

多功能路徑規劃演算法

A Multiple Purpose Path Routing Algorithm

詹景裕

國立臺北大學

電機工程研究所

gejan@mail.ntpu.edu.tw

李明哲

銘傳大學

資訊傳播工程學系

leemc@mcu.edu.tw

蔡維峻

國立臺北大學

資訊工程學系

doraemontsai@hotmail.com

陳永源

中華大學

資訊工程學系

chenyy@chu.edu.tw

摘要

路徑長度和轉彎次數一直是交通工具的主要運輸成本因素，路徑越長則所花時間越久；而每次轉彎必須減速也增加時間成本的支出。

然而，一條最快的路徑不應該只考慮距離長短和轉彎次數，實際上，每段路徑可以行進的速度也是一項重要的因素，但是同時考慮多項參數的路徑搜尋演算法為 NP-complete 的課題，故本文使用的方式為將多組參數整合成一個參數；以 Kirby 提出的構想及 Ahuja 等人改良之 Dijkstra 演算法為核心發展成多功能最佳路徑演算法，在一個 $n \times n$ 的網格圖中，其運算時間複雜度為 $O(n\sqrt{\log C})$ ，並可藉由調整相關變數來得到最短路徑、最少轉彎數路徑和最快路徑等多種不同的輸出結果路徑。

關鍵詞：最短路徑、網格圖、Kirby、Dijkstra 演算法

1. 緒論

在以往的導航系統當中，通常只考慮障礙物以及距離長短，但是在行進過程中，每逢轉彎的地方則必須減速，此舉將增加到達的時間；而每段道路實際上可行進的速度皆不一樣，更直接地影響到達目的地的時間，所以轉彎次數和行進速度，對於路徑規劃而言，兩者皆是不可忽略的因素。

本文的研究背景始於詹景裕等人 [1]在二維

運輸網路的路徑規劃中加入了轉彎的影響，除了減少轉彎次數，如何更進一步減少到達目的地的時間是本文所追求的目標。最佳路徑規劃的應用研究領域非常廣，Beattie [2]曾利用電子海圖 (Electronic Navigational Chart)及地理資訊系統 (Geographic Information System)技術來發展船舶自動導航系統。Fawcett 與 Robinson [3]曾針對道路地圖的繞行問題提出解決方法。動態環境的路徑搜尋問題則有 Aggarwal 與 Fujimura [4]、Aggarwal [5]及 Castro 等人 [6]提出不同的方法去深入探討及解決。

有關處理地圖資訊的形式，施保旭 [7]及 Eastman [8]的論著中均曾提及，主要可分為兩大類，一為向量圖形 (Vector Graphics) 的資料結構，另一則為網格化圖形 (Raster Graphics) 或稱其為蜂巢組織狀 (Cellular Organization) 的資料結構。由於使用的資料結構不同，相對用來找尋最短路徑的演算法也會不相同。若使用向量式資料結構的路徑規劃，對於物體間相互碰撞的關係並不容易判斷、會花費相當大的計算量且只可處理有稜有角的凸邊形，若為凹邊形則需將其分割為多個凸邊形，始可計算，最後所求出之路徑也未必為最佳路徑。因此，本文使用網格式資料結構去建構模擬環境。

本文結合了 Kirby 所提出的構想以及 Ahuja 等人所改進的 Dijkstra 演算法再加上距離、轉彎權重、動態可行進速度等變數來尋求最短路徑、最少轉彎數路徑和最快路徑的多功能最佳路徑

演算法，可通用於網格圖及交通運輸網路 (Transportation Networks)。其時間複雜度為 $O(n\sqrt{\log C})$ 。

2. 相關背景

在導航路徑規劃過程中，雖然每個網格代表相同之距離，但是每個網格所能行進的速度實際上卻都不一樣，找出同時考慮距離長度、適當轉彎數與行進速度的最佳化路徑，是本文研究所追求的目標。

