

# 階層式交通道路模型之應用

黃永發

朝陽科技大學  
資訊與通訊系教授  
yfahuang@cyut.edu.tw.

林佳怡

朝陽科技大學  
資訊與通訊系碩士  
s9930601@cyut.edu.tw

許重信

朝陽科技大學  
資訊與通訊系碩士班研究生  
billy10353@yahoo.com.tw

## 摘要

本文提出簡易階層式交通道路模型(Simple Hierarchy Route Networks Model, SHRNM), 依道路的等級、加上不同的行車速率及交通狀態, 利用Dijkstra演算法來進行最快路徑搜尋, 測試其可行性, 並與Google Map之最快道路功能比較, 並以台中市交通道路進行模擬, 來驗證所提出之道路模型之效益。由測試結果可知, 用9個路徑作比較, 得其平均平方誤差為 $9.5(\text{min}^2)$ 。

**關鍵詞：**最短路徑規劃、交通路徑規劃、Dijkstra 演算法、Google Map

## Abstract

In this paper, a simplified hierarchy road network model (SHRNM) is proposed for the optimal shortest road route discovery for transportation systems in metropolitan area. The Dijkstra algorithms are applied for the optimal path calculation. In our model, the metric of shortest path is the driving time. Thus, the shortest path can easily be found in proposed simple road traffic model. To verify the feasibility of SHRNM, the shortest path for the Taichung transportation roads, is performed to compare with Google Map. From the results, the mean square error (MSE) between SHRNM and Google Map is  $9.5 (\text{min}^2)$  with 9 routes discovery.

**Keywords:** Shortest Road Route Discovery, Transportation Systems, Dijkstra Algorithms, Google Map.

## 1. 前言

現今智慧型運輸系統(Intelligent Transportation Systems, ITS)的蓬勃發展及全球定位系統(Global Positioning System, GPS)導航裝置的普遍應用, 使得路徑規劃成為我們所依賴的功能。然而, 目前我們最常使用的Google

Map的路徑規劃與我的地圖等功能, 可提供使用者以最短的時間到達目的地。雖然Google Map搜尋路徑十分方便且準確度極高, 但遇到塞車時其效果將大打折扣。而在全球定位系統導航裝置中必須利用有限的資源對龐大的路網資料進行路徑規劃, 並在容許的時間內計算出可由起點到達終點之路徑, 提供導航裝置在行進過程中進行導航提示, 相對之下路徑的運算效率更顯得重要[1]。

目前對於到達目的地的要求, 主要有兩種, 最短距離行駛與最快行駛路徑為主要規劃來規劃出到目的地的路徑規劃, 對於現在有無法預期的路況事故或是塞車發生, 原本最短的路徑反而無法讓駕駛人快速到達目的地, 針對旅行路徑中盡可能降低所需花費的旅行時間, 這才是我們更重視的議題。

在現今路徑演算有許多不同演算法用以求解, 最短路徑問題求解之主要演算法有為Dijkstra[2], A\*[3]、階層式路徑搜尋法[7,8]。A\*使用啟發函式在計算過程中以直線距離預估與終點之距離, 並挑選預估值較佳的路段進入計算[4]。在節點分佈均勻的情況下, A\*效能表現極佳, 但A\*演算法仰賴一個假設: 兩點之間的最短距離是兩點之間的直線距離。當以行車時間作為路段的成本時, 這個假設未必能夠成立, 因此A\*的效能將會受到影響。

另一個方法在階層式路徑搜尋法中, 利用水波擴散搜尋的方式[8,9]的起始點與相鄰的每個節點, 像水滴滴在水面上, 以一般擴散出去的方式, 將每個鄰近的節點都給予階層編號, 有階層值產生時, 就達到方法的判斷出最小值, 這樣的方式雖然有可以讓運算速度提升, 但是在起點與終點之間距離很長時, 中間涵蓋節點數也相對提升, 所以當起點與終點之間距離越長, 階層式路徑搜尋的效率就越差[10]。

