

智慧型手機上多點實際道路路徑規劃

詹景裕^a、林政宏^a、胡英懋^{*a}、何嘉玲^b
 國立臺北大學電機工程學系^a
 臺北城市科技大學行銷與流通管理系^b

摘要—本文中主要的對象是宅配業者的配送運輸網路。我們可以歸納出以下問題：(1)以自我經驗為基準去規劃路徑，但由於營業據點的急速擴散，時常需要更動配送路線；(2)實際影響油耗情況，除了距離大小，還有轉彎成本、路燈波、迴轉道等因素；(3)抵達須配送的目的地時，還須考量去卸貨時間成本；(4)隨時隨地將導航裝置安置於任何車體上的便利性。本文去改善傳統以經驗法則為主的配送路線規劃。¹

一、緒論

這幾年來，電子商務業蓬勃發展，造成了新興的購物模式-配送服務。消費行為轉移，市場需求擴大，為了減少運輸成本的開銷，必須規劃出良好的運輸路線。在本文中主要的對象則是宅配業者的配送運輸網路，而目前的業者大多是採取經驗法則去規劃各自的配送路線，但所需考量交通因素過多，很容易規劃出運輸成本較高的路徑，因此僅僅只是依靠經驗法則是不夠的。在目前相關的研究大多針對距離的大小去規劃其路線，規劃出的路線不符實際路況，並且其便利性皆不如應用於智慧型手機上方便。研究相關領域的文章之後，我們可以歸納出以下的問題：(1)大部分送貨員都以自我經驗為基準去規劃路徑，但由於營業據點的急速擴散，時常需要更動配送路線，因此僅僅依靠經驗法則是不夠的；(2)在實際的道路情況底下，影響耗油及速度的因素不單只是距離大小，還有轉彎成本、路燈波、迴轉道等因素；(3)抵達須配送的目的地時，還須考量去卸貨時間成本；(4)隨時隨地將導航裝置安置於任何車體上的便利性。這些問題對於配送規劃皆是重要因素。

基於上述動機，我們找尋許多相關資料來針對這些問題做更深入的探討。而從過去相似的問題之研究，發現旅行銷售員問題 (Traveling Salesman Problem, TSP) 是配送問題的最基本問題型態，在學術及實際上都有廣泛的應用，例如工業製造流程 [15]、交通運輸排程 [10]、生物工程 [3] 等領域中，皆可將問題轉換成旅行銷售員問題 (TSP) 的型態來求解，以達到改善效率之結果。

本實驗室針對一般大眾作為對象，已在智慧型手機上成功實作出點與點之最快速路徑演算法、最經濟路徑演算法以及最佳實際路徑規劃演算法 [7] [8] [9]，其演算法已將許多實際道路因素考量完畢，能夠大量的縮小行車成本。

依此作為基礎，透過完善的單點路徑規劃作為參考依據，擴展至多點路徑規劃。本文以宅配業者作為服務的對象，去改善傳統以經驗法則為主的配送路線規劃，

透過整體的規劃後，就可大大的減少運輸成本。

二、背景與相關研究

2.1 旅行銷售員問題 (TSP)

旅行銷售員問題早期是在 1934 年由 Hassler Whitney 提出此問題 [4]。其中最主要的目的是希望能夠使整個拜訪過程所花費的成本為最少。Held 及 Karp [6]將旅行銷售員問題的定義如下：一位銷售員從公司所在的城市出發，欲拜訪住在 n 個不同地區中的客戶，在整個拜訪的過程中，銷售員必須拜訪每座城市一次，且僅能一次，拜訪完所有的城市後須回原本出發的城市。

旅行銷售員問題又可分為對稱性和非對稱性的結構，所謂的對稱性指的是 A 點到 B 的距離會是等價於 B 點到 A 點之間的距離，而非對稱性則是不等價。在本文中，為了貼近實際道路情況，會有上坡較為耗油，而下坡較省油的情況下，我們採用的是非對稱性的結構。

