

逢 甲 大 學
交通工程與管理學系碩士班
碩 士 論 文



模擬式動態交通指派模式
系統最佳化原則之研究

**Micro-simulation Based Dynamic Traffic
Assignment: The Principle of System Optimization**

指導教授：胡大瀛

研 究 生：李達樟

中 華 民 國 九 十 一 年 六 月

誌 謝

十幾年學生生涯，胡大瀛博士可說是影響我最深的恩師，胡老師亦師亦友，無論在研究上的指導督促，亦或是生活上的關心，總不厭其煩地給予叮嚀與關懷，在此獻上學生最深的感謝之意。

此外，口試期間承蒙陳惠國教授及魏慶地博士於百忙之中抽空指導，提供學生許多寶貴意見，並指導論文內容缺失之處，使論文更具完整性；論文計畫書與進度審查期間，所上李克聰博士及溫傑華博士細心審閱與謬誤之指正，在此一併致上最誠摯之謝意；另外感謝張湧老師於學生碩士班期間，無論在生活上或學習上給予許多的啟發與關心；也感謝所上所有老師及助教們在各方面的幫忙。

論文期間，尤其感謝同窗好友，百賢不厭其煩地提供程式上的協助，以及景文(Van)在實驗過程的寶貴意見，以及志霖(小胖)、昭維(Vie)、憲國、仲偉、筱葳、曉蘋(令果)、俊豪、俊樺(hero)學長、建昌(衝ㄟ)、仁旭在生活及課業上的互相關懷勉勵。

在逢甲六年，完成了我的學士及碩士學位，於大三下學期進入 Pluto，Pluto 感覺像個大家庭，胡老師猶如我們的大家長，我們經過無數「熬夜趕工」的夜晚，正因如此我們有如兄弟般情感，有「吃」同享、有難同當，因為其實大家也沒什麼朋友。感謝文能學長幫我細心校對，以及惠如在計畫案及專題的幫忙，另外要感謝英志(Ethan)學長，總是知道我的需求，在我最需要幫忙的時候，適時提供寶貴的意見及資料，更感謝許董讓我兩年內嚐盡各式美食，以致增加了五公斤；佳貝(Bruce)與仲強(Roger)在「精神」上的支持，實在是感激不盡，希望以後可以大家再去吃六點的早餐，也謝謝錦昌在繪圖上的協助；最後還有還有 yoyo 學姊在論文及各方面的意見、甄試期間收容我的 ddc 及芳瑜學姐、及怪怪的 Boat。

其實最要感謝的應是七年來一直陪伴著我的惠婷，數年來的風風雨雨，有了她的體諒、鼓勵與支持，我才得以順利完成論文。

最後謹將此份論文獻給我最親愛的家人，尤其是等不及與我分享這份喜悅及榮耀的父親，感謝父親給我最好的學習環境，在未來的日子裡，我會更加努力；還有永遠支持鼓勵我的母親及其他所有關心我的家人，唯有你們當我精神上的後盾，我才能成就這份論文，謝謝你們！

李達權 謹誌
於 Pluto lab
2002 年 7 月

摘 要

智慧型運輸系統希望透過對道路使用需求的管理，提供即時路況資訊給駕駛者，達到整體路網有效利用，在智慧型運輸系統架構下，強調「即時性」的交通資訊與交通控制策略，因此傳統靜態運輸系統分析方法已無法滿足資訊化交通路網(Information-Based Traffic Network)的分析需求。

資訊化交通路網內的依時性車流分佈取決於資訊之供給、駕駛者行為、路網特性、與交通控制策略；然而在此一複雜環境下如何正確且有效的分析交通車流，在這個複雜的環境下，動態交通指派的觀念被提出做為分析動態交通車流的主要方向，目前已發展的方法，包括最佳控制化理論、變分不等式、數學規劃法與模擬式指派程序等四種。本研究即根據模擬式動態交通指派之演算架構，探討系統最佳化 (System Optimization, SO) 指派原則。

在模擬式動態交通指派模式中，車輛根據依時性 OD 旅次產生，配合指派原則選取的路徑進行模擬。路徑的產生，除可透過交通指派步驟的直接應用外，經由路徑與車輛之配合，模擬可產生各時段下車輛在路網內移動的情形，此一依時性流量型態，可供其他指派原則的應用與產生。這個架構下，傳統式的路段績效函數之功能完全被模擬所取代，如路段時間、路口延滯時間的計算等。

路網指派原則，包括系統最佳化 (SO) 及使用者均衡 (UE) 指派原則，主要目的在分析系統管理目標與個人最佳化目標下之車流量分佈。系統最佳化指派原則，是在 ATMS 下管理者，期望透過規範性資訊的提供與交通控制策略或措施，來規範駕駛者使用提供之路徑，達到整體路網運作績效最佳化為目標。在系統最佳化的條件下，使用到的路徑其「邊際成本」將會趨於一致，然而在動態的環境下，如何定義與計算路徑邊際成本成為一個重要的課題。

目前各種對動態與靜態交通指派模式，系統最佳化指派原則的路段邊際成本之討論，皆有所差異，且各方法之邊際成本(marginal)表示方法，所得到的值，並無任何標準值得以驗證其結果合理性與差異性，本研究首先透過計算各方法之依時性邊際旅行時間得到初步比較，接著將各表示方法，應用於模擬式動態交通指派模式中，並作分析比較；最後，整合各表示方法之優點構建一新的邊際成本表示方法，於模擬式動態交通指派模式架構下，進行實際路網測試。

關鍵詞：模擬式動態交通指派模式、系統最佳化原則、邊際成本。

Abstract

The need for a methodology to provide time-dependent estimates of traffic flows and associated performance measures in networks arises in several problem areas, including Advanced Traffic Management systems (ATMS) and Advanced Traveler Information systems (ATIS). In order to properly capture dynamic, real-time flow distribution in an information-based network, the concept of dynamic traffic assignment has been proposed to extend traffic assignment techniques in a dynamic aspect.

Four different approaches, including mathematical formulations, optimal control theory, variational inequalities, and simulation-based approach, have been applied in developing Dynamic Traffic Assignment (DTA). Two assignment principles, namely, system optimization (SO) and user equilibrium (UE), have been widely accepted and used in transportation planning. The UE rule implies that at equilibrium no motorist can experience a lower travel time by unilaterally changing routes; the SO rule implies that the system overall travel time is minimum.

Within the micro-simulation based DTA procedure, an iterative scheme is developed to generate time-dependent least marginal cost path based on the simulation model, DYNASMART. The core of the procedure is DYNASMART, a simulation-assignment model, and the model allows consideration of individual trip maker decisions at the origin as well as at each node along the way to the destination, in addition to describing the dynamics of vehicular traffic flow.

In the procedure, link marginal costs are estimated and utilized within a time-dependent least marginal shortest-path algorithm. Thus, the estimation of link marginal cost is an important factor for micro-simulation based DTA. Different estimation approaches, such as dynamic link performance functions, queue-based delay estimation, and numerical estimation, have been developed for estimating link marginal cost. This research aims at exploring marginal cost calculations and optimizing the estimation in the micro-simulation based DTA procedure.

Numerical experiments on a test network are conducted to investigate the accuracy of link marginal cost. Relationship between time-dependent marginal travel time and path travel time are compared to obtain further understanding of marginal cost calculation characteristics of those approaches.

One new model is constructed based on the results of numerical experiments on the test network. Numerical experiments on an actual network are also conducted to investigate the accuracy and efficiency of the micro-simulation based DTA procedure using new marginal cost estimation model.

Keywords: system optimization, Dynamic Traffic Assignment, marginal cost.

目錄

目錄.....	
圖目錄.....	
表目錄.....	
第一章 緒論	1
1.1 研究背景.....	1
1.2 研究動機與目的.....	2
1.3 研究流程.....	3
第二章 文獻回顧	5
2.1 靜態交通指派模式.....	5
2.2 動態交通指派模式.....	7
2.3 邊際成本表示方法.....	8
2.3.1 動態系統最佳化路段邊際旅行.....	10
2.3.2 局部邊際延滯.....	11
2.3.3 局部邊際旅行時間.....	12
2.3.4 Peeta 之邊際成本表示方法.....	14
2.3.5 DYNASMART-X 中的邊際旅行時間.....	16
2.4 DYNASMART 模擬指派模式.....	20
2.4.1 DYNASMART 之主要模式架構與功能.....	21
2.4.2 DYNASMART 之輸入及輸出資料.....	24
第三章 研究方法	26
3.1 模擬式動態交通指派模式.....	26
3.1.1 模擬式動態交通指派模式之觀念性架構.....	26
3.1.2 模擬式 SO 動態交通指派演算法.....	28
3.2 邊際成本對模擬式動態交通指派演算之影響.....	30
3.2.1 模擬式動態交通指派模式中之邊際成本計算.....	31

3.2.2 邊際成本計算對模式之影響.....	31
3.3 邊際成本推估模式之特性與構想.....	32
第四章 邊際成本表示方法比較與分析.....	35
4.1 各表示方法之特性與比較.....	35
4.1.1 局部邊際延滯.....	35
4.1.2 局部邊際旅行時間.....	36
4.1.3 Peeta 之表示方法.....	36
4.1.4	
DYNASMART-X.....	37
4.1.5 各表示方法之特性比較.....	38
4.2 依時性邊際旅行時間之測試.....	40
4.2.1 實驗設計.....	40
4.2.2 路徑之依時性邊際旅行時間.....	43
4.2.3 測試結果.....	44
4.2.4 各方法合理性之比較.....	51
4.3 模擬式動態交通指派模式之測試.....	55
4.3.1 測試流程.....	55
4.3.2 實驗路網及條件.....	56
4.3.3 各方法實際路網測試結果.....	57
4.4 小結.....	60
第五章 模式構建及測試.....	61
5.1 模式之構建與路徑依時性邊際旅行時間測試.....	61
5.1.1 模式之構建.....	61
5.1.2 路徑依時性邊際旅行時間測試結果.....	63
5.2 模式於模擬式 SO 動態交通模式之測試.....	65
5.2.1 測試條件.....	65
5.2.2 測試結果.....	66

5.3 演算法之收斂.....	71
5.4 演算效率分析比較.....	73
5.4.1 模擬式 SO 動態交通指派模式演算步驟.....	73
5.4.2 演算法運算時間.....	75
5.5 模式之實際應用.....	76
5.5.1 道路系統容量變化.....	76
5.5.2 緊急事故之應用.....	78
5.6 小結.....	81
第六章 結論與建議.....	82
6.1 結論.....	82
6.2 建議.....	85
參考文獻.....	86
附錄.....	88
附錄一 本研究模式求解程式碼.....	88
附錄二 局部邊際延滯求解程式碼.....	96
附錄三 Peeta 之邊際成本表示方法求解程式碼.....	100
附錄四 局部邊際旅行時間求解程式碼.....	104
附錄五 依時性邊際旅行時間求解程式碼.....	109

圖目錄

圖 1.1	研究流程圖.....	4
圖 2.1	邊際成本基本定義示意圖.....	10
圖 2.2	局部邊際延滯推估示意圖.....	12
圖 2.3	局部邊際旅行時間示意圖.....	13
圖 2.4	即時資訊下邊際旅行時間推估示意圖.....	14
圖 2.5	上游路段邊際影響示意圖.....	19
圖 2.6	DYNASMART 模式架構.....	23
圖 2.7	DYNASMART 功能架構.....	24
圖 3.1	模擬式動態交通指派之觀念性架構.....	27
圖 3.2	模擬式 SO 動態交通指派之演算流程圖.....	30
圖 3.3	模式發展與測試流程圖.....	34
圖 4.1	路段邊際求解微分示意圖.....	37
圖 4.2	虛擬路網圖.....	40
圖 4.3	路徑依時性邊際旅行時間計算流程圖.....	44
圖 4.4	需求因子與平均停等、旅行時間關係圖.....	45
圖 4.5	不同需求因子及時間點加入路徑 7 之旅行時間圖.....	53
圖 4.6	路徑 7 需求因子 1.0 之依時性邊際旅行時間.....	53
圖 4.7	路徑 7 需求因子 2.0 之依時性邊際旅行時間.....	54
圖 4.8	路徑 7 需求因子 3.0 之依時性邊際旅行時間.....	54
圖 4.9	路徑 7 需求因子 4.0 之依時性邊際旅行時間.....	55
圖 4.10	動態交通指派模式示意圖.....	56
圖 4.11	實際路網圖.....	58
圖 4.12	需求因子 1.0 各表示方法每遞子迴之平均旅行時間.....	60
圖 5.1	鄰近路段邊際影響示意圖.....	62
圖 5.2	求解切線示意圖.....	63
圖 5.3	需求因子 1.0 依時性邊際旅行時間與旅行時間關係圖.....	64
圖 5.4	需求因子 2.0 依時性邊際旅行時間與旅行時間關係圖.....	64
圖 5.5	需求因子 3.0 依時性邊際旅行時間與旅行時間關係圖.....	65

圖 5.6	需求因子 4.0 依時性邊際旅行時間與旅行時間關係圖	65
圖 5.7	每模擬時段路網中車輛數	66
圖 5.8	需求因子 1.00 每遞子迴之平均旅行時間	69
圖 5.9	需求因子 2.75 每遞子迴之平均旅行時間	69
圖 5.10	需求因子 2.75 每遞子迴平均旅行時間變化	71
圖 5.11	模擬式 SO 動態交通指派模式程式執行流程	74
圖 5.12	道路容量改變路網中車輛每遞子迴之平均旅行時間	78
圖 5.13	模擬式 SO 動態交通指派應用於緊急事故決策之架構	80



表目錄

表 4.1	各計算方法比較表.....	39
表 4.2	比較路徑一覽表.....	42
表 4.3	不同需求因子路網特性表.....	45
表 4.4	不同需求因子下第四分鐘加入各路徑之旅行時間表.....	47
表 4.5	需求因子 1.0 第四分鐘加入各路徑依時性邊際旅行時間表..	48
表 4.6	需求因子 2.0 第四分鐘加入各路徑依時性邊際旅行時間表..	49
表 4.7	需求因子 3.0 第四分鐘加入各路徑依時性邊際旅行時間表..	50
表 4.8	需求因子 4.0 第四分鐘加入各路徑依時性邊際旅行時間表..	51
表 4.9	需求因子 1.00 各方法每遞子迴之平均旅行時間.....	59
表 5.1	需求因子 1.00 每遞子迴之平均旅行時間.....	68
表 5.2	需求因子 2.75 各方法每遞子迴之平均旅行時間.....	70
表 5.3	需求因子 2.75 各方法之數值實驗結果.....	70
表 5.4	需求因子 2.75 之平均旅行時間平均變化量.....	71
表 5.5	違反值比較表.....	72
表 5.6	需求因子 2.75 各方法之計算時間比較表.....	76
表 5.7	本研究模式改善演算效率百分比比較表.....	76
表 5.8	需求因子 2.75 不同道路容量下每遞子迴平均旅行時間.....	77
表 5.9	不同道路容量下實驗結果.....	78

第一章 緒論

1.1 研究背景

隨著世界各國的經濟發展，私人運具持有率快速成長，私人運具數量增加卻成為目前各國都市地區之最大問題，如何有效利用有限的道路資源成為各國重視的課題。

近年來所發展之智慧型運輸系統（U.S. D.O.T.，1999）所包含的先進交通管理系統（ATMS）與先進用路人資訊系統（ATIS），希望透過對道路使用需求的管理，提供即時路況資訊給駕駛者，達到整體路網有效利用，在智慧型運輸系統架構下，交通旅運規劃與分析需考慮資料的彈性與即時性，因為智慧型運輸系統強調「即時性」的交通資訊與交通控制策略，因此傳統靜態運輸系統分析方法已無法滿足資訊化交通路網（Information-Based Traffic Network）的分析需求。

基本上，資訊化交通路網內的依時性車流分佈取決於資訊之供給、駕駛者行為、路網特性、與交通控制策略；然而在此一複雜環境下如何正確且有效的分析交通車流，已是智慧型運輸系統發展中所面臨的重要課題。在這個複雜的環境下，動態交通指派的觀念被提出做為分析動態交通車流的主要方向，目前已發展的方法，包括最佳控制化理論、變分不等式、數學規劃法與模擬式指派程序等四種。本研究即根據模擬式動態交通指派之演算架構，探討系統最佳化（System Optimization, SO）指派原則。

在模擬式動態交通指派模式（胡大瀛，2001）中，車輛根據依時性 OD 旅次產生，配合指派原則選取的路徑進行模擬。路徑的產生，除可透過交通指派步驟的直接應用外，經由路徑與車輛之配合，模擬可產生各時段下車輛在路網內移動的情形，此一依時性流量型態，可供其他指派原則的應用與產生。這個架構下，傳統式的路段績效函數之功能完全被模擬所取代，如路段時間、路口延滯時間的計算等。

路網指派原則，包括系統最佳化（SO）及使用者均衡（UE）指派原則，主要目的在分析系統管理目標與個人最佳化目標下之車流量

分佈。使用者均衡指派原則之主要條件是：沒有駕駛者可以藉由路徑之改變而改善旅行時間或成本。系統最佳化指派原則，則是在 ATMS 下管理者期望透過規範性資訊的提供與交通控制策略或措施，來規範駕駛者使用提供之路徑，達到整體路網運作績效最佳化為目標，也就是說「系統中部分駕駛者必須犧牲本身之時間，以達系統最佳化」。在系統最佳化的條件下，使用到的路徑其「邊際成本」將會趨於一致，然而在動態的環境下，如何定義與計算路徑邊際成本成為一個重要的課題。

現有的動態交通指派模式有各種邊際旅行成本的表示方法，包含 Chen (1999) 動態系統最佳化 (DSO) 的邊際成本計算、Peeta(1994) 提到的依時性(time-dependent)邊際旅行時間，及美國德州大學奧斯汀分校所發展 DYNASMART-X 對路段邊際成本的計算方式，與局部邊際旅行時間(魏慶地，1999)，以及 Ghali and Smith(1993)利用車輛延滯觀念而建立的局部邊際延滯(local marginal delay, LMD)等方法，上述各方法皆為動態交通指派模式中，系統最佳化原則如何計算邊際成本之方法。

1.2 研究動機與目的

在智慧型運輸系統架構下，管理者必須透過即時之旅運規劃，利用先進旅行者資訊系統(ATIS)提供旅行者合理且正確的路徑導引，使整體運輸系統達到最佳化的目的，而駕駛者路徑的選擇，乃依據路徑之邊際成本進行路徑選擇，因此進行系統最佳化原則指派時，邊際成本之計算便顯現其重要性，因錯誤的邊際成本將造成指派路徑之錯誤，而使整個道路系統更加擁擠，故本研究希望透過探討分析目前各方法之優缺點，期望找出更正確及有效率之計算方式。

目前各種對動態與靜態交通指派模式，系統最佳化指派原則的路段邊際成本之討論，皆有所差異，然各方法之邊際成本(marginal)表示方法，所得到的值，並無任何標準值可以驗證或斷定其結果合理性與差異性，尤其現實生活中囿於時間與空間等因素，倘若要進行實地調查或實驗，則有其困難度，加上大規模調查等特性，在時間、空間、

物力與經費的考量下，於實際生活中取得影響駕駛人的邊際成本，實有其困難度，因此本研究透過計算各方法之依時性邊際旅行時間（Time-dependent marginal travel time）得到初步比較，以及各表示方法，應用於模擬式動態交通指派模式中之特性作分析比較。

根據上述說明本研究進行之目的為：

1. 透過整合性地比較各種邊際成本表示方法，期望得到各模式中合理且符合實際道路的邊際成本表示方法。
2. 各邊際成本表示方法的測試比較，進行方法改良與修正進而構建一模式，此模式將透過路網測試與修正得到能有效率地計算及真正反應實際邊際成本的方法。
3. 透過實際路網測試，達到模擬式動態交通指派模式之應用於實際路網圖中的可能性。
4. 提高未來應用模擬式動態交通指派模式，於交通控制與策略研擬時的可靠性與準確性。

1.3 研究流程

本研究之研究流程如圖 1.1，在界定研究方向與範圍之後，首先對研究所採用之模擬式動態交通指派模式及目前各種邊際成本計算方法進行回顧，其中邊際成本計算的方法，包含動態系統最佳化路段邊際旅行時間、Peeta 的邊際成本表示方法、DYNASMART-X 的方法、局部邊際延滯、局部邊際旅行時間等五種。在透過各種計算方法之分析與測試比較後，整合各方法之優點，發展出一新的模式，將新模式應用於模擬式動態交通指派模式中，並測試其合理性，最後做成結論與建議。

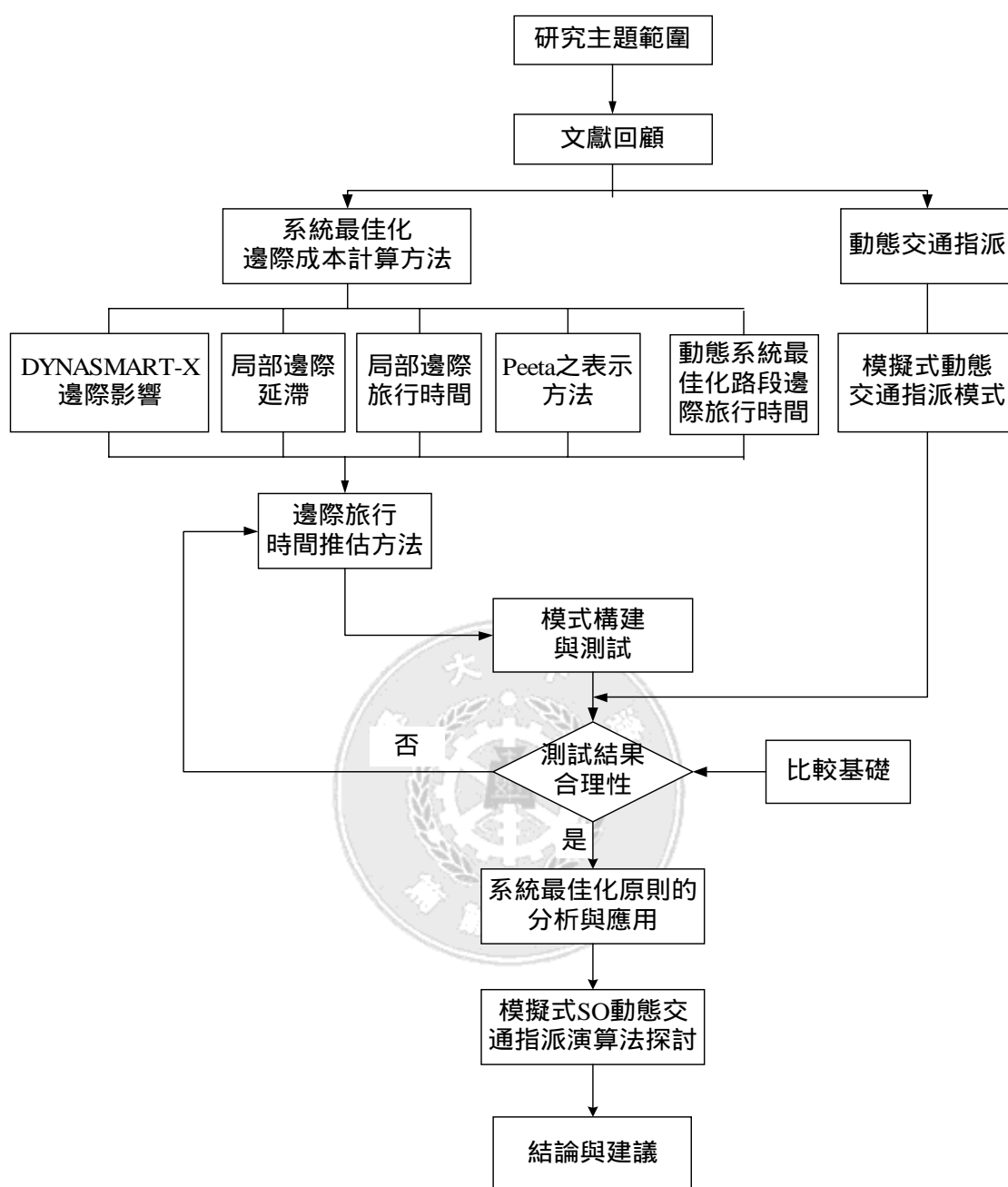


圖 1.1 研究流程圖

第二章 文獻回顧

本章將於 2.1 節中，描述傳統靜態交通指派的觀念；2.2 節回顧動態交通指派模式之求解方法；2.3 節中探討經濟學中，對「邊際成本之定義」，以及在動態交通指派模式，系統最佳化原則下各種「邊際成本」表示方法；並於 2.4 節中說明本研究所使用之交通模擬器 DYNASMART 主要功能與其輸入及輸出資料內容。

2.1 靜態交通指派模式

靜態交通指派假設路段交通流量和旅行時間不隨時間之變動而改變，主要目的為將一靜態起迄 OD 旅次表指派於路網中的路段，透過累計的方式獲得車流在路網分佈的情形。早期的路網指派只考慮最短路徑的使用。最早的指派方法，全有或全無 (all-or-nothing) 將 OD 表中的旅次全部指派到該 OD 對之最短路徑，後續的容量限制指派方法試圖改變此一不合理現象 (Sheffi, 1985)，但大部份的容量限制指派，缺乏理論上的根據，僅以啟發式 (heuristic) 方法求解，因此常得出不合理的解。

Beckmann et al.(1956)提出第一個使用者均衡 (user equilibrium, UE) 指派的數學模式，模式所獲得的解，具存在 (existence) 與唯一 (uniqueness) 等數學特性，並滿足 Wardrop (1952) 所提出的使用者均衡的條件：亦即沒有駕駛者可以藉由路徑的改變改善旅行時間或成本。除了使用者均衡指派的問題外，靜態交通指派的另一主要思考方向為規範性的問題，希望透過流量的分佈，達成運輸路網使用的系統最佳化 (system optimization, SO)。靜態交通指派可以 Frank-Wolfe 演算法求解，其他相關模式請參閱卓訓榮與李治輝(1995)。

一般而言，路網均衡模型係指根據 Wardrop(1952)的用路人或系統均衡原則，以及事先校估的路段績效函數，將路網上起迄對間的運輸需求，有系統地分配到路網各個路段上所構建的模型，使用者均衡指派的數學模式 (Sheffi, 1985)，如下所示：

$$\min Z(w) = \sum_a \int_0^{w^a} ta(w)dw \quad (2-1)$$

subject to

$$\sum_k f_{ijk} = f_{ij}, \forall i, j \quad (2-2)$$

$$f_{ijk} \geq 0, \forall i, j, k \quad (2-3)$$

$$w^a = \sum_i \sum_j \sum_k f_{ijk} \delta_{ijk}^a, \forall a \quad (2-4)$$

式中， w 代表路段流量之值； w^a 為路段 a 上的流量； $ta(.)$ 代表路段之時間函數，或稱績效函數（link performance function）， $ta(.)$ 一般是正、遞增，且為凸形函數。 f_{ij} 代表由起點 i 至迄點 j 的指派量， f_{ijk} 為（ $i \rightarrow j$ ）間第 k 路徑的流量，靜態路段 - 路徑指標以 δ_{ijk}^a 代表，可用於路徑流量（path flow）與路段流量（link flow）的換算。

式(2-1)之目標函數 $Z(w)$ ，是取路段績效函數的積分總和，此一函數並不具備經濟上、行為上的解釋，只能說是以數學方式來建立的函數，而此一函數又恰相當於使用者均衡(user equilibrium)的問題。式(2-2)的限制是流量的守恆；式(2-3)是非負解的限制，以求得合理的流量分佈；式(2-4)的條件將路徑流量與路段流量以 δ_{ijk}^a 的指標連結。Sheffi(1985)於所著的書中，將使用者均衡指派相關的問題，包含觀念、數學式、解法，有詳盡的探討，在此不再加以詳述。

除了使用者均衡指派的問題外，靜態交通指派的另一主要思考方向為規範性的問題，希望透過流量的分佈，達成運輸路網使用的系統最佳化。靜態系統最佳化(SO)的問題可以下式表示：

$$\min Z(w) = \sum_a w^a t_a(w^a) \quad (2-5)$$

subject to

$$\sum_k f_{ijk} = f_{ij}, \forall i, j \quad (2-6)$$

$$f_{ijk} \geq 0, \forall i, j, k \quad (2-7)$$

$$w^a = \sum_i \sum_j \sum_k f_{ijk} \delta_{ijk}^a, \forall a \quad (2-8)$$

式中所有的變數意義與前述 UE 的數學模式相同。系統最佳化的流量分佈並不代表系統均衡，部份使用者仍可藉由路徑的轉換達成縮短旅行時間的目的。因此達成在系統最佳化時，隱含著部份駕駛者願意犧牲自己的時間，以尋求交通系統的最佳化。

系統最佳化(SO)與使用者均衡(UE)指派的求解過程，基本上非常類似，使用者均衡(UE)的解為連結起、迄兩點間使用到的路徑，其平均成本（i.e.旅行時間）相等，系統最佳化(SO)的解為連結起、迄兩點間使用到的路徑，其邊際成本相等。根據 Ben-Akiva(1985)、Boyce(1984)靜態指派模式的主要缺點可歸納如下：

1. 無法描述交通擁擠或尖峰間的交通狀況。
2. 無法考慮在 ATIS/ATMS 的系統下，駕駛者可能的個別反應。

如何改善這些缺點並顧及系統最佳化(SO)與使用者均衡(UE)的特性，將是動態交通指派模式所探討的主要重點，第 2.2 節中即對動態交通指派模式之主要求解方法進行回顧。

2.2 動態交通指派模式

近年來在先進交通管理系統（ATMS）與先進用路人資訊系統（ATIS）之發展下，動態交通指派模式更顯得其重要性，主要是管理者，需要透過動態交通指派之分析，於最短時間內提供駕駛者相關的即時資訊，以達到系統管理之目標。動態交通指派模式，依照研究內容與求解方法之不同大致可分為數學規劃式、最佳控制化之動態指派模式、變分不等式之動態指派模式、與德州動態指派模式（胡大瀛，2001）。茲就各方法簡單介紹如下。

1. 數學規劃法

傳統動態交通指派(dynamic traffic assignment)數學模式，主要的觀念是由 Merchant 與 Nemhauser (M—N) 於 1978 年所提出。M—N 模式改良傳統指派方法，模式中假設一個迄點的情形，Merchant and Nemhauser (1978) 將上述問題化成一片斷連續(piecewise)方程式，以線性規劃方式求解。往後亦有如 Carey 與 Gahli and Smith(1995)對動態模式之探討。

2. 最佳控制化理論的動態指派模式(Optimal Control Theory Based Formulation)

這一類的理論以最佳控制化理論 (optimal control theory) 為基礎，導出連續性的最佳控制方程組。最佳控制理論的前身為變異微分

(calculus of variations)；許多控制理論被成功的應用在航空，太空的領域中。這類的方法假設在研究期間內，已知連續性，依時性的起訖點流量(continuous time-dependent OD flows)，理論的發展以系統最佳化與使用者均衡的動態指派為主。於數學式中將路段的流入率視為控制變數，而將路段的流出率視為路段上車輛數的函數，因此模型較為複雜。

3. 變分不等式之動態指派模式

在動態路網均衡問題上的研究，以變分不等式應用於動態旅運選擇模型，早期之作法框架在最佳化控制理論之下，而求解過程中，大半仍為轉換成一般非線性數學規劃模型求解。因此，其立論基礎遠不如直接將變分不等式的技巧，構建成動態旅運選擇的數學規劃模型。Chen(1999)在變分不等式與動態旅運需求模式的基礎下探討動態路網均衡模型，針對相關問題有深入的探討與分析，根據交通資訊型態，發展出三大類動態使用者最佳化旅運模型，包括確定性、隨機性、與模糊性等模型，並根據旅運決策來細分動態路徑指派，出發時間與路徑選擇等之模式求解與應用，模型具備簡單及數學證明嚴謹之特性，求解演算法則以巢狀對角法(nested diagonalization method, NDM)為中心，依模型的不同而予以調整，各個模型均數值範值加以說明對照，可說是未來發展相關研究的重要參考資料。

4. 德州模擬式動態指派模式(The Texas Approach)

德州大學奧斯汀分校於 1990 年開始發展動態指派模式。然而，這個數學模式之方程組無法以現有數學方式求解，為能有效的描述車流的特性與行為，因此以模擬的方式取代傳統的路段績效函數(link performance function)與路段上離開函數(exit function)。

2.3 邊際成本表示方法

根據靜態交通指派的定義，系統最佳化原則下，車輛路徑的選擇乃依據路徑的邊際成本，因此邊際成本的定義及計算正確與否便顯現其重要性。本節主要回顧經濟學中對「邊際成本」的定義，以及目前常應用於動態交通指派模式，系統最佳化原則指派的邊際成本表示方

法。

首先就經濟學上的定義而言，所謂邊際成本係指：「生產一單位產品，所需額外加入之成本。」(Elbert V. and Judith H. , 1995)，如圖 2.1，橫軸為產出量 (N)，縱軸為生產成本 (C)，總成本函數曲線 TC，由總成本函數曲線可知：產出單位 n 的成本為 C1，產出 n+1 單位的成本為 C2，則若欲將產出單位由原本的 n 單位提高至 n+1 單位，則該產品在 n 單位時的生產邊際成本可定義如式 (2-9)：

$$MC = \frac{C2 - C1}{(n+1) - n} = \frac{dTC}{dn} \quad (2-9)$$

若以數學角度定義，則某產品在產量 n 的邊際成本，可以產量與生產成本，所形成之總成本曲線上，對應產量 n 的該點切線的斜率(如圖 2.1 中 L 線) 定義之，如式 (2-9) 最右側的微分部分。

