國立臺灣大學工學院工業工程學研究所

碩士論文

Institute of Industrial Engineering

College of Engineering

National Taiwan University

Master’s Thesis

考量時間與轉乘次數之捷運搭乘路線規劃

Metro Route Planning Considering Travel Time and Number of Transfers

洪詩媛

Shih-Yuan Hung

指導教授: 洪一薰博士

Advisor: I-Hsuan Hong, Ph.D.

中華民國 114 年 7 月

July 2025

doi:10.6342/NTU202504286

目次

doi:10.6342/NTU202504286

圖次

doi:10.6342/NTU202504286

doi:10.6342/NTU202504286

表次

doi:10.6342/NTU202504286

誌謝

謝謝老師支持我的個人的安排，讓我的碩士生涯增添不少樂趣，也謝謝實驗室同學在我學業上有困難時提供幫助，希望大家未來都能順順利利。

doi:10.6342/NTU202504286

中文摘要

大眾運輸系統是現今社會不可或缺的一部分，捷運更是其中非常重要的一部分，隨著時間與技術的進步，捷運的規模越來越大，路網圖越來越複雜，相同起訖站間可能的路徑也越來越多，因此也面臨路徑推薦系統的升級，以提升旅客搭乘體驗。本研究針對臺北捷運系統，結合實際路網與班次時刻資料，建立一個考慮轉乘次數與搭乘時間的動態路徑規劃模型。透過 Dijkstra's Algorithm (Dijkstra, 1959) 與 Yen's Algorithm (Yen, 1971) 計算出多條時間最短且轉乘數最少的路徑，提供乘客多元選擇以改善搭乘體驗。

關鍵字: 路徑規劃、最短路徑、捷運、Dijkstra's Algorithm、Yen's Algorithm

doi:10.6342/NTU202504286

英文摘要

Public transportation systems are an indispensable part of modern society, with metro systems playing an especially important role. As time and technology advance, metro networks have become increasingly extensive and complex, resulting in a growing number of possible routes between the same origin and destination stations. This complexity presents a challenge for route recommendation systems, which must be upgraded to enhance the passenger experience. This study focuses on the Taipei Metro system, integrating the actual network and timetable data to develop a dynamic route planning model that considers both the number of transfers and travel time. By applying Dijkstra's Algorithm (Dijkstra, 1959) and Yen's Algorithm (Yen, 1971), the model identifies multiple shortest-time routes with minimal transfers, offering passengers a variety of options to improve their overall commuting experience.

Key words: Route Planning, Shortest Path, Metro, Dijkstra's Algorithm, Yen's Algorithm

doi:10.6342/NTU202504286

第一章 緒論

在現代都市中，大眾運輸系統扮演著不可或缺的角色，而其中以捷運系統尤為重要。捷運因其高效率、準時與環保等特性，已成為城市居民日常通勤與出行的首選方式之一。隨著城市人口密度上升與交通需求日益複雜，各地捷運系統的規模持續擴展，路網圖也愈加錯綜複雜。相同起訖站之間可能存在多條可行路徑，如何從眾多選項中提供最適合乘客的路徑，成為捷運營運單位與智慧交通系統設計者關注的焦點。

過去捷運路徑推薦多以靜態最短路徑演算法為主，未能充分考慮轉乘等待時間、不同列車班距、夜間營運限制等動態因素，使得推薦結果與實際搭乘經驗存在落差。此外，當面對非尖峰時段、夜間末班車或區間車限制時，靜態路徑可能導致乘客錯過列車、無法順利抵達目的地，降低整體搭乘體驗。因此，開發一套能反映實際班次與時間約束的動態路徑規劃模型，對於提升路徑推薦實用性與可靠性具有重大意義。本研究將深入探討最短路徑問題與設施規劃這兩個關鍵議題，這兩者在最佳資源使用和提升整體系統效能方面扮演著重要角色。最短路徑問題的研究，目的在尋找在複雜網絡結構中，從一個節點到另一個節點之間的最低成本或最短時間路徑。這一問題在物流配送、交通管理和網路等領域中格外重要，選擇合適的路徑能顯著降低運營成本並提高服務的快速性。本研究聚焦在大眾運輸之最短路徑問題，透過 Dijkstra's Algorithm (Dijkstra, 1959) 計算起訖站之間的最小成本路徑，最小成本包括最少轉乘次數與最短時間，同時結合時刻表動態計算不同搭乘時間的最短時間路徑。為提供旅客更好的搭乘體驗，進一步使用 Yen's Algorithm (Yen, 1971) 計算多個路徑，提供更多元的路徑選擇。本研究使用臺北大眾捷運股份有限公司之路網與時刻表作為實驗資料。