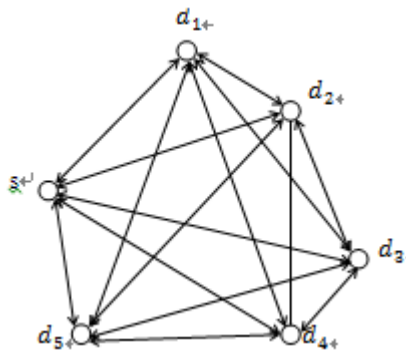
旅行銷售員問題是一個困難的組合問題。在 1979 年由 Gaery [5] 證實 TSP 為 NP-Complete 的問題。因此，當問題的個數被擴大時，解空間會急速增大，無法以一個多項式函數來求得最佳解。而旅行銷售員問題的解法可分成確切解法 (Exact Algorithm) [16] 與啟發式解法 (Heuristic Algorithm) [13] 兩大類。

最佳解法包含窮舉搜尋法 (Exhaustive Search Method)、分支界限法 (Branch and Bound Method) [14]、等。以上方法雖然可以取得確切解，但是隨著節點的增加，時間複雜度也會隨著指數型成長。

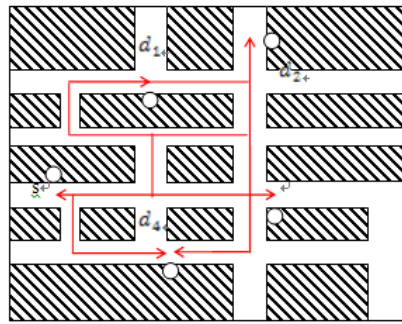
由於確切解法所耗費的時間複雜度相當大，因此有了啟發式解法，得以快速求取近似解。啟發式解法大概可以分成以下三類 [1]：(1) 路線建構模組 (Tour Construction)；(2) 路線改善模組 (Tour Improvement) [11]；(3) 綜合法 (Composite Procedure) [12]。路線建構模組是依據路網的距離或成本去產生初步的解法 [22][23]；路線改善模組則是針對任意初始路徑 (Initial Tour) 透過交換 (Exchange) 的方式取得較小成本的路徑；綜合法則是綜合上述兩種模組去取得最小路徑成本。

2.2 完全連接圖 (Complete graph)

在歐氏空間下，任何不同的兩個頂點皆有邊相連的簡單圖，稱之為完全連接圖 (Complete graph)。當使用者設定完各個目的點時，將起點 s 與目的點 d_i 建構出完全連接圖 $G(V, E)$ ， V 指的是起點 s 與目的點 d_i 的點集合， E 指的是連接所有 V 的邊集合。如圖一所示。本文考量的因素有很多，例如：有無對向車道、坡度、最少轉彎成本等，故不適用於歐氏空間下，因此建立出的完全連接圖有些許差異。為了作為規劃路徑的基準值之一，將起點



圖一：有向完全連接圖



圖二：基於實際道路之 Directed Complete graph

s 與目的點 d_i 建構出完全連接圖 $G^C(V^C, E^C)$, V^C 指的是起點 s 與目的點 d_i 的點集合, E^C 指的是連接所有 V^C 的邊集合。如圖二所示。

2.3 Edge Weight

在市面常見的導航裝置, 大多以距離作為參考的基準, 在 Edge Weight 也大多以距離作為其參考值。在本文中不單單只是以距離作為參考點, 而是透過最佳路徑演算法將眾多交通重要因素考量進來, 例如: 道路寬度、迴轉道、轉彎成本、坡度、綠燈波等。因為本文為非對稱結構, 所以在點與點之間會有兩個 Edge Weight。如圖三所示, 其中斜線節點為目的地節點, 其餘則為一般節點。