Dijkstra演算法是最常用的最佳演算法[2], 但當節點數多時, 共計算複雜度非常高, 實際上之應用有困難。但大部分Dijkstra演算法為目前演算法中的主架構, 針對每個路徑的搜尋, 使得搜尋結果較為理想。

在現今工商社會中，人口都集中於都市之中，所以都市人口之上下班及商業活動與市區道路交通有相對關係。因此，本論文以市區交通狀況建置一個簡易交通地圖模型，模型中依道路的等級、加上不同的行車速率、號誌及事故狀況的阻礙進行道路模擬，並以台中市交通道路為例，與Google Map道路行駛時間進行比較及驗證所提出之道路模型之效益。

## 2. 最短路徑

### 2.1 Dijkstra 演算法

Dijkstra's 演算法 [9,10]是以某一節點為出發點，計算從該節點出發到所有其他節點最短路徑時間。首先，Dijkstra 演算法主要在找出任兩點之間最短路徑。例如，當我們在某個城市中，將從一個城鎮至另一個城鎮，在這些道路中會有多種不同的選擇，但是所行駛距離也會有所不同，從地圖中某一點到達另一點，也可能存在許多可能的路徑，應用在 Dijkstra 演算法中可找出其最短路徑。

以圖1之7個節點之網路為例，圖中點與點之間的連線上為其成本權重值，以a點為起始點，g點為終點，下列說明如何應用 Dijkstra演算法尋找出最短路徑。

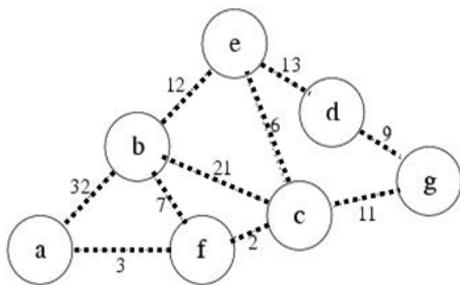


圖 1 各點關連權重圖

在 Dijkstra 演算法中把圖 1 所有點分成二個集合，為  $V = \{a, b, c, d, e, f, g\}$  及  $S = \{\}$ ，其中  $V$  集合是未查詢的節點集合， $S$  則是放查詢完成的節點集合。在圖 2 中，我們定義  $D_i[x]$  為第  $i$  次計算 a 點至 x 點之間最短距離， $l(f,a)$  為節點 f 到 a 點的距離，在演算法中欲為取出集合內的最小的值  $\min\{\}$ ，一開始所有節點都沒搜尋過，所以設置將節點陣列狀態時，將節點到任何節點都先設置無窮大，如表 1 所示。

表 1 節點陣列圖

	a	b	c	d	e	f	g
1	$\infty$						

步驟一:因起始點為 a，與 a 點相鄰節點為 b 點與 f 點兩點，則 b 點與 f 點第一次求得之距離為  $D_1[b]=32$  及  $D_1[f]=3$ ，如圖 2(a)(b)所示。當完成兩點搜尋後，在這兩點之間找出距離最短的值为  $D_1[f]$ ，則以 f 點相鄰的節點往下比較，以此類推。

步驟二: 將參考點設為 f 點後，與 f 點相鄰節點為 c 點與 b 點，透過 f 點，a 點至 b 的距離可求得為:

$$\begin{aligned} D_2[b] &= \min\{D_1[b], D_1[f] + l(b, f)\} \\ &= \min\{32, (3+7)\} = 10 \end{aligned} \quad (1)$$

節點 \ 步驟	b	c	d	e	f	g
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	<b>32</b>	$\infty$	$\infty$	$\infty$	<b>3</b>	$\infty$