本研究分為五個章節，第二章結為相關文獻回顧；第三章節為問題描述，說明最短路徑問題（Shortest Path Problem, SPP）與相關台北捷運資料；第四章節為研究方法，說明路網圖成本與演算法，並展示研究結果；第五章節為結論。

doi:10.6342/NTU202504286

第二章 文獻回顧

捷運系統在現代城市運輸中扮演著至關重要的角色，影響城市發展、交通效率與環境可持續性。有效的捷運路線規劃不僅能夠縮短通勤時間，還能減少交通擁堵與碳排放 (Bertolini et al., 2005)。隨著都市化的加速，城市對於捷運網絡的需求日益增加，因此研究最短路徑演算法在捷運規劃中的應用顯得尤為關鍵。

捷運路線規劃可追溯至傳統的圖論 (Graph Theory)，透過最短路徑演算法尋找最佳捷運路線。隨著計算機技術的進步，各種最短路徑與 K 短路徑演算法（如 Dijkstra's Algorithm、Yen's Algorithm）被廣泛應用於捷運系統的路線規劃與優化 (Ahuja et al., 1993; Zhan & Noon, 1998) 。這些演算法能夠考慮不同因素，如時間、距離、轉乘次數與乘客流量，以提供更符合實際需求的捷運路線規劃方案。

Dijkstra's Algorithm 由 Edsger W. Dijkstra 於 1959 年發表，是一種用於解決最短路徑問題的圖演算法。該演算法透過貪婪策略，逐步選取當前已知最短的節點進行更新，直至找到從起點到所有其他節點的最短路徑。由於其計算效率高，Dijkstra's Algorithm 被廣泛應用於各種領域，如交通網路規劃、電信網路設計與導航系統。

在大眾運輸系統中，Dijkstra's Algorithm 常用於捷運與公車路線的規劃，幫助乘客找到最快抵達目的地的路徑。Bozyigit et al. (2017) 與 Hadi et al. (2025) 探討了 Dijkstra's Algorithm 在公共交通網絡中的應用，並發現其在處理固定時刻表的路徑推薦問題上表現良好。此外，Farhan et al. (2019) 與 Ulum et al. (2025) 利用 Dijkstra's Algorithm 分析高速公路網路的最短路徑，進一步驗證了其在大規模交通網路中的可行性。然而，Dijkstra's Algorithm 僅能求解單一路徑，無法提供多條可行的替代路徑。因此，在多條捷運線路選擇與臨時路線變更的應用場景下，Dijkstra's Algorithm 的限制促使研究者尋求其他方法，如 Yen's Algorithm。

Yen's Algorithm 是一種求解 K 條最短路徑問題（K-Shortest Paths Problem, KSP）的演算法，能夠找出從起點到終點的前 K 條最短路徑。該演算法基於 Dijkstra's Algorithm，每次計算最短路徑後，通過調整部分路徑來尋找次短路徑，

doi:10.6342/NTU202504286

並重複該過程直至找到 K 條不同的路徑。Yen's Algorithm 廣泛應用於交通規劃。例如，Zhang et al. (2019) 與 Auer et al. (2020) 將 Yen's Algorithm 應用於電動車相關規劃。Zhang et al. 使用 Yen's Algorithm 規劃電動車充電樁位置，考慮電力系統成本與交通壅塞，規畫出能最小化成本的規畫路徑。Auer et al. 則考慮地形變化、電池壽命與充放電循環等因素，規劃電動車的最佳型行車路徑。Yu et al. (2024) 提出一個針對公車路網的同步最佳化模型，能夠有效降低旅客等候時間與公車操作成本，其中即使用 Yen's Algorithm 計算同一對起訖站的多種可能路徑。

在捷運系統的路徑推薦應用中，Yen's 演算法提供了比 Dijkstra's Algorithm 更具靈活性的選擇，尤其在遇到擁擠或突發狀況時，乘客可根據不同條件選擇適合的替代路徑。若更進一步考慮各節點的時間限制，即每個節點有規定可通過時間，則 SPP 擴展為時間限制最短路徑問題 (Time-Constrained Shortest Path Problem , TCSPP)，增加時間限制使複雜度上升。Idri et al. 針對多模式運輸網提出一種基於目標導向與時間依賴特性的最短路徑演算法，透過虛擬路徑與歐幾里得距離作為啟發式限制搜尋空間，以提升演算法效率與目標性。Guo et al. (2024) 提出一種改良 Dijkstra Algorithm 與混合整數規劃模型，以處理多模式運輸網中時間不確定性下的穩健最短路徑問題，實驗顯示其在效率與解的穩定性上皆具實用性。