就道路系統的邊際成本而言，系統的「產出」為「車輛加入道路系統的車輛數」，而成本為「道路系統所增加之成本」，故其定義的邊際成本是為「道路系統中，加入一輛車對系統之影響」。因此其邊際成本的計算，係將「因車輛加入道路系統後造成道路系統中其他車輛增加的旅行成本」，再加上「車輛本身之旅行成本」，便是造成整體系統所增加之邊際成本，以靜態交通指派模式中計算邊際成本的模式為例，如式 (2-10)。

$$t^a = t_a(w) + w^a \frac{\partial t_a(w)}{\partial w^a} + \sum_b w^b \frac{\partial t_b(w)}{\partial w^a} \quad (2-10)$$

上述(2-10)式由三部分組成：(1) $t_a(w)$ 為該車行駛於路段 a 上所需之旅行時間，(2) $\frac{\partial t_a(w)}{\partial w^a}$ 為因該車進入路段 a，造成路段 a 上每一輛車所額外增加之旅行時間，將其乘上路段 a 之流量 w^a 即為路段 a 其他所有車輛增加之旅行時間，(3) $\frac{\partial t_b(w)}{\partial w^a}$ 則是路段 a 增加一車輛造成其他路段 b 每輛車之額外之旅行時間，乘上路段 b 的流量 w^b 則是路段 b 上整體所增加的旅行時間。

式 (2-10) 即為道路系統中，整體邊際成本的表示方法，然最主要的關鍵，在於求解式中偏微分的值，目前有許多不同方法，求解式

中微分的近似值之邊際成本表示方法。

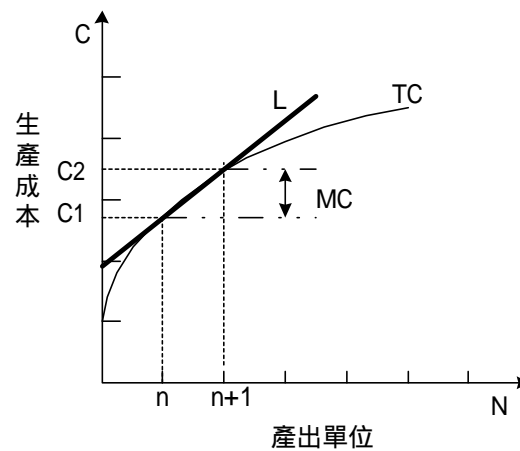


圖 2.1 邊際成本基本定義示意圖

如假設動態成本函數已知的情況下，直接對此已知之成本函數作偏微分後即得邊際成本者，如「動態系統最佳化路段邊際旅行時間 (Chen, 1999)」，茲於 2.3.1 節中說明；然而現實生活中，若要求得道路旅行時間函數並不容易，因此目前有許多避免使用成本函數的邊際成本表示方式，改以路口「延滯」(delay)觀念為計算依據者，如局部邊際延滯(Ghali and Smith, 1995)及局部邊際旅行時間(魏慶地, 1999)兩者，分別說明其主要觀念方法於 2.3.2 節及 2.3.3 節；最後，則是改以模擬之方式，得出推估路段邊際成本時所需的路段流量型態，如 Peeta (1994) 及 DYNASMART-X 的表示方法，如 2.3.4 節及 2.3.5 節所述。

2.3.1 動態系統最佳化路段邊際旅行時間(Chen, 1999)

Chen (1999) 所著的書中，對於動態系統最佳化 (Dynamic System-Optimal, DSO) 邊際成本的表示方法如下：

動態系統最佳化中，動態路段邊際旅行時間 $\hat{c}_a(t)$ 的定義為：於時間 t 在路段 a 上增加一輛車，對所有時間區段之所有路段，總旅行時間的影響，如式(2-11)所示：

$$\begin{aligned}\hat{c}_a(t) &= \frac{\partial \left(\sum_{a'} \sum_{t'} c_{a'}(t') u_{a'}(t') \right)}{\partial u_a(t)} = c_a(t) \frac{\partial u_a(t)}{\partial u_a(t)} + \sum_{a'} \sum_{t'} u_{a'}(t') \frac{\partial c_{a'}(t')}{\partial u_a(t)} \\ &= c_a(t) + \sum_{a'} \sum_{t'} u_{a'}(t') \frac{\partial c_{a'}(t')}{\partial u_a(t)}\end{aligned}\quad (2-11)$$

(2-11)式中,動態路段邊際旅行時間 $\hat{c}_a(t)$ 由兩部分組成,其中 $c_a(t)$ 為當路段流率為 $u_a(t)$,路段中加入一車輛時,該車輛的旅行時間;另外 $\sum_{a'} \sum_{t'} u_{a'}(t') \frac{\partial c_{a'}(t')}{\partial u_a(t)}$ 則是該加入車輛對其他時間 t' ,路網中其他路段 a' ,所影響的旅行時間。

此方法進行數值實驗求解邊際成本時,假設動態旅行時間函數(dynamic travel time function)存在,如(2-12)式:

$$c_a(t) = 1 + 0.01(u_a(t))^2 + 0.01(x_a(t))^2 \quad (2-12)$$

(2-12)式中的 $u_a(t)$,指在時間區段 t 中路段 a 的流入率, $x_a(t)$ 則是在時間區段 t 開始時路段 a 上的車輛數,其中 $x_a(t)$ 可被流入率的變數取代。上式之時間函數為可微分,故當動態系統最佳化路段邊際旅行時間計算時,直接對上述之動態旅行時間成本函數作偏微分,便可求得邊際成本。

2.3.2 局部邊際延滯(local marginal delay)

Ghali and Smith(1995)將局部邊際延滯應用在動態交通指派模式,主要係利用邊際延滯之觀念取代傳統邊際旅行時間函數,其主要觀念為,在尖峰時段,道路上的容量小於流入率時,便會形成等候(queue)的情況,進而導致延滯(delay),而利用延滯的計算作為邊際成本之表示方法,茲說明如后。

局部邊際延滯分成兩部分,一為該車輛進入路段後本身所遭遇到的延滯;另一部分為該車輛加入後,造成路段上其他車輛的延滯。如圖 2.2,假設流率為單位時間為 1 輛車,假設某車於時間點 t 進入路段並進行等候,則 t 時間之紓解率 $w(t)$ 固定的情況下,該車本身必須等待相當於 d 長的時間,直到該車開始起動駛離,此時該車即開始影

響路段上其餘車輛，直到 T 時間點，即 $v(t)$ 曲線與 $w(t)$ 曲線相交時。故一輛車進入道路系統本身的延滯為 d ，而造成其他車輛的延滯則為 $m-d$ 。

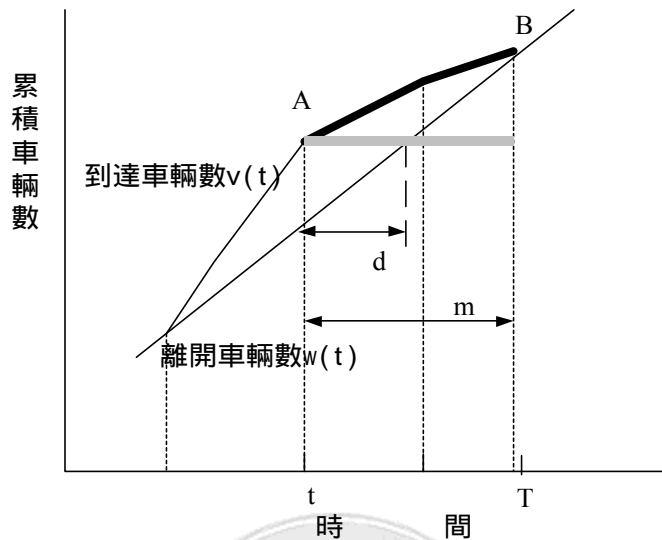


圖 2.2 局部邊際延滯推估示意圖

2.3.3 局部邊際旅行時間（魏慶地，1999）

魏慶地（1999）並未對整體路網的邊際旅行時間進行考量原因為：依時性整體的邊際旅行時間考慮路段上增加一車輛後對時間 t 之後的時段 ($\tau=t+1, \dots$) 的整體系統有何影響的情況下計算過於複雜，因此產生「局部邊際旅行時間」的計算。

魏慶地（1999）利用 Ghali and Smith(1995)局部邊際延滯的觀念視為車輛於路口的延滯，然後依據 DYNASMART 所使用的速率與密度關係圖得到一輛車加入路段對路段的影響，計算方法如下：

1. 路段行駛之邊際旅行時間

局部邊際旅行時間應用 DYNASMART 中根據速率—密度關係圖所得的速率，在路段上加入一車輛重新計算其密度、速率及路段旅行時間，再扣除未加入車輛前原有之旅行時間，所得到者即為新增一車輛於該路段增加之旅行時間，再乘以該車輛於路段行駛時間內路段上的車輛數即為路段之邊際旅行時間，運算過程假設該車輛行駛於路段上之時間內進入路段流量固定。

2. 路口等候的邊際旅行時間

在路口等候線的紓解過程，當新的車輛加入後，將增加其後加入等候線之車輛的等候時間，所增加之等候時間即為路口等候的邊際旅行時間。此邊際旅行時間是由離開路口車輛數與到達路口等候車輛數累積曲線求得，如下圖 2.3(a)中 $w(t)$ 、 $v(t)$ 分別代表離開、到達路口的曲線函數，在 t 時間到達路口等候所造成其後車輛所增加的等候時間，如圖 2.3(a)中之粗黑線部分。即圖 2.3(b)中 $\triangle ABC$ 與 $\triangle abC$ 的面積差，即相當於時間 $m (=T-t)$ 。

利用累積曲線計算路口邊際旅行時間需事先得知未來各時段的到達及離開車輛數，在即時資訊下並無法得知未來交通狀況，因此必須使用推估的方式。其推估的方式，為在時間 t 時間到達之車輛 i ，其邊際旅行時間為將時間 t 與其前三分鐘($t-3$)的累積到達與離開車輛數所構成的直線延長，若兩直線無法相交，則將時間 t 再加上車輛 i 之等候時間 d ，即 $t+d$ ，假設累積到達曲線水平線與累積離開曲線之交叉點 C 為等候紓解時點，進而推出邊際旅行時間 m ，如下圖 2.4 所示，然其等候時間較短，將造成較大之誤差。

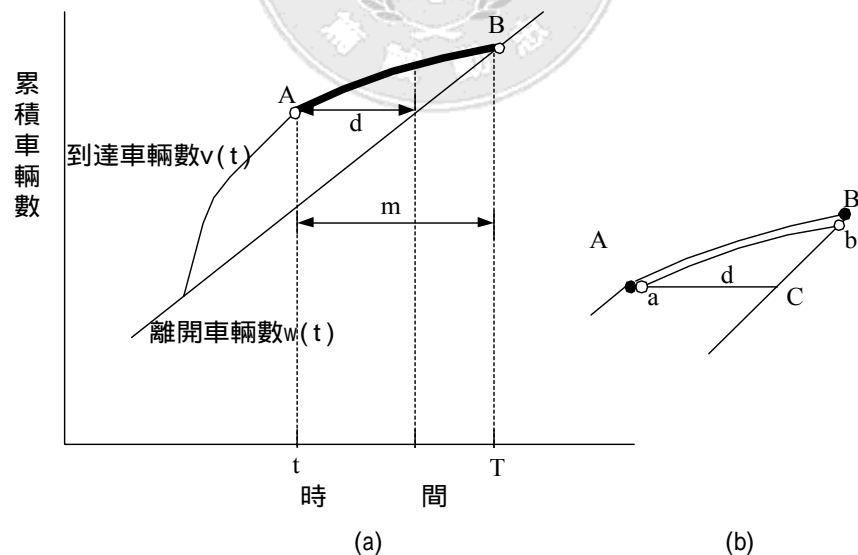


圖 2.3 局部邊際旅行時間示意圖

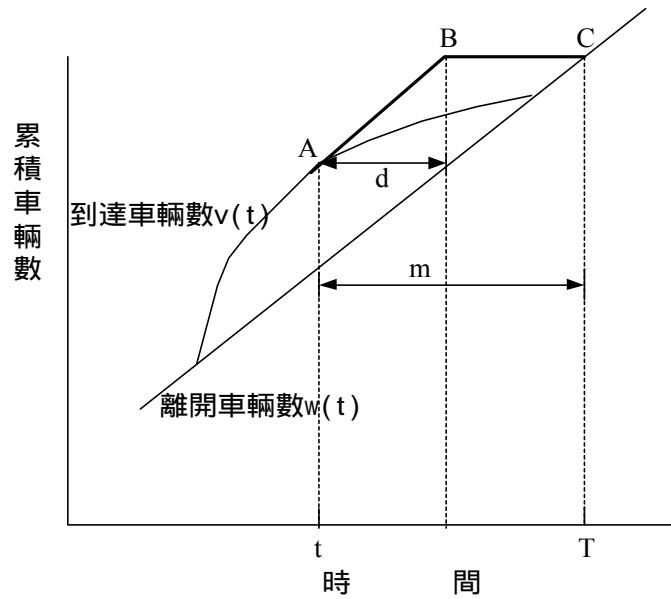


圖 2.4 即時資訊下邊際旅行時間推估示意圖

2.3.4 Peeta 之邊際成本表示方法

Peeta(1994)將靜態交通指派情況下邊際成本 t^a 以「每單位流量的增加對整個系統成本的改變量」定義之，如下式：

$$t^a = \frac{dz(w)}{dw^a}, \forall a \quad (2-13)$$

其中 $z(\cdot)$ 為系統總成本且為路段流率向量 w 的函數，所以假如整個系統的成本以系統的旅行時間來計算，則可以下式表示：

$$z(w) = \sum_c w^c t_c, c \in A \quad (2-14)$$

則由(2-12)、(2-13)整體路段邊際旅行時間如式 (2-10) 所示，為方便說明，再列出式 (2-10) 如下：

$$t^a = t_a(w) + w^a \frac{\partial t_a(w)}{\partial w^a} + \sum_b w^b \frac{\partial t_b(w)}{\partial w^a} \quad (2-10)$$

在不考慮路段間的彼此影響之情況下，可將(2-10)式簡化如(2-15)：

$$t^a = \frac{d[t_a(w^a) \cdot w^a]}{dw^a} = t_a(w^a) + w^a \frac{dt_a(w^a)}{dw^a} \quad (2-15)$$

其中， $t_a(w^a)$ 是傳統路段績效函數的流率與時間關係圖， $\frac{dt_a(w^a)}{dw^a}$ 則是在流量為 w^a 時的函數切線斜率。

就路徑而言，靜態狀況下路徑邊際旅行時間 T_{ijk} 是指當增加一輛車於起迄點為 i 、 j 的路徑 k 上時，造成系統增加的旅行時間。 T_{ijk} 可從以路徑為基礎的以下等式中求得：

$$z(f) = \sum_i \sum_j \sum_k f_{ijk} \cdot T_{ijk} \quad (2-16)$$

其中 T_{ijk} 為起迄點 i 、 j 的路徑 k 上的旅行時間， f_{ijk} 則是該路徑上之流量向量。則 T_{ijk} 可以(2-17)定義之：

$$T_{ijk} = \frac{dz(f)}{df_{ijk}}, \forall i, j, k \quad (2-17)$$

將(2-16)代入(2-17)，則特定的路徑 p ，起迄點 m 、 n ，該路徑 p 之邊際旅行時間如(2-18)式：

$$T_{mnp} = T_{mnp} + \sum_i \sum_j \sum_k f_{ijk} \cdot \frac{\partial T_{ijk}}{\partial f_{mnp}} \quad (2-18)$$

當路段之間的影響被忽略時，路徑的邊際旅行時間亦可由路徑上所有路段的邊際旅行時間加總求得(2-19)。

$$T_{ijk} = \sum_a t^a \delta_{ijk}^a \quad (2-19)$$

依時性的邊際旅行時間則因須考慮時間因素，故較靜態情況複雜，其依時性整體的邊際旅行時間 t^{ta} 如下：

$$\begin{aligned} t^{ta} = & T^{ta}(x) + d^{ta} \frac{\partial T^{ta}(x)}{\partial x^{ta}} + \sum_b d^{tb} \frac{\partial T^{tb}(x)}{\partial x^{ta}} \\ & + \sum_{\tau=t+1}^n d^{\tau a} \frac{\partial T^{\tau a}(x)}{\partial x^{ta}} + \sum_{\tau=t+1}^n \sum_c d^{\tau c} \frac{\partial T^{\tau c}(x)}{\partial x^{ta}}, \forall t, a \end{aligned} \quad (2-20)$$

其中前三項與靜態的情況類似，第一項中的 $T^{ta}(x)$ 為 x^{ta} 為 a 路段上的依時性車輛數，第二、三項的 $\frac{\partial T^{ta}(x)}{\partial x^{ta}}$ 與 $\frac{\partial T^{tb}(x)}{\partial x^{ta}}$ 則分別為在時間 t 於 a 與其他 b 路段上分別增加 d^{ta} 、 d^{tb} 輛車對 a 、 b 路段的上增加 d^{ta} 、 d^{tb} 的每一輛車增加的旅行時間，第四、第五項則是在 t 之後時段 ($\tau = t+1, \dots, n$)

新增一輛車對系統時間的影響。

依時性的路徑邊際旅行時間， T_{ijk}^t 為於時間 t 增加一輛車於 i, j 起迄點的路徑 k 上對系統增加的時間。將(2-18)式轉為依時性，可得(2-21)：

$$T_{mnp}^t = T_{mnp}^t + \sum_{\tau} \sum_i \sum_j \sum_k f_{ijk}^{\tau} \cdot \frac{\partial T_{ijk}^{\tau}}{\partial f_{mnp}^{\tau}} \quad (2-21)$$

表示為時間 t 時起迄點 m, n 的路徑 p 的邊際旅行時間，第二項為 t 時段後於路網中其他車輛的影響。相同的若不考慮路段之間的互相影響，則路徑的邊際旅行時間亦可以路段邊際旅行時間做加總求得。

2.3.5 DYNASMART-X 中的邊際旅行時間

DYNASMART-X (Center for Transportation Research U.T. at Austin) 中路段的邊際旅行時間定義為：「一額外車輛加入路段進行 L 的移動($L=1$ 為左轉， $L=2$ 為直行或右轉)所造成的總旅行時間的改變」，今假設總旅行時間的改變可以分成兩部分，分別為現有路段(tt_a)與上游鄰近路段(tt_{ups})的改變，則依定義構建模式如下：

$$mtt(a, ts, L) = \frac{\partial (tt_a + tt_{ups})}{\partial xp(a, ts, L)} \quad (2-22)$$

$$= \frac{\partial tt_a}{\partial xp(a, ts, L)} + \frac{\partial tt_{ups}}{\partial xp(a, ts, L)} \quad (2-23) \quad (2-24)$$

其中 tt_a ：現有路段上的總旅行時間；

tt_{ups} ：上游路段的總旅行時間；

$xp(a, ts, L)$ ：做 L 移動的總車輛數；

上式(2-23)則可定義為一額外 L 移動的車輛對現有路段產生的邊際影響，式(2-24)則定義為一額外 L 移動的車輛對上游路段產生的邊際影響。

(一)現有路段的邊際影響(marginal effect)

上述(2-23)式中隱含的邊際影響除額外加入車輛在路段上運行外

對現有路段上的影響，亦包括該加入車輛做轉向 L 對下游交叉路口的影響。所以當現有路段上有 $x_p(a, ts, \cdot)$ 的車輛數時，其總旅行時間 $T^{ts,a}$ 可表示如下：

$$T^{ts,a} = T_m^{ts,a} + T_{qL}^{ts,a} \quad (2-25)$$

其中 $T_m^{ts,a}$ 指車輛在路段上移動的時間，在此忽略車輛在行進中的車道變換的行為， $T_{qL}^{ts,a}$ 為在時間 ts 一輛做轉向 L 的車輛在等候線上紓解所需的時間。

則當增加一輛車進入現有路段進行左轉或直行右轉時，其現有路段的總旅行時間分別如(2-26)、(2-27)式：

$$T_l^{ts,a}(x) = T_m^{ts,a} + T_{ql}^{ts,a} \quad (2-26)$$

$$T_s^{ts,a}(x) = T_m^{ts,a} + T_{qs}^{ts,a} \quad (2-27)$$

而 $T_{qL}^{ts,a}$ 的計算可將等候作轉向 L 的車輛數除以紓解的流率，其中 $L=1$ 、 $L=2$ 分別代表左轉與直行右轉。

上述(2-23)式可推導如(2-28)式：

$$\begin{aligned} \frac{\partial tt_a}{\partial xp(a, ts, L)} &= \frac{\partial t[xp(a, ts, \cdot)] \cdot xp(a, ts, \cdot)}{\partial xp(a, ts, \cdot)} \\ &= t[xp(a, ts, \cdot)] + \frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, \cdot)} \cdot xp(a, ts, \cdot) \end{aligned} \quad (2-28)$$

其中 $t[xp(a, ts, \cdot)]$ 為當路段上有 $x_p(a, ts, \cdot)$ 的車輛數時的旅行時間，其等於(2-25)式的 $T^{ts,a}$ 。

因此 DYNASMART 中的現有路段的邊際成本，因為車輛轉向的不同而分成以下兩部分：

(1) 加入的車輛為左轉

$$\begin{aligned} \frac{\partial tt_a}{\partial xp(a, ts, 1)} &= t[xp(a, ts, \cdot)] + \frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, 1)} \cdot xp(a, ts, 1) \\ &= T_l^{ts,a} + \frac{\partial T_l^{ts,a}}{\partial X_l^{ts,a}} \cdot X_l^{ts,a} + \frac{\partial T_s^{ts,a}}{\partial X_l^{ts,a}} \cdot X_s^{ts,a} \cdot \alpha \end{aligned} \quad (2-29)$$

其中 $\frac{\partial tt_a}{\partial xp(a, ts, 1)}$ 為當在 ts 時間路段 a 加入一左轉車輛的邊際旅行時間， $T_l^{ts,a}$ 為當路段 a 在 ts 時間有 $x_p(a, ts, \cdot)$ 的車輛數且加入的車輛是左轉時，

估計的旅行時間； $\frac{\partial T_l^{ts,a}}{\partial X_l^{ts,a}} \cdot X_l^{ts,a}$ 為加入路段的車輛為左轉時，對左轉的車輛 $X_l^{ts,a}$ ，所增加的旅行時間； $\frac{\partial T_l^{ts,a}}{\partial X_l^{ts,a}} \cdot X_s^{ts,a} \cdot \alpha$ 為加入路段的車輛為左轉時，對右轉的車輛 $X_s^{ts,a}$ ，所增加的旅行時間，其中 α 為直行右轉車輛受左轉車輛影響的比例，在此設為 100%。

(2) 加入的車輛為右轉

$$\begin{aligned} \frac{\partial tt_a}{\partial xp(a,ts,2)} &= t[xp(a,ts,\cdot)] + \frac{\partial t[xp(a,ts,\cdot)]}{\partial xp(a,ts,2)} \cdot xp(a,ts,2) \\ &= T_s^{ts,a} + \frac{\partial T_s^{ts,a}}{\partial X_s^{ts,a}} \cdot X_s^{ts,a} + \frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot X_l^{ts,a} \cdot \beta \end{aligned} \quad (2-30)$$

其中 $\frac{\partial tt_a}{\partial xp(a,ts,2)}$ 為當在 ts 時間路段 a 加入一直行右轉車輛的邊際旅行時間， $T_s^{ts,a}$ 為當路段 a 在 ts 時間有 $xp(a,ts,\cdot)$ 的車輛數且加入的車輛是左轉時，估計的旅行時間； $\frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot X_s^{ts,a}$ 為加入路段的車輛為直行右轉時，對直行右轉的車輛 $X_s^{ts,a}$ ，所增加的旅行時間； $\frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot X_l^{ts,a} \cdot \beta$ 為加入路段的車輛為直行右轉時，對左轉的車輛 $X_l^{ts,a}$ ，所增加的旅行時間，其中 β 為左轉車輛受直行右轉車輛影響的比例，在此設為 100%。

然而如要求得上述式(2-29)、(2-30)的解，必須在式中所有偏微分可微的情況下方能求解，在 DYNASMART 中以近似值求解的方法求得各偏微分的值，其法為，利用旅行時間的值、車輛數及三個連續模擬時間 ts-1，ts 與 ts+1 進行各偏微分值的估計。

(二) 上游路段的邊際影響

所謂上游路段的邊際影響，指現有路段新加入一 L 轉向車輛對上游路段產生的影響，故將其視為「由於現有路段 a 上的容量限制，導致上游路段的等候」，因此當現有路段上的容量可容納上游路段的需求時，上游路段便無等候出現。

在 DYNASMART 中將現有路段上游的等候稱 Candidate Queue，當現有路段下游紓解車輛空出空間(free slots)後，便從 Candidate Queue 中加入新的車輛於現有路段 a 中，如下圖 2.5 中 A 為

Candidate Queue 的車輛數，而 B 為現有路段上的可容納空間(free slots)。

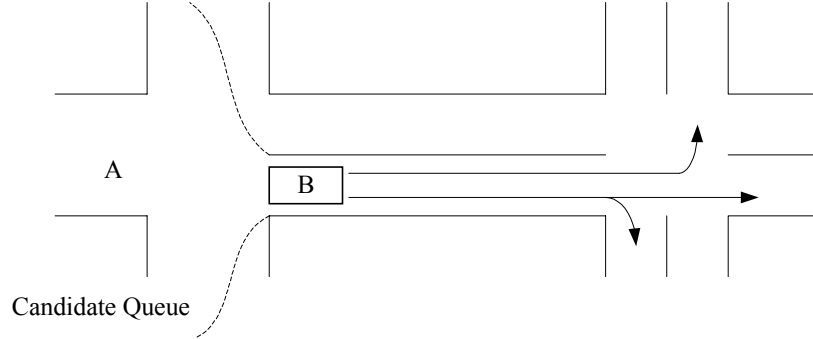


圖 2.5 上游路段邊際影響示意圖

根據以上的觀念及定義構建模式如下：

$$\frac{\partial t_{ups}}{\partial xp(a, ts, L)} = \frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, L)} \cdot \max\{0, (A - B)\} \quad (2-31)$$

其中 $\frac{\partial t_{ups}}{\partial xp(a, ts, 2)}$ 為現有路段增加一 L 轉向的車輛對上游路段的邊際影響； $\max\{0, (A - B)\}$ 則是上游鄰近路段等待進入現有路段的車輛數； $\frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, L)}$ 為現有路段上第一部車的延滯，可依進入車輛轉向不同分別推得式(2-32)、(2-33)。

$$\frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, 1)} = \frac{\partial T_s^{ts, a}}{\partial X_l^{ts, a}} \quad (2-32)$$

$$\frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, 2)} = \frac{\partial T_l^{ts, a}}{\partial X_s^{ts, a}} \quad (2-33)$$

將(2-32)與(2-33)的結果代入(2-31)中分別可得(2-34)與(2-35)。

$$\begin{aligned} \frac{\partial t_{ups}}{\partial xp(a, ts, 1)} &= \frac{\partial t[xp(a, ts, \cdot)]}{\partial xp(a, ts, 1)} \cdot \max\{0, (A - B)\} \\ &= \frac{\partial T_s^{ts, a}}{\partial X_l^{ts, a}} \cdot \max\{0, (A - B)\} \end{aligned} \quad (2-34)$$

其中 $\frac{\partial t_{ups}}{\partial xp(a, ts, 1)}$ 為一左轉車輛於 ts 時間加入現有路段 a 對上游路段的邊

際影響； $\frac{\partial T_s^{ts,a}}{\partial X_l^{ts,a}}$ 則為該左轉車輛對直行右轉車輛所造成的額外旅行時間；

$$\begin{aligned}\frac{\partial tt_{ups}}{\partial xp(a,ts,2)} &= \frac{\partial t[xp(a,ts,\cdot)]}{\partial xp(a,ts,2)} \cdot \max\{0, (A-B)\} \\ &= \frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot \max\{0, (A-B)\}\end{aligned}\quad (2-35)$$

與(2-34)式相同， $\frac{\partial tt_{ups}}{\partial xp(a,ts,2)}$ 為一右轉直行車輛於 ts 時間加入現有路段

a 對上游路段的邊際影響； $\frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}}$ 則為該右轉直行車輛對左轉車輛所造成的額外旅行時間；而上述兩式中的 $\max\{0, (A-B)\}$ 只能為正數，也就是說當上游的需求(A)大於現有路段的供給(B)時才會產生所謂的邊際影響，否則 $\max\{0, (A-B)\}$ 將為零。

(三)路段的邊際時間計算

將(一)與(二)所導出的現有路段及上游的邊際影響，依其加入車輛的轉向分別可得如式(2-38)、(2-39)：

$$mtt(a,ts,1) = T_l^{ts,a} + \frac{\partial T_l^{ts,a}}{\partial X_l^{ts,a}} \cdot X_l^{ts,a} + \frac{\partial T_s^{ts,a}}{\partial X_l^{ts,a}} \cdot X_s^{ts,a} \cdot \alpha + \frac{\partial T_s^{ts,a}}{\partial X_l^{ts,a}} \cdot \max\{0, (A-B)\} \quad (2-38)$$

$$mtt(a,ts,2) = T_s^{ts,a} + \frac{\partial T_s^{ts,a}}{\partial X_s^{ts,a}} \cdot X_s^{ts,a} + \frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot X_l^{ts,a} \cdot \beta + \frac{\partial T_l^{ts,a}}{\partial X_s^{ts,a}} \cdot \max\{0, (A-B)\} \quad (2-39)$$

2.4 DYNASMART 模擬指派模式 (胡大瀛, 1996)

上一節中本研究已對目前常用的邊際成本計算方式作詳細回顧與探討，本節則將分別對於本研究將使用的模擬器：DYNASMART 之主要模式之架構及其模擬之功能，以及其所需之輸入及輸出資料作說明。

2.4.1 DYNASMART 之主要模式架構與功能

本研究將以 DYNASMART 做為模擬路網中交通型態之模擬器，茲說明如下：

DYNASMART 是因應美國公路聯邦總署對 ATMS/ATIS 的規劃，所發展的綜合評估模式，其前身是 Chang et al. (1985) 用以模擬兩條平行道路的交通狀況之模擬模式，Jayakrishnan (1992) 將此一模式延伸至路網。Hu et al. (1992) 加強模擬與指派的整合，將交通控制策略加入此一模式，並改善模式運算效率。而根據 FHWA 的構想，模擬指派模式應具備下列的功能：

- 1.同時模擬市區街道與高速公路的交通。
- 2.同時進行多車種的指派。
- 3.反應個別駕駛者對交通資訊的反應。
- 4.能擷取與產生不同類型的交通資訊。

DYNASMART 主要是利用交通模擬來建立相對應的交通型態，並用以評估交通路網在不同假設下的績效，其採用模擬指派 (Simulation-Assignment) 的方式來指派依時改變的交通需求。迄今，DYNASMART 的應用，主要是以描述與評估在 ATIS/ATMS 架構下路網車流可能的變化；考慮的因素包含了交通資訊的產生與供給、交通控制策略、路線導引與駕駛者個體對系統的反應與決策。

如圖 2.6 所示，DYNASMART 包含了車流模式 (Traffic Flow Models) 的應用，路徑資訊產生 (Path Information Processing)、個體決策行為模式 (Individual Decision Behavioral Rules) 與可能的資訊供給法則 (Information Supply Strategies)。在這個架構下，旅次產生者所決定的出發時間 (Departure Time)，運具 (Model) 與使用路徑 (Route)，構成了一個三維的依時 OD 矩陣。DYNASMART 依時間次序，將私人車輛 (含汽、機車與貨車) 與公車陸續載入交通路網中，並按各車種的特性來移動車輛，直到車輛抵達目的地為止。在車輛移

動的過程中，交通資訊的傳輸與使用，亦將針對車輛特性加以區分。DYNASMART 架構中，包含三個主要元素：

1. 交通模擬

模擬階段，分成兩大部份，一是在路段中車輛的移動，另一是車輛進入路口的行為，我們可將兩者定義為路段移動 (Link Movement) 與路口轉換 (Node Transfer)。路段移動的主要目的在於捕捉車輛移動的情形，因此車輛速度的決定將是考慮的重點。路口轉換須能精確的反應交通控制的限制與混合車流組成之特性。DYNASMART 提供了大部份現有交控措施，其中可分為兩大項，市區街道與公路系統。

2. 路徑資訊的產生

根據交通資訊產生的策略，不同的路徑資訊，可由不同的駕駛者於旅次前或行進中使用。在這個元素中，具備考慮轉向成本的 k 條最短路徑將被用在此一模式，以確保模式在使用上之彈性 (Ziliaskopoulos, 1994)。