然而，同時考慮多項參數的路徑搜尋演算法是 NP-complete 的問題，於是本文運用 Kirby 所提出之模型來修改每個網格的資料結構，將每個網格以若干虛擬節點取代，簡化轉彎與行進方向等各種狀態，接下來應用 Dijkstra 演算法取得最快路徑。

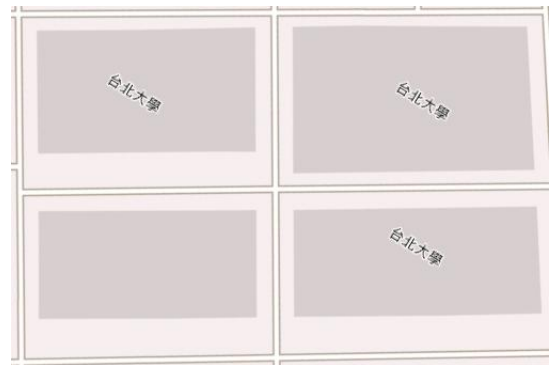
2.1 網格化

本文是建立於二維空間中的工作環境，因此，是以一個網格為單位。首先，將工作環境進行網格化處理，即對電子地圖中之地理環境經掃描後進行網格化，藉電腦化處理並建構成網格圖，電子地圖轉換成網格圖之網格化過程如圖 1(a)至 1(c)所示。

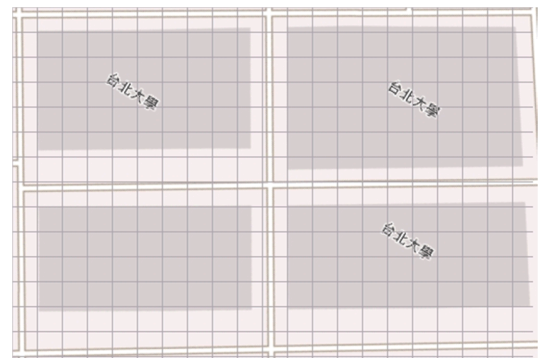
圖 1(a)為尚未網格化的原始電子地圖，白色區域代表可行走之道路，有顏色的區塊則為建築物。圖 1(b)將原始電子地圖分割為等大小網格狀。圖 1(c)為網格化後之地圖，白色網格為可行走的區域，黑色網格則代表障礙物。

2.2 Dijkstra 最短路徑演算法

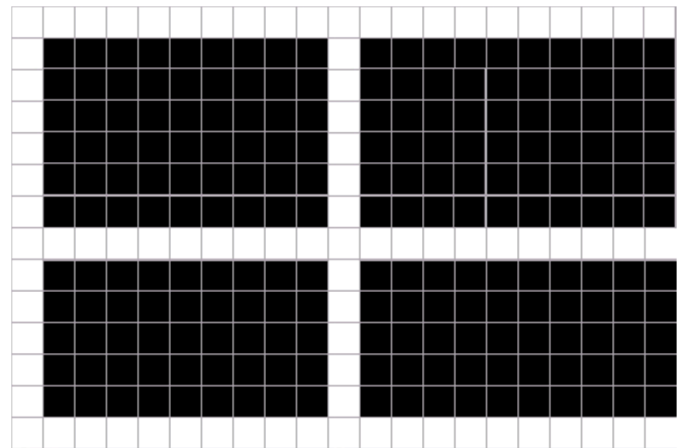
Dijkstra 所提出的最短路徑演算法，可適用於有向圖上搜尋最小成本路徑，只要有向圖上的所有鏈結花費為非負整數皆可使用 [8]。其主要想法是維持一個集合 S ，使之包含所有已找到由起點 s 至該節點最短路徑的節點，所有不屬於集合 S 的剩餘節點所成的集合稱之為 Q ，即 $Q = \{u$



