逢 甲 大 學
資 訊 工 程 學 系 碩 士 班
碩 士 論 文 (初 稿)

近似巡訪天際線路徑查詢

Approximate Patrolling Skyline Path Query

指導教授：陳奕中 _____
(指導教授認可簽名)

研 究 生：邱聖崴

中 華 民 國 一 百 零 六 年 三 月

摘要

本研究提出 *Approximate Patrolling Skyline Path Query*，簡稱 APSPQ 來解決過去鮮少研究討論的巡訪與多條件路徑查詢問題。所謂的巡訪指的是在多次的查詢中行經各個路段。例如下雪中的城市，鏟雪車必須要能夠行經積雪量較多的路段，而這些路段又被稱為巡訪路段，這會導致每一次查詢的結果都不盡相同，且在每一次查詢時都要能有效率的找到積雪量較多的路段，除此之外還必須考慮到每個路段上的條件，例如距離、花費時間或安全性等等，因此本研究會遇到下列問題(1)要如何快速得找出巡訪路段(2)巡訪路段若需要刻意繞路才能行經時該如何解決(3)多條件路徑查詢上的效率以及產生不合理路徑的問題。

在 APSPQ 中我們首先透過佇列的概念建立的一索引模型來快速取得巡訪路段，且避免了多次排序來取得積雪量較高的步驟，接著由此模型得到多個可供巡訪的路段後發展 *Confidence relevent Network Road Skyline Query* 來找出相較之下較不繞遠路的巡訪路段，此查詢是基於空間天際線的概念所發展。而在不同的應用與使用者皆對繞路有不同的容忍值，透過使用者所提供的容忍值，我們建立一路網來改善傳統天際線路徑的不合理以及效率上的缺陷。

關鍵詞：路徑規劃、巡訪路徑、天際線、天際線路徑、空間天際線

目錄

圖目錄

表目錄

第一章 緒論

在諸多的地理資訊應用中，路徑規劃隨著 Google Map 或是 Yahoo! Map 等服務的普及已成為了重要的一環，不同使用者都會透過這些服務解決路徑規劃的問題。而近年來對於路徑規劃系統也不再僅限於空間上的查詢，研究者們也開始考慮時間對路徑規劃系統所造成的影響。舉例來說在下雪中的城市路網，每個路段的積雪量會隨著時間增加，如圖 1.1 路網中每個路段皆記錄不同時間戳記時的平均積雪量，觀察從時間戳記 T＝0 的圖 1.1(a)到 T＝1 的圖 1.1 (b)，可以發現每個路段的平均積雪量都增加了 10 釐米，因此積雪量這個屬性是具時變性的。而在規劃鏟雪車的路徑時，為了避免有路段的平均積雪量過高以致於發生危險，鏟雪車的路徑應該優先經過平均積雪量較高的路段，若今天有一鏟雪車欲從圖 1.1 的 A 節點出發至 E 節點，觀察 T＝1 的(b)中 BE 是平均積雪量最高的路段，因此為了規劃鏟雪車的路徑能夠經過 BE，鏟雪車可能會走 ABE 來鏟除 BE 上的積雪，而鏟雪車經過 ABE 後，路段 AB 及 BE 的積雪量將降低為 0，如圖 1.1(c)所示。而其他路段在鏟雪車對 ABE 除雪時，其平均積雪量仍然會隨著時間增加，如圖 1.1(b)中路段的 AD 的積雪量從 40 變成如圖 1.1(c)中路段 AD 的 50。則隨著時間增加。倘若在 T＝2 時，鏟雪車欲再從 E 節點到 A 節點，那麼觀察 T＝2 的(c)中，平均積雪最高的路段為 CE，而能經過 CE 路段的路徑有 ECA 與 ECDA，當有多條路徑可以選擇時我們可以去計算路徑的總體平均積雪量，我們就能觀察到 ECA 相對 ECDA 的路徑總平均積雪量較多(ECA 為 200，ECDA 為 190)，代表鏟雪車行走 EAC 將能比起行走 ECDA 鏟除更多的雪，因此在 T＝2 時我們會認為 ECA 會是最佳的鏟雪路徑，而經過此次鏟雪後路網上各路段的積雪量將如圖 1.1(d)所示，那麼要如何在每次的查詢中快速得到具有時變性的平均積雪量較高的路段，並找出經過其路段且總平均積雪量最高的路徑即是本研究所探討的核心議題之

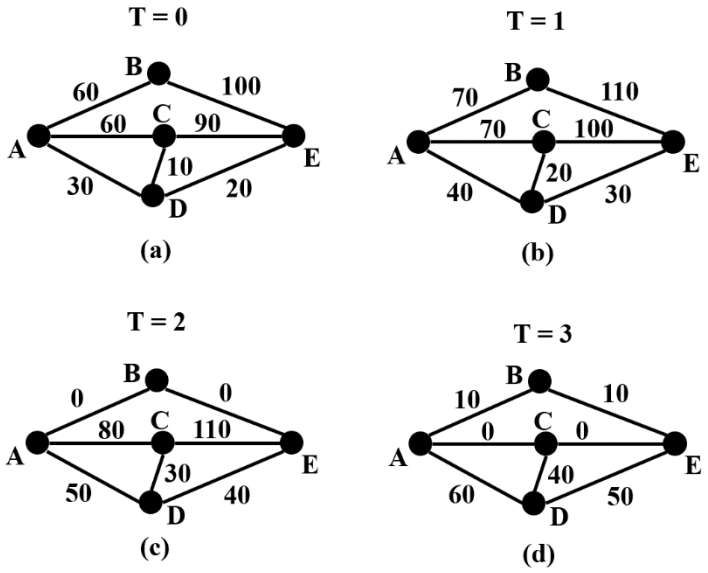

圖 1.1 路網圖積雪量變化示意圖

一。

上述的例子僅考慮到平均積雪量對路徑規劃的影響，然而鏟雪車在實際上路時還需要考慮到其他條件，例如行走距離或是平均耗油量等。舉圖 1.2 僅考慮路網上積雪量及鏟雪車行走距離的狀況為例，若今有一台鏟雪車從 A 出發至 E 時，他所有可行走的路徑如表 1.1 及各路段的總平均積雪量及總距離。對表 1.1 中的 ABE 與 ACE 兩條路徑來說，我們會認為 ACE 是比起 ABE 更好的鏟雪車路徑，因為 ACE 路徑上的平均積雪量大於 ABE 路徑(100 > 80)，且 ACE 路徑的距離小於 ABE 路徑(9 > 14)，接著我們觀察 ACE 與 ADCE 兩條路徑，可以發現 ACE 路徑的平均積雪量雖然小於 ADCE(100 < 140)，但 ACE 的距離卻小於 ADCE(9 < 14)，兩者並無法比較優劣。所以我們會將 ACE 與 ADCE 都呈現給使用者，讓使用者決定最後的鏟雪車路徑。

本論文提出了一個新的多條件路徑查詢來有效解決上述問題，並稱為巡訪天際線路徑查詢(*Patrolling Skyline Path Query*, PSPQ)，此查詢結合了巡訪與天際線路徑的兩大概念。所謂的巡訪是指必須找出那些較其他路段來說，離上次走訪時間最久而必須優先處理的路段，如同圖 1.1(a)中因時間較長未行走而導致平均積

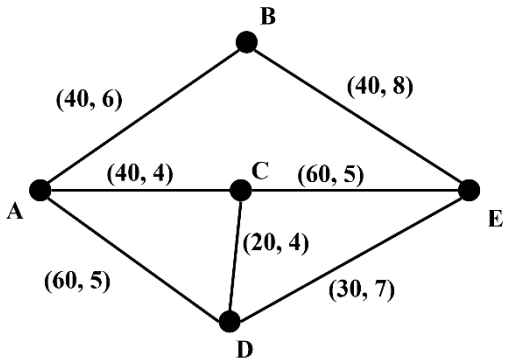

圖 1.2 多屬性路網示意圖

表 1.1 圖 1.2 A 至 E 的所有路徑以及其路徑屬性

path	(snowfall, distance)
ABE	(80, 14)
ACE	(100, 9)
ACDE	(90, 15)
ADCE	(140, 14)
ADE	(90, 12)

雪量較高的 BE 路段。為了以下說明方便，我們會將這些路段稱為巡訪路段。而天際線路徑[1][2]可以幫助使用者在含有多個條件路網上找出任一組起終點間適合移動的路徑。