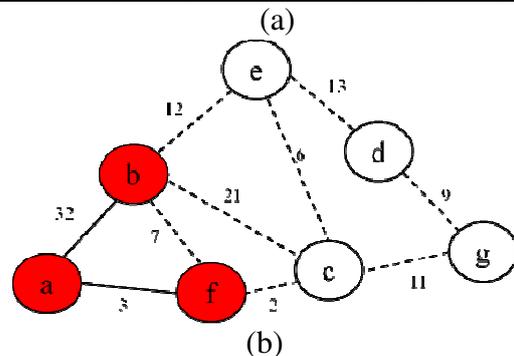


圖 2 第一步驟後(a)節點陣列值 (b)節點圖

此時，a 點至 b 點的距離，經過 f 點的(1)式計算後比原先距離  $D_1[b]=32$  小，所以取代之。而另一個相鄰節點為 c 點，透過 f 點，a 點至 c 點的距離可求得為:

$$D_1[c] = D_1[f] + l(c, f) = 3 + 2 = 5 \quad (2)$$

最後，因兩點距離最小的點為  $D_1[c]$ ，則以相同的方式以 c 點相鄰的節點往下比較。

步驟三:目前算出 a 到 c 為最短距離，所以選擇 c 點往下搜尋路徑，此時，將參考點設為 c 點後，與 c 點相鄰節點有三個為 b、e、g 點，透過 c 點，a 點至 b 點的距離可求得為:

$$\begin{aligned} D_3[b] &= \min\{D_1[b], D_1[c] + l(b, c)\} \\ &= \min\{10, (5+21)\} = 10 \end{aligned} \quad (3)$$

但在(1)式中， $D_2[b]=10$  是最短的距離，所以不取代，透過 c 點，a 至 e 點的距離可求得為:

$$D_1[e] = D_1[c] + l(e, c) = 5 + 6 = 11 \quad (4)$$

透過 c 點，a 至 g 點的距離可求得為:

$$D_1[g]=D_1[c]+l(g,c)=5+11=16 \quad (5)$$

步驟四:參考點設為 b 點後,與 b 點相連節點有一個為 e 點,透過 b 點, a 至 e 點的距離可求得為:

$$D_2[e]=\min\{D_1[e],D_3[b]+l(e,b)\} \\ =\min\{11,(10+12)\}=11 \quad (6)$$

但在(4)式中, $D_1[e]=11$  是最短的距離,所以不取代,找出距離最小的點為  $D_2[e]=11$ , 此時, a 至 e 點暫時設為  $D_1[e]=11$ , a 至 g 點暫時設為  $D_1[g]=16$ , 在陣列中找出距離最小的點為  $D_3[b]=10$ 。

步驟五:將參考點設為 e 點後,與 e 點相連節點有一個為 d 點,透過 e 點, a 點至 d 點的距離可求得為:

$$D_1[d]=D_2[e]+l(d,e)=11+13=24 \quad (7)$$

因與 e 點相連的節點只有 d 點,所以暫設  $D_1[d]=24$ , 接下來往 g 點進行比較。

步驟六:因  $D_1[g]$  小於  $D_1[d]$ , 故參考點設為 g, 與 g 點相連節點有一個 d 點, 圖 3 所示, 透過 g 點, a 至 d 點距離可求得為:

$$D_2[d]=\min\{D_1[d],D_1[g]+l(d,g)\} \\ =\min\{24,(16+9)\}=24 \quad (8)$$

在(7)式中, $D_1[d]=24$  已為最短距離,所以不取代,此時,因  $D_1[g]<D_2[d]$ , 已經確定目的地  $D[g]=16$  為最短距離,所以不再繼續尋找計算。

	a	b	C	d	e	f	g
0	∞	∞	∞	∞	∞	∞	∞
1	0	32	∞	∞	∞	3	∞
2	0	10	5	∞	∞	3	∞
3	0	10	5	∞	11	3	16
4	0	10	5	∞	11	3	16
5	0	10	5	24	11	3	16
6	0	10	5	24	11	3	16