3. 旅次產生的決策行為

模式中最大的優點是假設旅次產生者能根據自己的行為特性或派遣中心的指示去選擇路徑，因此在這一個元素中，可以反應路徑或路線之選擇。

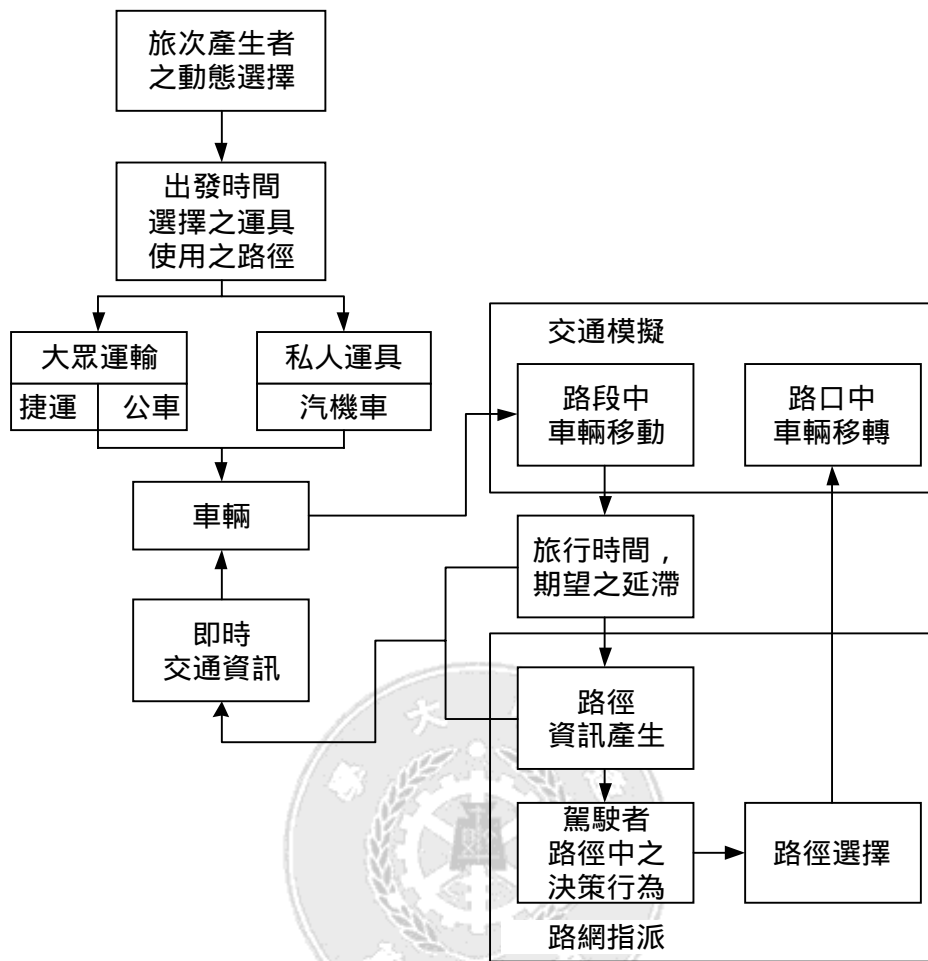


圖 2.6 DYNASMART 模式架構

DYNASMART 的功能結構，如圖 2.7 所示，其中包含了不同元素間的功能關係。圖中最基本的單元是車輛（或駕駛人）。在這個單元中包含了幾個不同的屬性：1.車輛靜態屬性：在車輛產生時建立，用以描述車輛靜態的特性，並且假設在模擬中是固定不變的項目，如車型、車內設備等；2.駕駛人行為屬性（behavior attributes）：駕駛者行為相關屬性反應出使用者的特性，這些行為屬性將應用於駕駛者路徑選擇的分析上。行為屬性包含了一個人的社經特性，如年齡、性別等；3.執行過程屬性（run-time attributes）：執行中屬性是用於模擬中，登錄並改變駕駛者旅次中的統計資料，可提供事後的分析。

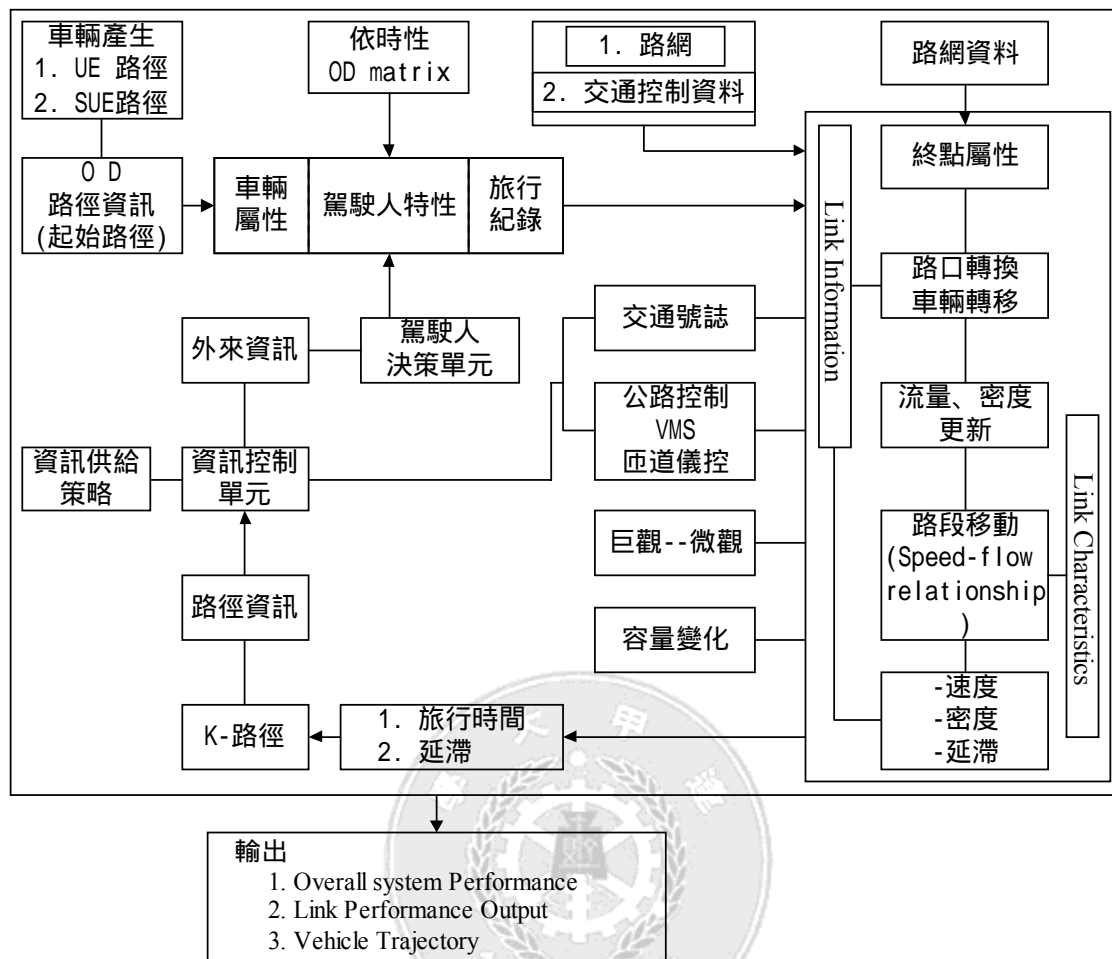


圖 2.7 DYNASMART 功能架構

2.4.2 DYNASMART 之輸入及輸出資料

茲簡單說明各項輸入及輸出資料如下，若欲詳細資料請參考 Hu (1996)。

DYNASMART 主要之輸入資料如下：

1. 路網特性資料

包含了基本資料，如節點數、節線數起迄點數目等，交通分行資料，如各節點對應的交通分區等，路段資料，如車道數、路段長度最大速度及飽和流率等等。

2. 交通需求資料

DYNASMART 所需之需求資料為一依時性的 OD 起迄資料，時間

之間隔與長短使用者可以自行依需要調整。

3. 模擬控制之參數

模擬中有許多控制策略可藉由參數之調整，來改變所模擬出來之各種不同情境，如需求因子、亂數產生、模擬長度及時段、車種百分比等。

4. 路口號誌之控制資料

使用者需對於每個路口不同之號誌情況給予不同控制型態參數，如時制資料、控制資料。

5. 其他相關資料

其他資料如匝道儀控、路段轉向資料、事件或肇事資料以及公車資料，使用者皆可依不同之需要給予資料。

DYNASMART 之主要輸出資料：

1. 總體性統計資料

提供使用者巨觀、總體之資訊，如平均旅行時間、停等時間、旅行距離。

2. 路段屬性資料

提供各路段各模擬時段之車流量、密度、速度之情形，並由這些資料可推估路網中車流之分佈情況。

3. 程式偵錯資料

DYNASMART 提供許多資訊予使用者，以供程式操作時進行偵錯及除錯功能。

第三章 研究方法

本研究主要在模擬式動態交通指派模式之基礎上，以各種邊際成本表示方法，分別做為模擬式動態交通指派模式進行系統最佳化指派時，計算路段邊際成本之方法，因此本章將於 3.1 節中，介紹模擬式動態交通指派模式之觀念性架構，以及系統最佳化指派原則之演算法；3.2 節中將分析模式中邊際成本之計算方法，並探討其正確性對模式可能產生之影響；最後 3.3 節中，根據第二章之回顧與 3.2 節中所得之初步比較結果，說明模式構建之初步構想，以及如何進行各方法之測試比較。

3.1 模擬式動態交通指派模式

模擬式動態交通指派模式（胡大瀛，2001）與其他求解方法最大之不同，在於其以模擬之方式取代傳統路段績效函數之方法，因此可以得到每一時段每一路段上之車流型態以做為依時性計算之基礎，茲就其主要觀念架構說明於 3.1.1；並於 3.1.2 說明本研究主要依據之系統最佳化原則之演算流程。

3.1.1 模擬式動態交通指派模式之觀念性架構

模擬式動態交通指派模式之觀念性架構如圖 3.1，模式中當路網進行動態交通指派時，路網中之車輛係利用依時性 OD 表產生，車輛路徑之產生與指派則是根據各種定義之規則分派路徑，這裡所指之各種規則可以為透過需求分析所產生之個體路徑選擇行為之選擇模式。而主要不同於其他動態交通指派模式的是，路網中之車流型態是以模擬模式 DYNASMART 求得，如路口延滯、路段旅行時間等，取代以往路段績效函數之功能，而由模擬所產生之資料是每個時段每條路段上之流量型態，以供各種路徑指派原則使用，而本研究即是利用此特性所產生之模擬結果，進行各種邊際成本表示方法之計算。

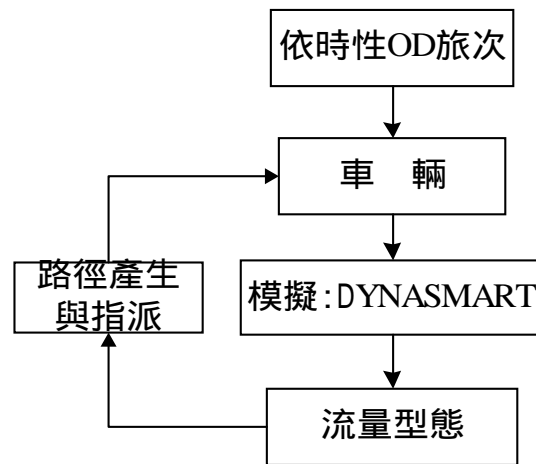


圖 3.1 模擬式動態交通指派之觀念性架構

路徑指派的原則，主要在反應道路使用者的特性，在智慧型運輸系統下，依可能的行為反應，大致有下列四類駕駛人：

1. 使用固定路徑的駕駛者，此種駕駛人不會輕易更換其使用之路徑，如一般通勤者、公車等等。
2. 以本身利益最佳之駕駛者，此類行為之駕駛者僅對本身最佳利益之路徑為考量；
3. 願意接受管理者所提供之規範性資訊，以達到系統最佳化的駕駛者；
4. 判斷道路即時資訊，加上本身駕駛經驗的駕駛者，選擇最佳路徑的駕駛者。

因此在模擬式動態交通指派模式中，可將上述行為以各車種觀念來分類指派原則，包含了系統最佳化、使用者均衡、可獲得即時資訊者與固定路徑使用者。茲說明如下：

1. 系統最佳化指派原則

係指路網中之車輛透過路徑之分配達到整體道路系統之最佳化，也就是這類之車輛是接受資訊中心之建議資訊的導引下進行路徑之選擇，而這 SO 路徑之產生，在模式中是以每個遞子迴（iteration）之「最小成本」路徑為依據，在此最小成本是依據路段之「邊際成本」計算得到。此邊際成本之計算即是本研究之主要研究重心，將詳述於

下一節。

2. 使用者均衡指派原則

此類車輛則是，使用者期望本身之旅行時間最短，故在指派過程所選擇之路徑是依據最短路徑（Shortest Path）作選擇。

3. 可獲得即時資訊之車輛

此類車輛具有車內資訊系統（in-vehicle information system），駕駛人可以依據所收到之道路資訊及本身之偏好進行路徑之選擇（胡大瀛，2001）。

4. 固定路徑之車輛

此類車輛有其固定之路徑，並不受其他外來資訊所影響，在模式之演算過程中，此類車輛被視為背景車輛。

在模擬式動態交通指派模式中，各種類車輛透過此四種不同的移動方式，獲得滿足其本身需求的路徑，而且其彼此路徑的選擇與使用，會造成車輛與車輛間及車輛與系統之間的相互影響，故在蒐尋 SO 與 UE 的路徑時，必須以遞子迴的方式讓系統達穩定狀態。

在本研究之應用上，將先對系統最佳化原則作分析探討，然後加入可獲得即時資訊的車輛，探討即時資訊之供給對系統最佳化的影響。

3.1.2 模擬式 SO 動態交通指派演算法

模擬式動態交通指派模式之演算法主要是根據 Frank Wolfe 演算法，但模擬指派模式是以依時性探討路網、車輛、路徑之關係，因此其特性有以下三點：

1. 依時性之 OD 資料

動態交通指派下之車輛進入路網及路網中移動之方式是具有依時性之行為，故模式中需要假設一依時性之 OD 資料以配合模式中交通依時性資訊之規劃。

2. 路段旅行時間與路口延滯

傳統之交通指派模式是以路段績效函數做為路段中之旅行時間，但動態交通指派之路段旅行時間及路口延滯皆隨著時間或者各種

不同交通策略而改變，故模擬式動態交通指派模式即以模擬之方式替代傳統之路段績效函數。

3.路徑之產生

傳統指派中最短路徑之計算並無依時性之特性，故無法應用於動態交通指派模式中，在此演算過程將依時性最短路徑演算法配合模擬之結果，始可應用於動態交通指派模式。

模擬式動態交通指派模式單一車種之演算流程如圖 3.2，首先讀入 OD 資料值及歷史性之路徑資料，根據此 OD 資料及路徑之資料建立 $XP(O,D,T,K,I)$ 之路徑指派，其中 $XP(O,D,T,K,I)$ 為指派時段 T 由起點 O 迄點 D 於 I 時段使用路徑 K 之車輛數；然後再根據 $XP(O,D,T,K,I)$ 透過 DYNASMART 進行交通模擬，並且紀錄各路段於各模擬時段之旅行時間及相關交通型態，如路段上車輛數及等候車輛數等；經過模擬之後便依照模擬產生之結果進行邊際旅行時間之計算，本研究即是利用模擬產生之結果計算不同邊際成本計算方法之邊際成本值進行比較分析；獲得各模擬時段各路段之邊際成本之後，即可計算最小邊際旅行時間路徑（the least marginal time-dependent path），利用此路徑進行全有或全無之指派產生之結果則是 $YP(O,D,T,K,I)$ ；在產生新的路徑後便需要更新車輛與路徑，檢查路徑是否已在現有路徑集合內，如未包含，則增加此一新路徑，若兩迴圈間車輛於路徑的分配比例維持不變或小於一收斂值，則視為可接受之結果，否則將繼續進行下一迴圈，主要收斂準則如下：

其中圖 3.2 更新路徑方法主要是以 MSA(Method of Successive Averages)作為更新路徑之方法，方法如式 (3-1)，式中各變數之解釋與上述相同，其中 $XP(O,D,T,K,I+1)$ 為更新後路徑上之車輛數：

$$XP(O,D,T,K,I+1) = \frac{1}{I+1} YP(O,D,T,K,I) + (1 - \frac{1}{I+1}) XP(O,D,T,K,I) \quad (3-1)$$

其收斂準則為 $XP(O,D,T,K,I)$ 與 $XP(O,D,T,K,I+1)$ 兩者差之總和小於一給定之違反值視為收斂。

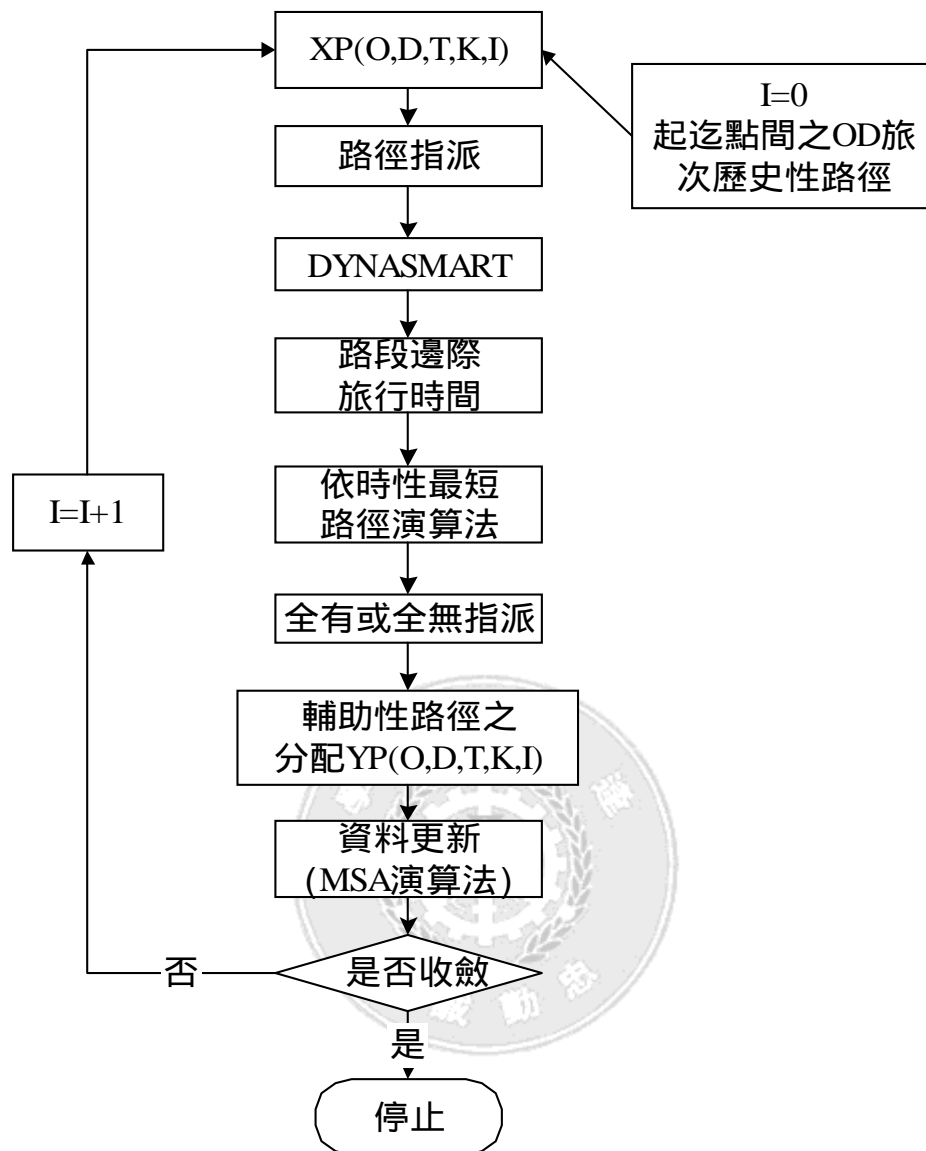


圖 3.2 模擬式 SO 動態交通指派之演算流程圖

3.2 邊際成本對模擬式動態交通指派演算之影響

本節於 3.2.1 將對其中邊際成本之計算作更深入之探討，並於 3.2.2 節中分析其對模式之影響。

3.2.1 模擬式動態交通指派模式中之邊際成本計算

模擬式動態交通指派模式（胡大瀛，2001）中系統最佳化原則所得到之路徑成本是依據路段之「邊際成本」選擇，所謂邊際成本是指一車輛加入路段a造成整體道路系統增加之總成本定義之，故可寫成如式（3-2）：

$$t^a = \frac{dZ(w)}{dw^a}, \forall a \quad (3-2)$$

其中， t^a 表路段 a 之邊際成本， $Z(\cdot)$ 為系統之總成本， w 則是流量向量。因此系統之總成本可以式（3-3）表示，其中 w^c 為路段之流量， $t_c(w)$ 為流量 w 下路段 c 之平均成本。

$$Z(w) = \sum_c w^c t_c(w), c \in A \quad (3-3)$$

式(3-2)之表示為路段之完全邊際成本(global link marginal)，代表了整個系統因為增加了旅行者所增加的成本。因此從(3-2)與(3-3)式可導出式(3-4)：

$$t^a = t_a(w) + w^a \frac{\partial t_a(w)}{\partial w^a} + \sum_b w^b \frac{\partial t_b(w)}{\partial w^a} \quad (3-4)$$

式(3-4)中 t^a 與式（3-2）定義相同， $t_a(w)$ 係指在路段 a 上的旅行時間 $\frac{dt_a(w)}{dw^a}$ 指因為旅行者增加而導致其他使用者的額外成本(時間)， $\frac{dt_b(w)}{dw^a}$ 是因為路段 a 的旅行者，而導致在其他路段 b 上使用者的額外成本。

在求解時如考慮時間因素，式(3-4)將變得更複雜，其運算方法無法以分析式求解，因此僅能以數值估計(approximation)的方式進行。

3.2.2 邊際成本計算對模式之影響

根據 3.1.2 節中 SO 之演算流程，可知式 3-4 邊際成本之計算將扮演一決定性之角色，主要影響模擬式動態交通指派模式在系統最佳化原則下計算車輛路徑之資料及計算最短邊際旅行路徑(Least Marginal Travel Time Paths)，如邊際成本之計算不合理將可能發生下列狀況：

1. 演算過程雖收斂型態一致，收斂速度過慢

也就是說收斂之方向為正確，但由於邊際成本計算方法可能過於複雜或過於簡略導致收斂計算之值收斂過慢，若收斂過慢結果將可能喪失時效性，導致模擬模式不符實際需要之問題。

2. 無法收斂或收斂結果不佳

若邊際成本計算與一般道路使用狀況不合理則將造成結果無法收斂或所收斂之結果與實際狀況相距過大，如此若交通管理者提供道路資訊給駕駛者則將使用路人行駛旅行時間較長之路徑。

因此本研究將主要針對目前各界對系統最佳化原則中「邊際成本」之計算作深入分析探討，在比較各計算方法中之優劣之同時，以一小型路網測試比較其計算之合理性，並擷取各方法中較接近實際道路使用之方法，然後發展一套最符合實際道路系統之模式與演算法，並應用於模擬式動態交通指派模式中，最後進行大型路網之測試。

3.3 邊際成本推估模式之特性與構想

根據第二章對的邊際成本推估方法之回顧與 3.2 節中之分析，本研究將構建之模式，首先應符合以下特性：

1. 符合計算邊際成本之基本定義，也就是「系統增加一輛車，對整體系統所造成的邊際效應」。
2. 符合動態的特性，即所構建之模式，須考慮動態交通指派資料之依時性，如 Peeta 之依時性邊際旅行時間，因為在不同時間進入系統對系統所造成的影響應該是有所差異，故研究中所構建之模式將朝此方向思考。
3. 模式求解的過程與時間應力求其時效性，模式未來將有應用於具有即時性的交通管理系統中之可能性，故須掌握求解的效率。
4. 合理性，模式應該力求能得到接近真正之邊際成本，故應盡量以能求得整體（global）系統之影響為主，而非回顧之計算方法中部分局部（local）求解之方法。

根據上述特性，本研究在模式之初步構想上將朝下列方向思考：

1. 模式將參考各方法之優缺點後擷取其中較佳之觀念及方法；
2. 模式之構建首先需考慮加入後對現有路段之影響；
3. 由於模式未來可能應用於擁擠路網中，因此車輛加入現有路段後對其他路段之影響應該予以考慮；
4. 由於模式未來將於模擬式動態交通指派模式中應用，因此模式之計算求解上，應符合模擬式動態交通指派模式之特性。

以上為本研究模式構建之初步構想，為能真正瞭解各方法的特性以利進行參考，因此本研究將透過一連串的實驗及分析，以得到各方法之優缺點，進而參考得到新的邊際成本計算方式，茲就說明實驗之測試流程如後。

本研究將分成兩階段進行，第一階段就各種邊際成本之計算方法進行深入分析，並且列出其輸入、輸出各項資料，並比較其複雜度，接著對各方法進行「依時性邊際旅行時間」之測試，並分析結果之差異性，以比較各方法所計算出的「最短依時性邊際旅行時間成本路徑」是否一致。

應用各方法於實際路網中，並以模擬式動態交通指派之系統最佳化指派原則演算法為基礎，分析比較各方法之適用與否；然後依據上述之分析及比較結果，構建新的邊際成本計算模式，並以新的邊際成本計算模式，於虛擬路網求得之依時性邊際旅行時間，與旅行時間作比較並觀察模式的合理性。

第二階段，本研究將利用實際路網，比較本研究構建之模式與構建模式時所參考的模式，在模擬式動態交通指派模式中之特性及結果，接著針對各種邊際成本計算方法於模擬式動態交通指派之演算時間效率及收斂特性作比較，如結果合理則停止，否則再回到各種參考模式之測試比較。

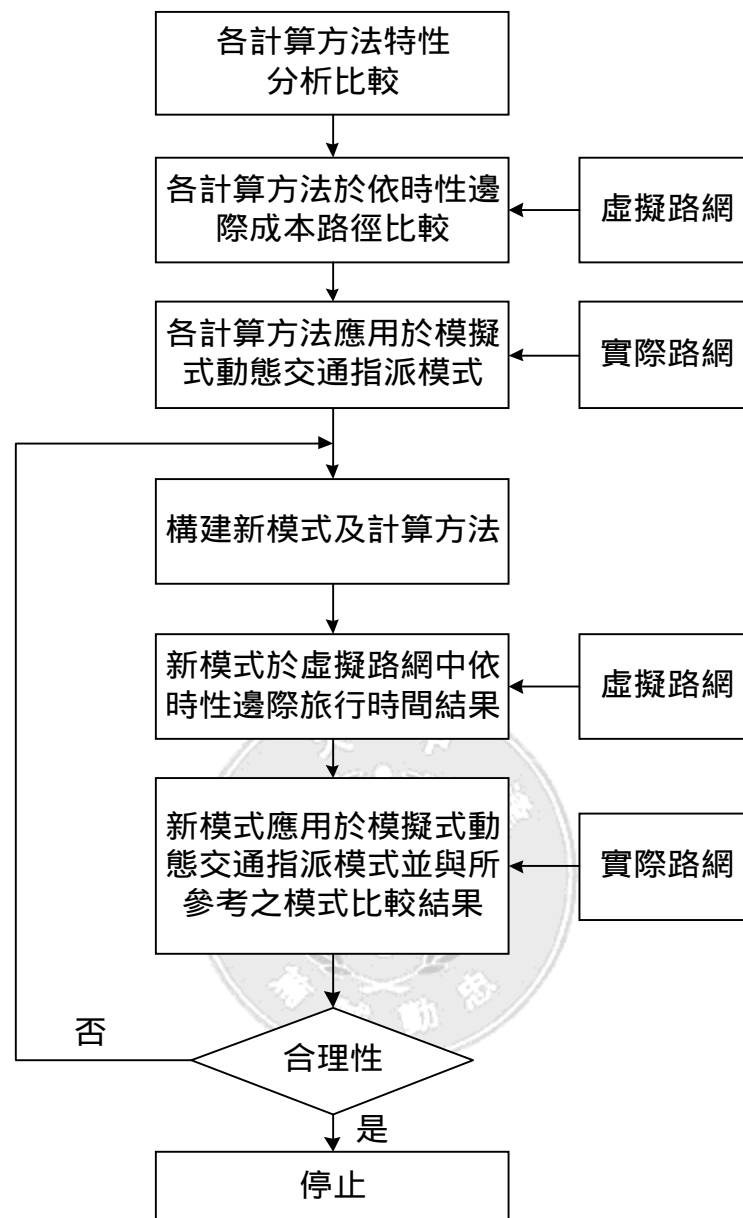


圖 3.3 模式發展與測試流程圖

第四章 邊際成本表示方法比較與分析

依據第三章之實驗流程，本章將針對各種邊際成本表示方法進行分析、測試與比較，4.1 節首先就各種邊際成本之計算方法進行深入分析，並且列出其輸入、輸出各項資料，並比較其複雜度。

4.2 節中利用虛擬路網，對各方法進行「依時性邊際旅行時間」之測試，並分析結果之差異性，以比較各方法所計算出的「最短依時性邊際成本路徑」。

第 4.3 節將應用各方法於實際路網中，並以模擬式動態交通指派之系統最佳化指派原則演算法為基礎，分析比較各方法之特性及結果。

4.1 各表示方法之特性分析與比較

本節之主要在釐清各方法所需要之各項資料，以便得知，各方法應擷取從 DYNASMART 模擬器所產生的若干結果，進行計算。

4.1.1 局部邊際延滯

局部邊際延滯之計算為單一車輛進入路段 l 之時間點 t ，離開路段 l 之時間點 T ，如車輛進入路段 l 之後即停止不動，則其停等之時間 d ，為本身遭遇之延滯，而其影響到別人之時間為 $m-d$ ，故計算局部邊際成本時所須之資料如下：

- (1) 車輛進入路段 l 之時間點 t ；
- (2) 離開路段 l 之時間點 T ；
- (3) 停等之時間 d ；

則依據上述 $T-t=m$ 可計算出 m 。將 m 與 d 相加後即為該車輛加入後產生之延滯。

4.1.2 局部邊際旅行時間

局部邊際旅行時間（魏慶地，1999）將邊際旅行時間分成路段行駛及路口等候兩部分。其中路段行駛乃利用 DYNASMART 中之速率密度關係方程式，將車輛數增加 1，然後再重新計算速率、密度及路段旅行時間，兩者相減後得邊際旅行時間，再乘上車輛數即可得路段之邊際旅行時間。故需要下列輸入資料：

1. 利用速率及密度之關係方程式增加一輛車之額外旅行時間；
2. 路段上之車輛數。

另外在路口等候部分則需車輛等候時間、等候線長度、紓解時間，故可由下列之輸入資料提供：

1. 等候線長度；
2. 紓解率

故將等候線長度除以紓解率便是等候線將紓散之時間。

4.1.3 Peeta 之表示方法

Peeta (1994) 由部分數值實驗發現，路段 a 加入一車輛對於其他路段及其他時段之影響並不明顯，故 Peeta 忽略時間及空間上之影響，因此式 (2-20) 將對其他時段及其他道路之影響忽略而得下式 (4-1)：

$$t^{ta} = T^{ta}(x) + x^{ta} \frac{\partial T^{ta}(x)}{\partial x^{ta}} \quad (4-1)$$

其中 $T^{ta}(x)$ 與 x^{ta} 皆可由 DYNASMART 模擬而得，而在時間點 t ，路段 a 加入車輛之邊際成本， $\frac{\partial T^{ta}(x)}{\partial x^{ta}}$ 則是由三個連續時間點 $t-1$ 、 t 、 $t+1$ 之車輛數為橫軸，路段 a 各時間點車輛數所對應之平均旅行時間為縱軸，如圖 4.1，將該路段之績效函數求出，然後求時間點 t 之切線而得，基於此觀念 Peeta(1994) 以 quadratic fit 之方式求得近似解，其方式與 4.1.3 節中 DYNASMART-X 所使用之方式相同，然此方法並未考量加入車輛之轉向問題。

另外，為考量演算結果之穩定性（stability），Peeta(1994)利用平均（averaging technique）之方法，計算平均的導數（derivative）及車輛數，然後將平均之結果作為式（4-1）中之導數及車輛數。