2.4 Node Weight

當配送員規劃配送路徑時, 很容易忽略掉整體時間的考量, 雖然目前有些導航裝置有預測行車時間的功能, 卻無法在行車時間內加入卸貨時間成本。在本文中, 我們將各個目的點 d_i 加入權重值 $W_{d_i}^C$ 來表示各個目的點的卸貨時間成本, 也可以根據道路方向的向量去運算, 加入至正確方向的 $W_{e_{i,j}}^C$ 上, 如圖 4 所示。就可以計算出加總之後的權重值 $W_{e_{i,j}}^C$ 。這樣就可以精確的判斷出抵達各目的點的時間, 也就可以提供送貨員路徑參考的資訊之一。

2.5 物流架構

由於本文的對象是宅配業者的配送運輸網路。規畫一個有效率、高品質的運輸系統, 首先必須要建構一個適當的營運網路, 以往的營運網路採用起點與目的點的直接網路運輸, 然而此種營運模式, 因每一運具的乘載率低, 故所耗用的運具較多。因此而有了軸幅式運輸網路 (hub-and-spoke network) [2] 的提出, 軸幅式運輸網路的結構提供人流或物流於中繼站轉運的功能, 替代運具低承載率的高運輸成本。

三、 TSP 多點實際路徑規劃演算法

以單點的最佳實際路徑規劃演算法為核心, 考慮多樣實際道路情況, 並且加入 Edge Weight 與 Node Weight 的資料結構, 最後透過多點實際路徑規劃演算法, 得到多點實際路徑。

Preprocessing: Road Map

Begin

Step1: 將電子地圖以適當化的解析度網格化, 產生 N 個網格

Step2: 將所有網格依其 R/B 來建構出有向圖 $G(V, E)$

End

Algorithm: TSP 多點實際路徑規劃演算法

Begin

Step1: 建立雙向完全連接圖 $G^C(V^C, E^C)$

Step1.1: 設定起點 s 和所有目的節點 $d_i, 1 \leq i \leq M$

Step1.2: 利用最佳實際路徑規劃演算法建構出起點 s 與所有目的點 d_i 之完全連接圖 $G^C(V^C, E^C)$

Step2: 將各個目的節點之權重值 $W_{d_i}^C$ 加入至 $W_{e_{i,j}}^C$ 得到完全連接圖之所有邊之綜合權重 $W_{e_{i,j}}^C$ 。

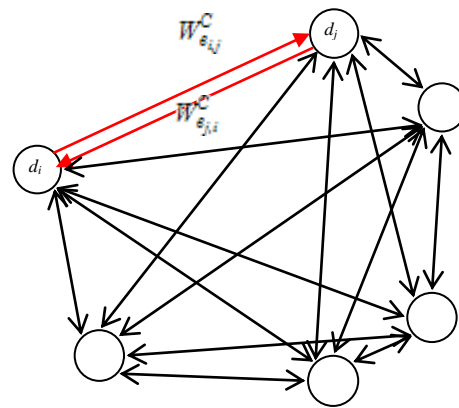
Step3: 建立多點實際路徑

從起始點 s 出發選擇距離該節點綜合權重最小的節點 d_i 連接, 接著以 d_i 為新的起點繼續連接綜合權重最小的節點, 直到所有目的點拜訪結束且回到起點為止。