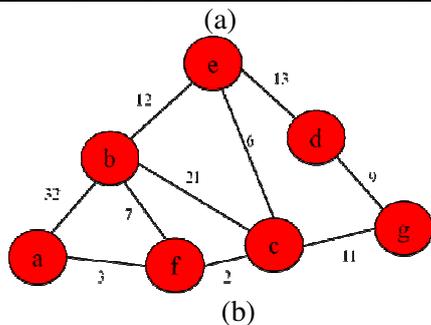


圖 3 完成尋找後 (a) 節點陣列值 (b) 節點圖

### 3. 實驗與結果

本研究針對市區道路建立一個簡易模型,以期能應用實際道路流量分析,並藉由模型來套用至目前台中市區交通道路中,來驗證所提出之道路模型之效益。

#### 3.1 SHRNM 與實際交通道路配置

為了使道路模型符合實際道路交通環境,首先需對實際道路作行駛時間之觀測與記錄,以對道路模型之多數適當調整,依照目前台中市道路交通地圖,依流量觀測與行駛時間路徑之分析。設定起點為朝陽科技大學,終點為東海大學,利用這兩點的路線來進行路徑選擇,利用 Google map 所顯示出來的四種不同道路路徑來與本文所提出的 SHRNM 進行比較。

在模擬中,將這次起點設定為朝陽科技大學,終點則設定為東海大學後,進行搜尋,Google map 會依照時間長短來顯示道路結果。將 SHRNM 設定成台中市交通地形,地圖以  $9 \times 9$  的節點矩陣作為此次模型,如圖 4 所示,在道路規劃中,主要是依照實際交通道路狀況進行劃分,依照道路限速來劃分成四種行駛道路分別為市區道路、外環道路、快速道路、高速公路四種模式,而這四種模式皆納入尋找最快道路時間考量。

透過 Google map 行駛時間來與 SHRNM 進行比較,每段道路的行駛時間都設置成一個隨機變數,其均勻地分佈  $[T_L, T_H]$ , 其中,  $T_L$  和  $T_H$  是分別為輕交通量和重交通流量的行駛時間,分別為道路類型中的限制行駛速度和各路段的行駛時間。

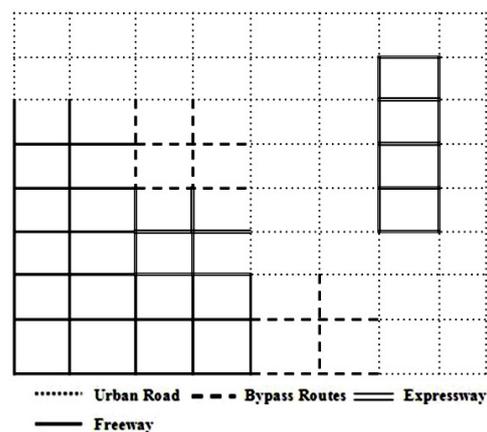


圖 4 台中市交通道路-SHRNM

在表 2 中,依照實際道路的行駛時間,我們分別以均勻分布之隨機變數作為所需之時間來進行實驗,依照交通量設置不同的參數,並將模組環境設定台中市市區道路來進行模擬。

表 2 四種路段行駛時間

道路名稱	最高限速	每段所需行駛時間 $[T_L, T_H]$
市區道路	限速 50km/hr	[5,7]分/段
外環道路	限速 70km/hr	[4.6,5.3]分/段
快速道路	限速 90km/hr	[3.2,4.7]分/段
高速公路	限速 110km/hr	[3.1,4.2]分/段