(a) 給定的電子地圖



(b) 網格化



(c) 網格化後之電子地圖

圖 1 電子地圖轉換成網格圖實例

$| u \in V-S \}$ ，接著由 Q 中找出包含最小花費的節點，用以加入集合 S ；不斷重複此一步驟直至到達目的節點為止。

因為 Dijkstra 使用線性排序，來尋找集合 Q 中擁有最小值的節點來加入集合 S ，故其時間複

雜度為 $O(V^2+E)$ ，其中 V 為節點總數， E 為鏈結總數 [8]。自此之後，許多相關的研究陸續被提出，這些研究大多針對排序方面加以改進，以期能在最短時間內搜尋擁有最小值的節點，其中，以 Ahuja 等人所提出的改進方式為最快 [9]。其改進方式為使用 Radix heap 配合 Fibonacci heap 來加以排序，使得改進後的 Dijkstra 最短路徑搜尋演算法的時間複雜度可達到 $O(E + V\sqrt{\log C})$ ，其中 V 為節點總數， E 為鏈結總數，而 C 為鏈結花費(edge weight)中的最大值。

3. 資料結構

這節將介紹 Kirby 所提出的構想。Caldwell 最早提出最少轉彎問題後 [10]，許多研究人員即對此一問題提出後續的研究。研究的重心大多集中在如何在網狀架構上表達轉彎的概念，並依此計算轉彎數，用以避免轉彎 [10,11]。其中，Kirby 提出了如圖 2 的構想，我們將每個十字路口網格改為一個由 8 個虛擬節點所組成的虛擬網路來代替。各節點間由鏈結(edges)互相連接，這些鏈結代表了網狀架構上的各種行進狀態，包括行進速度及行進方向等等，所須付出的時間成本權重，我們則以 $W_{Th}^v(C_{ij}, t)$ 表示，而 v 代表行進速度， T_h 代表為行進方向權重， $T_h = \{(T_L, T_R, T_S) | (Left, Right, Straight)\}$ ，分別代表左轉、右轉以及直行權重。使用者可以用常數或時間函式定義。

$$W_{Th}^v(C_{ij}, t) = (1+T_h) \times Dist / (1+V_E(C_{ij}, dir, t)) \quad (1)$$

其中， $Dist$ 為每網格固定之距離， $V_E(C_{ij}, dir, t)$ 為各網格可行進速度，速度會因 dir (方向)和 t (時間)而不同， T_h 為轉彎權重；為了避免 $W_{Th}^v(C_{ij}, t)$ 產生 0 或者是 ∞ ，故將 T_h 和 $V_E(C_{ij}, dir, t)$ 多加 1。但為了簡化問題，本文後段暫不使用時間函式來定義。