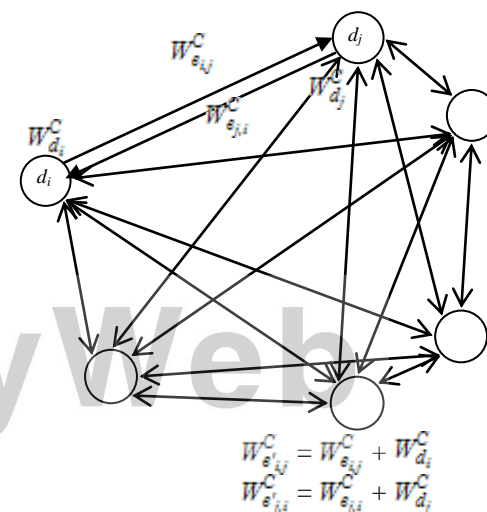
End

四、 效能分析

4.1 時間複雜度



圖三：Edge Weight 示意圖



$$W_{e_{i,j}}^C = W_{e_{i,j}}^C + W_{d_i}^C$$

$$W_{e_{j,i}}^C = W_{e_{j,i}}^C + W_{d_j}^C$$

圖四：Node Weight 示意圖及加入 Node Weight 至 Edge Weight

在Preprocessing 部份，將電子地圖網格化後，產生了 $I \times J$ 個網格，共有 N 個網格並將 Block 設定完成得到有向圖 $G(V, E)$ ，其時間複雜度為 $O(N)$ ，Preprocessing 只需執行一次即可。接下來，逐一推導每步驟時間複雜度，進而得到 TSP 多點實際路徑演算法的時間複雜度。

TSP 多點實際路徑規劃演算法的第一步驟建立完全連接圖是藉由最佳實際路徑規劃演算法求出各個點與點之間的權重值，而使用最佳實際路徑規劃演算法需花費 $O(\sqrt{N \log C})$ ，而 C 為鏈結權重中的最大值。起點和目的節點數量為 $M+1$ 個，則產生的鏈結數為 $(M^2+M)/2$ ，故推之 TSP 多點實際路徑規劃演算法第一步驟的時間複雜度為 $O(M^2 \sqrt{N \log C})$ 。

第二步驟加入節點權重 $w_{a_i}^C$ 至邊權重 $w_{e_{ij}}^C$ 中，只需要常數時間。而目的節點數為 M 個，由於為非對稱性結構，故所需加入邊權重的個數為 $2M$ 個，故推得第二步驟的時間複雜度為 $O(M)$ 。

第三步驟建立多點實際路徑，因需比對至各節點之權重值大小來做為參考依據，故第一次需比對 M 次，第二次需比對 $(M-1)$ 次，依此類推，則可推得第三步驟的時間複雜度為 $O(M^2)$ 。

綜合以上所述，我們推出 TSP 多點實際路徑規劃演算法的時間複雜度為 $O(M^2 \sqrt{N \log C})$ 。

4.2 空間複雜度

由於本計劃是以最佳路徑規劃演算法作為基礎，故資料結構與其相似，在 Preprocessing 部份，把電子地圖網格化後，產生 $I \times J$ 格網格，共有 N 格網格並將 Block 設定完成得到有向圖 $G(V, E)$ ，其空間複雜度為 $O(N)$ ，本 Preprocessing 只需執行一次即可。接下來，TSP 多點實際路徑規劃演算法空間複雜度如下計算。

第一步驟建立完全連接圖 $G^C(V^C, E^C)$ ，起點與目的地節點數之總和為 $M+1$ ，產生完全連接圖所需的鏈結數為 $(M^2+M)/2$ ，故第一步驟所需之空間複雜度為 $O(M^2)$ 。

第二步驟加入節點權重 $w_{a_i}^C$ 至邊權重 $w_{e_{ij}}^C$ 中，需儲存節點權重 $w_{a_i}^C$ 總數為 $M+1$ ，以及鏈結權重 $w_{e_{ij}}^C$ 總數為 $(M^2+M)/2$ ，因此第二步驟空間複雜度為 $O(M^2)$ 。

第三步驟建立多點實際路徑，針對第二步驟所儲存的資訊去做判斷，故第三步驟空間複雜度為 $O(M^2)$ 。

總結上述結果，我們得到 TSP 多點實際路徑規劃演算法的空間複雜度 $O(M^2)$ 。

五、智慧型手機應用實例與分析

本節將介紹實作於 Android 系統的 TSP 多點實際路徑演算法之結果，進行路徑規劃的地區為三峽區。如圖 6，本小節藉由最佳解之結果與本文 TSP 多點實際路徑演算法之結果進行效能比較。但由於旅行銷售員問題為 NP-Complete 的問題，因此假設在相同空間下，選擇目的地節點數目為 10 之較小範例，隨機產生 10 組範例，透過人為計算出最佳解之結果，並將相同範例帶入至本文之 TSP 多點實際路徑演算法。本文之 TSP 多點實際路徑演算法的路徑雖然比較長，但是平均長度差異相較於最佳解多了 9.4%。而在時間複雜度以及空間複雜度都有所改善，時間複雜度從 $O(M^2 2^M)$ 縮至 $O(M^2 \sqrt{N \log C})$ ，空