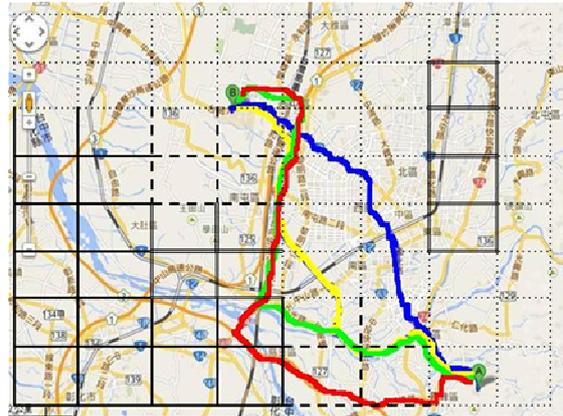
### 3.2 模擬結果

模擬中，設定起點 A 為朝陽科技大學，終點 B 為東海大學，依照四個不同類型道路行駛。

在模擬中，規劃出兩種道路行駛比較方式：(1) Google Map 規劃出最快路線 (2) SHRNM 最快道路行駛時間。Google Map 的即時交通路況則是利用全國交通即時路況及 GPS 手機使用者傳送的資料來判斷路況並預測目前行駛時間。SHRNM 最快道路行駛時間則是以四個道路進行模擬，利用 Dijkstra 演算法來計算出最快道路行駛時間。

實驗中，我們將 SHRNM 放入台中市區交通道路中，進行模擬比較，依照每個道路位置進行實驗，如圖 4-6 所示，行走路徑為市區道路時，因在台中市的交通地圖中市區道路佔了大部分，所以我們將 SHRNM 的道路設定擴大，而在外環道路方面，在台中市的交通地圖中，外環道範圍較大重疊性高，所以在 SHRNM 中將外環道路設定成先從起市區道路走一段路後，銜接至外環道路來行走至市區道路目的地，而快速道路及高速公路在台中市的交通地圖上，都在外圍部分，所以在 SHRNM 上，將快速道路設置成從起點市區道路走一段路後，接至外環道路或直接進入快速道路最後行走至市區道路目的地，最後將高速公路設置成從起點市區道路走一段路後，接至外環道路及快速道路或直接進入高速公路來行走，最後行走至市區道路目的地。此環境我們在行駛時間表中，提出六個時間點進行模擬，在這六個時間點主要包括：平日早、中、晚及假日早、中、晚的行駛時間，主要是要探討每個時段車流量不同是否會影響行駛時間。由圖 6 可知，時間為上午 7~9 時為上班的尖峰時期，所以在行駛時間都明顯增加，而在 SHRNM 中，透過 Google Map 的行駛時間，我們設定出四個不同道路類型的行駛時間，結果顯示，SHRNM 四種道路

平均行駛時間為約 50 分鐘，其中以高速公路 45.52 分鐘較快到達，所。在平日下午 13~15 時中，每個道路的行駛時間都較為縮短，所以屬於離峰時段，Google Map 與 SHRNM 則計算出最快道路為高速公路，在 SHRNM 中高速公路平均行駛時間為 44.11 分鐘，其次最快道路則為快速道路。



■ 市區道路 ■ 外環道路 ■ 快速道路  
■ 高速公路

圖 5 SHRNM 與台中市區交通道路配置圖

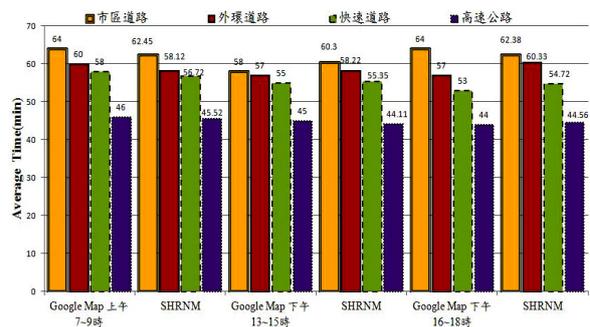


圖 6 平日之 Google Map 與 SHRNM 行駛時間比較

在平日下午 16~18 時中，為下班的尖峰時期，所以在行駛時間又都明顯增加，所以在選擇建議上一直為高速公路，而在平日行駛時間中我們觀察到 Google Map 與 SHRNM 的平均平方誤差 (Mean Square Error, MSE) 為  $1.22(\text{min}^2)$ 。而在假日行駛時間部份，上午 7~9 時行駛時間，如圖 6 所示，行駛時間明顯降低，市區道路行駛時間為 58 分比圖 7 中的行駛間降低了 7 分鐘。反觀，在假日的下午 13~15 時及下午 16~18 時的行駛時間裡，預測的行時間都明顯增加，Google Map 與高速公路的行駛時間增加為 49 分鐘，最後在假日行駛時間中我們觀察到 Google Map 與 SHRNM 的平均平方誤差 (Mean Square Error, MSE) 為  $1.97(\text{min}^2)$ 。