本論文提出了一個新的多條件路徑查詢來有效解決上述問題，並稱為巡訪天際線路徑查詢(*Patrolling Skyline Path Query*, PSPQ)，此查詢結合了巡訪與天際線路徑的兩大概念。所謂的巡訪是指必須找出那些較其他路段來說，離上次走訪時間最久而必須優先處理的路段，如同圖 1.1(a)中因時間較長未行走而導致平均積雪量較高的 BE 路段。為了以下說明方便，我們會將這些路段稱為巡訪路段。而天際線路徑[1][2]可以幫助使用者在含有多個條件路網上找出任一組起終點間適合移動的路徑，舉圖 1.2 來說，ABE、ACE 與 ADCE 都是 AE 間的可行路徑。其

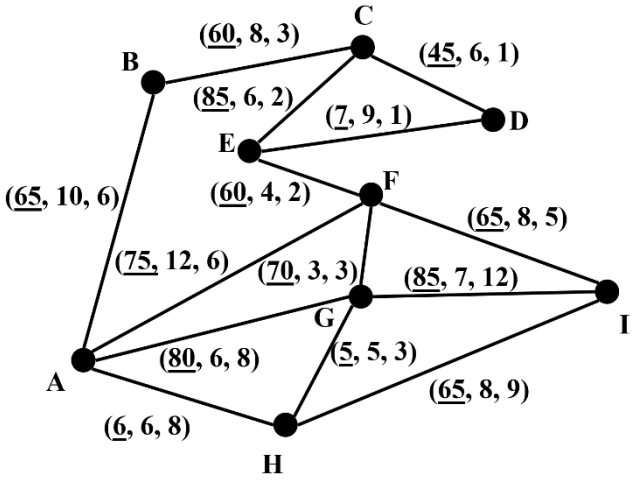

圖 1.3 PSPQ 定義示意圖

中若像 ACE 各條件都比 ABE 路徑來的好，我們就會稱 ACE 支配 ABE，反之若是 ACE 與 ADCE 兩條路徑在各條件各有優劣，我們就會稱 ACE 與 ADCE 互不支配，而天際線路徑即是找出任一起終點間所有不被支配的路徑，例如在表 1.1 中的天際線路徑為 ACE 與 ACDE。

根據上述兩個內容，PSPQ 的定義如下，給予一無向圖，圖中的每一個路段皆有一個信心指數與多個靜態維度，每個路段的信心指數落於 0 至 100 間，且信心指數越高代表此路段相對其他路段較不需要巡訪，反之則相反。那為了易於討論，本論文僅考慮信心指數隨著時間遞增而線性下降的情況。除此之外當一段路徑在走訪後，會將其路徑上的所有路段信心指數設為 100，代表短時間內不再需要走訪。PSPQ 的目的在於找出起終點中行經信心維度最低的路段之路徑，且這些路徑在考慮信心指數與靜態維度下必為天際線路徑。以下舉圖 1.3 為例，每個路段上皆有信心指數與靜態維度的距離以及平均耗油量共三個維度，而假設今天起點為 A 終點為 I 的情況下，我們可以將 A 至 I 部分的路徑用表 1.2 呈現，且表 1.2 中也列出路徑屬性的總合，如其中的 AGHI 即為 AG、GH 與 HI 路段的屬性的加總，且畫底線的數值表示為信心指數。利用圖 1.3 我們可以發現 GH 路段上的信心指數為所有路網中最低的路段，因此我們可以找出所有經過 GH 路段的路

表 1.2 圖 1.3 A 至 I 部分路徑與其路徑屬性

path	(confidence, distance, oid)
***AGHI**	(160, 11, 20)
***AHGI**	(106, 18, 23)
***AHGFI**	(156, 22, 19)
***AFGHI**	(225, 28, 21)
***ABCEFGHI**	(420, 44, 28)
***ABCDEFGHI**	(377, 53, 28)
ABCDEFI	(292, 45, 18)
ABCDEFGI	(382, 47, 28)

徑有 AGHI、AHGI、AHGFI、AFGHI、ABCEFGHI 與 ABCDEFGHTI 路徑(如表 1.2 中有星號標誌的路徑)，藉由天際線路徑的概念可以發現皆沒有其他路徑可以支配 AGHI、AGHI 與 AHGFI 路徑，因此這三條路徑即是我們巡訪天際線路徑。

PSPQ 能有效針對時變性屬性，如上述信心維度最低的路段，找出其天際線路徑，但基於傳統天際線路徑查詢的三個特性，會造成 PSPQ 在實際路網查詢時的困難。(1)天際線支配概念中，縱使維度數值只相差無幾還是會將較差的那筆資料支配，例如對圖 1.3 中 AH 路段就會被 GH 路段支配，但我們可以發現兩者的信心指數只差 1，代表這些路段的積雪量可能是相差無幾的，且鏟雪車應一併考慮到這些路段，因此該如何考慮信心指數低下的多個巡訪路段是本研究必須克服的問題。(2) 一般來說，大多數人在點與點之間的移動時不會刻意繞遠路，但在天際線路徑概念中，有一路徑只有要一維度特別優於其他路徑時，縱使路徑距離相當長依然會是天際線路徑結果，這也就是說，若使用天際線路徑來進行查詢，我們會找到許多不符合現實情況的路徑。也因此本論文也將針對此部分進行討論 (3)天際線的路徑查詢是相當耗費計算資源的查詢，尤其當城市路網路口與道路數上升時計算時間也會劇烈的成長，這將難以在合理的時間內給予使用者路徑。

因此在設計查詢演算法時，我們也需一併考慮此問題。

為了解決上述傳統天際線路徑查詢(1)的問題。我們定義一近似巡訪天際線路徑查詢 (*Approximate Patrolling Skyline Path Query*, APSPQ) 演算法來解決同時找尋多個巡訪路段的問題，首先我們必須先決定要巡訪的路段數 k 個，並找出信心指數最低的 k 個路段，並找出經過這 k 個路段其一的天際線路徑，即為 APSPQ 的結果。同樣舉具多維度的路網圖 1.3 起點 A 終點 I 為例，假設我們今天要找出三個巡訪路段，因此我們先找出信心指數最低的三個路段，分別是 GH、DE 與 AH，接著表 1.2 列出所有經過 GH、DE 與 AH 路段的路徑以及其路徑屬性，我們能觀察到路徑 AGHI、AHGI、AHGFI 與 ABCDEFI 為天際線路徑，因此我們會提供這些路徑給使用者，同時也為 APSPQ 的結果。

過往的研究中，也有人討論過巡訪與天線路徑查詢的問題，但這些方法都無法直接被應用到本論文中。首先針對巡訪路徑的相關研究，[1][2][3][4]皆提出巡訪的研究，其中[1][2]提出的方法成功的解決了在任意平面上巡訪的研究，但我們必須了解到本論文主要應用於道路路網中的應用，所考量的內容皆與道路屬性有關，但這些論文沒有針對道路屬性做討論，故難以應用於本論中，而[3][4]則提出路網上的巡訪查詢，但這些研究目的在於一次走訪全部路段，與本研究一次找出一個合適路段的概念概念不同，同時這些研究的做法也無法處理本論文路段屬性具時變性的問題。

在天際線路徑及相關查詢方面，已有一些人做討論，主要可分為無索引、索引以及人工智慧的方法在無索引方面，由 Tian 等人[1]提出的貪婪演算法，以及 Kriegel 等人[2]利用貪婪演算法與三角不等式找出天際線路徑，而 Chen 等人 [16] 基於廣度優先搜尋法開發 ParetoBFS 來尋找類似天際線路徑的結果。而索引方面則有 Chen 與 Lee[20]提出於路網上建立 One-hop tree 來加速天際線路徑查詢的時間。人工智慧的方法有 Jeddisaravi[15]提出使用多目標基因的演算法來解決多條

件路徑。此外，由於 Hansen[]的研究中提出了在圖中的多條件路徑查詢是一個 NP-Hard 的問題，因此 Shekelyan 等人[14]提出方法求出線性天際線路徑，並近似天際線路徑的結果。以上研究皆成功提出不同的方法解決天際線路徑查詢問題，但過往的天際線路徑研究中卻沒有可直接處理時變性信心指數的研究，進而無法滿足本論文近似巡訪天際線查詢的需求。