本文所使用的路徑時間成本計算方式，是採

累加的方式，將該路徑所經過的鏈結花費累計起來，即是該路徑的時間成本。令該路徑的時間成本為 $Cost_s^d$ 。

$$Cost_s^d = \sum_{l=0}^{d'} W_{Th}^v(C_{i(l), j(l)}, t) \quad (2)$$

雖然 Kirby 提出如此完備的虛擬網路來模擬節點間行進的行為，然而卻使用樹狀搜尋法來尋求最少轉彎路徑，造成非常高的時間複雜度 [10]。

Algorithm : Multiple Purpose Path Routing

BEGIN

Step 1 : 初始化

Step 1.1 : 將電子地圖網格化，並將非障礙物網格之 $Cost$ 設為 ∞ 。

Step 1.2 : 設定起點 s 和終點 d ，並判斷是否在障礙物上。

Step 1.3 : 使用 Kirby 的方法重新建構十字路口之網格，並計算每個鏈結的時間成本 $W_{Th}^v(C_{ij}, t)$ 。

Step 2 : 計算起點至每個虛擬節點的時間成本

Step 2.1 : 將位於起始節點的虛擬節點存到集合 S 。

Step 2.2 : 使用 Ahuja 等人改良過的 Dijkstra 演算法，直到遇到終點或無法擴張。

Step 3 : 選出目的網格中 $Cost_s^d$ 最小的節點

Step 3.1 : 若目的節點的 $Cost_s^d$ 皆為 ∞ ，則表示無法到達。

Step 3.2 : 選取 $Cost_s^d$ 最小的目的節點，其所花費的時間成本為起點至終點的最佳時間成本。

Step 4 : 回溯

從步驟 3 選出的目的節點開始，藉由步驟 2 所記錄的路徑資訊返回至起點即為最佳路徑。

END

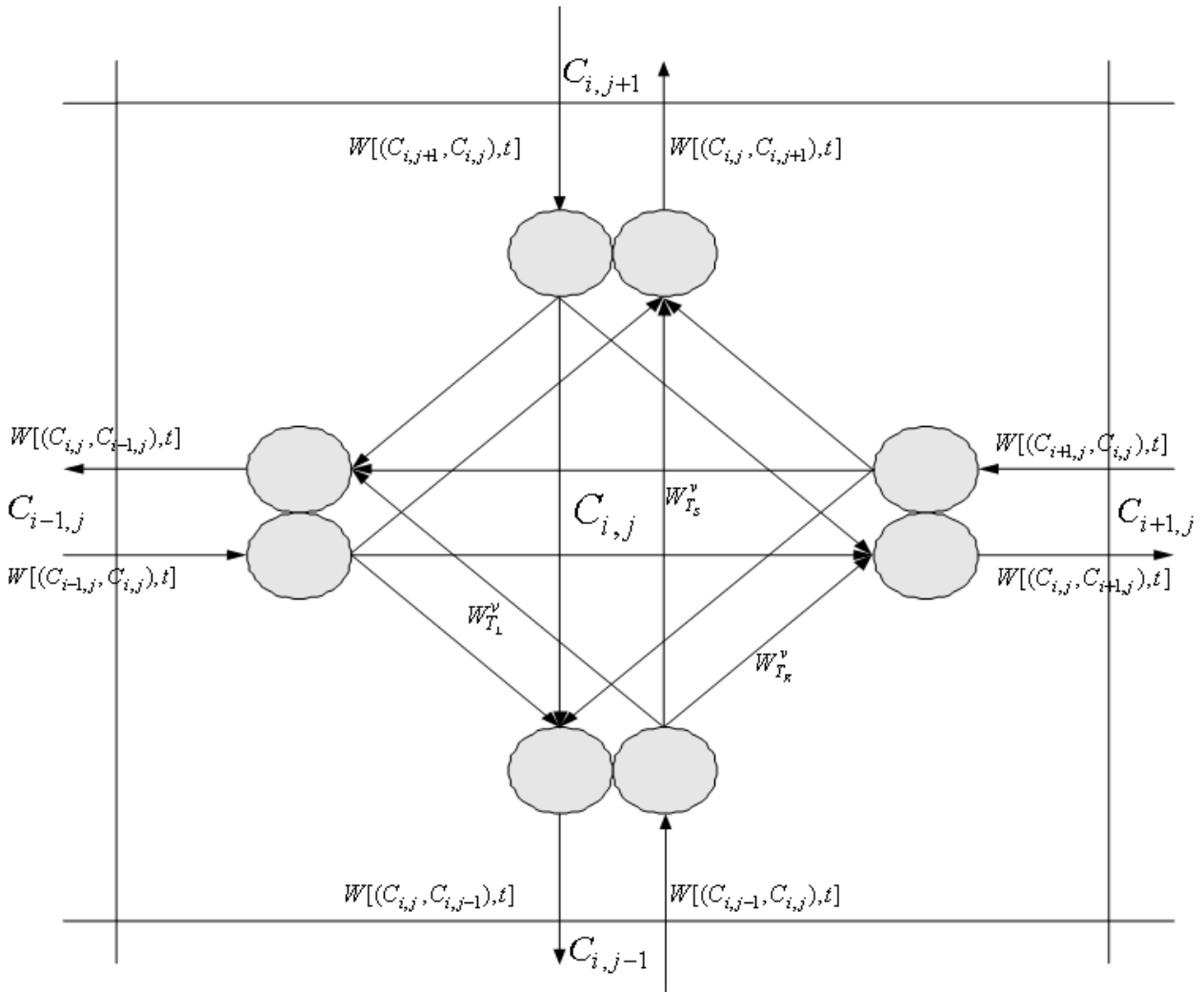


圖 2 演算法的節點資料結構示意圖

4. 效能分析

4.1 時間複雜度

我們假設網格圖上有 $n \times n$ 共 N 個網格，每個十字路口網格為由 8 個虛擬節點及 16 條鏈結所組成的虛擬網路，可行走之交通運輸網路為 n 個，而多功能最佳路徑演算法的第一步驟為初始化，只需常數時間。第二步驟藉由 Ahuja 所改進的 Dijkstra 演算法，來尋求最短路徑，其時間的複雜度為 $O(E + V\sqrt{\log C})$ ，因為可行走之 n 個網格