Dijkstra's Algorithm 與 Yen's Algorithm 在捷運路徑推薦領域均具有重要應用。Dijkstra's Algorithm 適用於靜態最短路徑計算，而 Yen's Algorithm 則適用於提供多條可行路徑，提升乘客的選擇彈性。隨著智慧交通技術的發展，未來的研究方向可能包括將這些演算法與即時資料整合，以進一步優化捷運路徑推薦系統。

doi:10.6342/NTU202504286

第三章 問題描述

隨著現代城市化進程的加速，全球範圍內的交通運輸需求日益增長。城市中的交通流量管理、道路規劃以及公共交通系統的有效運行變得尤為重要。在這些情況下，最佳化交通資源的配置、提升交通運輸效率，並減少乘客的出行時間成為了一個待解決的問題。最短路徑問題（Shortest Path Problem, SPP）作為工業工程中的經典問題，能夠在這些應用場景中發揮關鍵作用。

3.1 SPP 問題描述

最短路徑問題是一種在圖論 (Graph Theory)中非常重要的問題，主要目的是在一個有向或無向圖中，尋找從一個起始節點 (源點) 到一個終止節點 (目標點) 的最低成本或最短距離的路徑。在最短路徑問題中，網路圖由一組節點和一組邊組成，每一條邊都有一個與之對應的成本或成本。目標是找到一條從源點到目標點的路徑，使得這條路徑上所有邊的成本總和最小，簡單網路示意圖如圖 1，總共有六個節點，邊上數字為成本，如 A 點到 B 點的成本為 3。

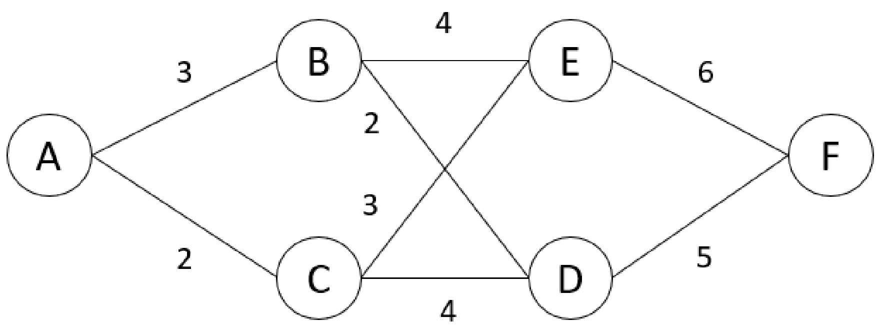


圖 1 網路示意圖

一般最短路徑問題中，計算成本只考慮兩節點間的成本，但隨著問題複雜化，需進一步考慮節點的可通過時段，若在非可行時段經過該節點，節點將有額外時間成本。當節點有此時段限制時，該問題即為 TCSPP，可應用於工廠自動運輸機器規劃與大眾運輸路徑規劃。圖 2 為具有時段限制之網路示意圖，中括弧內數字代表可通過該節點之起始時間與結束時間，舉例來說，可通過節點 A 之時段為時間單位 0 至 2 與時間單位 4 至 6。若從時間單位 0 時從 A 通至 B，因 AB 間成本為