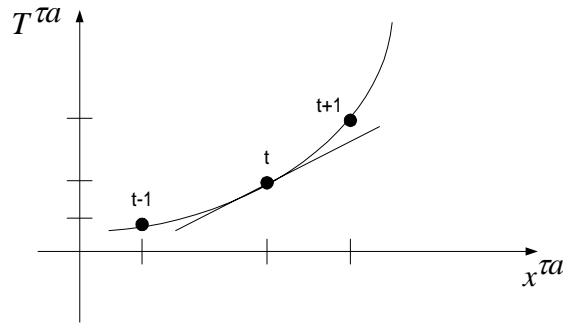


圖 4.1 路段邊際求解微分示意圖

4.1.4 DYNASMART-X

DYNASMART-X 利用三個連續之模擬時間點（ t_{s-1} 、 t_s 、 t_{s+1} ）之旅行時間及車輛數差去近似，將上述之三個模擬時間點分成兩個區段：A、B，其中 A 為時間 $[t_{s-1}, t_s]$ ；B 為時間 $[t_s, t_{s+1}]$ ，則可得到下列各式

$$\Delta TA_s^{ts,a} = T_s^{ts,a}(X) - T_s^{ts-1,a}(X) \quad (4-2)$$

$$\Delta TB_s^{ts,a} = T_s^{ts+1,a}(X) - T_s^{ts,a}(X) \quad (4-3)$$

$$\Delta TA_l^{ts,a} = T_l^{ts,a}(X) - T_l^{ts-1,a}(X) \quad (4-4)$$

$$\Delta TB_l^{ts,a} = T_l^{ts+1,a}(X) - T_l^{ts,a}(X) \quad (4-5)$$

$$\Delta XA_s^{ts,a} = X_s^{ts,a} - X_s^{ts-1,a} \quad (4-6)$$

$$\Delta XB_s^{ts,a} = X_s^{ts+1,a} - X_s^{ts,a} \quad (4-7)$$

$$\Delta XA_l^{ts,a} = X_l^{ts,a} - X_l^{ts-1,a} \quad (4-8)$$

$$\Delta XB_l^{ts,a} = X_l^{ts+1,a} - X_l^{ts,a} \quad (4-9)$$

以上式（4-2）為例，其等號左項之定義，為「加入直行或右轉額外車輛後，在時間區段 A 中，額外車輛增加之旅行時間」，而此額外增加之時間是由兩部分所組成：(a) 受直行右轉之額外車輛， $\Delta XA_s^{ts,a}$ 之影響；(b) 受左轉車輛， $\Delta XA_l^{ts,a}$ 之影響。故式（4-2）之定義將前述之 (a) 及 (b) 相加後便可導出如下式（4-10）：

$$\Delta TA_s^{ts,a} = \Delta XA_s^{ts,a} \times \frac{\partial T_s^{ts}(X)}{\partial X_s^{ts,a}} + \Delta XA_l^{ts,a} \times \frac{\partial T_s^{ts,a}(X)}{\partial X_l^{ts,a}} \quad (4-10)$$

依此類推其餘各式依定義導出如下：

$$\Delta TB_s^{ts,a} = \Delta XB_s^{ts,a} \times \frac{\partial T_s^{ts}(X)}{\partial X_s^{ts,a}} + \Delta XB_l^{ts,a} \times \frac{\partial T_s^{ts,a}(X)}{\partial X_l^{ts,a}} \quad (4-11)$$

$$\Delta TA_l^{ts,a} = \Delta XA_s^{ts,a} \times \frac{\partial T_l^{ts}(X)}{\partial X_s^{ts,a}} + \Delta XA_l^{ts,a} \times \frac{\partial T_l^{ts,a}(X)}{\partial X_l^{ts,a}} \quad (4-12)$$

$$\Delta TB_l^{ts,a} = \Delta XB_s^{ts,a} \times \frac{\partial T_l^{ts}(X)}{\partial X_s^{ts,a}} + \Delta XB_l^{ts,a} \times \frac{\partial T_l^{ts,a}(X)}{\partial X_l^{ts,a}} \quad (4-13)$$

其中，式(4-2)至式(4-9)皆可由 DYNASMART 模擬而得，故式(4-10)中之偏微分，便可由連續之時間區段 A 及 B（及式 4-10 與式 4-11）求聯立方程組而得，則式(2-26)及式(2-27)中之偏微分便可求出。

4.1.5 各表示方法之特性比較

就計算之複雜度而言局部邊際延滯、局部邊際旅行時間及 Peeta 之表示方法，此些方法中未考慮車輛於交叉路口不同轉向對其他車輛之影響，而 DYNASMART-X 中，考量交叉路口車輛之轉向行為應對路段中車輛有所影響，故將其區分為左轉及直行右轉，兩種不同車輛對現有及上游路段影響，因此計算上較為複雜。

再者 DYNASMART-X 考慮「局部整體」(partially global) 計算者，其餘各方法則皆採「局部」(local) 之方法，所謂局部整體係指計算之方法除現有路段之影響尚包含其他路段之影響；而局部乃指其計算範圍忽略對其他路段之影響，僅針對加入之車輛對現有路段之影響，其中 Peeta 雖然在構建模式時，強調車輛之加入將影響不同路段及時段皆會有影響，但最後求解時，忽略了時間及空間之影響，僅針對局部之邊際影響進行求解。

在上述之各方法之輸入值大部分皆可由 DYNASMART 模擬得到，因此本章主要在如何以各方法計算邊際成本整合 DYNASMART 模擬器的程式後進行測試。

今先就各方法之輸入、輸出資料及各計算方法之複雜度整理如表 4.1，其中複雜度其等級定義為：4 表示最為複雜，其次為 3，依次類推。

表 4.1 各計算方法比較表

比較項目 計算方法	輸入	輸出	複雜度
局部邊際延滯 (LMD)(Ghali and Smith, 1995)	1. 車輛進入路段 l 之時間點。 2. 離開路段 l 之時間點。 3. 停等之時間。	1. 單一車輛本身所遭遇之延滯 d 。 2. 車輛造成該路段上其他車輛之延滯。	1 local
局部邊際旅行時間(魏慶地, 1999)	1. 利用速率及密度之關係方程式增加一輛車之額外旅行時間。 2. 路段上之車輛數。 3. 等候線長度。 4. 紓解率。	由左欄之 1、2 項，可求得路段之邊際旅行時間。由 3、4 項則可得到路口之等候邊際旅行時間。	2 local
Peeta 之表示方法 (1994)	1. 同路段三個連續時間點之旅行時間。 2. 各模擬時間區間之車輛數及平均旅行時間。 3. 模擬之區間數。	1. 時間點 t 之切線斜率。 2. 系統模擬時間之平均車輛數及平均導數 (average derivatives)。	3 local
DYNASMART-X	1. 連續兩時間區段加入路段之車輛數。 2. 連續兩個時間區段，額外車輛加入後，其行駛時間之差。	分別得到加入路段之車輛為直行右轉及左轉時之邊際時間成本。	4 partially global

資料來源：本研究整理

4.2 依時性邊際旅行時間之測試

本節將利用一小型虛擬路網如圖 4.2，進行「依時性邊際旅行時間」之計算，比較各方法計算的結果是否一致，以求各方法之正確性比較。4.2.1 節將敘述本研究所使用的路網特性及實驗之基本假設條件；4.2.2 節則說明本研究所使用計算依時性邊際成本路徑之方法；最後 4.2.3 節利用 4.2.2 節產生各比較路徑結果，並在 4.2.4 節中以路徑的旅行時間作合理性的比較。

4.2.1 實驗設計

1. 虛擬路網基本特性及模擬環境

本研究進行之虛擬路網為 9 個節點、24 條節線，路段之距離皆為 800 公尺，路網中節點 1、3、7、9 為無號誌路口，其餘節點為交通控制皆為定時號誌之交叉路口，且每條節線皆為三車道的道路，車道限速 60KPH，如圖 4.3。研究中將以此路網進行「邊際成本」之模擬及各種計算方法之測試，目前虛擬路網之測試乃以虛擬之 OD 進行測試，並假設每一節點皆為一起迄分區。各計算方法之程式以 FORTRAN 程式語言撰寫，並在 COMPAQ Alpha workstation XP1000，記憶體 512MB，中央處理器為 500 MHz Alpha 21264 之工作站進行模擬。

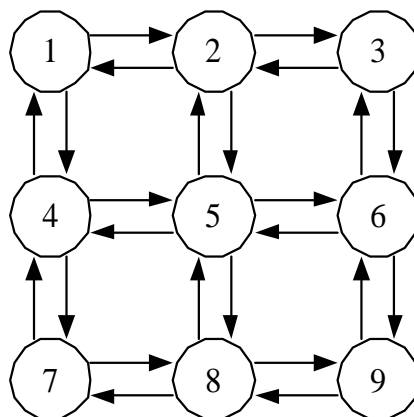


圖 4.2 虛擬路網圖

2 基本假設及條件

(1) 需求因子

實驗中各計算方法彼此獨立，產生在不同需求因子(demand factor)下路網中各路段之邊際成本，所謂需求因子(Demand Factor)係 DYNASMART 在控制路網中車輛數之主要因素，其詳細說明如下：

在一固定交通路網的結構下，車輛數的多寡將會造成不同的交通現象。車輛數越多，交通擁擠的情形越容易發生；車輛數越少，越接近自由流的狀況。透過需求因子的改變，可達到觀察路網容量與車流分佈的情形。

需求因子， f ，可定義為用以改變交通需求量的變數。使用 f 的目的在於分析路網在不同的需求下，可能會有的變化與反應。

$$D = f \times B$$

(4-14)

其中；

D ：考慮之總車輛數，

B ：基本交通需求量，

f ：需求因子。



在實驗中本研究將 f 值設定為：1.0、2.0、3.0、4.0，分別進行測試。

(2) 路徑之依時性邊際旅行時間比較

本研究期望透過虛擬路網中路徑的「依時性邊際旅行時間」比較，分析各種方法所計算的依時性邊際旅行時間的差異，倘若要將各分區之間的路徑一一陳列比較可能過於繁雜，為能清楚且簡易比較各方法進行指派時，所提供的邊際成本之敏感性，本研究選擇虛擬路網中，相隔最遠兩點之起迄對的路徑進行比照：節點 1 出發至節點 9，及從節點 7 出發至節點 3 兩者各 11 條路徑，如表 4.2。

選定比較路徑後，依據 DYNASMART 在不同需求因子下所產生之交通流量型態，進行路徑之依時性邊際旅行時間計算，所謂「依時性邊際旅行時間」，係指在某路徑中的各路段之邊際旅行時間，

是依據車輛到達之時間點所求得，即於不同時間點之車輛加入該路段時的邊際成本，會依時間之變化而改變。

本研究 DYNASMART 模擬器產生的交通特性資料，如流量及流率資料，乃以每一模擬時段（simulation interval）為單位（0.1 分鐘）輸出；研究中的指派時段（assignment interval）為一分鐘，也就是每一分鐘給予每一路段更新的邊際成本值。

另外，局部邊際旅行時間（魏慶地，1999）之計算是利用前三分鐘之車輛到達率推估第四分鐘後所可能產生之到達率，再進而推估邊際成本，故該方法無法得到前三分鐘之依時性邊際旅行時間，因此本研究為求比較之基準，乃統一以第四分鐘之後，所加入該路徑之依時性邊際旅行時間作比較，故本研究中所得到的依時性邊際成本路徑，皆為車輛在第四分鐘後加入路徑時將產生之依時性邊際成本。

表 4.2 比較路徑一覽表

路徑	起點	迄點	依序經過之節點	節點數
1	1	9	2,3,6	5
2	1	9	2,5,6	5
3	1	9	2,5,8	5
4	1	9	4,7,8	5
5	1	9	4,5,8	5
6	1	9	4,5,6	5
7	1	9	2,3,6,5,8	7
8	1	9	2,5,4,7,8	7
9	1	9	4,7,8,5,6	7
10	1	9	4,5,2,3,6	7
11	1	9	2,3,6,5,4,7,8	9
12	7	3	8,9,6	5
13	7	3	8,5,2	5
14	7	3	4,5,6	5
15	7	3	4,1,2	5
16	7	3	4,5,2	5
17	7	3	8,5,6	5
18	7	3	8,9,6,5,2	7
19	7	3	8,5,4,1,2	7
20	7	3	4,1,2,5,6	7
21	7	3	4,5,8,9,6	7
22	7	3	8,9,6,5,4,1,2	9

4.2.2 路徑之依時性邊際旅行時間

本研究依據比較路徑之起迄點，進行其依時性邊際旅行時間之計算，其計算方式如式（4-15）：

$$\begin{aligned} m_p^{t_0} &= m_{p1}(t_0) + m_{p2}(t_0 + t_1) + m_{p3}(t_0 + t_1 + t_2) + \cdots + m_{pn}(t_0 + t_1 + t_2 + t_3 + \cdots + t_{n-1}) \\ &= \sum_{i=1}^n m_{pi} \left(\sum_{k=0}^{i-1} t_k \right) \end{aligned} \quad (4-15)$$

式（4-15）中左手邊， t_0 為車輛加入路徑的時間點， p 為路徑編號，則 $m_p^{t_0}$ 係指在時間點 t_0 ，加入 p 路徑的依時性邊際旅行時間；式（4-15）右手邊， n 為 p 路徑中所有之路段總數， t_k ，則是車輛行駛於 p 路徑上 k 路段的旅行時間， $k=1,2,\dots,n$ ， $\sum_{k=1}^{i-1} t_k$ 為車輛到達路段 i 所使用之時間點，

則 m_{pi} 是該路段 i 於 $\sum_{k=1}^{i-1} t_k$ 時間點邊際成本，則 m_{pi} 之總和即為路徑 p 之依時性邊際旅行時間。

依時性邊際旅行時間，主要擷取車輛到達 k 路段時間點的邊際成本值加以累加，即為一路徑之依時性邊際成本，因此首先利用由 DYNASMART 產生的路段自由車流時間、路段流量及路口紓解率等交通資料，計算依時性旅行時間矩陣，再以各邊際成本計算方法計算得到之路段依時性邊際旅行時間矩陣，然後依式（4-15）計算各路徑的依時性邊際旅行時間，其流程如圖 4.3。

另外，研究中為能有一合理之比較依據，因此將計算出各比較路徑於 t_0 加入路徑 p 時的路徑旅行時間，作為各邊際成本計算方法所計算出的依時性邊際旅行時間之觀察基礎。

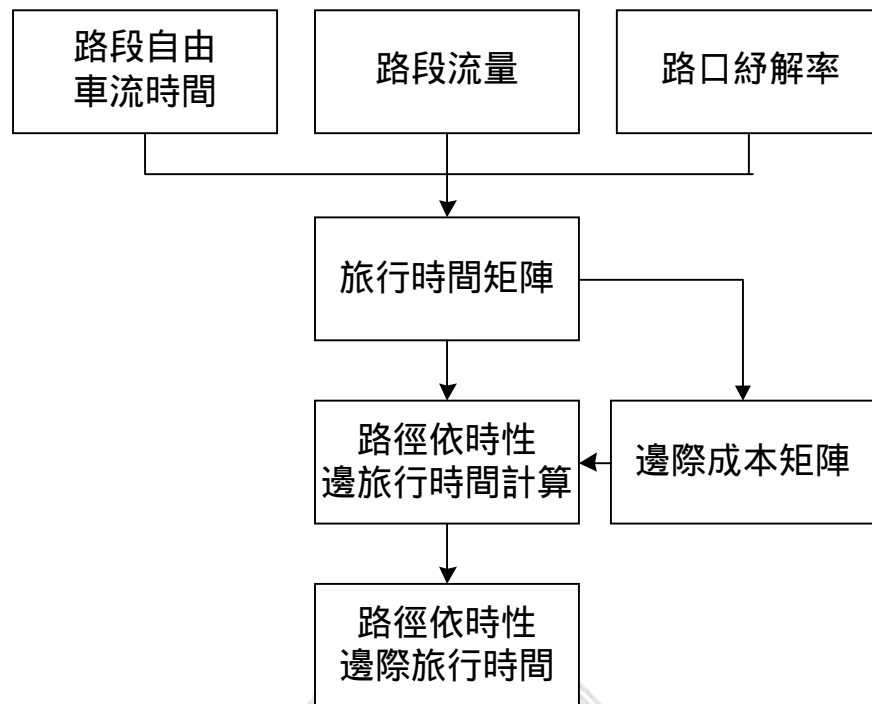


圖 4.3 路徑依時性邊際旅行時間計算流程圖

4.2.3 測試結果

為使各方法有一初步比較，本研究利用不同需求因子控制路網中車輛數，以分別測試不同車輛數下，各路徑利用不同方法所產生之依時性邊際旅行時間，進行最短路徑之比較。

在不同需求因子下，由 DYNASMART 作起始模擬（initial simulation）產生之交通特性如表 4.3 及圖 4.4 所示。表 4.3 中車輛數係指模擬期間產生之車輛數，包含已到達目的地車輛；模擬時間，則是自模擬起始時間至所有車輛皆已到達其目的地之結束時間，通常需求因子愈大路網中車輛的平均旅行時間及平均停等時間愈長，圖 4.4 即為需求因子與平均旅行時間及平均停等時間關係圖，皆呈遞增之現象。平均速度則在需求因子 3.0 情況下，隨需求因子愈大，速度明顯變慢，顯示需求因子較大的路網中，加入路網的車輛，將受路段上其他車輛的影響甚大。

表 4.3 不同需求因子路網特性表

需求因子	車輛數	模擬時間 (分鐘)	平均旅行時間 (分鐘)	平均停等時間 (分鐘)	平均旅行距離 (公里)	平均速度 (公里/時)
1.0	4,787	38.8	2.16	0.24	0.91	40.44
2.0	9,576	38.8	2.55	0.49	0.96	36.14
3.0	14,407	67.6	5.72	2.54	1.01	16.95
4.0	19,175	73.6	11.12	4.11	1.02	8.81

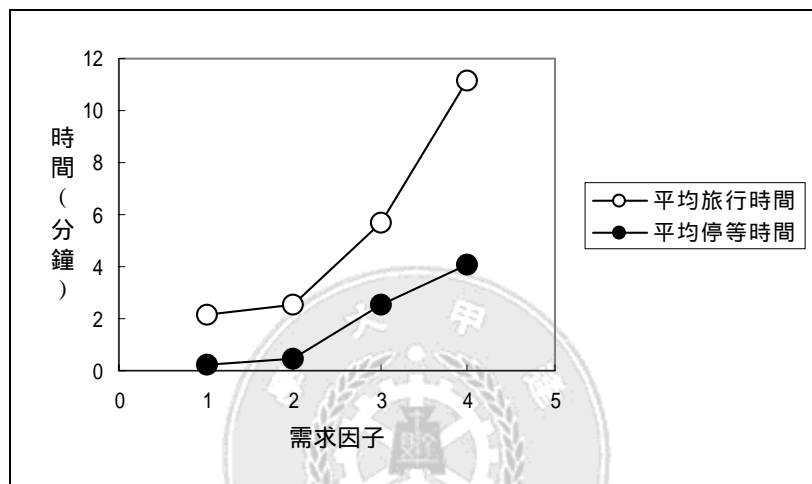


圖 4.4 需求因子與平均停等、旅行時間關係圖

若車輛於第四分鐘加入各路徑，則在不同需求因子下之依時性的路徑旅行時間，如表 4.4，最左欄為路徑編號，右手邊四欄則分別為需求因子 1.0、2.0、3.0、4.0 下之路徑旅行時間，由表 4.4 可看出需求因子愈大，各路徑之旅行時間成本愈大，這隱含該路徑已有產生交通擁擠狀況，因此在依時性邊際旅行時間之計算中將會反應此種情況。茲就得到之數值結果作最短路徑及旅行時間敏感度兩部分之比較及分析：

1.比較各方法得到之路徑依時性邊際旅行時間

各路徑利用四種方法於不同需求因子所計算出來的依時性邊際旅行時間，結果分別如表 4.5~表 4.8 所示，其中表 4.5 為需求因子 1.0 各路徑在第四分鐘的依時性邊際成本，起點為節點 1 迄點為節點 9（路

徑 1~路徑 11) 的比較路徑中，DYNASMART-X 的結果路徑 1 時間最短，Peeta 之表示方法結果為路徑 2 最短，局部邊際延滯及局部邊際旅行時間所得到之結果則路徑 2 及路徑 3 最短。

其他需求因子 2.0、3.0、4.0 所得到之結果如表 4.6~表 4.8，綜合表 4.5 至表 4.8 之數值結果，可以得到各方法之最短路徑結果並無太大差異。

2.比較各方法對路徑旅行時間之敏感度

為能明確看出各方法對擁擠路徑之敏感度，綜合比較表 4.5~4.8，各方法在不同需求因子下各路徑得到的依時性邊際旅行時間的結果可知，Peeta 之表示方法與 DYNASMART-X 之方法，對於擁擠路徑之反應較穩定，其餘兩者則較大變化；另外在數值大小上，局部邊際延滯及局部邊際旅行時間的結果稍大。

為使各方法能合理且正確地比較，本研究進一步以各路徑之旅行時間作為較基礎，與各邊際成本表示方法所得到的依時性邊際旅行時間，進行比較，如 4.2.4 節。

表 4.4 不同需求因子下第四分鐘加入各路徑之旅行時間表

需求因子 路徑	1.0	2.0	3.0	4.0
1	3.45	3.85	4.32	4.82
2	3.45	3.68	4.53	4.36
3	3.43	3.81	4.45	4.63
4	3.50	4.56	8.01	8.17
5	3.54	4.25	6.22	6.19
6	3.49	4.20	6.25	5.88
7	5.28	6.20	7.14	9.44
8	5.29	6.06	8.79	9.18
9	5.38	6.74	11.59	12.28
10	5.52	6.31	8.94	10.62
11	7.09	8.31	10.78	17.00
12	3.77	5.04	7.49	12.20
13	3.54	4.83	6.92	9.53
14	3.68	4.38	5.26	5.89
15	3.59	4.38	5.19	6.50
16	3.64	4.21	5.10	5.75
17	3.65	5.00	6.93	9.58
18	5.47	7.32	10.40	20.22
19	5.44	7.36	10.02	15.74
20	5.39	6.76	7.55	9.50
21	5.40	6.83	9.22	12.23
22	7.59	9.55	14.61	29.95

(單位：分鐘)

表 4.5 需求因子 1.0 第四分鐘加入各路徑依時性邊際旅行時間表

方法 路徑	DYNASMART-X	Peeta 之 表示方法	局部 邊際延滯	局部邊際旅 行時間
1	3.31	3.54	99.41	30.34
2	4.90	3.46	6.48	26.82
3	4.90	3.55	8.88	29.27
4	4.87	3.54	6.83	20.15
5	5.81	3.56	9.13	28.10
6	5.81	3.47	6.74	25.64
7	6.54	5.36	104.91	68.95
8	7.29	5.40	11.57	35.56
9	8.16	5.28	9.64	33.09
10	7.49	5.31	101.95	39.75
11	8.93	7.22	107.60	75.24
12	4.88	3.62	8.50	31.69
13	5.71	3.64	96.17	24.66
14	5.02	3.54	7.97	25.99
15	3.37	3.71	102.97	31.70
16	5.08	3.57	98.16	27.21
17	5.66	3.61	5.98	23.45
18	8.17	5.39	101.79	69.07
19	7.34	5.65	105.98	43.39
20	6.58	5.45	15.07	41.07
21	7.53	5.37	15.70	49.62
22	9.79	7.40	111.60	87.80

(單位：分鐘)

表 4.6 需求因子 2.0 第四分鐘加入各路徑依時性邊際旅行時間表

方法 路徑	DYNASMART-X	Peeta 之 表示方法	局部 邊際延滯	局部邊際旅 行時間
1	3.39	3.63	18.26	56.85
2	5.01	3.70	15.53	33.10
3	5.19	3.75	19.59	25.14
4	5.12	7.07	28.72	32.04
5	6.39	4.04	24.70	24.50
6	6.22	3.99	20.64	32.46
7	6.85	5.77	26.89	69.05
8	7.55	9.17	29.66	57.84
9	8.47	9.07	28.49	49.06
10	7.98	5.85	28.05	77.81
11	9.21	11.18	36.96	101.76
12	5.28	8.21	25.24	27.46
13	5.89	7.43	15.65	67.52
14	5.42	4.62	10.84	29.98
15	3.59	4.07	14.82	71.11
16	5.55	4.54	14.35	67.79
17	5.76	7.51	12.14	29.72
18	8.69	10.21	33.31	85.43
19	7.57	9.34	22.18	96.02
20	6.85	6.08	15.99	54.89
21	8.46	7.38	27.76	36.78
22	10.37	12.12	39.84	113.92

(單位：分鐘)

表 4.7 需求因子 3.0 第四分鐘加入各路徑依時性邊際旅行時間表

方法 路徑	DYNASMART-X	Peeta 之 表示方法	局部 邊際延滯	局部邊際旅 行時間
1	3.34	4.30	17.14	152.84
2	6.00	4.21	7.30	21.41
3	6.35	4.25	7.01	28.50
4	5.24	21.10	25.88	54.62
5	6.86	4.86	17.74	50.24
6	6.51	4.82	18.03	43.16
7	6.99	6.53	21.07	173.15
8	8.65	23.77	22.10	55.47
9	8.64	24.51	28.99	73.02
10	8.54	9.40	31.86	230.75
11	9.30	26.05	36.16	200.12
12	5.40	20.25	15.73	43.01
13	6.24	25.16	11.06	214.60
14	5.60	5.02	17.85	40.05
15	3.56	4.23	3.78	166.11
16	5.97	7.51	18.19	209.21
17	5.88	22.67	9.06	45.44
18	9.06	24.94	5.49	225.39
19	7.75	25.17	4.96	194.09
20	7.88	6.23	10.29	53.11
21	8.76	5.99	22.12	59.28
22	10.58	25.10	15.73	95.15

(單位：分鐘)

表 4.8 需求因子 4.0 第四分鐘加入各路徑依時性邊際旅行時間表

方法 路徑	DYNASMART-X	Peeta 之 表示方法	局部 邊際延滯	局部邊際旅 行時間
1	3.63	4.49	11.02	32.06
2	5.43	4.70	10.49	31.68
3	5.20	4.94	10.02	29.47
4	5.41	19.85	24.27	58.49
5	7.03	21.95	15.28	47.18
6	7.27	21.70	15.75	49.39
7	6.78	5.84	17.86	50.13
8	8.27	5.70	22.49	60.97
9	8.99	24.01	30.29	96.10
10	9.40	22.75	21.69	67.59
11	9.88	6.56	29.67	83.78
12	5.76	11.46	27.33	67.83
13	6.49	4.04	20.18	85.31
14	6.11	7.37	12.45	44.27
15	3.89	6.71	16.31	46.31
16	6.55	6.16	12.92	45.22
17	6.05	5.25	19.71	84.36
18	10.30	11.35	32.55	84.60
19	8.54	7.51	26.30	107.53
20	7.38	9.18	21.24	62.89
21	9.23	17.94	25.03	68.66
22	11.35	13.47	49.58	127.07

(單位：分鐘)

4.2.4 各方法合理性之比較

本研究根據 4.2.2 節的路徑依時性邊際旅行時間計算方法，得到各比較路徑於不同時間點，加入車輛的依時性邊際旅行時間與旅行時間，由於各方法產生各路徑的依時性邊際旅行時間結果，型態類似，因此本研究僅就其中一條路徑，進行說明各方法的差異性。

根據各比較路徑的結果，本研究挑選第七條路徑作說明，選擇此路徑之原因，乃因其結果可明顯看出各方法的差異性。該路徑所得到的不同時間點加入路徑之旅行時間結果如圖 4.5，由圖 4.5 可看出當路網中所使用之需求因子愈大（車輛數愈多），則其旅行時間愈高。

再者，將路徑 7 以各邊際成本表示方法於需求因子分別為 1.0、2.0、3.0、4.0 等四種不同車輛數情況下，所得到之「依時性邊際旅行時間」，與該路徑的旅行時間比較，結果如圖 4.6~4.9。圖 4.6 至 4.9 之橫軸代表車輛加入路徑 7 的時間點，縱軸則是表示旅行時間的值(含旅行時間以及依時性邊際旅行時間)。

由圖 4.6 即可發現各需求因子 1.0 情況下，在模擬時間(simulation time)內，Peeta 的表示方法與 DYNASMART 的方法之依時性邊際旅行時間變化情況，與旅行時間比較之結果，較局部邊際延滯與局部邊際旅行時間的變化來得合理，也就是說當路徑之旅行時間愈高時，路徑依時性邊際旅行時間亦隨路徑旅行時間變高，但由圖 4.6 中發現，後兩者(局部邊際延滯與局部邊際旅行時間)的方法顯然有較大的差異性，因此在需求因子 2.0、3.0 及 4.0 將後兩者的比較省略。

在需求因子(2.0、3.0、4.0)愈大的情況下，DYNASMART-X 與 Peeta 的方法所得之依時性邊際旅行時間，亦開始在旅行時間較大的時間點有較大的變化，局部邊際延滯與局部邊際旅行時間則仍有較敏感的反應。因此可推論 DYNASMART-X 與 Peeta 的方法，對於邊際成本的反應較其餘兩者合理。

由於本研究希望所構建之模式，未來可實際應用於動態交通指派模式中，因此本研究於 4.3 節中，將各方法實際應用於模擬式動態交通指派系統最佳化指派原則演算法中，以測試各方法的特性，以利構建新模式之參考。