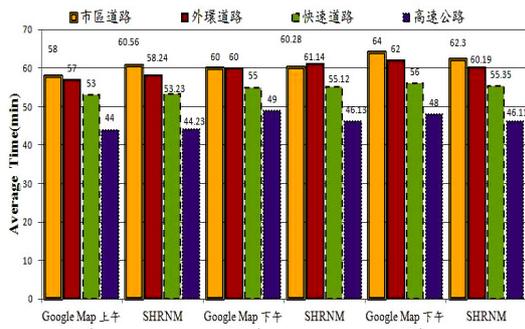


圖7 假日之Google Map與SHRNM行駛時間比較

為了使道路模型符合實際道路交通環境，我們分別以不同五個起點進行模擬，如圖8所示。在模擬結果中，如圖9所示，A至終點時SHRNM行駛時間為25分鐘，行駛道路為市區道路，而B點至終點時Google Map行駛時間為18分鐘比SHRNM行駛時間快4分鐘。C點與D點至終點時，SHRNM行駛時間為28與23分鐘，而在E點至終點中，SHRNM行駛結果為30分鐘比Google Map快，所以Google Map與SHRNM的平均平方誤差(Mean Square Error, MSE)為11.69(min<sup>2</sup>)。

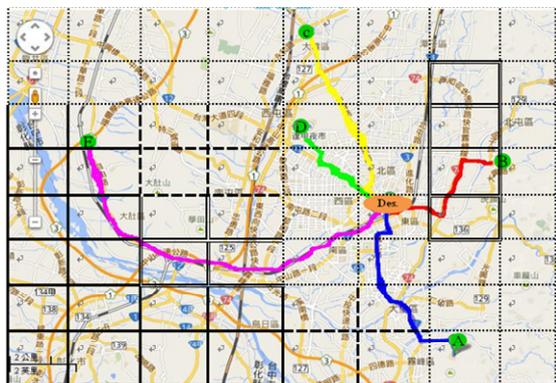


圖8 以Google Map為地點配置圖

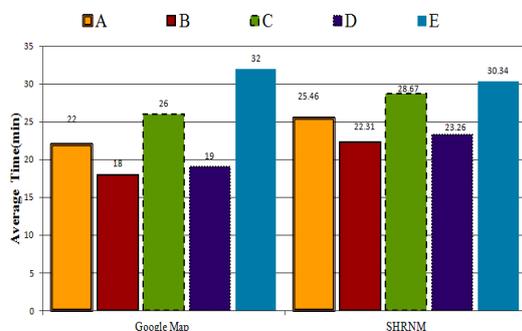


圖9 Google Map與SHRNM五個地點行駛時間比較

接下來，以SHRNM的道路模型的端點進行

模擬，我們在SHRNM中以節點為定點，來進行模擬，如圖10所示，我們分別設置四個起點行駛至所設定的終點，將行駛結果來與實際道路交通作比較。

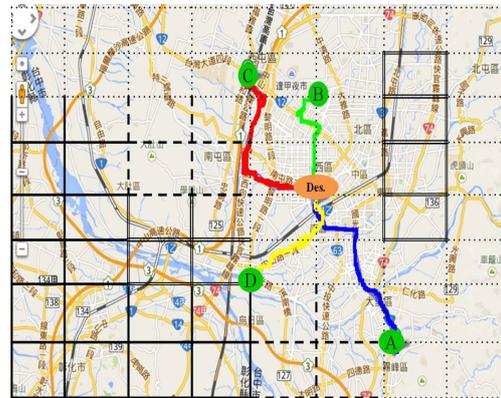


圖10 以SHRNM為地點配置圖

在模擬結果中，如圖11所示，A至終點時SHRNM行駛時間為22分鐘，行駛道路為市區道路，Google Map則所算出的行駛時間為25分鐘，而B點與C點至終點時Google Map行駛時間分別為17分鐘、19分鐘與SHRNM行駛時間相比，行駛時間較為相近。D點至終點時，SHRNM行駛時間13分鐘，而Google Map行駛時間15分鐘在行駛時間上差距不大。在模擬結果中，Google Map與SHRNM的平均平方誤差(Mean Square Error, MSE)為6.3(min<sup>2</sup>)，比Google Map端點進行模擬誤差值低，由此可知，SHRNM行駛道路的時間況可以運用在台中市交通道路上。

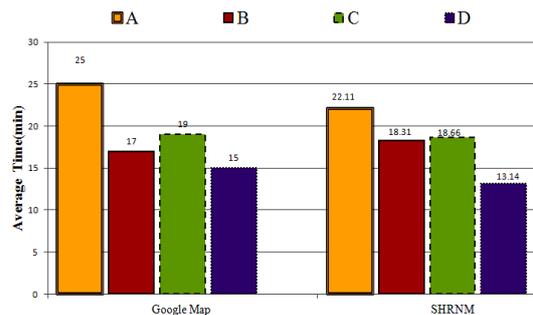


圖11 Google Map與SHRNM四個地點行駛時間比較

#### 4. 結論

本論文提出簡易交通道路分層模型(SHRNM)，模型中依道路的等級以及不同的行車速率，針對不同行駛條件進行從各道路到達終點的的道路時間模擬，快速找出最佳路線外，並以實際的台中市交通道路，利用Google Map預測行時間進行模擬，結果顯示，雖無法像Google Map所計算出的最快路徑精準，但在

SHRNM中，所模擬出的最佳路線確實能提供不同道路的行駛時間並與實際交通的行駛時間相符。

### 參考文獻

- [1] C. Zhang, F. Zhang, J. Ren and J. Ren, "Multidimensional Traffic GPS Data Quality Analysis Using Data Cube Model," *International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, pp. 307-310, December, 2011.