doi:10.6342/NTU202504286

3，抵達 B 為時間單位 3 時，但此時無法通過 B，須等到時間單位 5 時才能通過 B，AB 間成本即增加 2 單位。

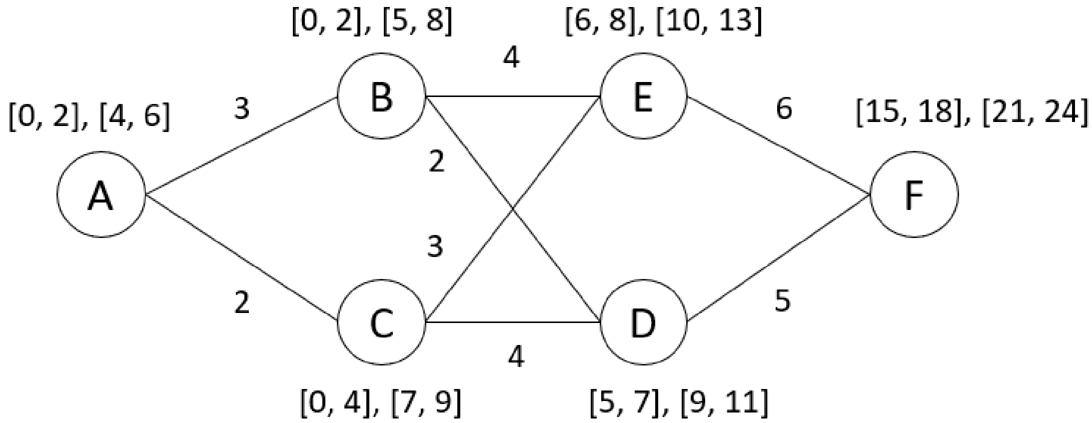


圖 2 時段限制之網路示意圖

針對不同問題與資料，可以製作不同的網路圖與其成本，並將其規劃為一最短路徑問題，進而使用相關演算法求解。交通運輸之路徑規劃即是一個廣為人知的最短路徑問題之應用。

如今大眾運輸設施成為眾多民眾日常交通很重要的一部分，不論是公車、捷運或輕軌，都是目前社會不可或缺的設施。捷運路徑規劃不只能夠提供旅客推薦的乘車路線，改善成乘車體驗，對公司而言也是很重要的資訊，若是能夠知道旅客在特定起訖站間所選擇的路徑，便能夠協助公司進行人流估計與班表設計，提高營運效率，也可以進一步分析旅客行為，藉此提升服務質量，並針對旅客偏好推出相關優惠以提升旅客忠誠度。

3.2 臺北捷運資料與路網圖成本說明

臺北大眾捷運股份有限公司 (北捷) ，為臺灣首間提供大眾捷運系統服務之企業，以開放創新之觀念與服務，支持城市居住、工作、休閒及運輸等機能發展為願景 (臺北大眾捷運股份有限公司, n.d.-c) ，其路網範圍涵蓋臺北市與新北市，規模為全台最大。目前北捷路網包含文湖線、淡水信義線、松山新店線、中和新蘆線及板南線等 5 條，共 117 個車站(西門站、中正紀念堂站、古亭站及東門站等 4 個轉

doi:10.6342/NTU202504286

乘站於不同路線共用站體計為 1 站，其餘轉乘站計為 2 站) (臺北大眾捷運股份有限公司, n.d.) ，總路網長度為 131.1 公里。目前臺北市與新北市 (雙北市) 都會區內，除北捷所營運之 5 條路線外，尚有一條由新北市政府捷運工程局 (新北捷) 負責之環狀線，營運中車站共 14 站，總路線長度為 15.4 公里 (維基媒體專案貢獻者, 2025) ，圖 3 為北捷與新北捷營運中之路網圖。

北捷所開發之台北捷運 GO 應用程式，提供旅客查詢特定起訖站間之推薦搭乘路線，目前北捷使用方法為線下查詢 (off-line)，事先將所有車站間之可能路線計算出來，再篩選出最佳路徑，篩選條件為最短旅程時間 (不包含轉車候車時間) 或最少轉乘次數。目前僅提供單一路徑，為了提供旅客更多元且即時的路徑選擇，本研究使用 Yen's Algorithm 即時 (real-time) 計算多個不同的路徑選擇。

doi:10.6342/NTU202504286

圖 3 台北捷運路網圖 (臺北大眾捷運股份有限公司, n.d.)

最短路徑問題中，節點間成本 (成本) 之計算方法相當重要，圖 4 為現有台北捷運 GO 應用程式之時間成本計算方法，節點為各車站，節點間成本為兩站間的行駛時間，當行駛到某一車站時，捷運將會停留一段時間，此時間稱為停留時間，為旅客上下車之時間，若是需要轉乘，旅客須從原本所搭乘路線之月台步行至要轉乘

doi:10.6342/NTU202504286

路線月台，此步行時間稱為轉乘時間。需要注意的是，當旅客抵達轉乘站時，不會計算該站之停留時間，只計算轉乘時間。圖 4 中從起站 (Origin, O) 到訖站 (Destination, D) 之時間成本為行駛時間 + 停留時間 + 轉乘時間。本研究延伸此問題，根據北捷之需求，為了提升旅客搭車體驗，需要能夠即時計算多種路線，將候車時間納入計算，提供最短旅程時間與最少轉乘次數路徑。