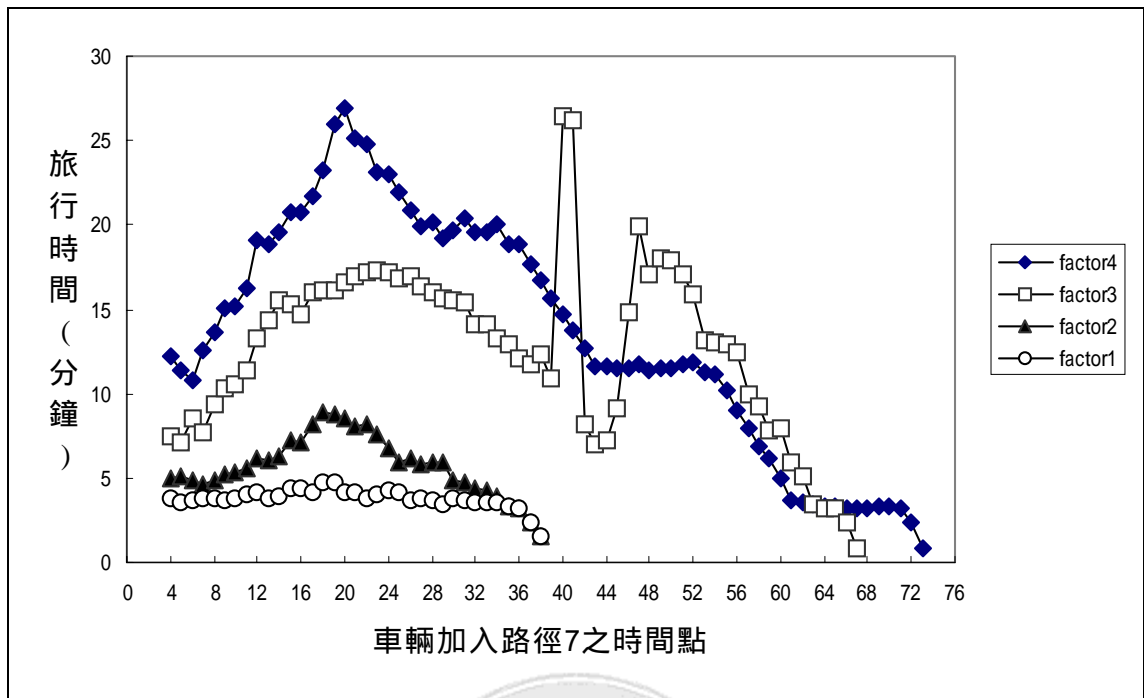


圖 4.5 不同需求因子及時間點加入路徑 7 之旅行時間圖

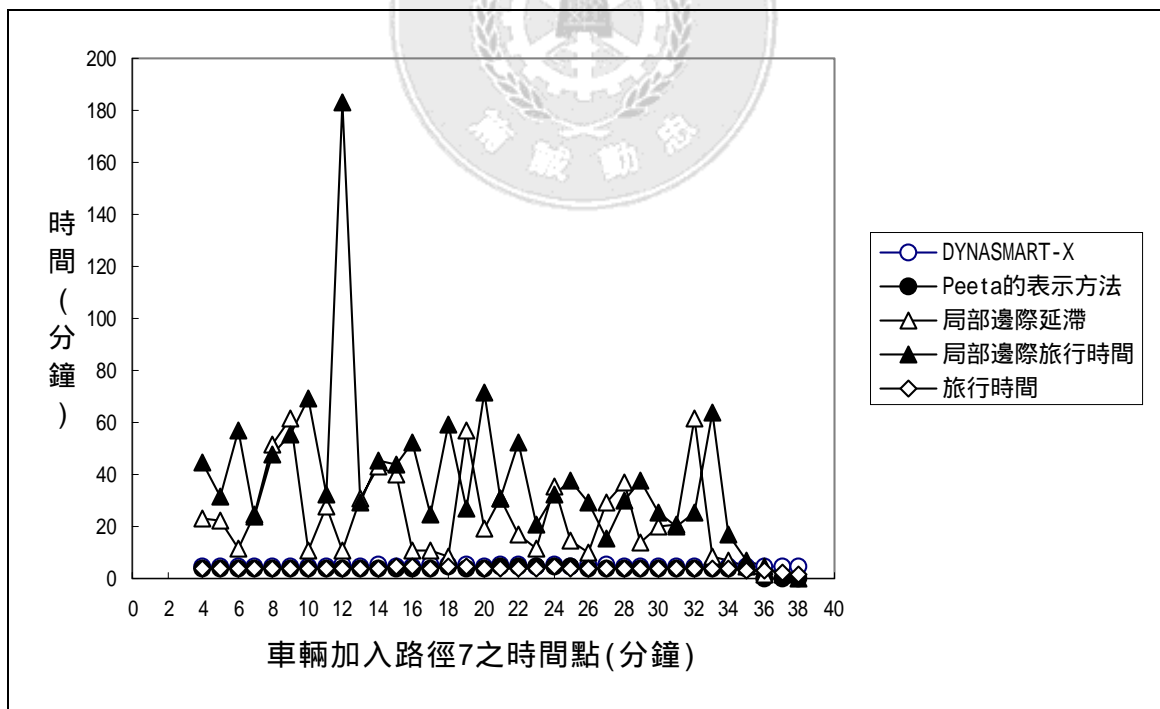


圖 4.6 路徑 7 需求因子 1.0 之依時性邊際旅行時間

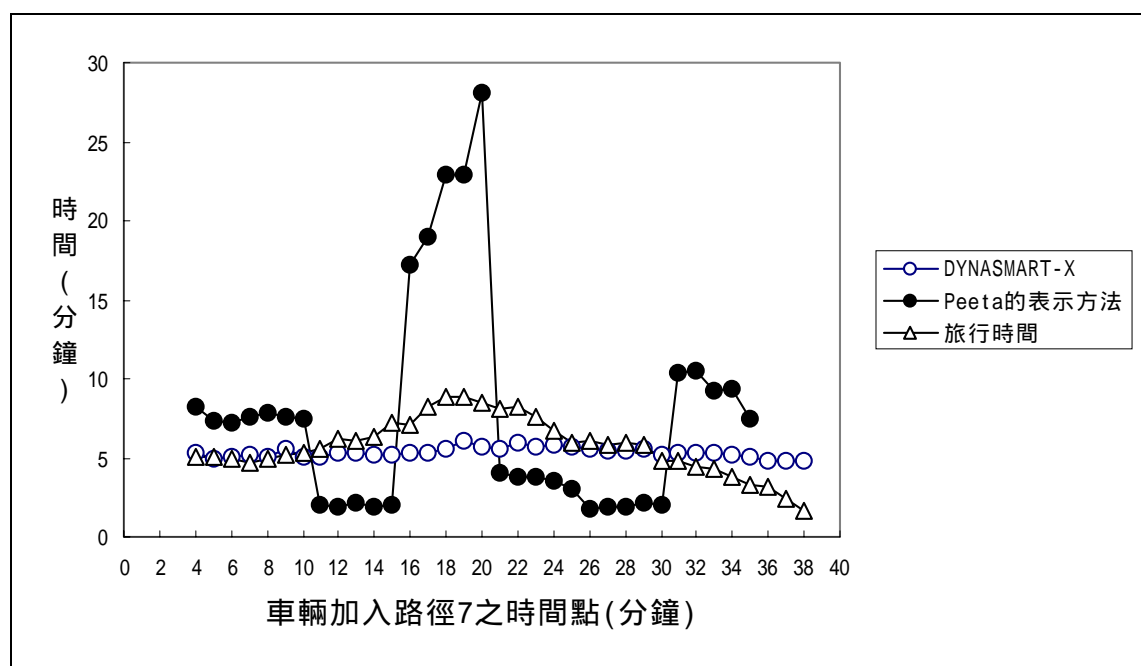


圖 4.7 路徑 7 需求因子 2.0 之依時性邊際旅行時間

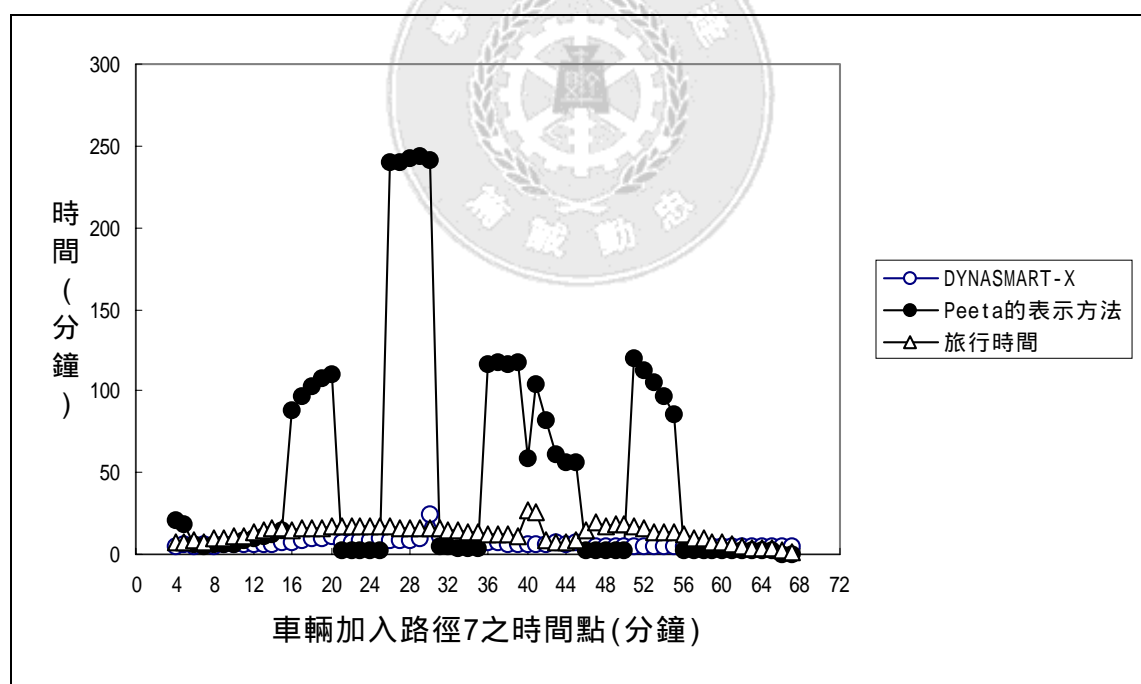


圖 4.8 路徑 7 需求因子 3.0 之依時性邊際旅行時間

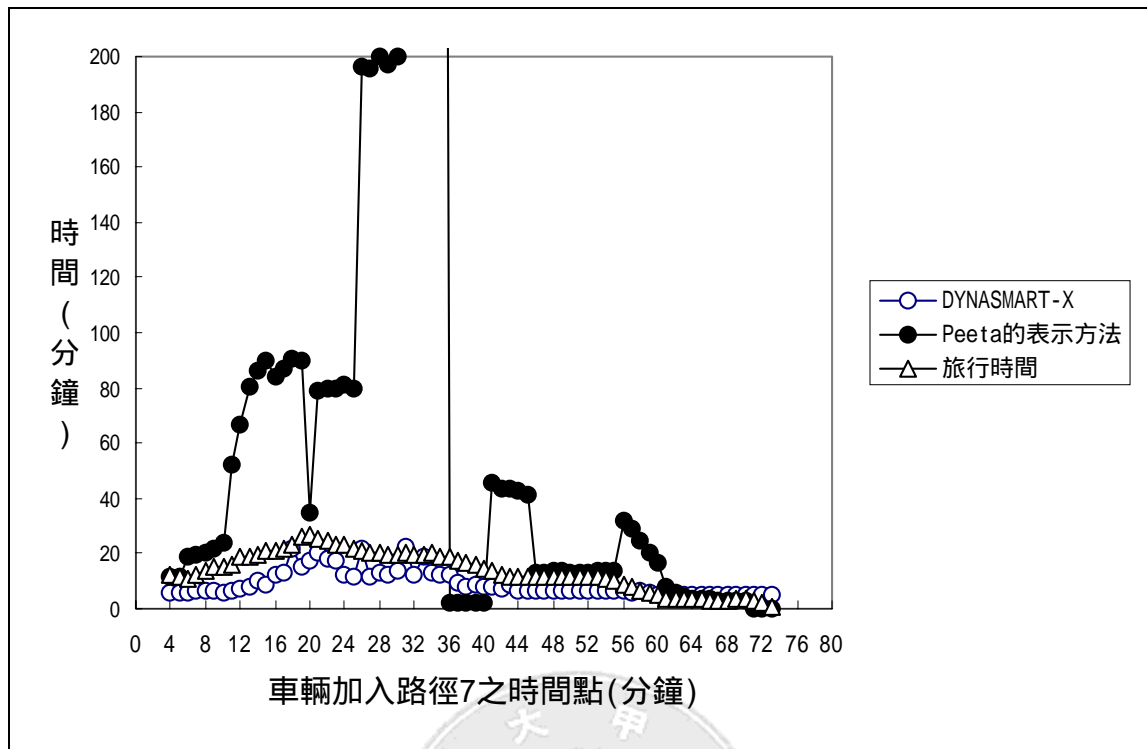


圖 4.9 路徑 7 需求因子 4.0 之依時性邊際旅行時間

4.3 模擬式動態交通指派模式之測試

根據 4.2 節之初步分析比較後，為再深入探討各邊際成本表示方法的特性，本節將針對上述各種方法，於模擬式動態交通指派模式系統最佳化原則演算法作測試，主要測試其是否適用於演算法中。

4.3.1 測試流程

在此 DTA 之主要架構流程，係利用交通模擬工具模擬路網中的基本的道路流量型態，根據道路中之交通狀況計算路段邊際成本，將路段之邊際成本當成路段之懲罰值（link penalties）選擇最小的邊際成本，進行所謂的依時性最小邊際成本路徑的計算（Ziliaskopoulos and Mahmassani, 1992a, 1992b），經過邊際成本之計算後，便依據邊際成本最小的路徑進行交通指派，然後再根據指派過後之路網再重新模擬路網中車流之型態，如圖 4.10，如此不停循環，直到達系統最佳化之

目標。

一個較佳之邊際成本計算方法，將會影響上述之指派模式收斂速度（或稱達系統最佳化目標），故本節將進行所有方法進行測試與比較分析，在模擬式動態交通指派模式之系統最佳化演算基礎下，比較何種方法之收斂特性較為合理。

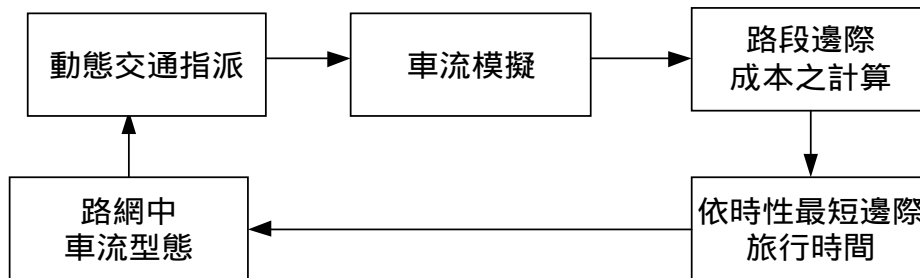


圖 4.10 動態交通指派模式示意圖

4.3.2 實驗路網及條件

本研究以一實際之都市交通路網進行數值實驗，如圖 4.11，希望瞭解各種計算方法之計算方法在大型路網下之應用與特性。此一路網為德州達拉斯附近的都市路網，其中包括 178 節點、441 節線、與 13 個交通分區，無號誌路口有 62，停等標誌有 55，定時控制號誌有 61 個，此外尚有 8 個進口匝道(on-ramp)儀控。模擬環境與第四章之測試環境相同，為 COMPAQ Alpha workstation XP1000。

本研究採用之需求因子為 1.00，車輛數為 6,136 輛，主要目的在測試各模式所產生之結果是否合理且適用，以及應用之特性。

4.3.3 各方法實際路網測試結果

各方法應用於模擬式 SO 動態交通指派之結果每趟子迴之平均旅行時間如表 4.9、圖 4.12，由結果圖 4.12 可以看出需求因子 1.00 局部邊際延滯及局部邊際旅行時間兩者之結果與其他方法差異甚大，可能原因為局部邊際延滯的原本使用上，其路徑與車輛更新並不需要以 MSA 的方法更新，因此應用於模擬式動態交通指派模式演算法導致產生了 MSA 蒐尋最佳解的方向錯誤；另外局部邊際旅行時間其推估方法需利用前三分鐘的累積車輛到達曲線，推估到達曲線與離開曲線交點，因此在推估上產生誤差而導致邊際成本計算誤差。

其餘兩者 DYNASMART-X 與 Peeta 的表示方法之結果皆較為合理；但 Peeta 的表示方法所產生之平均旅行時間結果又較 DYNASMART-X 稍高，故可以驗證，邊際成本計算之考慮愈完整，則計算結果愈符合實際狀況之邊際成本，故應用於模擬式動態交通指派模式的系統最佳化原則，便產生愈佳之結果（即整體路網車輛平均旅行時間愈低）。

因此根據 4.2 節與本節測試的結果，本研究將於第五章中利用各方法測試過程的特性構建模式。

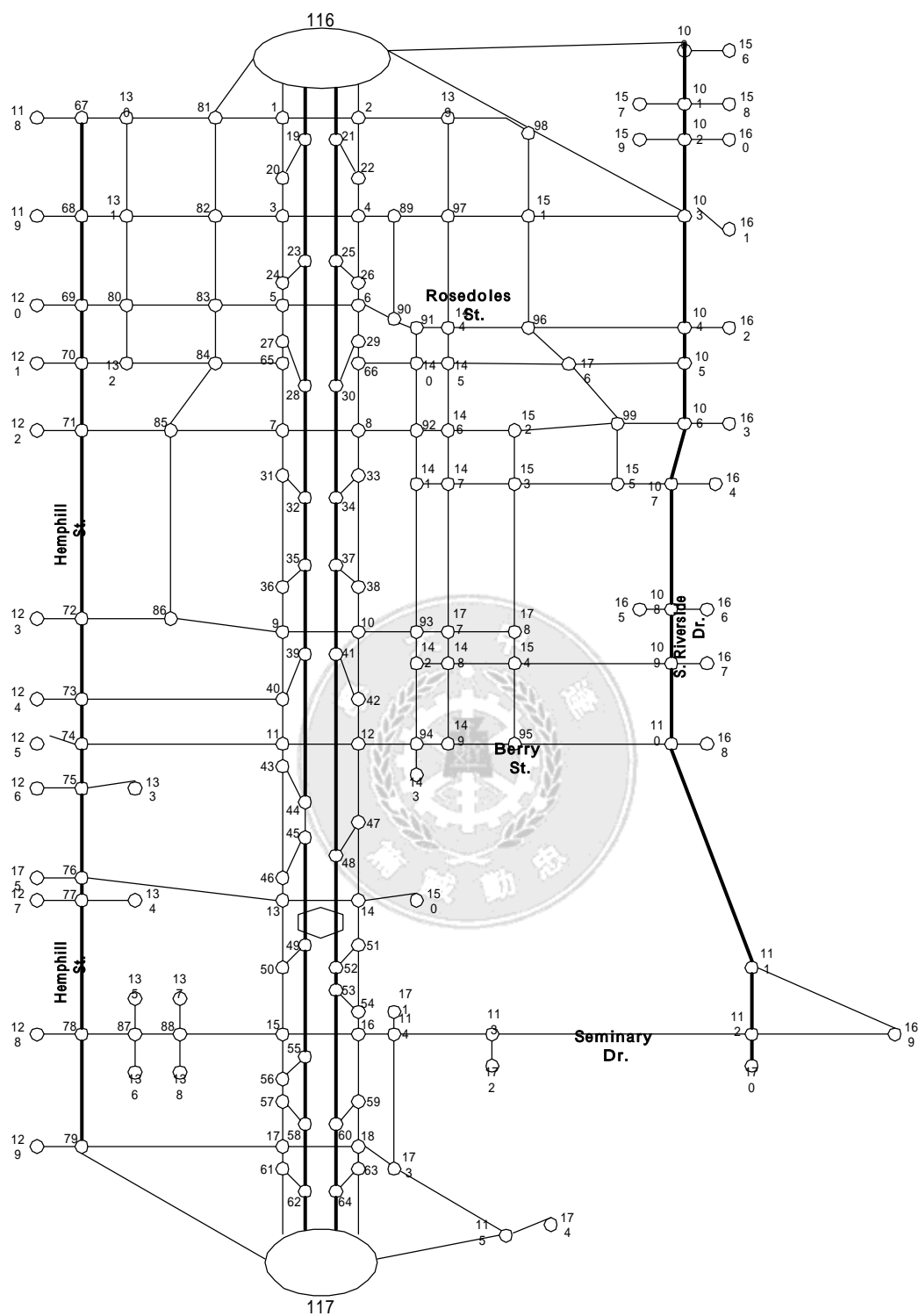


圖 4.11 實際路網圖

表 4.9 需求因子 1.00 各方法每遞子迴之平均旅行時間

方法 遞子迴	DYNASMART-X	Peeta 之 表示方法	局部邊際 旅行時間	局部 邊際延滯
0	9.02	9.02	9.02	9.02
1	7.46	7.58	9.62	9.25
2	7.36	7.58	11.02	10.44
3	7.27	7.51	13.06	12.30
4	7.27	7.49		
5	7.40	7.50		
6	7.31	7.67		
7	7.29	7.45		
8	7.27	7.48		
9	7.27	7.48		
10	7.26	7.45		
11	7.28	7.49		
12	7.34	7.38		
13	7.31	7.52		
14	7.24	7.48		
15	7.27	7.40		
16	7.24	7.40		
17	7.28	7.39		
18	7.32	7.38		
19	7.31	7.40		
20	7.30	7.54		
21	7.36	7.41		

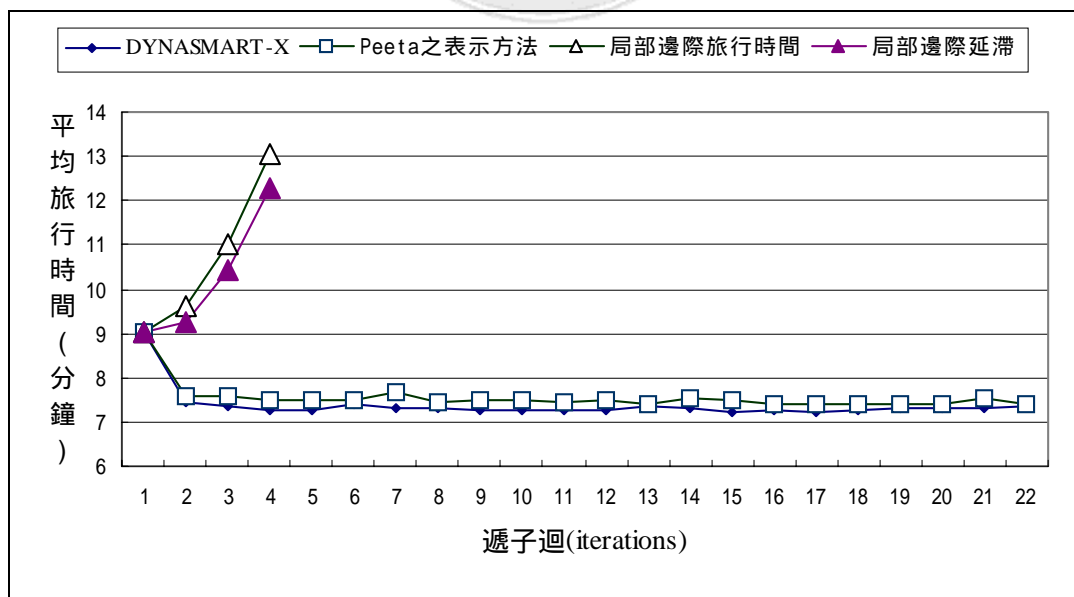


圖 4.12 需求因子 1.0 各表示方法每遞子迴之平均旅行時間

4.4 小結

本節主要分析及測試各計算方法合理性，及各模式於模擬式動態交通指派模式中的適用性。首先就各模式之求解進行分析探討之結果，發現 DYNASMART-X 的方法最為複雜，最單純者為局部邊際延滯。

路徑依時性邊際旅行時間之測試結果，以第四分鐘加入各比較路徑車輛的依時性邊際旅行時間比較時，發現各方法得到所有比較路徑中最短之結果差距不大，但發現局部邊際延滯及局部邊際旅行時間之值偏高。

然而為進一步比較各模式的合理性，研究中再以不同時間點，加入比較路徑的旅行時間，與研究中所計算的依時性邊際旅行時間作比較，結果發現，局部邊際延滯與局部邊際旅行時間的結果與旅行時間之變化有較大的誤差。

最後以各模式應用於模擬式動態交通指派模式中，觀察各模式於模擬式動態交通指派模式之適用性，結果發現局部邊際延滯與局部邊際旅行時間的測試結果有所誤差，原因可能為兩者的求解方法於模擬式動態交通指派模式的方法無法得到較佳之結果，其餘兩者之結果則較合理。

第五章 模式構建及測試

依據第四章 4.2 節與 4.3 節之測試結果，5.1 節將進行構建邊際成本模式，以提供有效率且精確的方法，並以此新模式，進行路徑之依時性邊際旅行時間的測試以觀察模式之正確性；5.2 節實際應用新模式於模擬式動態交通指派模式，並與構建時所參考的計算模式比較分析；5.3 節說明實驗之收斂特性；5.4 節則與其他模式在演算法上之運算效率作比較；5.5 節中以實例說明，未來如何應用本研究構建之邊際成本模式，結合模擬式動態交通指派系統最佳化原則演算法（以下稱「模擬式 SO 動態交通指派模式」）於實例中。

5.1 模式之構建與路徑依時性邊際旅行時間測試

本節將根據第四章的比較結果，擷取各表示方法之優點進行構建模式，並作路徑依時性邊際旅行時間與路徑旅行時間之比較。

5.1.1 模式之構建

本研究依據 4.3、4.4 節中各方法比較分析之後，再依實驗過程之經驗與實驗結果構建一模式，此模式主要參考 Peeta 的表示方法與 DYNASMART-X 兩者的觀念構建模式。茲說明構建此模式之主要方向。

Peeta 的表示方法，忽略了車輛加入現有路段後對其上游之影響，而 DYNASMART-X 中之現有路段影響分左右轉車輛之計算方式又過於複雜，但在擁擠路網中，現有路段上加入車輛對於鄰近(neighboring)路段應該有一定程度之影響，因此對上游路段影響之構想的確有其必要性，然而如要對於整體(global)系統之邊際影響計算可能過於複雜，因此模式考慮現有路段加入車輛對上游路段之影響，也就是將現有路段上所剩餘可容納上游鄰近路段之影響納入考慮。

故本研究主要參考 Peeta 表示方法中之現有路段影響及 DYNASMART-X 之上游路段影響之觀念構建模式，如式 (5-1)：

$$\tau^{ta} = \left\{ T^{ta}(x) + d^{ta} \frac{\partial T^{ta}}{\partial x^{ta}} \right\} + \left\{ \frac{\partial T^{ta}}{\partial x^{ta}} \times \max\{0, (A - B)\} \right\} \quad (5-1)$$

() ()

其中，

τ^{ta} ：係指在時間點 t ，車輛加入現有路段 a ，系統增加之旅行時間；

T^{ta} ：車輛於時間點 t ，加入路段 a 之旅行時間；

d^{ta} 、 x^{ta} ：時間點 t 路段 a 上之車輛數；

A ：上游路段加入現有路段 a 之車輛數；

B ：現有路段上剩餘之容量。

式 (5-1) 中之第 部分為加入現有路段之車輛，對現有路段 a 上之影響，其中之以 d^{ta} 、 x^{ta} 分別皆表示車輛數，原因為，在估算此部分時， d^{ta} 是採用平均車輛數之方式計算，故以不同符號區分。

第 部分則是參考 DYNASMART-X 中對於上游路段之影響，但本研究將其左右轉之區別忽略，僅考量加入之總車輛數；如圖 5.1 當 A 空間大於 B 空間時，對上游路段之影響即是兩者之差再乘上現有路段之邊際成本，也就是說當 A 小於 B 時，對上游路之影響便不存在，即第 部分值為零。

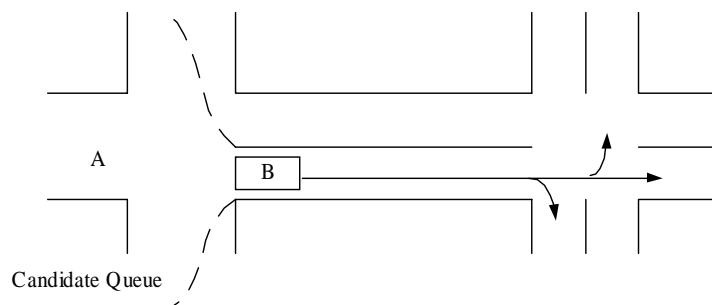


圖 5.1 鄰近路段邊際影響示意圖

模式之求解方法上，在第 部分雖根據 Peeta 的之求解觀念，但本研究在求解第 部分之偏微分部分，改以內插函數之求解方法，其主要方法為：取得三點不同車輛數及其旅行時間之後，進行計算得到

路段車輛與旅行時間之函數式 (5-2)，並作一次微分後得式 (5-3)，如圖 5.2。

$$f(x) = bx^2 + cx + d \quad (5-2)$$

$$f'(x) = 2bx + c \quad (5-3)$$

最後將時間點 t ，路段 a 上之車輛數 x^{ta} 代入，求得時間點 t ，該函數之切線斜率，則此切線斜率即為式 (5-1) 中偏微分之部分；第二部分則是根據 DYNASMART-X 中之計算方法，但不同於 DYNASMART-X 則是本研究並未考慮加入車輛之轉向，僅考慮上游路段加入現有路段之總車輛數。

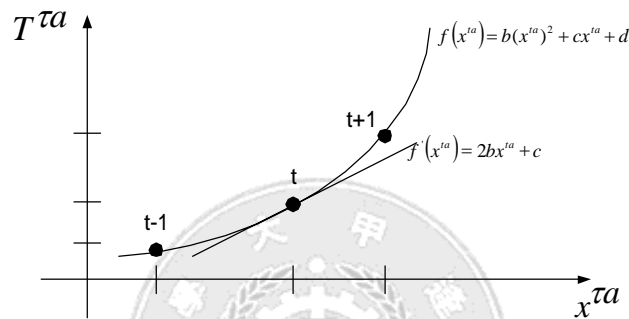


圖 5.2 求解切線示意圖

5.1.2 路徑依時性邊際旅行時間測試結果

為確認本研究所構建之模式的正確性及合理性，以本研究構建之模式，利用簡易路網 (圖 4.2)，計算各路徑不同需求因子下之依時性邊際旅行時間，在此，同樣選擇表 4.2 的路徑 7 作為說明。路徑 7 不同需求因子下，得到之依時性邊際旅行時間之結果，如圖 5.3~圖 5.6。

圖 5.3 及圖 5.4 分別為需求因子 1.0 及 2.0 的情況，其結果與 4.2.4 節中 Peeta 的表示方法所產生之結果差距不大，可能是路網中之車輛尚未產生擁擠狀況，而使本研究所構建模式中對其他鄰近路段影響未產生其應有的效果。

圖 5.5、及圖 5.6 則為需求因子 3.0 與 4.0 之結果，本研究模式產生之結果與 4.2.4 節中 Peeta 的結果開始有所差異；另外綜合四個圖形結果，發現在旅行時間最高的情況下，本研究所構建的模式，產生之依時性邊際旅行時間，亦反應此現象，故本研究所構建之模式，應可

合理及正確計算路徑邊際成本。

另外由實驗結果顯示，實驗過程中所使用的比較基礎：路徑旅行時間，與各邊際成本表示方法計算出來的依時性邊際旅行時間之間，有相關性，因為在路徑旅行時間增加的情況下，邊際旅行時間將隨之增加，然由於研究範圍與時程之限制，未來應建立兩者間之關係，以使比較基礎更加嚴謹。

測試本研究構建之模式的正確性及合理性後，為能驗證本研究提出之模式，可否適用於模擬式動態交通指派模式中，本研究將於 5.2 節中以本研究構建之模式，於模擬式動態交通指派模式下進行測試，並與本研究主要參考的計算方法：DYNASMART-X 及 Peeta 的表示方法產生之結果，作比較與分析。

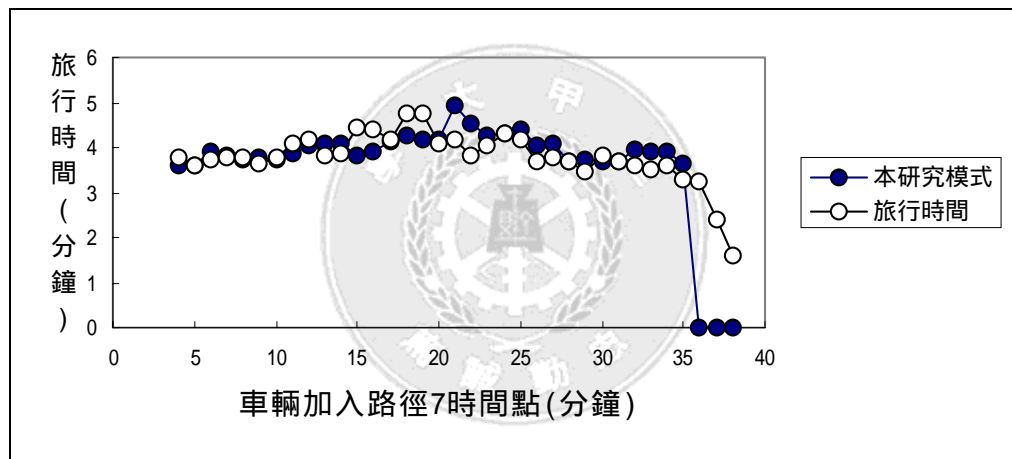


圖 5.3 需求因子 1.0 依時性邊際旅行時間與旅行時間關係圖

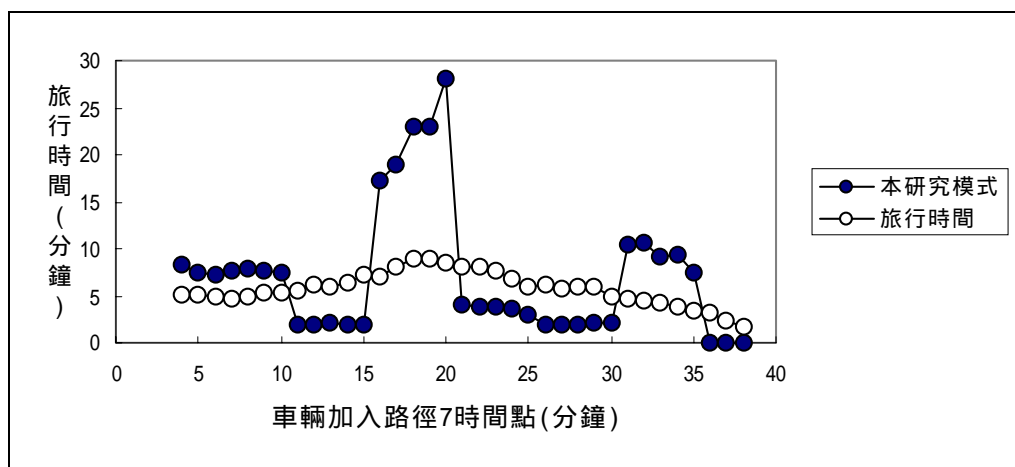


圖 5.4 需求因子 2.0 依時性邊際旅行時間與旅行時間關係圖

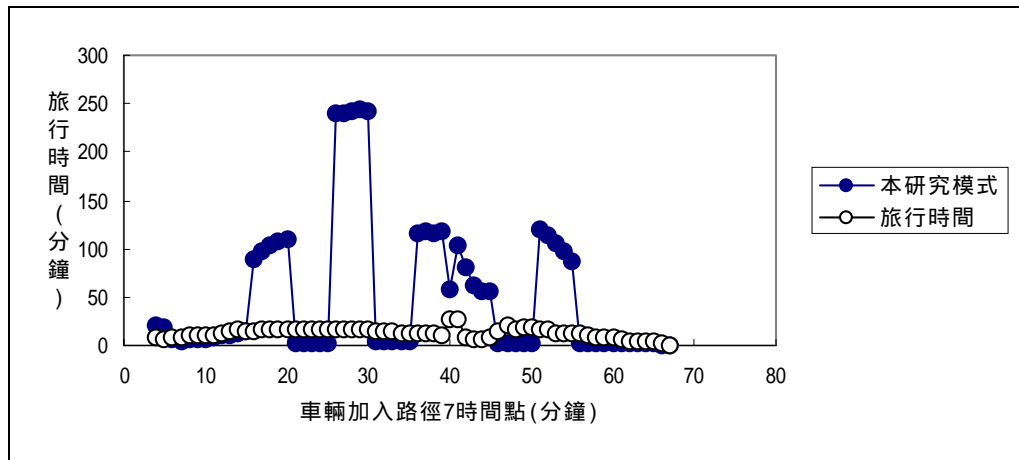


圖 5.5 需求因子 3.0 依時性邊際旅行時間與旅行時間關係圖

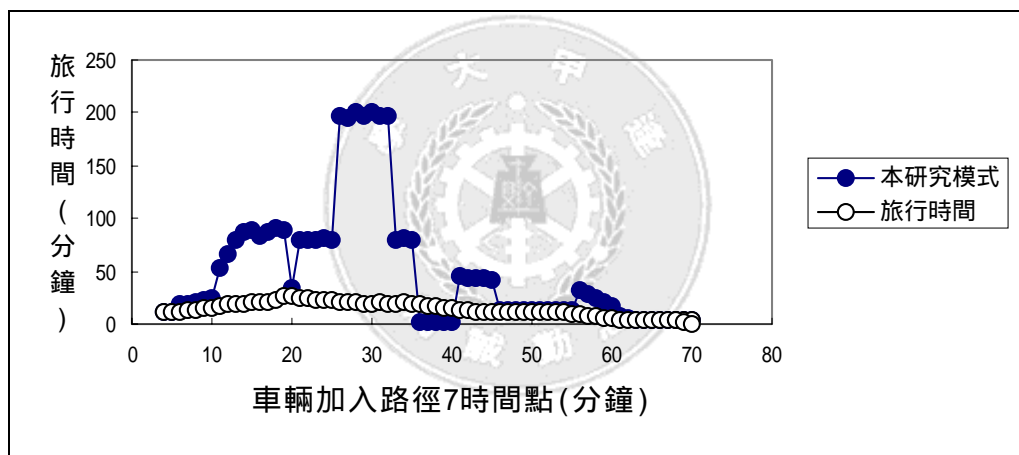


圖 5.6 需求因子 4.0 依時性邊際旅行時間與旅行時間關係圖

5.2 模式於模擬式 SO 動態交通模式之測試

5.2.1 測試條件

此節將以本研究所構建之模式與 DYNASMART-X 及 Peeta 的方法三種模式，利用實際路網圖（圖 4.11）於不同需求因子下，進行模擬式 SO 動態交通模式之測試比較與分析。