間複雜度也從 $O(M^2 2^M)$ 縮小至 $O(M^2)$ 。

六、結論

本文針對配送業者考量各種複雜的交通路況加入許多道路通用模型，並且搭配針對物流架構所建構的 TSP 多點實際路徑演算法，便可有效的規畫出較小成的輸送路徑。

參考文獻

- [1] L. Bodin, B. Golden, A. Assad and M. Ball, "Routing and scheduling of vehicles and crews: the state of the art," *Computers and Operations Research*, Vol. 10, pp. 63- 211, 1983.
- [2] D. Chen, S. Li, F. Qiu and N. Xu, "Research on hub-and-spoke logistics network construction: An empirical analysis from Zhejiang Province in China," *International Conference of Logistics Systems and Intelligent Management*, Vol. 2, pp.1003-1007, 2010.
- [3] M. Dorigo, V. U. Vrije, B. Brussels and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, Vol. 1, no. 1, pp. 53-66, 1997.
- [4] M. M. Flood, "The Traveling Salesman Problem," *Operations Research*, Vol.4, pp. 61-75, 1995.
- [5] M. R. Gaery and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Company, 1979.
- [6] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of Society Industrial and Applied Mathematics*, Vol. 10, pp. 196-210, 1962.
- [7] G. E. Jan, M. C. Lee, S. C. Hsieh and Y. Y. Chen, "Transportation Network Navigation with Turn Penalties," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, No. 0421, pp. 1224- 1229, July 2009.
- [8] G. E. Jan, K. Y. Chang and I. Parberry, "Optimal Path Planning for Mobile Robot Navigation," *IEEE/ASME Transaction on Mechatronics*, Vol. 14, No. 9, pp. 925- 936, Aug. 2008.
- [9] G. E. Jan, M. C. Lee, Y. S. Huang and B. S. Lin, "Practical Route Planning on the Smart phones," *2011 World Academy of Science, Engineering and Technology*, pp. 24-26, Jun. 2011.
- [10] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza and S. Dizdarevic, "Genetic algorithm for the traveling salesman problem: a review of representations and operators," *Artificial Intelligence Review*, Vol. 13, no. 2, pp. 129-170, 1999.
- [11] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, Vol. 21, pp. 498-516, 1973.
- [12] S. Lin, "Computer Solutions of the Traveling Salesman Problem," *The Bell System Technical Journal*, Vol.44, pp. 2245-2269, 1965.
- [13] C. Mlandraki and R. B. Diall, "A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem," *European Journal of Operational Research*, Vol. 90, Issue 1, pp. 45- 55, 1996.
- [14] T. Volgenant and R. Jonker, "A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation," *European J.of Operational Research*, Vol. 9, Issue 1, pp. 83- 89, 1982.
- [15] Q. Wang, L. Xiong, H. Liu and J. Liang, "Improved particle swarm algorithm for TSP based on the information communication and dynamic work allocation," *Asian Journal of Information Technology*, Vol. 5, no. 11, pp. 1191-1196, 2006.
- [16] G. J. Woeginger, "Exact Algorithms for NP-Hard Problems: A Survey," *Lecture Notes in Comp. Sci.*, Vol. 2570, pp. 185-207, 2003.



(a)設定需拜訪的目的地節點



(c)走向第二點



(b)走向第一點



(d)依照規劃路徑走完所有點

圖6：Android實作之運行圖