要處理本論文所提出的 APSPQ 會遇到以下兩個難點。(1)因為路網上的路段非常多，且每個路段都有時變性的屬性，因此不同時間下的查詢結果可能不同，那麼該如何在使用者下完查詢的極短時間內找到最必須要巡訪的多個路段便成為一個非常重要的課題。(2)上述所提及傳統天際線路徑查詢可能耗時過久及查詢結果可能對使用者來說可能是繞遠路而不合理的情況也必須克服。針對難點(1)(2)本研究提出類型一的近似巡訪天際線查詢演算法(*type-1 Approximate Patrolling Skyline path query algorithm*, 1APS)來克服這兩個問題，針對難點(1)我們發展一循序信心指數模型來解決，此模型是基於佇列與信心指數隨時間變化的特性發展而成，並能快速地找出不同查詢時間點下，全路網中信心指數低下路段，進而加快整體的查詢速度。接著針對難點(2)，本研究提出了將原始路網縮減成子路網的概念來克服，所謂的子路網是指整個路網中，使用者從起點經巡訪路段到終點的可接受移動路徑範圍，如圖 1.4 虛線包含範圍及為子路網，因此我們不需檢查所有原始路網的節點與路段，只需針對子路網內的節點與路段找出天際線路徑，進而大幅加速天際線路徑查詢時間。

上述我們所提的 1APS，的確能有效克服時變性屬性與傳統天際線路徑的問題，然而找出的必行經的巡訪路段可能需要刻意繞路，如圖 1.3 的 DE 路段，都相較於其他兩個巡訪路段 GH 與 AH 來的遠的許多，因此必須合理的濾除掉這些路段。而此難點我們提出類型二的近似巡訪天際線查詢演算法(*type-2 Approximate Patrolling Skyline path query algorithm*, 2APS)來解決，我們設計一合

適路段挑選階段，並利用空間天際線的概念將不合理的路段如圖 1.3 的 DE 濾除，使得 APSPQ 所找出的路徑更能被使用者給接受。因此，本研究提出的基本 1APS 演算法比起 PSPQ 能考慮更多巡訪路段外，還能快速的找到近似天際線路徑，接著提出 2APS 演算法解決 APSPQ 會產生不合理巡訪路段，導致繞遠路的問題。在本論文的實驗模擬也印證了 2APS 能更有效的處理此類問題。

總結來說本研究的貢獻含以下四點。

- 定義一個新的時間空間查詢與其相關的近似查詢。
- 利用一資料結構來有效率地取得信心指數低下的路段與巡訪路段後更新其信心值。
- 藉由合適路段挑選階段將起訖點與信心指數低下的路段的路網距離中找出適用於此次的巡訪路段。
- 利用容忍子路網來找出近似解，克服傳統天際線路徑查詢會遇到的耗時過久以及產生不合理路徑的問題。

本研究共分成七章，第二章為與本研究有相關的研究以及文獻探討，第三章定義本研究所用到的參數，第四章介紹本研究所提出的 1APS 演算法，第五章則是本研究針對 APSPQ 所提出的 2APS，第六章說明本研究的模擬實驗，第七章則對本研究做結論以及未來研究。

第二章 相關研究

本研究與路徑規劃、多條件路徑以及索引模型領域相關。其中與路徑規劃相關的論文有[1]、[2]、[3]與[4]，不過本研究強調於多條件下的路徑規劃，因此與多條件路徑規劃相關的論文有[5]、[6]、[7]、[8]、[9]與[10]。而在此類路網等複雜度隨點與邊數量高度成長的資料又常以索引模型解決效率問題，與此相關的研究有[11]、[12]、[13]與[14]。

2.1 路徑規劃

從學者 Dijkstra[6]在圖上提出著名的 Dijkstra 演算法來找尋兩點間的最短距離後，各式相關的路徑規劃演算法被陸續地提出。而此類的圖搜尋問題著名的演算法還有 A*演算法[7]，其特點在於改善 Dijkstra 的缺點採用了最佳優先搜尋法，並使用至終點期望值來評估該走的方向，使得搜尋的節點數與時間消耗有明顯的改善。而後 D*演算法[8]在基於 A*上進一步的改良，D*演算法強調在動態資料時的路徑規劃，在走訪路徑的過程中不停地進行 Dijkstra 演算法，來找出不同時間點的最短路徑。

除了依據路段屬性找尋兩點間的路徑規劃外，也有許多路徑規劃的研究採用蟻群演算法[9][10][11][12]，此演算法的特點在模仿蟻群的行經路徑會留下費洛蒙的特徵來找出最終收斂的路徑，適合運用在存有歷史路徑資料或是有連續查詢的狀況下。

而 Rao 等人[3][4]所提出有關鏟雪的研究中是與本研究最為相關的研究，[3]提出了單一台鏟雪車如何在單一次的走訪中最有效率的行經每個路段。並提出了使用基因演算法來解決此問題。其也有延伸此議題的研究[4]，此研究中將規畫多台鏟雪車同時出動的問題，同時也採用了基因演算法找出多台鏟雪車的路徑。

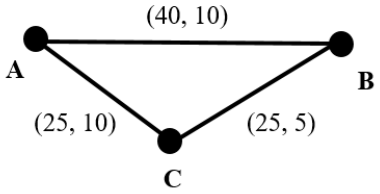

圖 2.1 片段天際線概念

2.2 多條件路徑

過往的路徑規劃往往只著重於單一條件最佳的情況下，Tian 等人[1]與 Kriegel 等人[2]將傳統天際線[13]的概念延伸至路網上並提出天際線路徑相關的演算法。Tian 等人的研究[1]試著存下每一次走訪路徑時的片段路徑，如圖 2.2 中展示當片段路徑 AB 已經支配當下走訪 ACB 時，此片段路徑 ABC 將不需要再被展開，因為無論 ACB 後續的路徑如何變化，都會被 AB 的後續路徑所支配。而 Kreigel[2]等人則提出以貪婪演算法並利用三角不等式來加速找出天際線路徑的結果。

而天際線路徑有一特性即是當路往複雜度隨之上升時，所找出的天際線路徑數量也會劇烈的上升，因此 Shekelyan[14]等人在計算天際線路徑時並不非採用傳統的天際線結果，而是求出線性天際線結果。圖 2.2 展示了線性天際線與傳統天際線的差異性，可以看到在相同的資料集下，線性天際線找出的結果會明顯少於傳統天際線。此篇研究採用線性天際線路徑的原因在於減少傳統天際線路徑過多的問題，而採用的方法則是使用凸包的概念發展 Linear Skyline Concex Hull (LSCH)演算法，在建構時會加入新的維度，並與單一條件最短路徑做結合來找出是否存在線性天際線結果。

除了有天際線路徑的研究可以處理多條件路徑外，Jeddisaravi 等人的研究[15]意在找出可移動機器人在多條件下的移動路徑。此研究採用了基於 A*演算法發展多目標基因演算法結合混沌數列，來同時考慮距離、機器人移動的平滑度以及安全性三個條件來找出合適的路徑，但此研究並非處理含節點與路段的路網

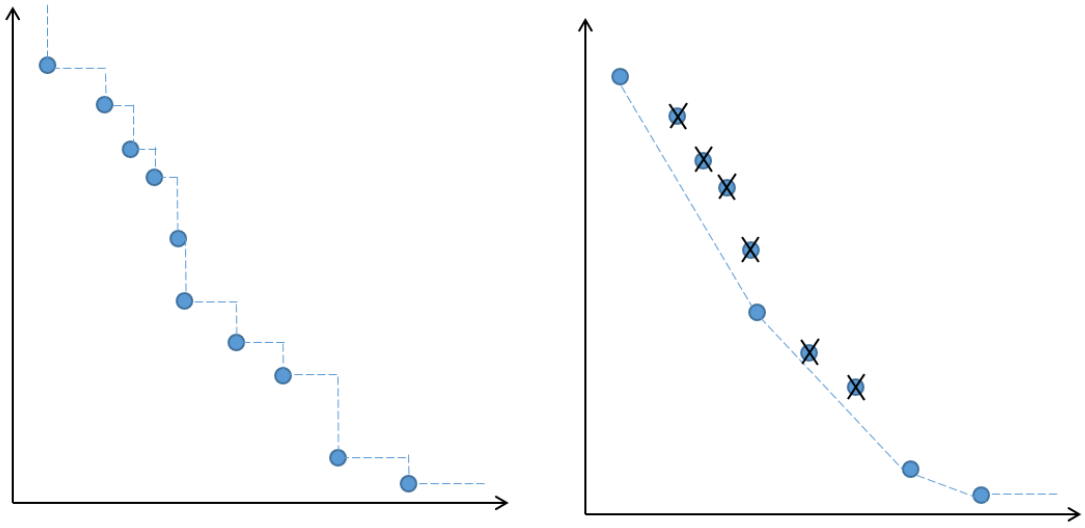

圖 2.2 傳統天際線與線性天際線示意圖

上，而是在開放式的地圖上找尋多條件路徑。

而與天際線路徑概念相彷的研究有 Pareto-optimal 路徑[16]意在找出路徑，此路徑結果等同於天際線路徑，而此篇發展方法基於廣度優先搜尋來發展一 ParetoBFS 來找出 Pareto-optimal 路徑，並適用於軟體定義網路的圖結構上。