實驗中，需求因子採 1.0 及 2.75 兩者不同的需求因子大小，主要目的在測試不同車輛數下，邊際成本模式的特性。在需求因子較大的

情形下其模擬時間 (simulation time) 愈長所需要的矩陣及陣列愈大，因此受限於系統記憶體不足之情況下，研究中以記憶體可支援之上限：需求因子 2.75 進行測試。

此節所使用之測試路網為實際路網 (圖 4.11)，實驗之需求因子 1.00 者，車輛數為 6,136 輛，需求因子 2.75 者，車輛數 16,783 輛，圖 5.7 為不同需求下，路網中車輛數的型態如圖 5.7，需求因子 1.0 (虛線) 時，路網中車輛數最多達約 2,000 輛車，需求因子 2.75 (實線) 則最多約達 8,700 輛。

另外需求因子愈小，所需模擬區間 (simulation interval) 愈少，如圖中需求因子 1.0 的模擬區間約 500 左右；需求因子 2.75 則約 1,200 個模擬區間。

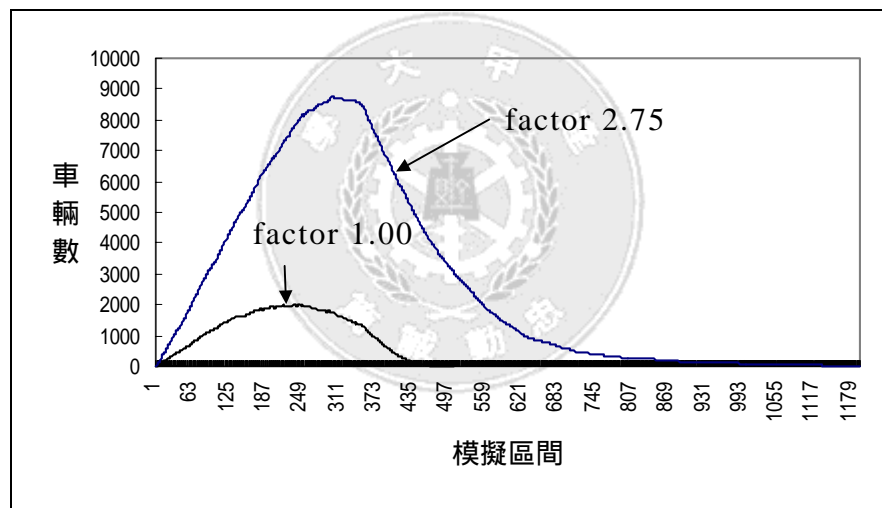


圖 5.7 每模擬時段路網中車輛數

5.2.2 測試結果

表 5.1 為需求因子 1.0，每遞子迴之平均旅行時間表，圖 5.8 則為利用表 5.1 所繪出，由圖 5.8 之結果可以發現：由於需求因子為 1.00 時路網中道路並未達擁擠之情況，因此 DYNASMART-X、Peeta 之表示方法及本研究所構建之模式，三種方法之差異性不大，但由表 5.1 其所呈現之結果，每遞子迴的平均旅行時間皆低於 DYNASMART 的起始模擬 (第 0 遞子迴) 產生之平均旅行時間，故尚稱合理。

為求正確比較，本研究以 2.75 的需求因子進行實驗，以使路網中部分道路可以產生擁擠狀況，以利比較三種邊際成本表示方法之優劣。

需求因子 2.75 的實驗結果（每遞子迴平均旅行時間），如表 5.2。由表 5.2 的結果，DYNASMART-X 所得到的結果以第六個遞子迴 12.91 分鐘最低，Peeta 的表示方法，最低為第 16 遞子迴的 14.24 分鐘最低，而本研究模式的結果則以第 7 遞子迴的 14.59 最低，結果顯示仍以 DYNASMART-X 的方法可得到最佳的結果，原因為 DYNASMART-X 的邊際成本模式由於考量最完整，因此在計算邊際成本上精準度較高，致使結果較佳。

在需求因子 2.75 之結果中，本研究模式與 Peeta 之表示方法之結果兩者相較下，並無顯著差距，故觀察模擬過程時，尚餘之道路空間（free slot）與車輛進入路段的需求量之比較，道路空間仍遠大於需求量，因此無法顯示出本研究模式第一部分（對鄰近路段影響）之特性，因此推測此為可能係造成上述結果的因素。因此未來如能克服系統資源限制的困難，使用較擁擠之路網，應可呈現出本研究模式明顯優於 Peeta 之表示方法產生的結果。

表 5.3 則是需求因子 2.75 各方法產生的實驗結果，平均旅行時間以 DYNASMART-X 的方法最低，本研究方法的結果稍高；平均停等時間以 Peeta 的方法最高；平均旅行距離則以本研究模式產生之結果較高。

另外，由各方法每遞子迴平均旅行時間的變化比較，如圖 5.10，需求因子 2.75 每遞子迴平均旅行時間變化，將其每遞子迴平均旅行時間變化量加總後再平均，得表 5.3，由表 5.3 可看出 DYNASMART-X 之平均變化量最小約 9.89%，再者為本研究所構建之模式為 10.61%最後為依時性邊際成本的 13.09%。

由上述結果可知：邊際成本之計算愈接近實際道路狀況的邊際成本，則平均旅行時間之變化愈趨於穩定，本研究所構建模式平均變化量與 DYNASMART-X 相近，而且較 Peeta 之表示方法之結果佳，因此可以推論，本研究所構建之模式計算結果，比 Peeta 之表示方法接近實際之邊際成本，但本研究之變動比 DYNASMART-X 之方法稍高。

表 5.1 需求因子 1.00 每遞子迴之平均旅行時間

方法 遞子迴	DYNASMART-X	Peeta 之表示方法	本研究之模式
0	9.02	9.02	9.02
1	7.46	7.58	7.58
2	7.36	7.58	7.59
3	7.27	7.51	7.51
4	7.27	7.49	7.52
5	7.40	7.50	7.51
6	7.31	7.67	7.37
7	7.29	7.45	7.63
8	7.27	7.48	7.48
9	7.27	7.48	7.53
10	7.26	7.45	7.47
11	7.28	7.49	7.42
12	7.34	7.38	7.59
13	7.31	7.52	7.47
14	7.24	7.48	7.58
15	7.27	7.40	7.41
16	7.24	7.40	7.50
17	7.28	7.39	7.47
18	7.32	7.38	7.50
19	7.31	7.40	7.48
20	7.30	7.54	7.53
21	7.36	7.41	7.50

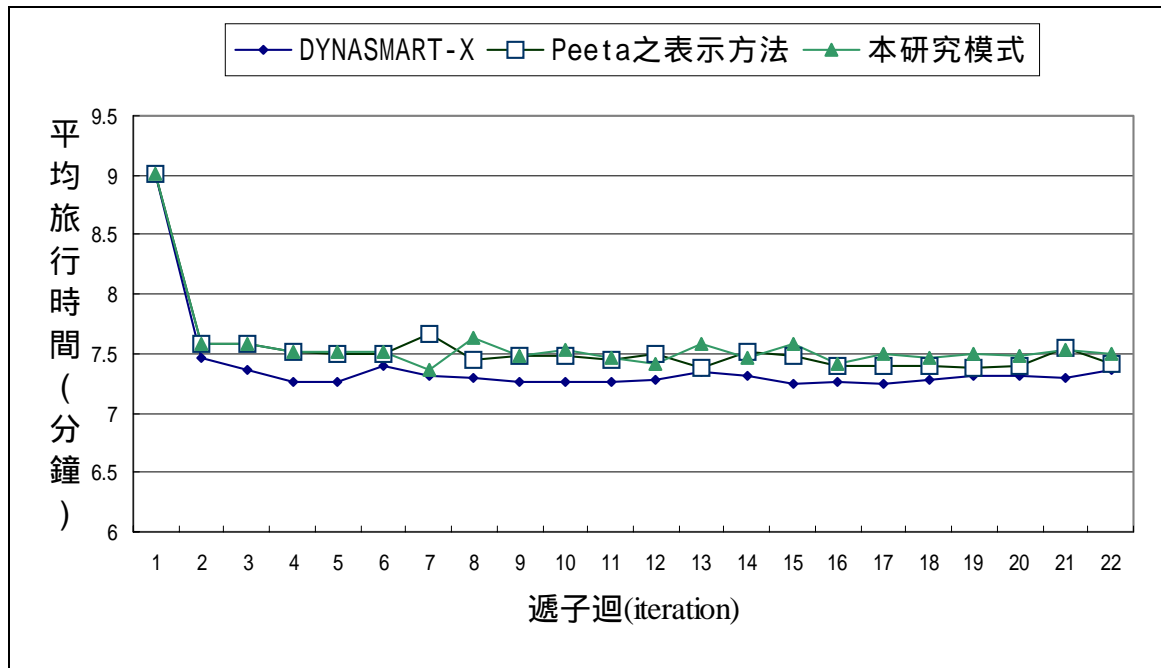


圖 5.8 需求因子 1.00 每遞子迴之平均旅行時間

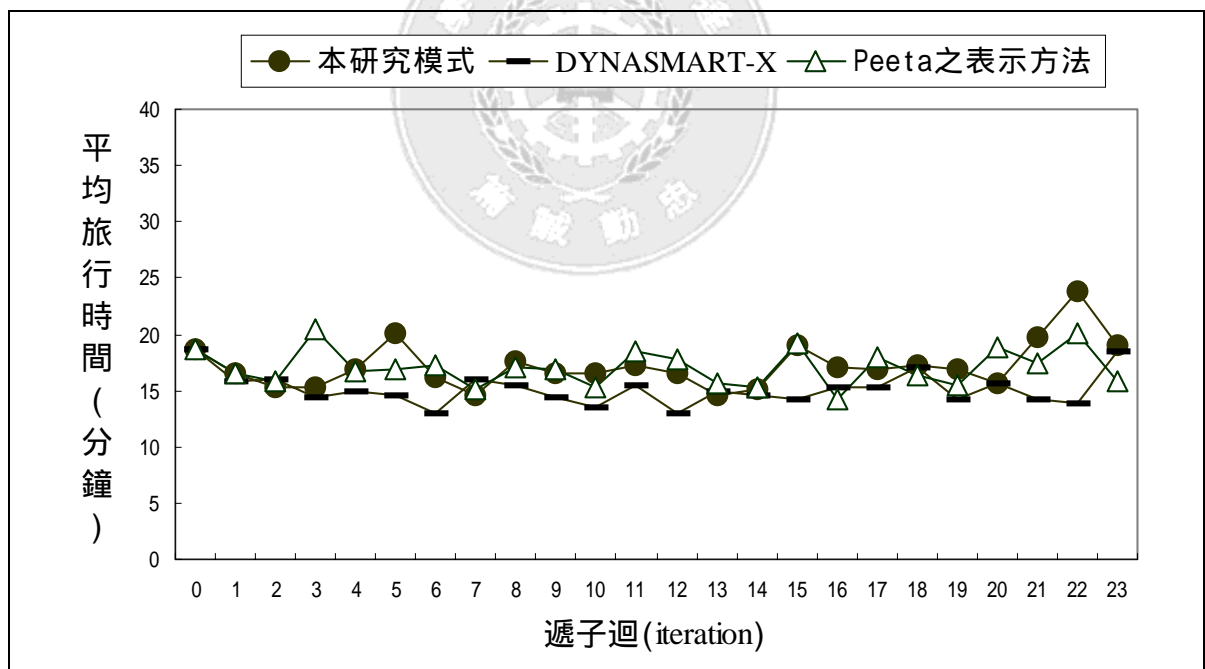


圖 5.9 需求因子 2.75 每遞子迴之平均旅行時間

表 5.2 需求因子 2.75 各方法每遞子迴之平均旅行時間

方法 遞子迴	DYNASMART-X	Peeta 之表示方法	本研究之模式
0	18.71	18.71	18.71
1	15.75	16.52	16.47
2	16.07	15.79	15.33
3	14.40	20.47	15.22
4	14.96	16.67	16.82
5	14.66	16.86	20.14
6	12.91	17.30	16.18
7	15.98	15.04	14.59
8	15.42	17.05	17.54
9	14.40	16.82	16.59
10	13.56	15.25	16.54
11	15.46	18.45	17.26
12	13.00	17.75	16.57
13	14.93	15.58	14.59
14	14.51	15.37	15.05
15	14.16	19.24	18.97
16	15.24	14.24	17.10
17	15.31	17.94	16.91
18	17.01	16.29	17.18
19	14.25	15.38	16.82
20	15.73	18.86	15.61
21	14.31	17.48	19.66

表 5.3 需求因子 2.75 各方法之數值實驗結果

項目 \ 方法	DYNASMART-X	Peeta 之 表示方法	本研究 之模式
平均旅行時間 (分鐘)	12.91	14.24	14.59
平均停等時間 (分鐘)	7.79	7.95	7.76
平均旅行距離 (公里)	3.92	4.03	4.05

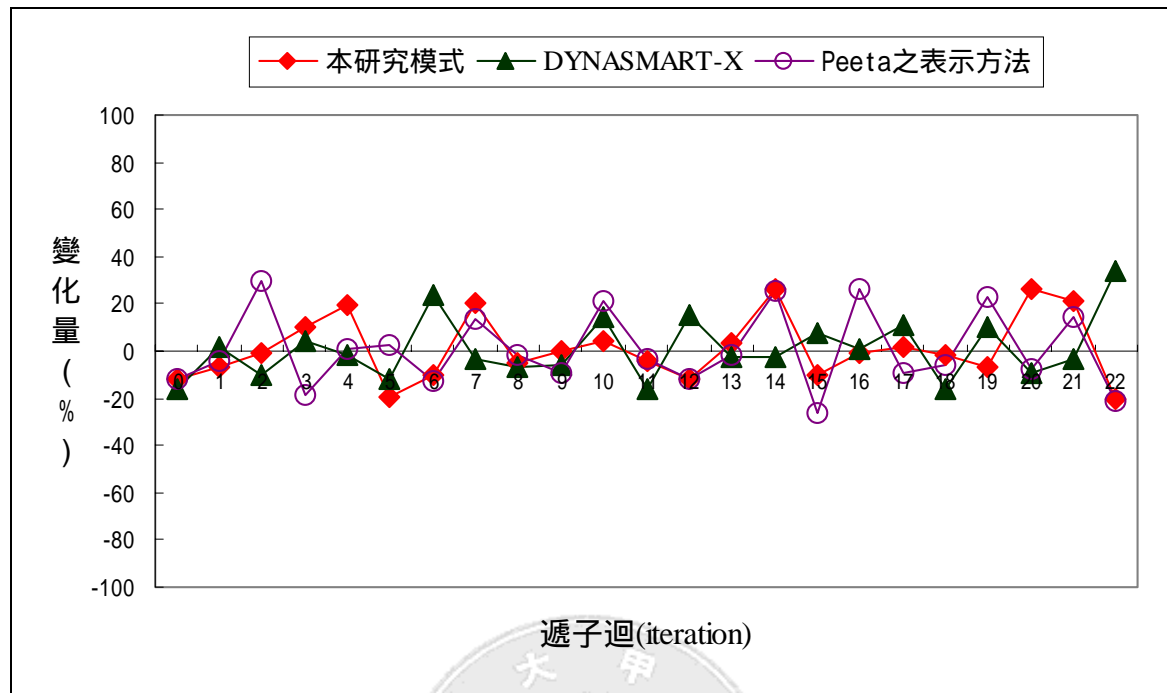


圖 5.10 需求因子 2.75 每遞子迴平均旅行時間變化

表 5.4 需求因子 2.75 之平均旅行時間平均變化量

計算方法	DYNASMART-X	Peeta 的表示方法	本研究之模式
平均旅行時間 平均變化百分比	9.89%	13.09%	10.61%

5.3 演算法之收斂

研究中所使用之演算法乃使用一「違反值」之累計總數，以確定演算是否收斂，其認定違反值之準則為：指派時段 T 由起點 O 迄點 D 於 I 階段使用路徑 K 之車輛數 $XP(O,D,T,K,I)$ 與指派時段 T 由起點 O 迄點 D 於 $I+1$ 階段使用路徑 K 之車輛數 $XP(O,D,T,K,I+1)$ 兩者之差大於 0.5 時則給以一違反值。

由上述，則違反值的定義為：兩遞子迴間指派時段 T 由起點 O 至迄點 D 使用 K 路徑的車輛數不相同者。本研究中所訂定收斂的違反值為 50。

5.2 節中需求因子 2.75 的測試，違反值如表 5.5。由表 5.5，DYNASMART-X 之最小值為第 15 個遞子迴的 1943，Peeta 的依時性邊際旅行時間最小值為第 16 個遞子迴的 1351，本研究模式的最小值則為第 12 及第 15 遞子迴的 1885。

各方法在每遞子迴的違反值雖有越來越低的趨勢，但並無法如預期往更低的違反值收斂，甚至在最後產生上下振盪之情況，可能因素為，演算法中 MSA 之更新路徑方法雖為穩定的更新路徑方式，但無法迅速使違反值下降。因此本研究所訂的違反值將是決定演算法收斂的因素，若訂定過高則演算法收斂雖快，但可能無法得到最佳結果；若訂定過低則可能導致演算法一直振盪不定。

表 5.5 違反值比較表

參數 遞子迴	DYNASMART-X	Peeta 的 表示方法	本研究模式
1	2110	2018	2020
2	2269	2284	2309
3	2159	1820	2155
4	2058	2053	2045
5	1977	1964	1974
6	2010	1867	1887
7	2018	1820	1944
8	1989	1451	1936
9	1987	1886	1936
10	1988	1839	1943
11	1962	1530	1914
12	1959	1905	1885
13	1976	1920	1896
14	1943	1902	1887
15	1943	1590	1885
16	1953	1351	1957
17	1944	2006	1985
18	1960	1976	1943
19	1973	1918	1987
20	2008	1570	1970
21	1977	1908	1960

5.4 演算效率分析比較

在本節中，本研究首先將於第 5.3.1 節探討模擬式 SO 動態交通指派模式，每一執行程式之功用，並於第 5.3.2 節中計算各步驟執行時間之結果並作比較分析。

5.4.1 模擬式 SO 動態交通指派模式演算步驟

為能詳細比較出 DTA 在演算過程時，在每個步驟所花費之計算時間，遂對本研究所採用之模擬式動態交通指派模式詳細演算流程作探討。在 SO 之指派流程中主要分「起始階段」及「模式主要運算階段」，如圖 5.11。茲就針對此兩階段說明如下。

1. 起始階段

起始階段主要目的在於作指派前將所需之各項資料準備就緒，並形成演算過程所須要之資料格式，其第一步主要是 DYNASMART 根據給定之依時性起迄模擬產生初步路網之交通型態如路徑及車輛數，接著則是改變參數讓下一步驟可以使用，利用 DYNASMART 所產生之各項交通型態資料，產生 SO 指派所需之路徑車輛數，然後將歷史之路徑資料儲存至路徑資料庫。

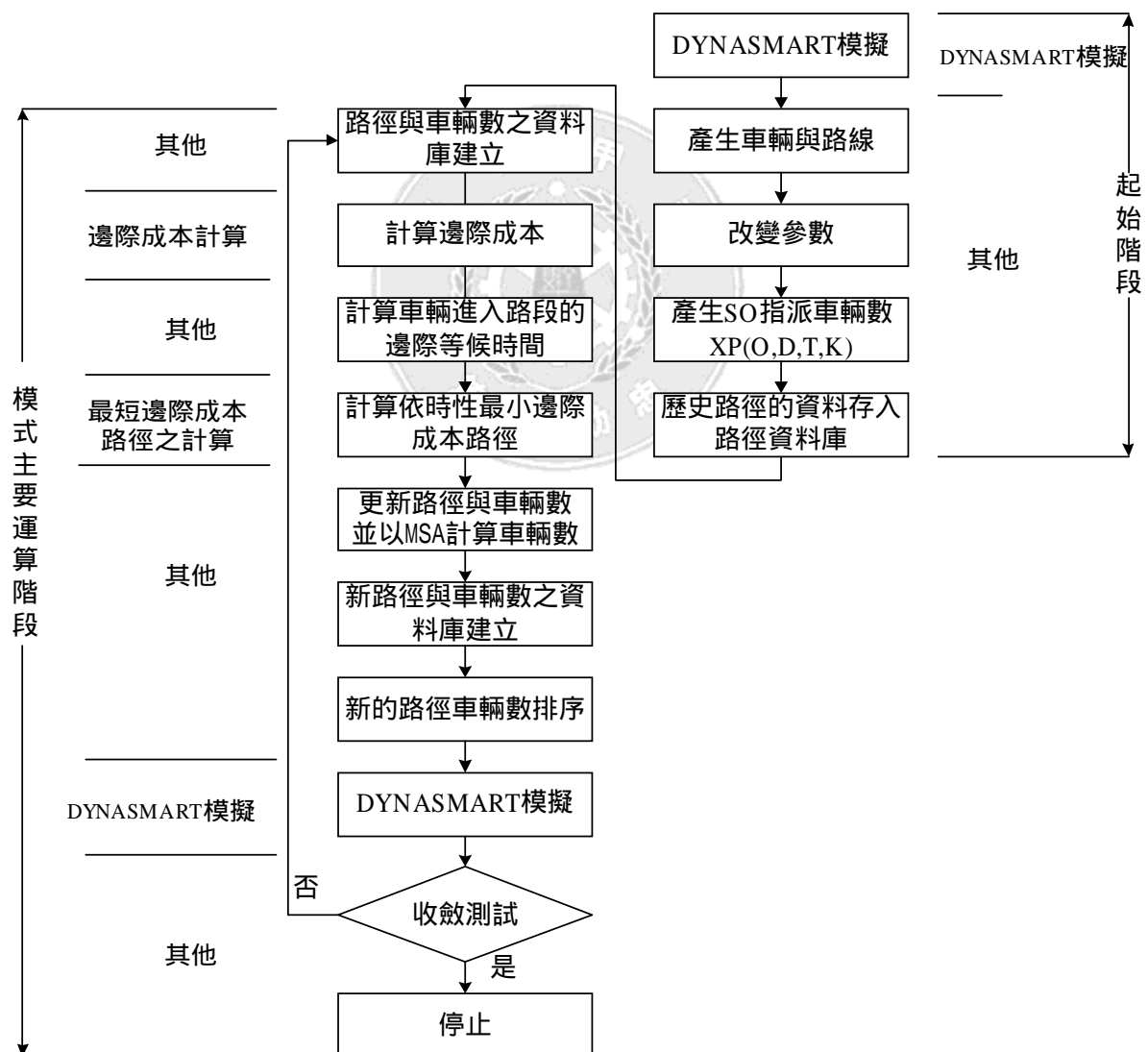
2. 模式主要運算階段

經過起始階段之各項資料準備齊全後，便可進行模擬式 SO 動態交通指派核心之演算部分，在此階段中，由所提供之交通型態資料及邊際成本運算方法不斷演算，直到經過演算所得到之結果能被接受為止。

首先利用起始階段所產生之車輛數及路徑資料庫建立路徑車輛數資料庫，然後利用所產生之各項路段與交通型態，如車輛數、旅行時間等等，依各種不同之邊際成本計算方式計算每條路段上之邊際成本，接著計算每條產生車輛之路段上之車輛等候進入路網之邊際等候時間時間，並將其加入路段之邊際成本中。

根據上述路段之邊際旅行時間及平均旅行時間，進行所謂之「依時性最短邊際旅行時間路徑」之計算，所使用之演算法則是由美國德州大學奧斯汀分校所發展之演算法 (Ziliaskopoulos and Mahmassani, 1992a, 1992b)，下一步則是更新路徑，首先以前述依時性最短邊際成本路徑演算法所產生之新路徑更新路徑資料庫，然後以 MSA(Method of successive Averages)路徑上之車輛數。

再者根據上述所產生之路徑以及車輛數建立 DYNASMART 模擬器所需要之資料庫，並進行模擬，如果所得到模擬之結果不能符合研究所能接受之收斂範圍，則繼續下一個遞子迴；如已符合收斂準則，則停止演算。



5.4.2 演算法運算時間

依據 5.3.1 節中各步驟之分析，本研究將各步驟之運算時間計算後，依各步驟主要運算時間佔總時間之比例大致分為四大部分：第一部分為 DYNASMART 模擬時間；第二部分為邊際成本計算時間，此部分亦為本研究主要探討者；第三部分為依時性最短邊際成本路徑計算時間；第四部分則為其他步驟時間之總和。在程式執行流程中，此四大部分之區分如圖 5.11，計算四大部分程式執行時間之結果如表 5.6。

表 5.6 為需求因子 2.75 之執行時間統計結果，表 5.6 所呈現之結果，以邊際成本計算時間來看，DYNASMART-X 所需之執行時間總計需要 1193.7 秒平均每遞子迴需 56.84 秒，而 Peeta 之表示方法則總計為 367.1 秒平均每遞子迴需 17.48 秒，本研究模式之執行時間總計為 417.6，平均每遞子迴需 19.9 秒。在邊際成本計算之總時間上 DYNASMART-X 的方法顯然較其他兩者多出許多，因此相較之下本研究模式之計算時間可以大幅減少原來 DYNASMART-X 中邊際成本所須之計算時間。

如就整體演算法之執行總時間，則 DYNASMART-X 為 4758.2 秒，Peeta 的方法為 3956.1，本研究模式的方法為 3966.4 秒，本研究模式的方法可大幅減少 791 秒。

表 5.7 即為本研究所構建之模式相對於原本 DYNASMART-X 所使用的邊際成本計算方法在計算時間上的改善百分比，本研究模式單就邊際成本來看即可改善 65.4%，而對於所有執行時間則可改善以 DYNASMART-X 之方法執行總時間的 17.4%。

由上述結果，本研究所構建的模式可大幅改善原本 DYNASMART-X 中所使用的邊際成本計算方法的執行時間，因此模式未來在應用上可有許多彈性，如即時資訊下的交通分析等等。

由表 5.6 結果，本研究亦發現 DYNASMART 之模擬時間佔執行總時間之比例甚高，未來在模擬式 SO 動態交通指派模式的應用上，從 DYNASMART 之模擬時間著手，則整體演算法將更適於即時資訊下之各種應用。

表 5.6 需求因子 2.75 各方法之計算時間比較表

比較項目 \ 方法		DYNASMART-X	Peeta 之表示方法	本研究模式
DYNASMART 模擬時間	總計	2542.1	2690.6	2649.1
	平均	121.1	128.1	126.1
邊際成本計算時間	總計	1193.7	367.1	417.6
	平均	56.8	17.5	19.9
依時性最短邊際成本路徑計算	總計	542.1	433.7	429.5
	平均	25.8	20.7	20.5
其他	總計	480.3	465.7	470.2
	平均	22.9	22.2	22.4
執行所需總時間		4758.2	3956.1	3966.4

(單位：秒)

表 5.7 本研究模式改善演算效率百分比比較表

	本研究模式改善百分比
邊際成本計算時間	65.4%
執行所需總時間	17.4%

5.5 模式之實際應用

本節主要針對模擬式 SO 動態交通指派模式如何實際應用作說明，在 5.5.1 節將路網結構作改變以測試整體道路系統的績效變化；5.5.2 節，以一例說明未來可以如何應用模擬式 SO 動態交通指派模式於實例中。

5.5.1 道路系統容量變化

除應用即時資訊測試對模式的影響外，本研究為測試，如道路施工封閉、發生事故或災害等造成道路容量改變後對於整體路網的變化及影響，因此將圖 4.11 中高速公路的容量改變，然後進行測試。

實驗中，「道路容量增加」，係將高速公路雙向的車道數各增加一車道，「道路容量縮減」，則是將高速公路雙向的車道數各縮減一車

道，在此實驗中並未考慮即時資訊的車輛，僅就依據系統最佳化原則之車輛作探討，實驗結果如表 5.8 為不同道路容量下，每遞子迴的所有車輛之平均旅行時間，圖 5.12 則為每遞子迴與平均旅行時間關係圖。

由表 5.8，其中原路網最低者為第 7 個遞子迴之 14.59 分鐘；縮減道路容量最低為 16.30 分鐘；增加道路容量最低為 14.12 分鐘，實驗結果顯示增加道路容量，的確可改善整體平均旅行時間，但與原路網比較下，改善幅度仍有限。本測試之實驗結果如表 5.9，發現道路容量增加停等時間亦隨之減少。

表 5.8 需求因子 2.75 不同道路容量下每遞子迴平均旅行時間

方法 遞子迴	原路網	縮減道路容量	增加道路容量
0	18.71	18.62	18.66
1	16.47	17.75	15.86
2	15.33	17.86	16.08
3	15.22	16.30	15.55
4	16.82	21.24	16.01
5	20.14	19.84	14.90
6	16.18	16.68	14.12
7	14.59	18.11	15.82
8	17.54	17.36	16.59
9	16.59	16.60	16.71
10	16.54	18.02	16.50
11	17.26	19.10	14.71
12	16.57	20.33	20.42
13	14.59	18.18	16.78
14	15.05	19.22	16.45
15	18.97	16.35	14.30
16	17.10	17.19	14.78
17	16.91	18.63	16.68
18	17.18	16.63	15.10
19	16.82	21.81	17.49

表 5.9 不同道路容量下實驗結果

項目	原路網	增加道路容量	縮減道路容量
平均旅行時間	14.59	14.12	16.30
平均停等時間	7.76	7.34	9.10
平均旅行距離	4.05	4.06	4.11

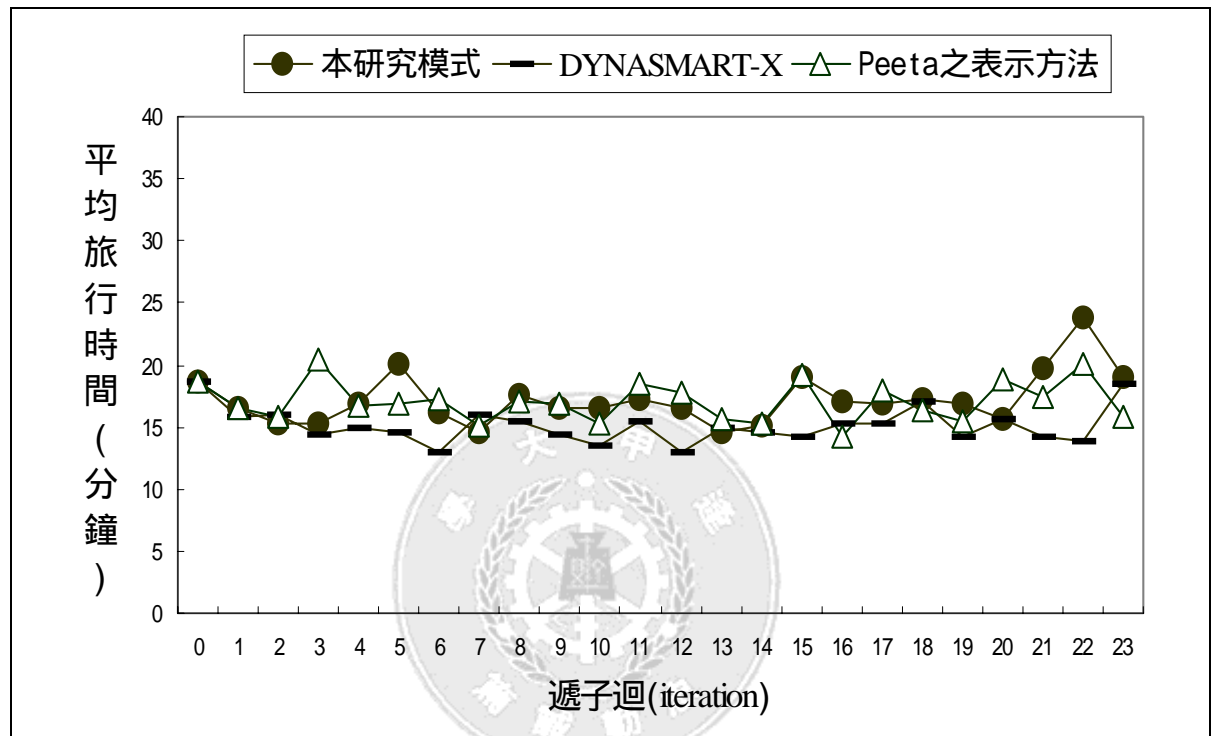


圖 5.12 道路容量改變路網中車輛每遞子迴之平均旅行時間

5.5.2 緊急事故之應用

模擬式 SO 動態交通指派模式可以應用之範圍及彈性甚廣，諸如各種需要路網分析與指派之功能者，本章即以緊急事故之決策作未來實例應用之說明。

緊急事故由於屬於偶發性，往往對於人員或財產造成相當大之損失，因此需要由政府統一分配救援物質或人員，始不至於造成資源浪費之情況。而在緊急事故發生後最重要的問題之一 交通問題便油然而生，諸如人員調派與車輛調派問題、救援物資及醫療人員之運補、