其鏈結總數為 $16n$ ，節點總數為 $8n$ ，而 C 為常數，是故多功能最佳路徑演算法第二步驟的時間複雜度為 $O(n\sqrt{\log C})$ ；第三步驟由目的網格中其組成的節點選出擁有最小時間成本的節點，使其時間成本成為由起點到此節點的最小時間成本，故第三步驟僅需常數時間；第四步驟，我們使用回溯的方法來輸出最小成本路徑，最糟的情形將不多於 $2n$ 次計算，即可找出最小成本路徑，是故第四步驟的時間複雜度為 $O(n)$ 。

由以上所述，我們可知多功能最佳路徑演算法的時間複雜度為 $O(n\sqrt{\log C})$ 。

4.2 空間複雜度

多功能最佳路徑演算法的第一步驟為初始化，需將網格圖轉換成 Kirby 模型，最多每個網格鏈結總數為 $16n^2$ ，節點總數為 $8n^2$ ，是故第一步驟的空間複雜度為 $O(n^2)$ 。第二步驟藉由 Ahuja 所改進的 Dijkstra 演算法，來尋求最短路徑，因為最多有 $8n^2$ 個節點，所以第二步驟的空間複雜度為 $O(n^2)$ ；第三步驟僅需從目的網格中挑選最少時間成本的節點，其空間複雜度為 $O(1)$ ；最後的第四步驟，我們使用回溯的方法來輸出結果路徑，最糟的情形將不多於 n^2 個網格，即可回溯至起點，是故第四步驟的空間複雜度為 $O(n^2)$ 。

由以上所述，我們可知最小成本路徑演算法的空間複雜度為 $O(n^2)$ 。

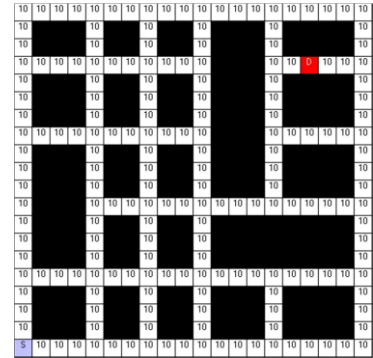
5. 實例

圖 3 為演算法的執行範例。圖 3(a) 為網格化地圖之初始狀態與起點終點並包含動態交通資訊。當 $T_h = 0, V_E = constant$ ，演算法得到之路徑為最短路徑，如圖 3(b) 所示。而當 $T_h \gg Dist, V_E = 0$ ，演算法得到之路徑為最少轉彎數路徑，如圖 3(c) 所示。假定 $T_h = \{(T_L, T_R, T_S) | (1, 0.5, 0)\}$ ，且 V_E 為時變函數，則演算法得到之路徑為時間成本最少之路徑，如圖 3(d) 所示。

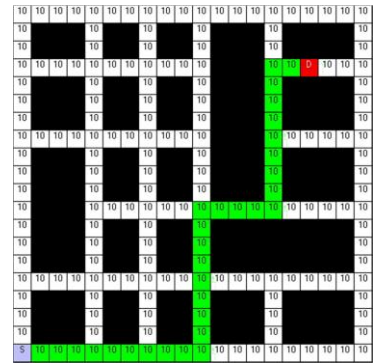
6. 結論與未來發展

本文所提出的演算法，是針對二維網狀架構及交通運輸網路上尋找最短路徑、最少轉彎數路徑和最快路徑而設計，其原理建構在 Kirby 所提出的模型上，利用其模型簡化眾多參數使其可以運用 Dijkstra 演算法計算所需路徑。