2.3 索引模型

為了找出空間上相關的資料點，Guttman 等人[17]提出了以最小矩形(*Minimum Bounding Rectangle*, MBR)為基本結構的 R-tree，此樹結構每一個非葉節點均為一個 MBR，而葉節點為每一個真實節點。MBR 的特色在於會將空間中相近的點建立於同一 MBR 中，而在走訪 R-tree 時就能快速找出空間上相近的節點並快速濾除其他 MBR 的節點。

而傳統 R-tree 並未考慮到物體會移動的性質，因此 Zhang 等人[]發展了 Time Parameterized R-tree (TPR-tree)來解決移動物體的問題。TPR-tree 中的中間節點會將每個 MBR 的範圍、移動速度、子節點的指標與 ID 記錄下來，葉節點則會記錄每個物體的基本資訊、座標、移動速度以及 ID。當物體移動後可能會超出原本所在的 MBR，那麼 TPR-tree 在記錄 MBR 的範圍及移動速度下，我們就能確保

移動的物體依然會在 MBR 的範圍內。

由於路網結構會隨著節點以及路段數上升而隨之複雜，Zhong 等人[18][19]發展了一種專用於路網上的索引樹 G-tree，其葉節點會包含樹個道路節點，而每一中間節點會紀錄從此中間節點對外節點至其他節點的最短距離。用以解決大型路網在找尋最短路徑或是最近鄰居點效率不彰的問題。

在多條件路徑上的索引結構，也有由 Chen 與 Lee[20]提出 One-hop tree 研究，此研究有別於 R-tree，採用的是真實路網的連接關係，由此連接關係中找出鄰居點並成為此樹的中間節點並計算其中的天際線路徑以及對外路徑，從而在走訪路徑時，加速濾除不必要的道路節點，因此可加速天際線路徑的查詢。

第三章 定義

定義一. 節點： 所有路網上道路與道路的連接處，用 $\mathbf{V}=\{v_1,v_2,\ldots,v_{|V|}\}$ 表示所有為圖中的節點，其中 $|\mathrm{V}|$ 為所有節點的數量。 ■

定義二. 路段： 所有路網上節點與節點的連接線，用 $\mathbf{E}=\{e_1,e_2,\ldots,e_{|E|}\}$ 表示為所有圖中的路段，其中 $|\mathrm{E}|$ 代表所路段的數量。而 $e_i=\{v_{i1},v_{i2}\}$ 可以代表路段，其 e_i 是由兩個節點 $v_{i1},v_{i2}\in\mathbf{V}$ 所構成。 ■

定義三. 路段屬性： 路段 $e_i\in\mathbf{E}$ 的路段屬性用 $\mathbf{Attr}(e_i)=\{w_{i1,}w_{i2},\ldots,w_{id,}c_i\}$ 來表示，其中 w 代表為靜態維度，表示為那些不會隨時間變動的維度，d 代表為靜態維度的數量，c 則代表信心維度，n 為信心維度的數量，並且信心維度 c 皆隨時間線性下降，範圍為 $0\le c\le 100$。 ■

定義四. 多屬性路網圖： 給予一個城市路網圖且每個路段上皆有多個路段屬性被稱為多屬性路網圖(*Multi-attribute network graph*, MAG)，可以表示為 $\mathcal{G}(\mathbf{V},\mathbf{E},\mathbf{Attr}(\mathbf{E}))$，其中 **V** 表示為路網上的所有節點，**E** 表示為路網上的所有路段，且 $\mathbf{E}\subset\mathbf{V}\times\mathbf{V}$，$\mathbf{Attr}(\mathbf{E})$ 代表每一路段上的屬性。 ■

定義五. 路徑： 路徑可以用 $\mathbf{P}=\{p_1,p_2,\ldots,p_{|P|}\}$ 表示，其中每條路徑可表示為 $p_k=(e_{k1},e_{k2},\ldots,e_{k|P|})$，其中 P 則表示為路徑節點的數量，並由多個路段 $e\in\mathbf{E}$ 組成且任兩路段 $e_{ki},e_{kj}\in p_k$ 皆必須成立 $e_{ki}\neq e_{kj}$，且 $e_k\in p_k$ 不會形成迴路。 ■

定義六. 路徑屬性： 每一條路徑 $p_k=(v_{k1},v_{k2},\ldots,v_{k|P|})$ 的花費可以表示成 $\mathbf{Attr}(p_k)=\{pw_{k1,}pw_{k2},\ldots,pw_{kd,}pc_k\}$，並可以由以下兩式求得 pw 與 pc。

$$pw_i=\sum_{j=0}^{|P|-1}w_{ji}$$

$$pc_i = c_i$$

其中 pw_{ji} 代表路段 e_j 的靜態維度， pc_i 則代表信心維度。 ■

定義七. 支配：如果有一維度大小為 N 的資料點 $data_i = (w_{i1},\ldots,w_{i|N|})$，其中 w 代表此資料的屬性，若 d_i 可以支配於 d_j ，那麼下式成立。

$$d_i.w_{ik} \le d_j.w_{jk} \text{ 對每一個 } k = 1,\ldots,n$$ ■

定義八. 天際線：若資料點集合 $\mathbf{Data} = \{data_1, data_2,\ldots, data_n\}$ 中不存在任何一點 $data_j \in \mathbf{Data}$ 可以支配 $data_i \in \mathbf{Data}$ ，那麼 d_i 就是天際線且 $data_j \neq data_i$ 。 ■

定義九. 天際線路徑：若路徑集合 $\mathbf{P} = \{p_1, p_2,\ldots, p_{|P|}\}$ 中不存在任何一條路徑 $p_j \in \mathbf{P}$ 的 $\mathbf{Attr}(p_j)$ 可以支配路徑 $p_i \in \mathbf{P}$ 的 $\mathbf{Attr}(p_i)$，那麼 p_i 就是天際線路徑，且 $p_j \neq p_i$ 。 ■

第四章 研究方法

本研究首先提出 APSPQ 演算法。我們用圖 4.1 的流程圖來說明 APSPQ 演算法。首先 APSPQ 演算法可以分成離線初始化查詢以及線上查詢兩個部分。在離線初始化查詢部分，核心在於建立循序信心指數模型，此模型可以管理各路段具時變性的信心指數，我們將路網上所有的路段(如圖 4.1 中的 A 部分)以及其信心指數建立循序信心指數模型(如圖 4.1 中的 B 部分)。之後在線上查詢部分，當有一個新的查詢時，使用者會提供相關資訊進行查詢(如圖 4.1 中的 C 部分)。接著在挑選巡訪路段(圖 4.1 中的 D 部分)中會找出目前信心指數較低的必經巡訪路段。有了巡訪路段後，就可以進行容忍子路網的建構(圖 4.1 中的 E 部分)來解決天際線路徑特性所造成的問題，最後我們就能由此路網中找出天際線路徑，並且當使用者挑選其中一條路徑行經後，循序信心指數模型將會被更新 (圖 4.1 中的 F 部分)。如此一來即完成一次的 APSPQ 演算法。