交通管制問題，茲說明如下

1. 人員調派與車輛調派問題

在緊急事故發生後，無論是否為救援車輛或人員皆會往路網集中，如此非但會造成道路擁擠而且可能會讓救援車輛及人員擔誤救災時效。

2. 救援物資及醫療人員之運補

在搶救時間限制下，救援物資或醫療人員之運補，可能由於路網之擁擠卻無法有效地取得替代道路或彈性的運補路線而造成延誤。

3. 交通管制問題

在 1.與 2.中所述之情況若欲針對某緊急或救援事故做交通管制，則往往無法配合實際車流之狀況進行管制。

整合上述之問題，在緊急事故之決策支援系統中架構路網分析之功能刻不容緩。

傳統之交通指派偏於靜態上之處理，因此欠缺對於指派過程與時間之考慮，模擬式 SO 動態交通指派模式主要根據系統最佳化之路徑選擇原則，將依時性之 OD 依次分配到交通路網中，而且車輛之移動是根據駕駛人行為決策之結果，故可反應即時資訊下駕駛人之反應。

模擬式 SO 動態交通指派模式在災害發生時期之決策支援系統中將扮演著重要角色，因為無論是災害之反應期亦或復原期皆必須透過預建於模式庫中之模擬式 SO 動態交通指派模式對路網車流之分析，以提供交通疏散及交通管制措施之參考。

模擬式 SO 動態交通指派模式於緊急事故之決策支援架構如圖 5.15，模式庫透過 GIS 之資料庫、地理圖形庫，擷取模擬式 SO 動態交通指派演算法所需之各項資料後，將資料輸入模式庫並進行演算，並將所得到之結果透過交通型態分析，將道路系統之車流狀況，如可能產生之擁擠或交通事件，呈現於 GIS 圖形上，提供決策者清楚且明確的相關決策分析之資料。

於緩和災難期，決策支援系統可配合 ATIS 之即時資訊蒐集建立資料庫，透過模擬式 SO 動態交通指派模式提供救援車輛及人員最佳之路徑於第一時間抵達災難現場消除災害原因。

準備規劃與訓練期，利用模擬式 SO 動態交通指派模式進行路網指派產生交通疏散計劃後，在反應期以 ATMS 給予路網中道路使用者規範性資訊，以達系統最佳化之目的。

最後於復原期，模擬災後可能的交通型態，並作車輛規範性資訊之提供，使車輛避免進入災區或佔用物質及人員運補之路線，務使災害儘速復原，並使交通狀況不致癱瘓。

本研究所發展的邊際成本計算模式，在緊急事故之決策支援中將扮演著關鍵角色，因決策支援系統中之車輛若依照系統最佳化原則進行路徑選擇，則其選擇依據為路徑的邊際成本，若成本計算誤差過大，則可能導致分析結果誤差及策略研擬錯誤，使整體路網更形擁擠，運補作業延誤。

另外在計算效率上，本研究所構建之模式可以改善原先 DYNASMART-X 中邊際成本之複雜計算，縮短路段邊際成本計算時間，故應可實際應用於講求時效性的緊急事故之決策支援中。

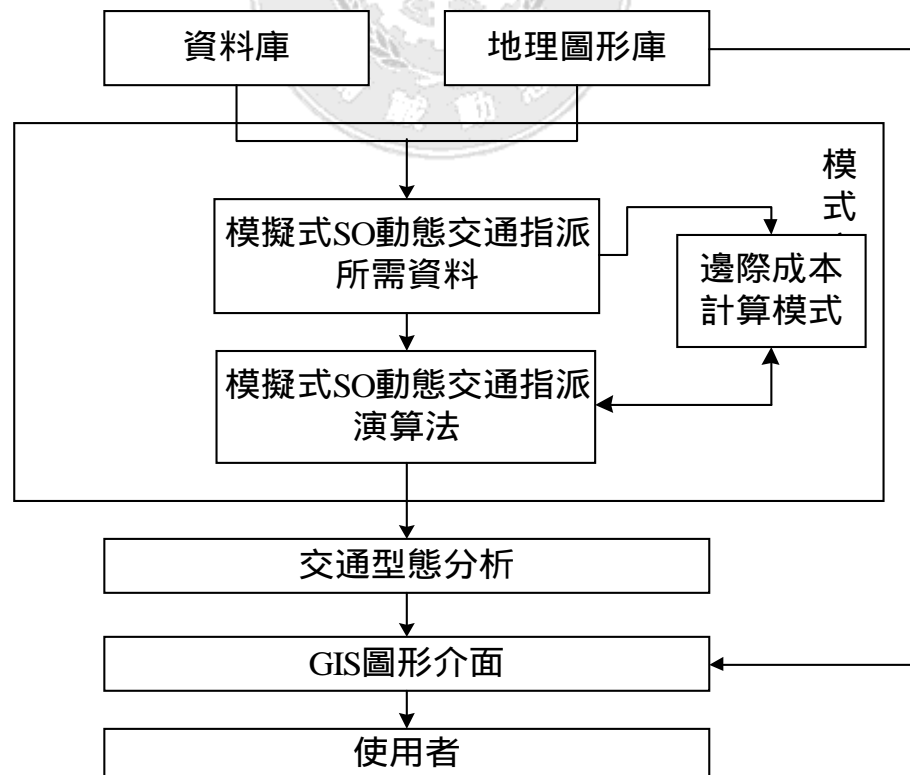


圖 5.13 模擬式 SO 動態交通指派應用於緊急事故決策之架構

5.7 小結

本研究應用第四章，各邊際成本表示方法的測試結果，構建模式，模式中除反應車輛對現有路段之影響外，亦考慮對鄰近路段的影響。而為求合理性，本研究利用所構建之模式，計算路徑不同時間點下之依時性邊際旅行時間，並與各路徑不同時間點產生之旅行時間比較，初步結果顯示，本研究構建之模式之反應與路徑之擁擠狀況一致。

為能實際應用於模擬式動態交通指派模式中，本研究將新模式應用於模擬式 SO 動態交通指派模式中，並與本研究構建模式時，所參考之另外兩模式：DYNASMART-X 的方法及 Peeta 的表示方法，同樣應用於模擬式 SO 動態交通指派模式中之結果作比較，結果顯示，在需求因子 1.0 時，三種模型的反應一致，亦反應本研究所構建模式的適用性；然而需求因子 1.0 時的結果，由於路網擁擠程度不足，三者的特性未明顯差異，因此本研究再採用較高需求因子進行測試，但受限於系統記憶體資源，本研究僅能將需求因子提高至 2.75。

需求因子 2.75 之測試結果，DYNASMART-X 的方法所得之結果優於本研究構建之模式，主要原因在於 DYNASMART-X 的計算上，考慮的確比本研究所構建之模式完整，其考慮完整性的結果導致，後來的演算法效率的比較，就邊際成本的運算上，本研究所構建之模式，卻能改善 DYNASMART-X 方法 65.4% 的計算時間，而演算法的總執行時間則改善 17.4%，故本研究所構建之模式在實際應用上仍具有許多彈性與空間。

另外，本研究為測試本研究所構建之邊際成本模式是否能實際應用在各種不同情況的交通分析，本研究測試一道路容量改變的實驗，實驗結果反應出，道路容量增加對系統平均旅行時間的確可以改善，但幅度並不大；道路容量縮減，則會增加系統的平均旅行時間，因此可確定本研究之模型在應用上的正確性。

最後，本研究提出一實例說明本研究如何結合模擬式動態交通指派模式應用於緊急事故決策，以供未來使用模擬式 SO 動態交通指派模式之參考。

第六章 結論與建議

6.1 結論

本研究回顧各種邊際成本的計算方法，經過各模式特性分析後，發現目前邊際成本的表示方法，其求解方法大致分三大類，第一類為假設在道路成本函數已知之情況下，求得成本函數之導數而得到邊際成本函數的「動態系統最佳化路段邊際旅行時間」，第二類則是利用「延滯」的觀念做為求解方法之局部邊際延滯與局部邊際旅行時間；最後一類則是以模擬方式求解，如 Peeta 的表示方法及 DYNASMART-X 所使用的求解方法。由於研究使用的模擬方法無法求得第一類方法的成本函數，因此研究中以第二類及第三類的表示方法為主要構建模式的參考。

為能建立正確而且合理的邊際成本計算方法，本研究以路徑的依時性邊際旅行時間與路徑旅行時間的變化比較各模式之合理性，並以模擬式動態交通指派模式測試各邊際成本模式之適用性，其結果如下：

1. 路徑依時性邊際旅行時間測試之結果

依時性邊際旅行時間測試之結果，以第四分鐘加入各比較路徑車輛的依時性邊際旅行時間進行比較時，發現各方法得到所有比較路徑中最短之結果差距不大。

然而單以第四分鐘為基準之依時性邊際成本比較，無法比較各模式的合理性，故研究中再以不同時間點加入比較路徑的旅行時間，與研究中所計算的依時性邊際成本比較，結果發現局部邊際延滯與局部邊際旅行時間的結果與旅行時間之變化有較大的誤差。

2. 模擬式動態交通指派測試結果

為得知各邊際成本模式，應用於模擬式動態交通指派模式的特性，因此再以各邊際成本模式應用於模擬式動態交通指派模式中，觀

察各模式於模擬式動態交通指派模式之適用性，結果發現局部邊際延滯與局部邊際旅行時間的測試結果有所誤差，原因可能為兩者的求解方法於模擬式動態交通指派模式的方法無法得到較佳之結果，其餘兩者之結果則較為合理。

另外本研究應用第四章對各邊際成本計算方法的測試結果構建模式，模式中除反應車輛對現有路段之影響外，亦考慮對鄰近路段的影響。但為求新構建模式的合理性及適用性，以新模式進行依時性邊際旅行時間及其他測試結果如下：

1. 新構建模式之依時性邊際旅行時間測試

為求合理性，本研究利用所構建之模式計算路徑不同時間點下之依時性邊際旅行時間，並與各路徑不同時間點產生之旅行時間比較，初步結果顯示，本研究構建之模式之反應與路徑之擁擠狀況一致。

2. 新構建模式於模擬式 SO 動態交通指派模式的測試

為能實際應用於模擬式動態交通指派模式中，本研究將新模式應用於模擬式 SO 動態交通指派模式中，並與本研究構建模式時，所參考之兩模式：DYNASMART-X 的方法及 Peeta 的表示方法作比較，結果顯示在需求因子 1.0 時，三種模型的反應一致，亦反應本研究所構建模式的適用性；然而需求因子 1.0 時的結果，由於路網擁擠程度不足，因此本研究再採用較高需求因子進行測試，但受限於系統記憶體資源本研究僅能將需求因子提高至 2.75。

需求因子 2.75 之測試結果，DYNASMART-X 的方法所得到之結果優於本研究之方法，主要原因在於 DYNASMART-X 的模式，的確考慮的因素，比本研究所構建之模式完整，然而在演算法效率的比較，就邊際成本的運算上，本研究所構建之模式卻能改善 65.4% 的計算時間，而演算法的總執行時間則改善 17.4%，故本研究所構建之模式在實際應用上仍有許多彈性與空間。

3. 模式於即時資訊下的測試

本研究為測試智慧型運輸系統下即時資訊對 SO 動態交通指派呈現的反應，本研究給予具不同比例之具即時資訊系統之車輛進行測試，結果發現，具即時資訊車輛的路網中，SO 的車輛旅行時間會有減少趨勢，而且路網中具即時資訊之車輛比例愈高，整體道路系統的平均旅行時間愈低。

4. 模式實際的應用

另外，本研究為測試本研究所構建之邊際成本模式是否能實際應用在各種不同情況的交通分析，本研究測試一道路容量改變的實驗，實驗結果反應出，道路容量增加對系統平均旅行時間的確可以改善，但幅度並不大；道路容量縮減，則會增加系統的平均旅行時間，因此可確定本研究之模型在應用上的正確性。

最後，本研究提出一實例說明本研究如何結合模擬式動態交通指派模式應用於緊急事故決策，以供未來使用模擬式 SO 動態交通指派模式之參考。

本研究結果，主要提供管理者，於進行交通管理時，能得到正確分析資料的方法，以提供駕駛人精確的即時資訊，然而在現實情況下，駕駛人是否遵守管理者給予之規範性資訊，卻具許多不確定因素，因為可能由於個體需求之起迄點不同，而導致同路段上的車輛，有不同的邊際成本，使得如何透過指派，達系統目標的情況更加複雜。因此，後續研究，除使本研究模式在邊際成本計算更完整之外，在指派上應可考慮其他配套措施，如道路擁擠收費等，使動態交通指派之結果，能更具實用性。

6.2 建議

根據實驗過程的經驗，本研究提出幾點建議如下：

1. 研究中局部邊際延滯及局部邊際旅行時間的結果，與其他方法特性差異較大，未來應進一步探討與分析，主要影響兩者結果的因素。
2. 研究過程由於系統記憶體的限制，發現本研究之路網無法達到真正擁擠之情況，因此未來應克服在系統記憶體上的限制，以一真正擁擠之路網進行測試，以求模式在應用上的彈性。
3. 在模擬式動態交通指派模式的測試中，演算法之收斂為一重要課題，除違反值門檻值的設定外，另外演算法中，路徑更新的方法亦是未來應該研究的方向，因為實驗過程中每遞子迴的違反值，除受邊際成本計算精準度的影響外，路徑更新的方法亦是影響演算法收斂的主要因素之一。
4. 未來應實際應用新構建模式於更多之實例中，如結合號誌控制及其他交控措施，以測試模式在不同的之應用下，可能產生的特性，提高模式實例應用的可能性。
5. 本研究所構建之模式在運算效率上雖可大幅改善原本 DYNASMART-X 的方法的運算時間，但求解結果卻不如該方法，因此未來，應在求解結果與演算效率去得到一平衡點，始能使演算達較佳結果。
6. 本研究主要在於觀察模擬式動態交通指派模式，在系統最佳化原則時，模式利用不同邊際成本表示方法，做為路段邊際成本計算依據時的特性與結果，未來應再探討使用者均衡指派原則之特性與結果，以探討兩者在模擬式動態交通指派模式之特性。

參考文獻

- 1.卓訓榮，李治輝（1995），「靜態路網交通量指派模式與求解法之回顧」，運輸計劃季刊，第二十四卷，第三期，頁283~頁298。
- 2.胡大瀛（2001），「模擬式動態交通指派模式之研究」，運輸計劃季刊，第三十卷，第一期，頁1~頁32。
- 3.胡大瀛（1996），「動態路網模擬指派模式之建立：以 DYNASMART 為例」，運輸學刊，第九卷，第三期，頁1~頁24。
- 4.韓復華、胡大瀛（1987），「路網疏散模式研究與微電腦決策輔助系統之建立」，中華民國運輸年會第二屆學術論文研討會，頁36。
- 5.魏慶地（1999），行前交通資訊逐日動態模擬之研究，國立台灣大學土木工程學研究所博士論文。
- 6.Ben-Akiva, M. (1985) "Dynamic Network Equilibrium Research", *Transportation Research*, Vol. 19A, No. 5/6, pp. 429-431.
- 7.Beckmann, M.J., McGuire, C.B., and Winston, C.B. (1956), *Studies in the Economics of Transportation*, Yale University Press, New Haven, Connecticut.
- 8.Boyce, D.E. (1984), "Urban Transportation Network Equilibrium and Design Models: Recent Achievements and Future Prospects", *Environment and Planning A*, Vol. 16, pp. 1445-1474.
- 9.Boyce, D.E. (1989), "Route Guidance Systems for Managing Urban Transportation Networks: Review and prospects", presented at the tenth Italian Regional Science Association Conference, Rome, Italy.
10. Byung-Wook W., Roger L.T., David B. and Terry L. Friesz (1996), "A Comparison of System Optimum and User Equilibrium Dynamic Traffic Assignments With Schedule Delays", *Transportation Research Part C*, Vol. 3, No. 6, pp. 389-411.
11. Chang, G. L., Mahmassani, H. S. and Herman R. (1985), "A Macroparticle Traffic Simulation Model to Investigate Peak-Period Commuter Decision Dynamics," *Transportation Research Record* 1005, pp. 107-120.
12. Chen, H. K. (1999), *Dynamic Travel Choice Models: A Variational Inequality Approach*, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag.
13. Center for Transportation Research The University of Texas at Austin, Texas (1994), *Sensitivity Test and Guidelines for The System Optimal Dynamic Traffic Assignment Procedures for ATIS/ATMS*, Technical Report DTFH61-90-C00074-BX.

14. Elbert V. B. and Juith H. B. (1995), *Economics: The Science of Common Sense*, South-Western College Publishing.
15. Ghali M.O. and Smith M.J. (1995), "A Model for The Dynamic System Optimum Traffic Assignment Problem", *Transportation Research Part B* Vol. 29B, No.3, pp.155-170.
16. Hu, T. Y., (1996), *A Guide to DYNASMART*, Technical Report, Center for Transportation Research, The University of Texas at Austin.
17. Hu, T. Y., Rothery, R. W., and Mahmassani, H. S. (1992), *DYNASMART: Dynamic Network Assignment-Simulation Model for Advanced Road Telematics*, Technical Report DTFH61-90-C-0074-TWP1, Center for Transportation, The University of Texas at Austin.
18. Jayakrishnan, R. (1992), *In-Vehicle Information System for Network Traffic Control: A Simulation Framework to Study Alternative Guidance Strategies*, Ph. D. Dissertation, The University of Texas at Austin.
19. Jayakrishana R., Wei K. Tsai and Anthony Chen (1995), "A Dynamic Traffic Assignment Model with Traffic-flow Relationships", *Transportation Research Part C*, Vol. 3, No. 1, pp. 51-72.
20. Merchant, D. K. and Nemhauser, G. L. (1978), "Optimality Conditions for a Dynamic Traffic Assignment Model", *Transportation Science*, Vol. 12, NO. 3, pp. 200-207
21. Peeta, S. (1994), *System Optimal Dynamic Traffic Assignment in Congested Networks with Advanced Information System*, Ph.D. Dissertation, The University of Texas at Austin.
22. Sheffi, Y. (1985), *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, NJ.
23. U.S. Department of Transportation (1999), *ITS Mission Definition*, Federal Highway Administration, Washington, DC,.
24. Wardrop, J.G. (1952), "Some Theoretical Aspects of Road Traffic Research", *Proceedings, Institution of Civil Engineers* II(1), pp. 325-378.
25. Ziliaskopoulos, A. and Mahmassani, H.S. (1992a), "Design and Implementation of a Shortest Path Algorithm with Time-Dependent Arc Costs", *Proceedings of 5th Advanced Technology Conference*, Washington D.C.,1992, pp. 1072-1093.
26. Ziliaskopoulos, A. and Mahmassani, H.S. (1992b), "A Time-dependent Shortest Path Algorithm For Real-Time Intelligent Vehicle/Highway System Applications", *Transportation Research Record* 1408, pp.94-100.
27. Ziliaskopoulos, A. (1994), *Optimum Path Algorithm on Multidimensional Networks: Analysis, Design Implementation, and Computational Experience*, Ph.D. Dissertation, The University of Texas at Austin.

附錄一 本研究模式求解程式碼

```

subroutine marginal()
implicit none
C *****
C ** This is a prg to calculate time-dependent marginal,          **
C ** it is the method developed by Peeta.                        **
C ** EDITED BY DA,TUNG-LI                                       **
C **                                                            **
C **                                                            **
C ** Mar. 26, 2002                                              **
C *****

real M,N
real A,B,I,J,JJ,K,L,P,Q,R,T,U,W,Z,PA,PB,RA
real XK(10)
REAL XKA(10)
REAL XKB(10)
REAL XKC(10)
REAL YK(10)
REAL FP(10)
REAL PS
REAL PSUM(10)
REAL S(10)
REAL SA(10)
REAL C(10)
REAL MATA(10,10)

C*****
C*** VARIABLES WHICH USED TO CALCULATE MARGINAL****
C* ETT: ESTIMATED TRAVEL TIME
C* VOLM: VEHICLES ON LINK ON THE CERTAIN TIME.
C* NUM_LINK: NUMBER OF LINKS
C* SLOPE: THE MARGINAL OF THE ADDITIONAL VEHICLE
C* MTT: MARGINAL TRAVEL TIME OF THE CERTAIN LINK
C* AVG_SLP: AVERAGE DERIVATIVE
C* MVG_AVSP: MOVING AVERAGE OF THE DERIVATIVE
C*****

REAL TTM_MTX(1000,1000)
REAL ETT(0:1000,1000)
INTEGER volm(0:1000,1000)
INTEGER NUM_LINK
REAL TIME(1000)
INTEGER TIM
INTEGER LINK
real v(10)
REAL SLOPE(1000,1000)
REAL DT_MTT(1000,1000)
REAL AVG_SLP(1000,1000)
REAL MVG_AVSP(1000,1000),avg_flow(0:1000,1000)
real tt_slot(0:1000,1000)
real intoo(0:1000)
real freeslot(0:1000)
real diff(0:1000,1000)
INTEGER INTVL
INTEGER SIM_INT
real AVG_VEH(1000,1000)

```

```

INTEGER TMP,SS,ST
INTEGER KK,LL,NN,ROW_NUM,MM
real temp
integer sim_tim,int_tmp,avg_int,int
real nebrg_mgn(1000,1000)
integer avfltmp(1000)
REAL MTT
integer li_nu,nu_sim
parameter (li_nu=441)
parameter (nu_sim=1000)

DIMENSION MTT(li_nu,nu_sim,2)
integer nst,na
COMMON /MTT/MTT
COMMON /NST/NST
COMMON /NA/NA
real new_veh(1000,500)
real new_slp(1000,500)
  if(nst.gt.1000)then
    nst=1000
  else
    nst=nst
  end if

SIM_TIM=NST
NUM_LINK=NA

C*****
C Read travel time matrix.
C*****
      rewind(79)
      DO 1018 I=0,sim_tim-1
      READ(79,*)TIME(I)
      READ(79,'(10f8.3)') (tt_slot(I,J),J=1,NUM_LINK)
1018  CONTINUE
      rewind(76)
C*****
C* read average flow rate.(fort.76)
C*****
      read(76,'(20i4)')(avfltmp(i),i=1,num_link)
      do 113 i=0,sim_tim-1
      read(76,*)time(i)
      read(76,'(10f8.3)') (avg_flow(i,j),j=1,num_link)
113  continue
C*****
C* reading volume on all links.
C*****
      rewind(31)
      DO 1016 I=0,sim_tim-1
      READ(31,*)TIME(I)
      READ(31,'(10I5)') (volm(I,J),j=1,num_link)
1016  CONTINUE

```

```

C*****
C*   calculate travel time.
C*****
      do i=0,sim_tim-1
      do j=1,num_link
      if (avg_flow(i,j).eq.0) then
      avg_flow(i,j)=0.005
      end if

      ETT(i,j)=tt_slot(i,j)+(volm(i,j)/(60*avg_flow(i,j)))

      end do
    end do
C*****
C * The number of initial solution is 3, because using t-1, t, t+1.      *
C*****
      M=3
C*****
C * READ THE XK(TIME INTERVAL) & YK(LINK TRAVEL TIME) OF THE FUNCTION.      *
C*****

C*
C* WHEN TIM=1, IT MEANS THAT SIMULATION TIME IS 0.1 IN DYNASMART.
C*
      DO 1000 TIM=1,sim_tim-1
      DO 1002 LINK=1,NUM_LINK
      v(1)=volm(TIM-1,link)
      v(2)=volm(TIM,link)
      v(3)=volm(TIM+1,link)
      if((v(1).eq.v(2)).or.(v(1).eq.v(3)).or.(v(2).eq.v(3))) then
      if ((v(1).ne.v(3)).and.(v(2).ne.v(3))) then
      do 9001 i=40,sim_tim
      if((volm(i,link).ne.v(2)).and.(volm(i,link).ne.v(3)))then
      v(1)=volm(i,link)
      XK(0)=v(1)
      XK(1)=v(2)
      XK(2)=v(3)
      YK(0)=ETT(i,LINK)
      YK(1)=ETT(TIM,LINK)
      YK(2)=ETT(TIM+1,LINK)
      goto 9999
      else
      M=3
      end if
9001  continue
      else if((v(1).ne.v(2)).and.(v(2).ne.v(3)))then
      do 9002 i=40,sim_tim
      if((volm(i,link).ne.v(2)).and.(volm(i,link).ne.v(3)))then

      v(1)=volm(i,link)
      XK(0)=v(1)
      XK(1)=v(2)
      XK(2)=v(3)
      YK(0)=ETT(i,LINK)
      YK(1)=ETT(TIM,LINK)
      YK(2)=ETT(TIM+1,LINK)

```

```

        goto 9999
    else
        M=3
    end if
9002    continue

    else if((v(1).ne.v(2)).and.(v(1).ne.v(3))) then
        do 9003 i=40,sim_tim
            if((volm(i,link).ne.v(1)).and.(volm(i,link).ne.v(2)))then

                v(3)=volm(i,link)
                XK(0)=v(1)
                XK(1)=v(2)
                XK(2)=v(3)

                YK(0)=ETT(tim-1,LINK)
                YK(1)=ETT(TIM,LINK)
                YK(2)=ETT(i,LINK)
                goto 9999
            else
                M=3
            end if
9003    continue
        else
            do 9004 i=40,sim_tim
                if((volm(i,link).ne.v(1)).and.(volm(i-20,link).ne.v(1))
& .and.(volm(i,link).ne.volm(i-20,link))) then

                    v(1)=volm(i-20,link)
                    v(3)=volm(i,link)

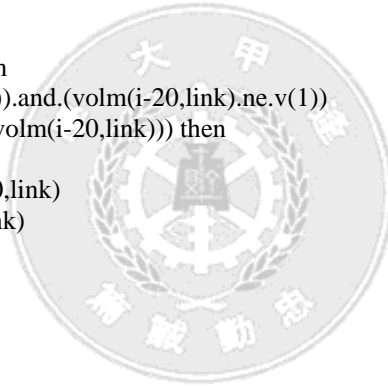
                    XK(0)=v(1)
                    XK(1)=v(2)
                    XK(2)=v(3)

                    YK(0)=ETT(tim-1,LINK)
                    YK(1)=ETT(TIM,LINK)
                    YK(2)=ETT(i,LINK)
                    goto 9999
                else
                    M=3
                end if
9004    continue
            end if
        else

            XK(0)=v(1)
            XK(1)=v(2)
            XK(2)=v(3)

            YK(0)=ETT(TIM-1,LINK)
            YK(1)=ETT(TIM,LINK)
            YK(2)=ETT(TIM+1,LINK)
        end if

```



```

C
C MAKE A NEW ARRAY FOR SOURCE XK ARRAY.
C
9999  DO 101 L=0,M-1
      XKA(L)=XK(L)
      XKC(L)=XK(L)
101  CONTINUE

C *****
C *   THE F'(X)                               *
C *****
      DO 2000 J=0,M-1
      PSUM(J)=1
      DO 2001 K=0,M-1
      IF(J.NE.K) then
        PSUM(J)=(XK(J)-XK(K))*PSUM(J)
      END IF
2001  CONTINUE
2000  CONTINUE

      DO 602 T=0,M-1
      XK(T)=XKA(T)
602  CONTINUE

      DO 103 P=1,M
      IF(P.EQ. 1) THEN
        DO 202 Q=0,M-2
        XKB(Q)=XKA(Q+1)
202  CONTINUE
        GOTO 301

      ELSE IF(P .GE. 2) THEN
        DO 204 PB=0,P-2
        XKB(PB)=XKC(PB)
204  CONTINUE
        IF(P .EQ. M) THEN
          GOTO 301
        ELSE
          RA=P-1
          DO 203 R=RA,M-2
          XKB(R)=XKC(R+1)

203  CONTINUE
          goto 301
        END IF
      END IF

301  S(0)=1
      SA(0)=1
      DO 902 Q=1,M-1
      SA(Q)=0
902  CONTINUE

```



```

DO 401 T=0,M-2
DO 402 U=0,M-2
S(U+1)=(XKB(T)*SA(U))+SA(U+1)
402 CONTINUE

DO 901 L=0,M-1
SA(L)=S(L)
901 CONTINUE
401 CONTINUE

W=M-1
DO 404 Z=1,M      !PUT THE OUPUT INTO MATRIX
MATA(P,Z)=(YK(P-1)/PSUM(P-1))*(SA(Z-1))
404 CONTINUE
103 CONTINUE
DO 701 B=1,M
DO 702 A=1,M-1
MATA(A+1,B)=MATA(A,B)+MATA(A+1,B)
702 CONTINUE
701 CONTINUE

C*
C* CALCULATE THE PARTIAL DERIVATIVE FOR EACH LINK ON THE CERTAIN TIME
C*

SLOPE(TIM,LINK)=(2*MATA(3,1)*volm(TIM,LINK))+MATA(3,2)
1002 CONTINUE
1000 CONTINUE
C*****
C* AVERAGING TECHNIQUE
C*****
C*
C* AVERAGING DERIVATIVE
C*

int_tmp=sim_tim/30
DO 1005 INTVL=1,int_tmp
TMP=INTVL*30
DO 1007 LINK=1,NUM_LINK
DO 1006 SS=TMP,TMP-28,-1
SLOPE(SS-1,LINK)=SLOPE(SS,LINK)+SLOPE(SS-1,LINK)
1006 CONTINUE
AVG_SLP(INTVL,LINK)=SLOPE(TMP-29,LINK)/30
1007 CONTINUE
1005 CONTINUE
avg_int=int_tmp/5
DO 1008 INTVL=1,avg_int
TMP=INTVL*5
DO 1009 LINK=1,NUM_LINK
DO 1010 SS=TMP,TMP-3,-1
AVG_SLP(SS-1,LINK)=AVG_SLP(SS,LINK)+AVG_SLP(SS-1,LINK)
1010 CONTINUE
MVG_AVSP(INTVL,LINK)=AVG_SLP(TMP-4,LINK)/5
1009 CONTINUE
1008 CONTINUE

```

```

        avg_int=int_tmp/5
        do i=1,avg_int
        do j=(i*50),(i*50-49),-1
        do link=1,num_link
            new_slp(j,link)=mvg_avsp(i,link)
        end do
        end do
        end do

C*
C* AVERAGING NUMBER OF VEHICLES ON THE CERTAIN LINK
C*
        DO 1011 INTVL=1,int_tmp
        TMP=INTVL*30
        DO 1012 LINK=1,NUM_LINK
        DO 1013 SS=TMP,TMP-28,-1
            VOLM(SS-1,LINK)=VOLM(SS,LINK)+
& VOLM(SS-1,LINK)
1013    CONTINUE
        AVG_VEH(INTVL,LINK)=VOLM(TMP-29,LINK)/30
1012    CONTINUE
1011    CONTINUE
        do i=1,int_tmp
        do j=(i*30),(i*30-9),-1
        do link=1,num_link
            new_veh(j,link)=avg_veh(i,link)
        end do
        end do
        end do

C*****
C* ADD THE LAST PART OF DYNASMART-X, THE EFFECT OF THE
C* NEIGHBORING(UPSTREAM) LINKS.
C*****
C* READING SPACE OF FREESLOT(FORT.25) AND NUMBER OF VEHICLES WISH GOING
C* INTO THE CURRENT LINKS(FORT.23) FROM NEIGHBORING LINKS.
C*
        rewind(23)
        rewind(25)
        do i=0,sim_tim-1
            read(23,*)
            read(23,*)
            read(25,*)
            read(23,2301) (intoo(a),a=1,num_link)
2301    format(10f8.3)
            read(25,2501) (freeslot(a),a=1,num_link)
2501    format(10f8.3)
            read(23,*)
            read(25,*)
            do a=1,num_link
                if ((intoo(a)-freeslot(a)) .gt. 0.0) then
                    diff(i,a)=intoo(a)-freeslot(a)
                else
                    diff(i,a)=0.0
                endif
            end do
        end do
        end do

```



```

C*
C*  CALCULATE MARGINAL COSTS OF EFFECT OF NEIGHBORING LINKS.
C*
      do tim=1,sim_tim-1
      do a=1,num_link
      nebrg_mgn(tim,a)=diff(tim,a)*slope(tim,a)
      end do
      end do
C*
C*  CALCULATE THE MARGINAL COSTS FOR EACH LINK
C*
      DO 1003 TIM=1,sim_tim-1

      DO 1004 LINK=1,NUM_LINK
      DT_MTT(TIM,LINK)=ETT(TIM,LINK)+
& (NEW_VEH(TIM,LINK)*NEW_SLP(TIM,LINK))
& +NEBRG_MGN(TIM,LINK)
1004  CONTINUE
1003  CONTINUE
1014  CONTINUE
      TMP=SIM_TIM-(avg_int*5)

      IF(TMP.GE.1)THEN
      DO TIM=(avg_int*5),SIM_TIM-1
      DO LINK=1,NUM_LINK
      DT_MTT(TIM,LINK)=ETT(TIM,LINK)+NEBRG_MGN(TIM,LINK)
      END DO
      END DO
      ELSE
      TMP=TMP
      END IF
C*
C*  CHECK THE MARGINAL COSTS IF GREATER THAN EQUAL TO TEMP 999.0000.
C*
      temp=999.0000
      DO 1019 TIM=1,int_tmp
      DO 1020 LINK=1,NUM_LINK
      IF (DT_MTT(TIM,LINK).GT.TEMP)THEN
      DT_MTT(TIM,LINK)=TEMP
      ELSE IF(DT_MTT(TIM,LINK).LT.0)THEN
      DT_MTT(TIM,LINK)=0
      ELSE
      DT_MTT(TIM,LINK)=DT_MTT(TIM,LINK)
      END IF
1020  CONTINUE
1019  CONTINUE
      do i=1,nst
      do j=1,na
      mtt(j,i,1)=dt_mtt(i,j)
      mtt(j,i,2)=mtt(j,i,1)
      END DO
      END DO
      return
      END
  