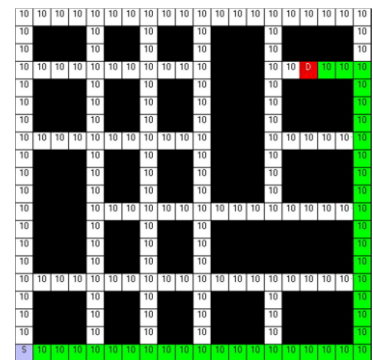
由於在路徑規劃中，若只專注於其中一項因素，通常會造成時間成本的增加，故本文將距離、轉彎和速度定義出路徑的時間成本計算方法，進而發展為多功能最佳路徑演算法。使用者可以依照不同需求，調整各參數之權重得到最短



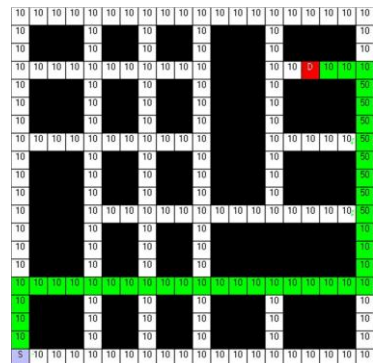
(a) 選定起點與終點之初始圖



(b) 最短路徑



(c) 最少轉彎路徑



(d) 最快路徑

圖 3 多功能最佳路徑演算法範例

路徑、最少轉彎數路徑或是時間最短路徑。

未來可以使用Android系統以達到在Google Phone上執行，利用手機的便利性和方便攜帶性達到更高效能的道路導航系統；另一方面可從2D發展成3D，應用層面則可拓展至航空、航海及其他領域。

參考文獻

- [1] Gene Eu Jan, Ming Che Lee, S. G. Hsieh, Yung Yuan Chen, "Transportation Network Navigation with Turn Penalties," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Singapore, July 14-17, pp. 1224-1229, 2009.
- [2] J. H. Beattie, "The future of electronic charts in merchant ships," *The Journal of Navigation*, vol. 48, pp. 335-348, 1995.
- [3] J. Fawcett and P. Robinson, "Adaptive routing for road traffic," *IEEE Computer Graphics & Applications*, vol. 20, pp. 46-53, 2000.
- [4] K. Fujimura, "Motion planning using transient pixel representations," in *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, vol. 2, 1993, pp. 34-39.
- [5] N. Aggarwal and K. Fujimura, "Motion planning amidst planar moving obstacles," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 1994, pp. 2153-2158.
- [6] D. Castro, U. Nunes, and A. Ruano, "Reactive local navigation," in *Proceedings 28th Annual Conference of the IEEE Industrial Electronics Society (IECON' 02)*, Sevilla, 2002, pp. 5-8.
- [7] 施保旭, *地理資訊系統*, 初版. 台北: 儒林圖書有限公司, 1995.
- [8] C. M. Eastman, "Vector versus raster: a functional comparison of drawing technologies," *IEEE Computer Graphics & Applications*, vol. 10, pp. 68-80, 1990.
- [9] K. M. Ravindra, K. Ahuja, J. B. Orlin, and R. E. Tarjan, "Faster algorithms for the shortest path problem," *Journal of ACM*, vol. 37, pp. 213-223, 1990.
- [10] R. F. Kirby and R. B. Potts, "The Minimum Route Problem for Networks with Turn Penalties and Prohibitions," *Transportation Research*, vol. 3, pp. 397-408, 1969.
- [11] B. V. Martin, F. W. Memmott, and A. J. Bone, "Principles and techniques of predicting future demand for urban area transportation," M.I.T. Report, 1961.