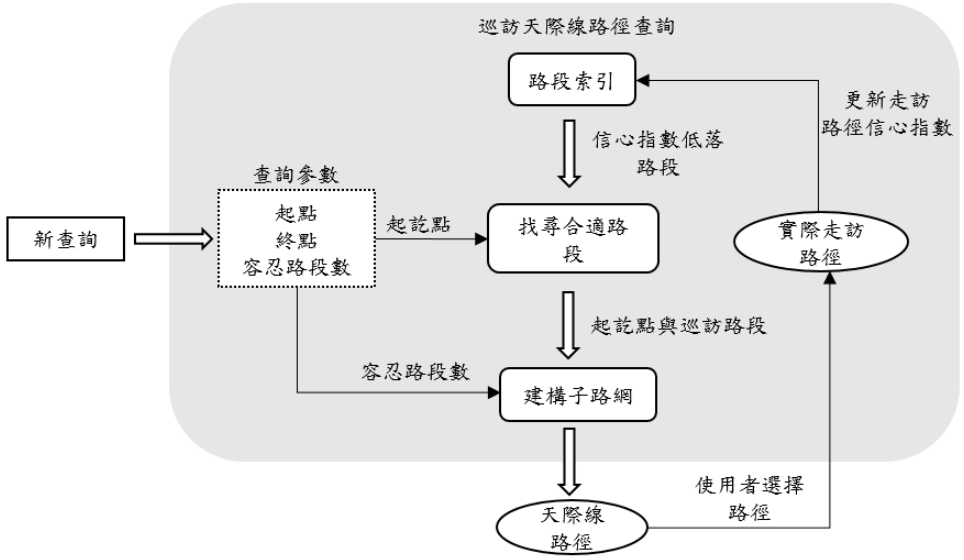

圖 4.1 巡訪天際線路徑查詢流程圖

4.1 取得信心指數低落路段

為了避免有路段過久未被巡訪，本研究想試圖找出當前信心指數較低的路段，但由於每一個路段上的信心指數都會隨著時間下降，因此若必須找出當前急需巡訪的路段，也就是要找出信心指數低下的路段就必須在每一次查詢前依據各路段的信心指數由小至大排序路段來找出前幾個急需被更新的路段。由此可知這是非常耗費時間的，除了每一次信心指數的排序會耗費時間外，排序後還必須推至其信心指數所對應的路段我們才可得知該巡訪的路段。因此我們將基於佇列的概念設計一路段索引，此路段索引只在第一次建立的時候得到各路段的信心指數，接下來此索引將不再花費時間依據信心指數排序路段，而是利用時間下降的信心指數的特性以及佇列先進先出的概念來有效率的維護此路段索引。此部分可對應至圖 4.1 的路段索引。

此索引模型如同圖 4.2 中 Q_1 對應的是表 4.1 的路段，會將所有路段以信心指數作為排序依據，並且此處排序為由小至大，也就是說此模型會將信心指數較地的路段放於前面，在此我們應定義一 k 值，來決定我們每次要從此索引模型中

表 4.1 更新前路段上的信心指數

edge id	confidence
e_1	52
e_2	12
e_3	66
e_4	8
e_5	45
e_6	82
e_7	93
e_8	86
e_9	33
e_{10}	21

表 4.2 更新後路段上的信心指數

edge id	confidence
e_1	50
e_2	100
e_3	64
e_4	6
e_5	43
e_6	100
e_7	91
e_8	84
e_9	31
e_{10}	19

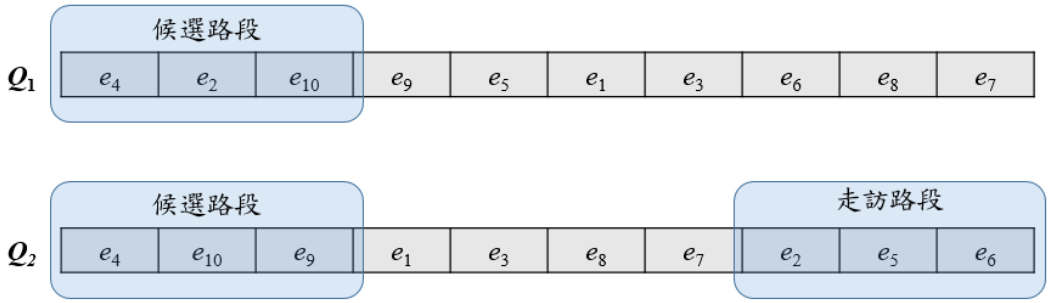

圖 4.2 路段索引示意圖

設定幾個候選路段，如圖 4.2 中 k 值為 3，因此 Q_1 的候選路段為 e_2、e_4 與 e_{10}。

然而在計算完最終的路徑並行走後，路段的信心指數應該被更新，如同表 4.2 即是表 4.1 在更新後的信心指數，而此時我們所定義的索引模型將會被更新成圖 4.2 的 Q_2，其中此次實際走訪的路段為 e_2、e_5 與 e_6，因此我們從此索引中找出這些路段移除並插至尾端。我們可以發現此索引模型並不需要依賴表 4.2 的信心指數，也就是說縱使路段被更新，我們僅需要將所走訪的路段移除並放入此索引模型的尾端，那麼其結果會等同於表 4.2 所示的信心指數。此舉將可使得每一次的查詢後我們皆可以快速地得到下次的備選路段，且完全不需要進行耗時排序行為。

4.2 找尋合適路段

當我們由上一段找出備選路段後，我們並不能直接對這些路段進行巡訪，這是因為在備選路段中的路段可能是離起訖點較遠的路段，若我們依然使用這些路段找出路徑，則會造成繞遠路的不合理情形，這種繞遠路的路徑參考價值相當低，如 PSPQ 於圖 4.1 的找尋合適路段中，因為每次查詢的起迄點不同，因此必須在所有信心指數低下的路段中找尋出適合此次查詢起訖點的路段。

在查詢信心指數低下的路段後，我們找到路段的狀況可能如同圖 4.4，*s* 與 *t* 各為起訖點，假設欲找出備選路段數為 3 的狀況下我們找到 A, B 與 C 三個應該優先處理的路段，但可以顯而易見的發現 C 無論是對 *s* 或是 *t* 都必須繞遠路才能經過 C，這對大部分使用者在選擇路徑上都是相當不合理的。若用於鏟雪路徑中，鏟雪車更應該專注在不刻意繞遠路的路徑來達到最高效益。為了找出合理的巡訪路段我們發展了信心相關之路網天際線查詢 (*Confidence relevent Network Road Skyline Query*, CNRSQ)來解決此類問題。此部分可對應至圖 4.1 的找尋合適路段。

4.2.1 空間天際線

空間天際線查詢[5]旨在找到在空間上不被支配的資料點，空間上不被支配代表的是查詢點與資料點間的直線距離，我們用圖 4.3 來舉例，圖中有兩查詢點分別為 Q_1 與 Q_2，以及三個資料點 A、B 與 C，而我們可以得到每一個資料點對上兩個查詢點的直線距離如表 4.3，可以發現到此處就可以運用傳統的天際線概念找出 A 與 B 兩點皆沒有被其他點支配，A 與 B 就可以表示成空間天際線。

若套用至本研究中，即可以假設各資料為上一階段所得的備選路段，而這些備選對應至起訖點皆有一路網距離，我們就能用空間天際線的概念找出離起訖點相對其他路段較近的那些路段為巡訪路段。

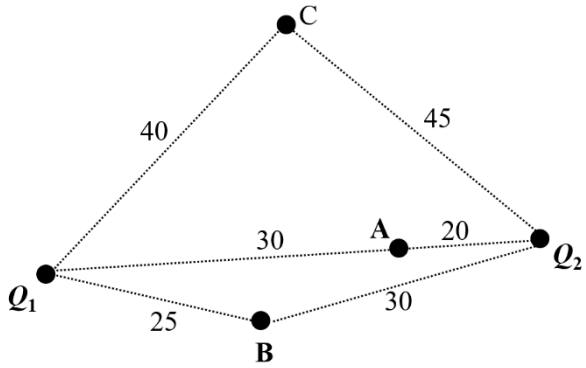

圖 4.3 空間天際線概念

表 4.3 空間天際線概念

point	Q_1	Q_2
A	30	20
B	25	30
C	40	45

4.2.2 信心相關之路網天際線

而傳統空間天際線查詢所使用的是直線距離，由於直線距離於路網中可能難以代表真實的路網距離，如圖 4.5 中，Q 點為查詢點，而 A 點到 Q 點的直線距離為 100 但路網距離卻需要 350，但 B 點到 Q 點的直線距離需要 150 而路網距離只需要 200，若採用直線距離計算將會取得 A 而不是 B，但以路徑來說所行走的會是路網距離，在此我們使用最短距離來取代直線距離。因此套用到我們的問題就可以將起訖點當成空間天際線中的查詢點，並用路網距離來取代直線距離。

再者[21]中所提出的概念為路網距離與文本相關程度所衍生出的新維度，並提及所使用之函式計算能有效表達空間上與文本上相關之資料點，這與我們的研究議題不謀而合，本研究探討的是路段優先順序，因此原先[21]相關性數值為文字相關性，並無法直接應用於本問題，本研究將延續相同概念，替換文字相關性成本研究所著重的信心指數，以符合巡訪路徑需求。