```

附錄二 局部邊際延滯求解程式碼

```

subroutine lmd()
c*****
c* This is a program to calculate LMD.
c* Edited by D.T. Li.
c*****
    integer k,divid,l
    integer volm_row,que_row,tim_int,row_num,num_link
    real delay
    real temp
    real time(1000),dtime(1000)
    integer volm(1000,1000),outflow(1000,1000)
    real volg(1000,1000)
    real lnk_que(1000,1000),arvl_rt(1000,1000),arvl_cum(1000,1000)
    real dptr_cum(1000,1000),slp_arvl(1000,1000),slp_dptr(1000,1000)
    real intr_mgn(1000,1000),lmd_mtt(1000,1000)
    integer nst,na
    common /nst/nst
    common /na/na
    integer sim_tim
    integer tmp
    parameter (li_nu=441)
    parameter (nu_sim=1000)
    real mtt
    dimension mtt(li_nu,nu_sim,2)
    common /mtt/mtt

    if(nst.gt.1000)then
        nst=1000
    else
        nst=nst
    end if
    num_link=na
    sim_tim=nst
    divid=num_link/10
    l=num_link-(divid*10)

c-
c- read the volume matrix. (fort.31)
c-
    rewind(31)
    k=divid-1
    volm_row=1
    do 200 i=1,sim_tim
        read(31,*) time(i)
        do 201 j=0,k
            read(31,'(10I5)') (volm(volm_row,j*10+n),n=1,10)
201        continue
            read(31,'(10I5)') (volm(volm_row,divid*10+m),m=1,l)
            volm_row=volm_row+1
200    continue

```



```

c-
c- read the outflow matrix(fort.39)
c-
      do 202 i=1,sim_tim
        read(39,*) dttime(i)
        k=divid-1
        do 203 j=0,k
          read(39,'(10I8)') (outflow(i,j*10+n),n=1,10)
203      continue
          read(39,'(10I8)') (outflow(i,divid*10+m),m=1,l)
202  continue
        rewind(30)
c-
c- read the volume generation matrix. (fort.30)
c-
      k=divid-1
      do 204 i=1,sim_tim
        read(30,*)time(i)
        do 205 j=0,k
          read(30,'(10f8.1)') (volg(i,j*10+n),n=1,10)
205      continue
          read(30,'(10f8.1)') (volg(i,divid*10+m),m=1,l)
204  continue
        do 208 tim_int=2,sim_tim
          do 209 link=1,num_link
            arvl_rt(tim_int,link)= volm(tim_int,link)+outflow(tim_int,link)
            &-volm(tim_int-1,link)
209          continue
208        continue

c-
c- cumulative arrival curve
c-

c* initial 0.1 time interval cumulative
      do 214 link=1,num_link
        arvl_cum(1,link)= arvl_rt(1,link)
214      continue

c* after 0.1 time interval arrival cumulative
      do 212 tim_int=2,sim_tim
        do 213 link=1,num_link
          arvl_cum(tim_int,link)= arvl_rt(tim_int,link)
          &+arvl_cum(tim_int-1,link)
213      continue
212      continue

c-
c- cumulative departure curve
c-
c* initial 0.0 minute interval

      do 215 link=1,num_link
        dptr_cum(1,link) = outflow(1,link)
215      continue

```



```

c* after 0.0 time interval departure cumulative
  do 216 tim_int= 2,sim_tim
    do 217 link= 1,num_link
      dptr_cum(tim_int,link)=(outflow(tim_int,link)
&+dptr_cum(tim_int-1,link))
217   continue
216   continue
c*****starting marginal calculation*****
c
c- calculation of marginal for each assignment time-interval
c-
  tmp=sim_tim/10
  do 400 tim_int=4,tmp
    row_num=tim_int*10
    arvl_row=row_num-30
    do 401 link=1,num_link
      lnk_que(row_num,link)=arvl_cum(row_num,link)
&      -dptr_cum(row_num,link)
      if (lnk_que(row_num,link).GT.0) then

c* check the slope of arrival and departure curve
        slp_arvl(row_num,link)=(arvl_cum(row_num,link)
&      -arvl_cum(arvl_row,link))/3
        slp_dptr(row_num,link)=(dptr_cum(row_num,link)
&      -dptr_cum(arvl_row,link))/3
        if (slp_dptr(row_num,link).eq.0) then
          slp_dptr(row_num,link)=0.000001
        else
          slp_dptr(row_num,link)= slp_dptr(row_num,link)
        endif
        if (slp_arvl(row_num,link).LT.slp_dptr(row_num,link))then

c          *** situation (a): the slope of arrival > 0 ***
          if (slp_arvl(row_num,link).GT.0) then
            intr_mgn(row_num,link)=((arvl_cum(arvl_row,link)
&      -dptr_cum(arvl_row,link))/(slp_dptr(row_num,link)
&      -slp_arvl(row_num,link)))-3

c          *** situation (b): the slope of arrival = 0 ***
          else if(slp_arvl(row_num,link).EQ.0) then
            intr_mgn(row_num,link)=((arvl_cum(arvl_row,link)
&      -dptr_cum(arvl_row,link))/
&      slp_dptr(row_num,link))-3

c          *** situation (c): the slope of arrival < 0 (negative slope) ***
          else
            intr_mgn(row_num,link)=((arvl_cum(arvl_row,link)
&      -dptr_cum(arvl_row,link))/
&      (slp_dptr(row_num,link)-slp_arvl(row_num,link)))-3
          endif
        else
c* calculate "d"
          delay=lnk_que(row_num,link)/slp_dptr(row_num,link)
          tt=(delay*slp_arvl(row_num,link))/slp_dptr(row_num,link)
          intr_mgn(row_num,link)=delay+tt
        end if
      end do
    end do
  end do

```

```

else
  intr_mgn(row_num,link)=0
end if
401   continue
400   continue
do 700 i=4,tmp
  p=i*10
do 701 j=1,num_link
  lmd_mtt(p,j)=intr_mgn(p,j)
701 continue
700 continue
do i=4,tmp
  p=i*10
do j=1,num_link
  if (lmd_mtt(p,j).gt.999.0000)then
    lmd_mtt(p,j)=999.000
  else
    lmd_mtt(p,j)=lmd_mtt(p,j)
  end if
end do
end do

  do i=1,nst
  do j=1,na
  if(i.lt.41)then
    mtt(j,i,1)=lmd_mtt(j,4)
    mtt(j,i,2)=lmd_mtt(j,4)
  else
    mtt(j,i,1)=lmd_mtt(j,i/10)
    mtt(j,i,2)=mtt(j,i,1)
  end if
  END DO
  END DO
return
end

```



附錄三 Peeta 之邊際成本表示方法求解程式碼

```

subroutine peeta()
implicit none
C *****
C ** This is a prg to calculate marginal cost,
C ** it is the method developed by Peeta.
C ** EDITED BY DA,TUNG-LI
C *****
real M,N
real A,B,I,J,JJ,K,L,P,Q,R,T,U,W,Z,PA,PB,RA

C*****
C***** VARIABLES WHICH USED TO CALCULATE MARGINAL ***
C* ETT: ESTIMATED TRAVEL TIME
C* VOLM: VEHICLES ON LINK ON THE CERTAIN TIME.
C* NUM_LINK: NUMBER OF LINKS
C* SLOPE: THE MARGINAL OF THE ADDITIONAL VEHICLE
C* MTT: MARGINAL TRAVEL TIME OF THE CERTAIN LINK
C* AVG_SLP: AVERAGE DERIVATIVE
C* MVG_AVSP: MOVING AVERAGE OF THE DERIVATIVE
C*****
REAL TTM_MTX(1000,1000), ETT(0:1000,1000)
INTEGER volm(0:1000,1000)
INTEGER NUM_LINK, TIM ,LINK
REAL TIME(1000) ,v(10)
REAL SLOPE(1000,1000)
REAL DT_MTT(1000,1000)
REAL AVG_SLP(1000,1000)
REAL MVG_AVSP(1000,1000),avg_flow(0:1000,1000)
real tt_slot(0:1000,1000)
INTEGER INTVL
INTEGER SIM_INT
real AVG_VEH(1000,1000)
INTEGER TMP,SS,ST
INTEGER KK,LL,NN,ROW_NUM,MM
real temp
integer sim_tim,int_tmp,avg_int,int,li_nu,nu_sim
integer avfltmp(1000)
REAL MTT
parameter (li_nu=441)
parameter (nu_sim=1000)
DIMENSION MTT(li_nu,nu_sim,2)
integer nst,na
COMMON /MTT/MTT
COMMON /NST/NST
COMMON /NA/NA
real new_veh(1000,500), new_slp(1000,500)

if(nst.gt.1000)then
nst=1000
else
nst=nst
end if
SIM_TIM=NST
NUM_LINK=NA

```

```

C*****
C Read travel time matrix.
C*****
      rewind(79)
      DO 1018 I=0,sim_tim-1
      READ(79,*)TIME(I)
        READ(79,'(10f8.3)') (tt_slot(I,J),J=1,NUM_LINK)
1018  CONTINUE
      rewind(76)
C*****
C* read average flow rate.(fort.76)
C*****
      read(76,'(20i4)')(avfltmp(i),i=1,num_link)
      do 113 i=0,sim_tim-1
      read(76,*)time(i)
      read(76,'(10f8.3)') (avg_flow(i,j),j=1,num_link)
113  continue
C*****
C* reading volume on all links.
C*****
      rewind(31)
      DO 1016 I=0,sim_tim-1
      READ(31,*)TIME(I)
      READ(31,'(10I5)') (volm(I,J),j=1,num_link)
1016  CONTINUE
C*****
C* calculate travel time.
C*****
      write(*,*)'start calculating travel time'
      do i=0,sim_tim-1
      do j=1,num_link
      if (avg_flow(i,j).eq.0) then
      avg_flow(i,j)=0.005
      end if
      ETT(i,j)=tt_slot(i,j)+(volm(i,j)/(60*avg_flow(i,j)))
      end do
      end do
C*****
C * The number of initial solution is 3, because using t-1, t, t+1.      *
C*****
      M=3
C*****
C * READ THE XK(TIME INTERVAL) & YK(LINK TRAVEL TIME) OF THE FUNCTION.      *
C*****
C*
C* WHEN TIM=1, IT MEANS THAT SIMULATION TIME IS 0.1 IN DYNASMART.
C*
      DO 1000 TIM=1,sim_tim-1
      DO 1002 LINK=1,NUM_LINK
      v(1)=volm(TIM-1,link)
      v(2)=volm(TIM,link)
      v(3)=volm(TIM+1,link)
      if(v(1).eq.v(3))then
      do i=0,sim_tim

          if(volm(i,link).ne.v(1))then
          v(3)=volm(i,link)

```

```

        if ((V(1)-v(3)).eq.0)then
            temp=0.001
            slope(tim,link)=(ETT(TIM-1,LINK)-ETT(I,LINK))/temp
            go to 1002
        else
            slope(tim,link)=(ETT(TIM-1,LINK)-ETT(I,LINK))/(v(1)-v(3))
            go to 1002
        end if
    else
        M=M
    END IF
end do

else
    M=M
end if

1002 CONTINUE
1000 CONTINUE
C*****
C* AVERAGING TECHNIQUE
C*****
C* AVERAGING DERIVATIVE
    int_tmp=sim_tim/30
    DO 1005 INTVL=1,int_tmp
        TMP=INTVL*30
        DO 1007 LINK=1,NUM_LINK
            DO 1006 SS=TMP,TMP-8,-1
                SLOPE(SS-1,LINK)=SLOPE(SS,LINK)+SLOPE(SS-1,LINK)
1006 CONTINUE
                AVG_SLP(INTVL,LINK)=SLOPE(TMP-29,LINK)/30
1007 CONTINUE
1005 CONTINUE
        avg_int=int_tmp/5
        DO 1008 INTVL=1,avg_int
            TMP=INTVL*5
            DO 1009 LINK=1,NUM_LINK
                DO 1010 SS=TMP,TMP-3,-1
                    AVG_SLP(SS-1,LINK)=AVG_SLP(SS,LINK)+AVG_SLP(SS-1,LINK)
1010 CONTINUE
                    MVG_AVSP(INTVL,LINK)=AVG_SLP(TMP-4,LINK)/5
1009 CONTINUE
1008 CONTINUE
            avg_int=int_tmp/5
            do i=1,avg_int
                do j=(i*150),(i*150-149),-1
                    do link=1,num_link
                        new_slp(j,link)=mvg_avsp(i,link)
                    end do
                end do
            end do
        C* AVERAGING NUMBER OF VEHICLES ON THE CERTAIN LINK
        DO 1011 INTVL=1,int_tmp
            TMP=INTVL*30
            DO 1012 LINK=1,NUM_LINK
                DO 1013 SS=TMP,TMP-28,-1
                    VOLM(ss-1,LINK)=VOLM(ss,LINK)+
& VOLM(ss-1,LINK)

```



```

1013  CONTINUE
      AVG_VEH(INTVL,LINK)=VOLM(TMP-29,LINK)/30
1012  CONTINUE
1011  CONTINUE

      do i=1,int_tmp
      do j=(i*30),(i*30-29),-1
      do link=1,num_link
          new_veh(j,link)=avg_veh(i,link)
      end do
      end do
      end do

C* CALCULATE THE MARGINAL COSTS FOR EACH LINK.
      DO 1003 TIM=1,sim_tim-1
      DO 1004 LINK=1,NUM_LINK
          DT_MTT(TIM,LINK)=ETT(TIM,LINK)+
          & (NEW_VEH(TIM,LINK)*NEW_SLP(TIM,LINK))
1004  CONTINUE
1003  CONTINUE
1014  CONTINUE

      TMP=SIM_TIM-1-(avg_int*5)
      IF(TMP.GE.1)THEN
          DO TIM=(avg_int*5),SIM_TIM
          DO LINK=1,NUM_LINK
              DT_MTT(TIM,LINK)=ETT(TIM,LINK)
          END DO
          END DO
      ELSE
          TMP=TMP
      END IF

C* CHECK THE MARGINAL COSTS IF GREATER THAN EQUAL TO TEMP 999.0000.
      temp=999.0000
      DO 1019 TIM=1,int_tmp
      DO 1020 LINK=1,NUM_LINK
          IF (DT_MTT(TIM,LINK).GT.TEMP)THEN
              DT_MTT(TIM,LINK)=TEMP
          ELSE IF(DT_MTT(TIM,LINK).LT.0)THEN
              DT_MTT(TIM,LINK)=0
          ELSE
              DT_MTT(TIM,LINK)=DT_MTT(TIM,LINK)
          END IF
1020  CONTINUE
1019  CONTINUE
      do i=1,nst
      do j=1,na
          mtt(j,i,1)=dt_mtt(i,j)
          mtt(j,i,2)=mtt(j,i,1)
      END DO
      END DO

      return
      END

```

附錄四 局部邊際旅行時間求解程式碼

```

subroutine wei()
implicit none
c***** document*****
c*****
c This program is the method of calculation of marginal
c with the wei's method which was edited by dtli.
c This program has two parts. The first part, using the velocity
c of each time interval to calculate the "link" marginal traveling cost.
c The second part calculates the intersection delay.
c*****

c-----
c The first part of this program is the
c calculation of "link" marginal cost.
c-----

c***** variables definition*****
c*****
c velcty :the velocity of current time slice
c empty: this variable is used reading the first line of each time interval
c time: time interval (0.1 minute)
c num_link: total number of links in the network
c row_num: the velocity matrix's row number,
c each row is the velocity of all links in a time interval
c link_mgn: marginal cost of each link

c- claim variables
c***** variables for the first part*****
real velocity(1000,1000), time(1000), empty(1000)
integer i,j,k,l,m,n,p,q
integer row_num, num_link, divid, clmn
real link_mgn(1000,500)

c***** variables for the second part*****
integer volm(1000,500), outflow(1000,500), volg(1000,500)
real arvl_rt(1000,500), lnk_que(1000,500), tt(500,2)
integer volm_row, que_row, tim_int, link ,delay ,arvl_row
real intr_mgn(1000,500) ,arvl_cum(1000,500) ,dptr_cum(1000,500) ,dtime(1000)
real temp
integer tmp
integer sim_tim
real wei_mtt(1000,500),tt_slot(1000,500)
real avg_flow(1000,500),diff_tt(1000,500)
integer int_tmp,tim
integer li_nu,nu_sim
parameter (li_nu=441)
parameter (nu_sim=1000)
integer nst,na
common /nst/nst
common /na/na
real mtt
dimension mtt(li_nu,nu_sim,2)
common /mtt/mtt
real avfltmp(1000)

```

```

        if(nst.gt.1000)then
            nst=1000
        else
            nst=nst
        end if

        num_link=na
        sim_tim=nst-1
c*****
c- read the volume matrix. (fort.31)
c*****
        rewind(31)
        divid=num_link/10
        k=divid-1
        l=num_link-(divid*10)
        volm_row=1
        do 200 i=1,sim_tim
            read(31,*)time(i)
            do 201 j=0,k
                read(31,'(10I5)') (volm(volm_row,j*10+n),n=1,10)
201            continue
                read(31,'(10I5)') (volm(volm_row,divid*10+m),m=1,l)
                volm_row=volm_row+1
200        continue
        rewind(76)
c*****
c* read average flow rate.(fort.76)
c*****
        k=divid-1
        row_num=1
        read(76,'(20i4)')(avfltmp(i),i=1,num_link)
        do 113 i=1,sim_tim
            read(76,*)time(i)
            do 114 j=0,k
                read(76,'(10f8.3)') (avg_flow(row_num,j*10+n),n=1,10)
114            continue
                read(76,'(10f8.3)') (avg_flow(row_num,divid*10+m),m=1,l)
                row_num=row_num+1
113        continue
        rewind(79)
c*****
c* read travel time of free slot.(fort.79)
c*****
        DIVID = NUM_LINK/10
        K= DIVID-1
        L= NUM_LINK-(DIVID*10)
        ROW_NUM=1
        DO 115 I=1,sim_tim
            READ(79,*)TIME(I)
            DO 116 J=0,K
                READ(79,'(10f8.3)') (tt_slot(ROW_NUM,J*10+N),N=1,10)
116            CONTINUE
                READ(79,'(10f8.3)') (tt_slot(ROW_NUM,DIVID*10+M),M=1,L)
                ROW_NUM=ROW_NUM+1
115            CONTINUE

```

```

C*****
c-- calculation of "link" marginal cost link_mgn(i,k)
C*****
      do 900 i=1,sim_tim
      do 901 p=1,num_link
      if(avg_flow(i,p).eq.0) then
      avg_flow(i,p)=0.001
      end if
      if(avg_flow(i+1,p).eq.0)then
      avg_flow(i+1,p)=0.500
      end if
      diff_tt(i,p)=(tt_slot(i+1,p)+(volm(i+1,p)/(60*avg_flow(i+1,p))))
      &-(tt_slot(i,p)+(volm(i,p)/(60*avg_flow(i,p))))
901   continue
900   continue

      do 102 i=1,sim_tim
      do 103 p=1,num_link
      if (volm(i,p).eq.volm(i+1,p)) then
      link_mgn(i,p)=diff_tt(i,p)
      else
      link_mgn(i,p)=diff_tt(i,p)*volm(i,p)/(volm(i+1,p)-volm(i,p))
      end if
103   continue
102   continue

c-----
c The second part of this program is the
c calculation of "intersection" marginal cost.
c-----

C*****
c- read the outflow matrix(fort.39)
C*****
      rewind(39)
      do 202 i=1,sim_tim+1
      read(39,*) dtime(i)
      k=divid-1
      do 203 j=0,k
      read(39,'(10I8)') (outflow(i,j*10+n),n=1,10)
203   continue
      read(39,'(10I8)') (outflow(i,divid*10+m),m=1,l)
202   continue
      rewind(30)
C*****
c- read the volume generation matrix. (fort.30)
C*****
      k=divid-1
      do 204 i=1,sim_tim
      read(30,*)time(i)
      do 205 j=0,k
      read(30,'(10f8.1)') (volg(i,j*10+n),n=1,10)
205   continue
      read(30,'(10f8.1)') (volg(i,divid*10+m),m=1,l)
204   continue

```

```

c*****
c- calculation of arrival rate.<arrival=inflow+volum generation>
c*****
c-   arrival rate(t+1)=volume(t+1)+outflow(t+1)-volume(t)
      do 208 tim_int=2,sim_tim
        do 209 link=1,num_link
          arvl_rt(tim_int,link)= volm(tim_int,link)+outflow(tim_int,link)
          &-volm(tim_int-1,link)
209       continue
208     continue
c-
c- cumulative arrival curve
c-
c* initial 0.1 time interval cumulative
      do 214 link=1,num_link
        arvl_cum(1,link)= arvl_rt(1,link)
214     continue
c* after 0.1 time interval arrival cumulative
      do 212 tim_int=2,sim_tim
        do 213 link=1,num_link
          arvl_cum(tim_int,link)= arvl_rt(tim_int,link)
          &+arvl_cum(tim_int-1,link)
213       continue
212     continue

c- cumulative departure curve

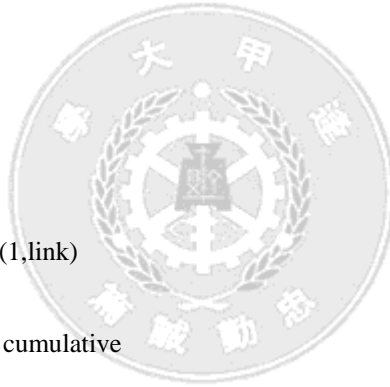
c* initial 0.0 minute interval

      do 215 link=1,num_link
        dptr_cum(1,link) = outflow(1,link)
215     continue

c* after 0.0 time interval departure cumulative
      do 216 tim_int= 2,sim_tim
        do 217 link= 1,num_link
          dptr_cum(tim_int,link)=(outflow(tim_int,link)
          &+dptr_cum(tim_int-1,link))
217       continue
216     continue

C*****starting marginal calculation*****
c
c- searching the point of that dptr cumulative being equal to arvl cumulative.
c-
      do i=1,num_link
        do j=1,2
          tt(i,j)=0
        end do
      end do
      do i=1,sim_tim
        do j=1,num_link
          intr_mgn(i,j)=0.000
        end do
      end do

```



```

do 400 j=1,num_link
do 401 i=2,sim_tim
if (dptr_cum(i,j).ne.arvl_cum(i,j)) then
  tt(j,1)=i-1
  go to 402
else
  tt(j,1)=tt(j,1)
end if
401 continue
402 do k=i,sim_tim
  if((dptr_cum(k,j).eq.arvl_cum(k,j)).and.(dptr_cum(k,j).ne.0))then
    tt(j,2)=k
  else
    tt(j,2)=tt(j,2)
  end if
end do
400 continue
do 403 j=1,num_link
do 404 i=1,sim_tim
if((i.lt.tt(j,1)).and.(i.lt.tt(j,2))) then
  intr_mgn(i,j)=real(0)
else if(i.eq.tt(j,1))then
  intr_mgn(i,j)=(tt(j,2)-i)/10
else if((i.gt.tt(j,1)).and.(i.lt.tt(j,2))) then
  intr_mgn(i,j)=(tt(j,2)-i)/10
else if(i.ge.tt(j,2))then
  intr_mgn(i,j)=real(0)
else
  intr_mgn(i,j)=0.000
end if
404 continue
403 continue
do 700 i=1,sim_tim
do 701 j=1,num_link
  wei_mtt(i,j)=link_mgn(i,j)+(volm(i,j)*intr_mgn(i,j))
701 continue
700 continue
  temp=999.0000
  DO 1019 TIM=1,sim_tim
  DO 1020 LINK=1,NUM_LINK
  IF (wei_mtt(TIM,LINK).GT.TEMP)THEN
    wei_mtt(TIM,LINK)=TEMP
  ELSE if (wei_mtt(TIM,LINK).LT.0)THEN
    wei_mtt(TIM,LINK)=0
  ELSE
    wei_mtt(TIM,LINK)=wei_mtt(TIM,LINK)
  END IF
1020 CONTINUE
1019 CONTINUE
  do i=1,nst
  do j=1,na
    mtt(j,i,1)=wei_mtt(i,j)
    mtt(j,i,2)=mtt(j,i,1)
  END DO
  END DO
return
end

```



附錄五 依時性邊際旅行時間求解程式碼

```

program tdlmsp
implicit none
c*****
c This is a program to calculate the time-dependent marginal travel time of the all paths
c of the given OD pair.
c
c EDITED BY DTLI APR. 3,2002
c*****

integer i,j,k,l,n,m,temp
integer num_link,divid,row_num
integer path(100,100),orgn(100),des(100)
integer linkid(100,100),link(100)
integer sim_tim
real avg_flow(0:1000,1000),tt_slot(0:1000,1000)
integer volm(0:1000,1000)
real time(0:1000)
real mgn_lnk(1000,1000)
real trvl_tim(0:1000,1000)
dimension mgn_path(1000,30,30)
real mgn_path
integer en_tim
dimension cum_tim(1000,30,30)
real cum_tim

write(*,*)'Input number of links.'
read(*,*)num_link
write(*,*)'Input length of simulation time.'
read(*,*)sim_tim
c*****
c* Read the test path.(testpath.dat)
c*****
OPEN(UNIT=100,FILE='testpath.dat',STATUS='OLD')
do 001 i=1,12
read(100,'(6I3)') (path(i,j),j=1,6)
001 continue
do 002 i=13,20
read(100,'(8I3)') (path(i,j),j=1,8)
002 continue
do 003 i=21,22
read(100,'(10I3)') (path(i,j),j=1,10)
003 continue
OPEN(UNIT=101,FILE='linkid.dat',STATUS='OLD')
do 004 i=1,24
read(101,'(3I5)')link(i),orgn(i),des(i)
004 continue
c*****
c* identify the link's i.d. of each path.
c*****
do 100 i=1,12
do 101 j=1,4
do 102 k=1,24
if((path(i,j+1).eq.orgn(k)).and.(path(i,j+2).eq.des(k)))then
linkid(i,j)=link(k)
end if

```

```

102   continue
101   continue
100   continue

      do 103 i=13,20
      do 104 j=1,6
      do 105 k=1,24
      if((path(i,j+1).eq.orgn(k)).and.(path(i,j+2).eq.des(k)))then
          linkid(i,j)=link(k)
      end if
105   continue
104   continue
103   continue

      do 106 i=21,22
      do 107 j=1,8
      do 108 k=1,24
      if((path(i,j+1).eq.orgn(k)).and.(path(i,j+2).eq.des(k)))then
          linkid(i,j)=link(k)
      end if
108   continue
107   continue
106   continue

c*****
c* read the marginal cost of each link on certain time.(output.100)
c*****
      divid=num_link/10
      k=divid-1
      l=num_link-(divid*10)
      OPEN(UNIT=102,FILE='output.100',STATUS='OLD')
      temp=sim_tim/10
      row_num=1
      do 109 i=1,temp-1
      read(102,*) time(i)
      do 110 j=0,k
      read(102,'(10f8.4)') (mgn_lnk(row_num,j*10+n),n=1,10)
110   continue
      read(102,'(10f8.4)') (mgn_lnk(row_num,divid*10+m),m=1,l)
      row_num=row_num+1
109   continue

c*****
c* read volume on link.(fort.31)
c*****
      OPEN(UNIT=31,FILE='fort.31',STATUS='OLD')
      k=divid-1
      row_num=0
      do 111 i=0,sim_tim
      read(31,*)time(i)
      do 112 j=0,k
      read(31,'(10I5)') (volm(row_num,j*10+n),n=1,10)
112   continue
      read(31,'(10I5)') (volm(row_num,divid*10+m),m=1,l)
      row_num=row_num+1
111   continue

```



```

c*****
c* read average flow rate.(fort.76)
c*****
      OPEN(UNIT=76,FILE='fort.76',STATUS='OLD')
      k=divid-1
      row_num=0
      do 113 i=0,sim_tim
        read(76,*)time(i)
        do 114 j=0,k
          read(76,'(10f8.4)') (avg_flow(row_num,j*10+n),n=1,10)
114      continue
          read(76,'(10f8.4)') (avg_flow(row_num,divid*10+m),m=1,l)
          row_num=row_num+1
113  continue

c*****
c* read travel time of free slot.(fort.79)
c*****
      DIVID = NUM_LINK/10
      K= DIVID-1
      L= NUM_LINK-(DIVID*10)
      OPEN(UNIT=79,FILE='fort.79',STATUS='OLD')
      ROW_NUM=0
      DO 115 I=0,sim_tim
        READ(79,*)TIME(I)
        DO 116 J=0,K
          READ(79,'(10f8.4)') (tt_slot(ROW_NUM,J*10+N),N=1,10)
116  CONTINUE
          READ(79,'(10f8.4)') (tt_slot(ROW_NUM,DIVID*10+M),M=1,L)
          ROW_NUM=ROW_NUM+1
115  CONTINUE

c*****
c* Calculate the travel time of each link.
c*****
      do 117 i=0,sim_tim
        do 118 j=1,num_link
          trvl_tim(i,j)=tt_slot(i,j)+(volm(i,j)/(60*avg_flow(i,j)))
118  continue
117  continue
c*****
c* calculate cumulative travel time of path 1~path 12.
c*****
c*
c* calculate cumulative travel time of path 1~path 12.
c*
      temp=sim_tim/10
      do 900 k=4,temp
        do 119 i=1,12
          cum_tim(k,i,1)=trvl_tim(k*10,linkid(i,1))
          do 120 j=2,4
            cum_tim(k,i,j)=trvl_tim((k*10+(cum_tim(k,i,j-1))*10),linkid(i,j))
            &+cum_tim(k,i,j-1)
120  continue
119  continue

```

```

c*
c* calculate cumulative travel time of path 13~path 20.
c*
    do 121 i=13,20
        cum_tim(k,i,1)=trvl_tim(k*10,linkid(i,1))
        do 122 j=2,6
            cum_tim(k,i,j)=trvl_tim((k*10+(cum_tim(k,i,j-1)*10)),linkid(i,j))
            &+cum_tim(k,i,j-1)
122    continue
121    continue
c*
c* calculate cumulative travel time of path 21~path 22.
c*
    do 123 i=21,22
        cum_tim(k,i,1)=trvl_tim(40,linkid(i,1))
        do 124 j=2,8
            cum_tim(k,i,j)=
            &trvl_tim((40+(cum_tim(k,i,j-1)*10)),linkid(i,j))+cum_tim(k,i,j-1)
124    continue
123    continue
900    continue
c*****
c* Begin to calculate marginal cost of each given path.
c*****
c*
c* calculate time-dependent marginal travel cost of path 1~path 12.
c*
    OPEN(UNIT=200,FILE='pthmgn.out',STATUS='old')
    do 901 k=4,temp
        do 200 i=1,12
            mgn_path(k,i,1)=mgn_lnk(k,linkid(i,1))
            do 201 j=2,4
                mgn_path(k,i,j)=mgn_lnk(k+(cum_tim(k,i,j-1)/10),linkid(i,j))
201    continue
200    continue
c*
c* calculate time-dependent marginal travel cost of path 13~path 20.
c*
    do 202 i=13,20
        mgn_path(k,i,1)=mgn_lnk(k,linkid(i,1))
        do 203 j=2,6
            mgn_path(k,i,j)=mgn_lnk(k+(cum_tim(k,i,j-1)/10),linkid(i,j))
203    continue
202    continue

c*
c* calculate time-dependent marginal travel cost of path 21~path 22.
c*
    do 204 i=21,22
        mgn_path(k,i,1)=mgn_lnk(k,linkid(i,1))
        do 205 j=2,8
            mgn_path(k,i,j)=mgn_lnk(k+(cum_tim(k,i,j-1)/10),linkid(i,j))
205    continue
204    continue

```

```

c*
c* Summation of time-dependent marginal cost of each path.
c*

      do 206 i=1,12
      do 207 j=4,2,-1
mgn_path(k,i,j-1)=mgn_path(k,i,j)+mgn_path(k,i,j-1)
207  continue
206  continue
      do 208 i=13,20
      do 209 j=6,2,-1
mgn_path(k,i,j-1)=mgn_path(k,i,j)+mgn_path(k,i,j-1)
209  continue
208  continue
      do 210 i=21,22
      do 211 j=8,2,-1
mgn_path(k,i,j-1)=mgn_path(k,i,j)+mgn_path(k,i,j-1)
211  continue
210  continue
c*****
c* write output.
c*****
      write(200,*)'time=',k
      do 301 i=1,22
write(200,*) 'path',i,'.',mgn_path(k,i,1)
301  continue
901  continue

      end

```