- [2] D. Fan and P. Shi, "Improvement of Dijkstra's Algorithm and Its Application in Route Planning," *Proceedings of Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp.1901-1904, 2010.
- [3] I. Chabini and S. Lan, "Adaptations of the A\* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," *Proceedings of IEEE Transactions on Intelligent Transportation Systems*, pp.60-74, Mar. 2002.
- [4] N.A.M. Nordin, N. Kadir, Z. A. Zaharudin and N. A. Nordin, "An Application of the A\* Algorithm on the Ambulance Routing," *Proceedings of IEEE Colloquium on Humanities, Science and Engineering Research (CHUSER)*, pp.855-859, Dec. 2011.
- [5] G. J. Shen, "An Intelligent Hybrid Forecasting Model for Short-term Traffic Flow," *Proceedings of the World Congress on Intelligent Control and Automation*, pp.486-491, July 2010.
- [6] M. K. Mainali, S. Mabu and K. Hirasawa, "Evolutionary Approach for the Traffic Volume Estimation of Road Sections," *Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp.100-105, 2010.
- [7] 余家豪(2007)，行動代理人訊息寄送之容錯，碩士論文，朝陽科技大學。
- [8] 劉後欣(2004)，行動代理人訊息傳輸之保證，碩士論文，朝陽科技大學。
- [9] 張君瑋(2008)，多階層式主機與無線感測網路基礎下以最短搜尋時間法規最適交通路徑，碩士論文，朝陽科技大學。
- [10] 林芳昌、張君瑋，多階層式主機與無線感測網路基礎下以最短搜尋時間法規最適交通路徑，台灣網際網路研討會(TANET)，義守大學，pp. 1-6，2008。