式子(1)為[21]所提出之文本相關空間天際線各屬性點的計算公式，q_i 代表查詢點，i 為查詢點的數量，p_j 代表的是資料點 j 為資料點之數量，$\mathrm{d}(q_i, p_j)$ 代表資料點對查詢點的直線距離，而 $\omega(Q.\psi, q_j.\psi)$ 則代表文本相關程度。

$$\mathrm{St}(q_i, p_j) = \frac{\mathrm{d}(q_i, p_j)}{\omega(Q.\psi, q_j.\psi)} \qquad (1)$$

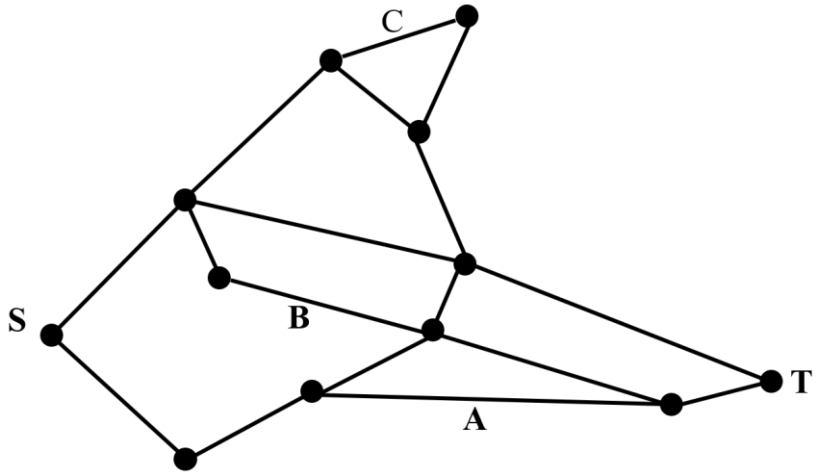

圖 4.4 不合理路段示意圖

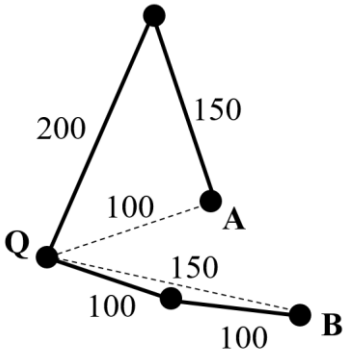

圖 4.5 直線距離與路網距離示意圖

由於本研究需要首先要將直線距離 $d(q_i, p_j)$ 換成路網最短距離，接著將 $\omega(Q.\psi, q_j.\psi)$ 替換成每一資料點的 d 值大小以表示我們路段的優先程度。最後可得公式(2)用以我們計算適用優先路徑的值。$q_{(s,t)}$ 代表起訖兩點，s 代表起點，t 則是終點，$E_{j(s,t)}$ 代表的是每一優先路段上的兩點，j 為優先路段之數量，$\mathrm{sp}(q_{(s,t)}, E_j)$ 代表起訖點至路段之路網最短距離， $E_j.d$ 代表路段 E_j 上 d 值。

$$\mathrm{cnr}(q_{(s,t)}, E_{j(s,t)}) = \frac{\mathrm{sp}(q_{(s,t)}, E_{j(s,t)})}{E_j.d} \qquad (2)$$

並可如下定義十一定義 CNRSQ 支配，然而運用公式(2)的值進行天際線運算能同時結合路段離起訖點的距離以及其路段上的信心指數，此方法相較其他方式下的計算方式有效[21]，藉此能找出此次查詢下較為合適的路段。

定義十一. CNRSQ 支配

給予一個查詢 Q

$$p_i dom_Q p_j \Leftrightarrow \forall q_k \in Q : cnr(q_k, p_j) < \mathrm{cnr}(q_k, p_j) \qquad (3)$$

我們用來表 4.4 與表 4.5 來說明我們如何利用 CNRSQ 來找出合適的路徑，我們首先觀察表 4.4，由於每一個位於 MAG 的路段都是由兩個節點所組成，因此表中的每一點都是由信心指數低下的路段所組成，例如 e_1 這個路段就可以被拆成 e_{1s} 與 e_{1d}，而表 4.4 中 S 即代表此節點與此次查詢的起點路網最短距離，T 就代表與終點的路網最短距離，D 則是信心指數，因為 e_{is} 與 e_{id} 代表的都是同一個路段，因此信心指數會相同，如表中 e_{1s} 與 e_{1d} 的 D 都是 20，我們可以利用公式(2)將表 4.4 推至表 4.5，例如表 4.5 中 e_{1s} 的 $\mathrm{cnr}(q_s, e_i)$ 就是由表 4.4 中 e_{1s} 的 S / D (30 / 20 = 1.5)，$\mathrm{cnr}(q_t, e_i)$ 則是表 4.4 中 e_{1s} 的 T / D (20 / 20 = 1)，接著我們在透過定義十一找出表 4.5 的天際線，會得到 e_{1s} 與 e_{2s}，我們透過 CNRSQ 後發現距離較遠的 e_3 的兩個節點都會被支配，因此我們就能在此階段得到 e_1 與 e_2 兩個合理的巡訪路段。

4.3 建構子路網

經由上述章節我們已經能夠得到合理的巡訪路段，不過若直接進行天際線路徑查詢將會產生不合理路徑的情況，表 4.6 為**錯誤! 找不到參照來源。**起點為 A 終點為 L 的天際線路徑查詢結果，即代表任一路徑都無法被其他路徑支配，但仔細觀察就發現路徑 p_2 與路徑 p_3 的軌跡難以符合一般人移動的習慣，我們應該將此屏除，進一步發現大部分的人也不見得會選擇路徑 p_4，根據不同者使用者有不同習慣，我們很難去抉擇是否該保留或摒棄路徑。因此本研究為了解決上述問題發展一動態資料結構容忍子路網(*Magnanimous Sub-Road Network*, MSRN)，試圖將以使用者的提供的容忍路段數建構合理的路徑搜索子路網，使得天際線路徑查

表 4.4 信心指數低下路段離起訖點距離與信心指數

point	S	T	D
e_{1s}	30	20	20
e_{1d}	25	30	20
e_{2s}	40	45	40
e_{2d}	40	50	40
e_{3s}	60	70	30
e_{3d}	70	80	30

表 4.5 CNRSQ 示意圖

point	$\mathrm{cnr}(q_s, e_i)$	$\mathrm{cnr}(q_t, e_i)$
e_{1s}	1.5	1
e_{1d}	1.25	1.5
e_{2s}	1	1.125
e_{2d}	1	1.25
e_{3s}	2	2.33
e_{3d}	2.34	2.67

表 4.6 天際線路徑示表

path id	path
p_1	A, B, C, D, E, L
p_2	A, B, C, E, D, F, G, J, L
p_3	A, B, C, D, F, G, H, I, K, J, L
p_4	A, B, C, E, D, J, L
p_5	A, G, F, D, E, L
p_6	A, G, J, L

詢結果更加貼近每一位使用者的需求。此部分可對應至圖 4.1 的建構子路網。

4.3.1 容忍子路網

MSRN 為一基於 MAG 上之子圖，其中由初始節點、內層節點與外層節點所組成，而無論是哪種節點皆是基於原始路網節點之上，在 MSRN 中外層節點可能會轉成內層節點。MSRN 並非為一或固定的路網，而是跟著每一次查詢下的起迄點、各路段的信心指數與使用者容忍的路段數構成。圖 4.9 中描繪了 MSRN 的建構過程，而圖 4.10 以虛線所涵蓋的範圍即為建構完成的 MSRN，以下將詳細

介紹 MSRN 的擴增、組成以及建構。

4.3.2 擴增

如上述 MSRN 必須由多個參數來決定，而使用者容忍的路段數即會構成 MSRN 的大小。而 MSRN 在建構時必須一步步延伸其路段，這樣的行為稱之為擴增。每一次擴增都會向外延伸一個路段，因此根據使用者容忍的路段數會決定擴增的次數，並逐漸增加 MSRN 的範圍。

4.3.3 內層節點

內層節點為 MSRN 節點中狀態不變的點，一但任一節點被標記為內層節點後即不會再改變其內層節點的性質。如圖 4.6 中擴增次數為 0 時內層節點為空集合，而在圖 4.7 中擴增次數為 1 時的內層節點為 a 與 b。

4.3.4 外層節點

位於外層節點的點為 MSRN 最外層的節點，每一個外層節點在未來皆有可能轉變成內層節點，並且同時代表著 MSRN 與未被 MSRN 所包含原始路網的界線。如圖 4.6 中擴增次數為 0 時 a 與 b 即為外層節點，而在圖 4.7 中擴增次數為 1 時外層節點為 c, d, e, f, g, l, n。

4.3.5 初始節點

在建構 MSRN 中需要的參數，初始節點數量必須不為零，在擴增次數為 0 時，初始節點同時也為外層節點，而在擴增次數大於 0 時的狀況初始節點即轉為內層節點。如圖 4.6 之位於原始路網 a 與 b 節點即為此 MSRN 的初始節點。

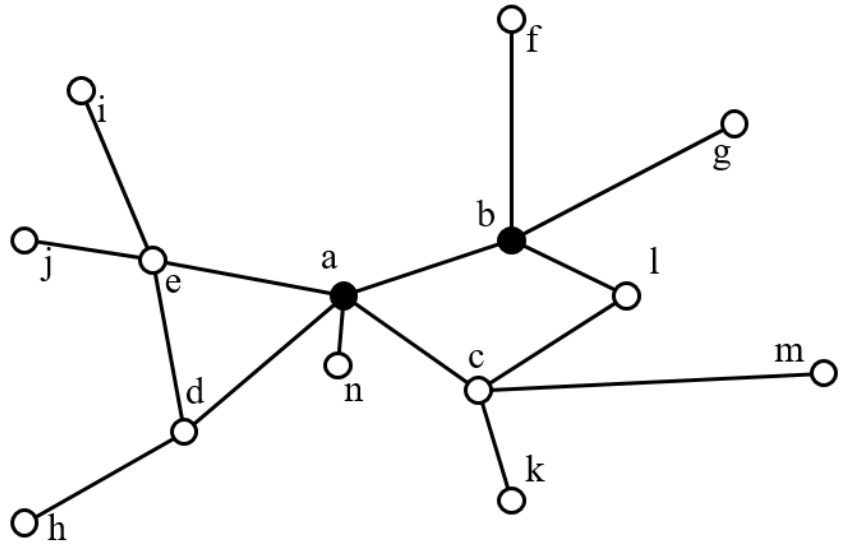

圖 4.6 MSRN 擴增次數 0 時

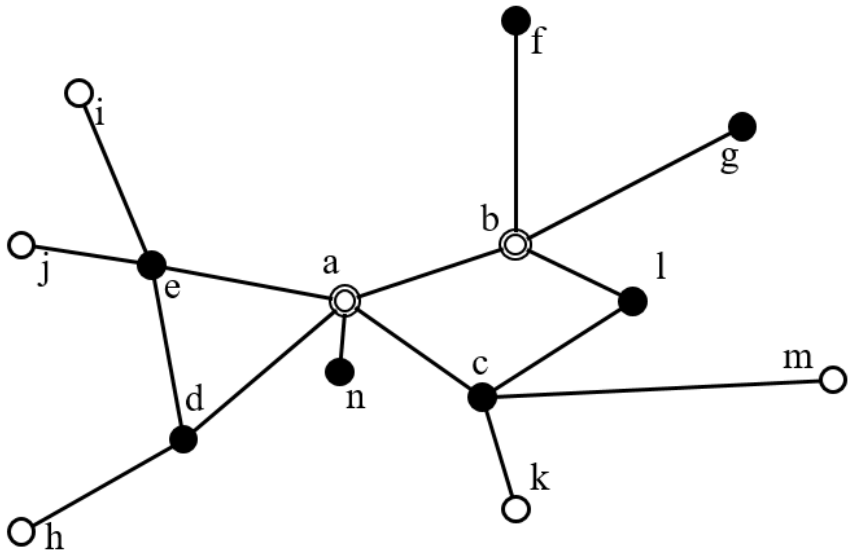

圖 4.7 MSRN 擴增次數 1 時

4.3.6 擴增容忍子路網

除了在 MSRN 的外層節點數為空集合時，MSRN 皆可以做擴增的動作，而每次擴增的次數也都代表 MSRN 總結點數的增長。在一次擴增時時，外層節點會選取所有自己的鄰居節點，接著排除其中是 MSRN 內層或外層節點的鄰居節點。在此動作後每一外層節點皆可以分成兩種狀況

- 有新鄰居點

在此狀況下，因存在新鄰居點所以外層節點得以被這些鄰居點所取代，舊有的外層節點將轉成內層節點，而這些被選取的鄰居點則轉成外層節點。如圖 4.8 (A)中 a, b, c, d 點起始點 Q 擴增次數為 1 時的外層節點，而圖 4.8 (B)表示擴增次數為 2 時因 c 有新鄰居 f，因此在經過此次擴增後，c 點本身轉為內層節點並且 f 將成為新的外層節點。

- 無新鄰居點

此狀況表示為當前外層節點的鄰居點皆為內層節點或是為其他的外層節點，因此直接將此節點轉為內層節點後不做任何擴增動作。換句話說，這個狀況即是 MSRN 某一外層節點已到達原始路網的邊界，無法進行擴增。如圖 4.8 (A) a 點為擴增次數 1 時的外層節點，而當圖 4.8 (B)擴增次數為 2 時因 a 的鄰居點皆為內層節點(Q 點)或外層節點(b 點)，因此在無新鄰居狀況僅將 a 點本身轉為內層節點，即 a 節點不會在向外擴增，下次擴增時也不需考慮為內層節點的 a 點。

4.3.7 建構容忍子路網

為了建構 MSRN，我們需要兩個參數，初始參數以及容忍路段數。

- 初始節點

大部分人在選擇路徑時都會考慮最短路徑，因此 MSRN 的初始節點也由路網最短路徑組成，我們會在得到 CNRSQ 的結果路段後，計算起訖點至這些路段的最短距離，並以這些最短距離上的每一節點當成 MSRN 的初始節點。

- 容忍路段數

容忍路段數代表的是使用者願意繞行的距離，根據不同使用者所提供不同的容忍路段數即會建構成不同大小的 MSRN，而 MSRN 也會隨著此路段數越大而越大。

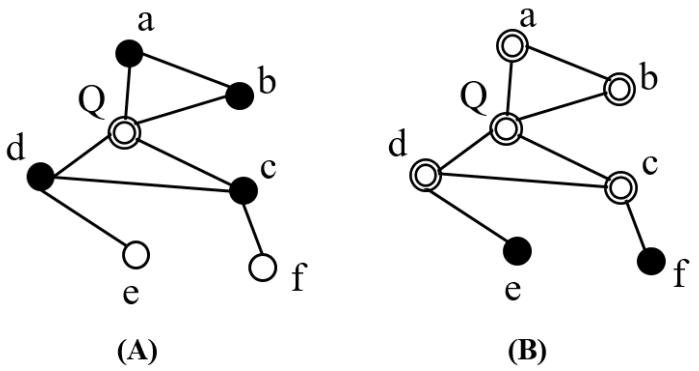

圖 4.8 鄰居點示意圖

圖 4.9 與圖 4.10 敘述了建構 MSRN 的過程，圖中為一完整路網，今天有查詢起訖點分別為 S 與 T，並且使用者容忍路段數為 1。我們藉由 CNRSQ 所找到路段為 FG 與 CD，因此我們分別找到以起訖點分別到 FG 與 CD 的最短路徑為 SFGT 與 SBCD，接著以最短路徑上的節點當成 MSRN 的初始節點如圖 4.9，接著因為使用者容忍路段數為 1 因此我們向外做一個擴張，結果如圖 4.10。因此我們就得到此次查詢的 MSRN。

4.4 巡訪天際線路徑查詢

以上介紹了 APSPQ 的各個部分，因此要進行一次的 APSPQ 首先我們先透過路段的索引模型快速的取得信心指數低落的路段 k 個，來保證查詢路徑時會經過這些路段以更新信心指數，接著在路網上信心指數低落的路段可能散布於路網各處，因此我們針對這些路段上節點離起訖點路網距離與信心指數做 CNRSQ 來找出合理的巡訪路段，最後我們可以透過起訖點與這些路段的最短路網距離路徑為基礎，結合使用者可容忍繞路的路段數建構出容忍子路網。至此，我們還需要進行天際線路徑查詢，此處所運用的天際線路徑查詢演算法為 Chiu 等人[22]的研究。為了要能夠讓天際線路徑必經過巡訪路段，我們必須於每一個路段進行天際線路徑查詢。但每個巡訪路段上皆有兩個節點，起訖點也為兩個節點，如此一來就會有四種天線查詢的組合，但由於於 CNRSQ 時我們已找出各巡訪路段上節點距離起訖點的距離，因此我們就能快速比較節點上的點對於起訖點的距離來找

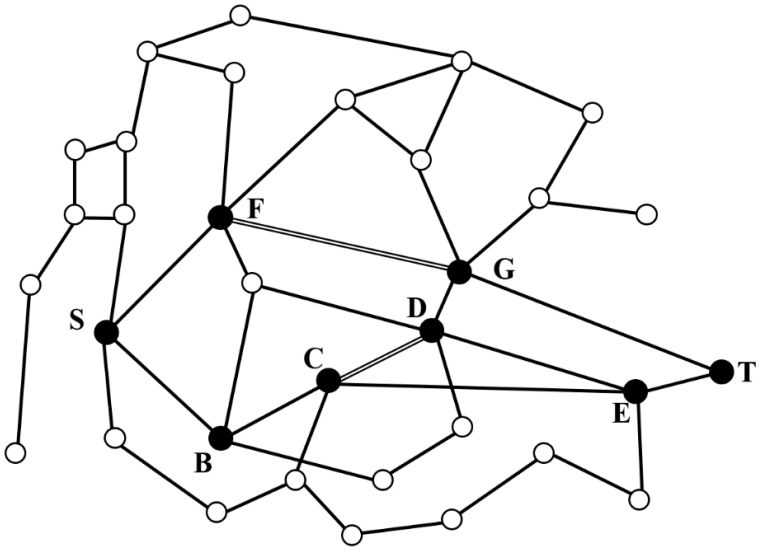

圖 4.9 MSRN 建構中示意圖

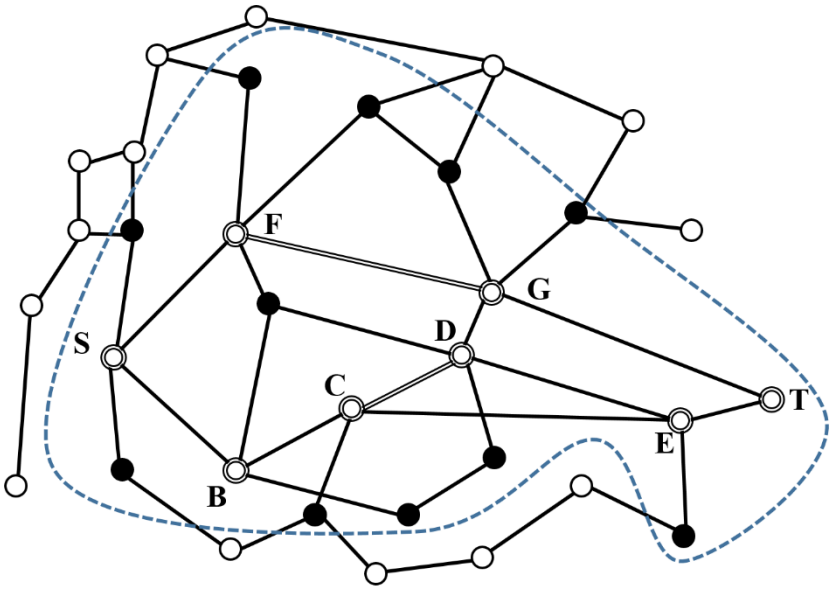

圖 4.10 MSRN 建構完成示意圖

出適當的天際路徑起訖點，如在上述中的計算可以得到表 4.4 的 e_{1s} 距離終點 T 的距離為 20 小於 e_{1d} 距離終點 T 的距離 30，因此於經過 e_{1s} 天際線路徑就為起點為 S 終點為 e_{1d} 與起點為 T 終點為 e_{1s} 的組合。依此類推找出所有巡訪路徑的天際線組合後即為 PSPQ 的結果。然後在使用者實際選擇一條天際線路徑後，將會更新路段索引。

參考文獻

[1] Y. Tian, K. C. K. Lee, and W.C. Lee, "Finding skyline paths in road networks," in *Proc. ACM Int. Conf. Adv. Geograph. Inf. Syst. (SIGSPATIAL)*, 2009, pp. 444-447.

[2] H.-P. Kriegel, M. Renz, and M. Schubert, "Route skyline queries: A multipreference path planning approach," in *Proc. IEEE Int. Conf. Data Eng.* (*ICDE*), Mar. 2010, pp. 261-272.

[3] T. M. Rao, S. Mitra, J. Zollweg, and F. Sahin, "Computing optimal snow plow route plans using genetic algorithms," in *Proc. IEEE Int. Conf.* Systems, Man and Cybernetics (*SMC*), Oct. 2011, pp. 2785-2790.

[4] T. M. Rao, S. Mitra, and J. Zollweg, "Route allocation for multiple snowplows using genetic algorithms," in *Proc. IEEE Int. Conf.* Systems, Man and Cybernetics (*SMC*), Oct. 2012, pp. 29-34.

[5] M. Sharifzadeh and C. Shahabi, "The spatial skyline queries," in *Proc. 32nd Int. Conf. Very Large Data Bases*, 2006, pp. 751-762.

[6] Dijkstra, E. W., " A note on two problems in connexion with graphs," in *Numerische Mathematik*, 1959, pp. 1:269-271.

[7] Hart, P. E., Nilsson, N. J. and Raphael, B., "A formal basis for the heuristic determination of minimum cost Paths," in *Proc. IEEE Transactions on Systems Science and Cybernetics SSC*, 1968, 100-107.

[8] S. Anthony, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf.* Robotics and Automation, 1994, pp. 3310-3317.

[9] Y. Xun, M. Liu, Y. Zhou, R. Wang and W. Zhang., "Ant colony based on cat swarm optimization and application in picking robot path planning," in *Proc. IEEE Int. Conf. Software Engineering and Service Science* (*ICSESS*) 2016, pp. 162-165.

[10] Y. Z. Cong and S. G. Ponnambalam, "Mobile robot path planning using ant colony optimization," in *Proc. IEEE Int. Conf. Advanced Intelligent Mechatronics*, July 2009, pp. 851-856.

[11] R. Rashid, N. Perumal, I. Elamvazuthi, M. K. Tageldeen, M.K.A. Ahamed Khan and S. Parasuraman, "Ant colony based on cat swarm optimization and application in picking robot path planning," in *Proc. IEEE Int. Conf. Robotics and Manufacturing Automation (ROMA),* 2016, pp. 1-6.

[12] K.Liu and M. Zhang, "Path planning based on simulated annealing ant colony

algorithm," in *Proc. IEEE Int. Conf. Symposium on Computational Intelligence and Design (ISCID)* 2016, pp. 461-466.

[13] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. IEEE 17th Int. Conf. Data Eng.* (*ICDE*), Apr. 2001, pp. 421-430.

[14] Shekelyan, M., Jossé, G. and Schubert , M., Kriegel, "Linear path skylines in multicriteria networks," in *Proc. IEEE Int. Conf. Data Eng.* (*ICDE*), 2015, pp. 459-470.

[15] K. Jeddisaravi, R. J. Alitappeh, and F. G. Guimarães "Multi-objective mobile robot path planning based on A* search," in *Proc. IEEE Int. Conf. Computer and Knowledge Engineering (ICCKE),* 2016, pp. 7-12.

[16] X. Chen, H. Cai, T. Wolf, "Multi-criteria routing in networks with path choices," In *Proc. IEEE Int. Conf Network Protocols (ICNP)*, 2015, pp. 334-344.

[17] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in *Proc. ACM Special Interest Group Manage. Data (SIGMOD)*, 1990, pp. 322-331.

[18] R. Zhong, G. Li, K.-L. Tan, and L. Zhou, "G-tree: An efficient index for knn search on road networks," in *Proc. 22nd Int. Conf. Inf. Knowl. Manage*., 2013, pp. 39–48.

[19] R. Zhong, G. Li, K.-L. Tan, L. Zhou, and Z. Gong "G-tree: An efficient index for knn search on road networks," in *Proc. IEEE Int. Conf. Data Eng.* (*ICDE*), Aug. 2015, pp. 2175-2189.

[20] Y.-C. Chen and C. Lee "Skyline path queries with aggregate attributes," in *Proc. IEEE Access*, Aug. 2016, pp. 4690-4706.

[21] Shi, J., Wu, D., Mamoulis, N.: "Textually relevant spatial skylines," in *Proc IEEE Trans. Knowl. Data Eng*., 2016, 224–237.