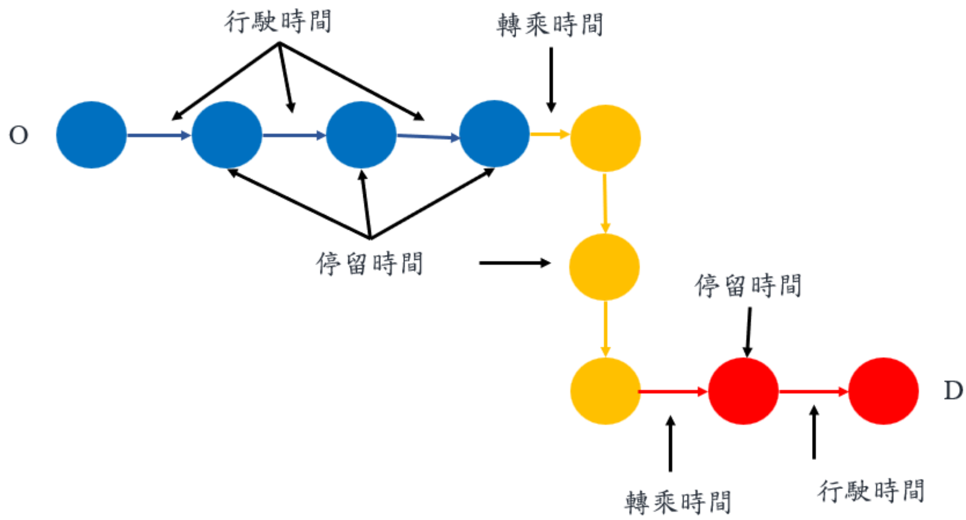


圖 4 原有時間成本計算示意圖

本研究使用之路網圖包含文湖線、淡水信義線、松山新店線、中和新蘆線、板南線及環狀線，其中文湖線與環狀線為中運量路線，其餘為高運量路線。所有車站都有相對應之代碼，一班車站有一個代碼，轉乘站則有兩個代碼，分別代表不同路線，車站代碼如表 1 (臺北大眾捷運股份有限公司, n.d.)。代碼開頭英文字母代表該車站所屬路線，R、BL、O、G、BR 與 Y 分別代表淡水信義線、板南線、中和新蘆線、松山新店線、文湖線與環狀線，數字部分為車站在所對應路線之代號，本研究皆使用該代碼表示路網中的車站節點，需要注意的是，地圖上轉乘站指顯示為一站，但本研究使用之路網資料將轉乘站視為至少兩個節點。

doi:10.6342/NTU202504286

表 1 車站代碼 (臺北大眾捷運股份有限公司, n.d.)

車站名稱	代碼	車站名稱	代碼	車站名稱	代碼
象山	R02	善導寺	BL13	景美	G05
台北 101/世貿	R03	忠孝新生	BL14/O07	萬隆	G06
信義安和	R04	忠孝復興	BL15/BR10	公館	G07
大安	R05/BR09	忠孝敦化	BL16	台電大樓	G08
大安森林公園	R06	國父紀念館	BL17	小南門	G11
東門	R07/O06	市政府	BL18	北門	G13
中正紀念堂	R08/G10	永春	BL19	南京復興	G16/BR11
台大醫院	R09	後山埤	BL20	台北小巨蛋	G17
台北車站	R10/BL12	昆陽	BL21	南京三民	G18
中山	R11/G14	南港	BL22	松山	G19
雙連	R12	南港展覽館	BL23/BR24	動物園	BR01
民權西路	R13/O11	南勢角	O01	木柵	BR02
圓山	R14	景安	O02/Y11	萬芳社區	BR03
劍潭	R15	永安市場	O03	萬芳醫院	BR04
士林	R16	頂溪	O04	辛亥	BR05
芝山	R17	古亭	O05/G09	麟光	BR06
明德	R18	松江南京	O08/G15	六張犁	BR07
石牌	R19	行天宮	O09	科技大樓	BR08
唭哩岸	R20	中山國小	O10	中山國中	BR12
奇岩	R21	大橋頭	O12	松山機場	BR13
北投	R22	三重國小	O50	大直	BR14
新北投	R22A	三和國中	O51	劍南路	BR15

doi:10.6342/NTU202504286

復興崗	R23	徐匯中學	O52	西湖	BR16
忠義	R24	三民高中	O53	港墘	BR17
關渡	R25	蘆洲	O54	文德	BR18
竹圍	R26	台北橋	O13	內湖	BR19
紅樹林	R27	菜寮	O14	大湖公園	BR20
淡水	R28	三重	O15	湖州	BBR21
頂埔	BL01	先嗇宮	O16	東湖	BR22
永寧	BL02	頭前庄	O17/Y18	南港軟體園區	BR23
土城	BL03	新莊	O18	十四張	Y08
海山	BL04	輔大	O19	秀朗橋	Y09
亞東醫院	BL05	丹鳳	O20	景平	Y10
府中	BL06	迴龍	O21	中和	Y12
板橋	BL07/Y16	新店	G01	橋和	Y13
新埔/新埔民生	BL08/Y17	新店區公所	G02	中原	Y14
江子翠	BL09	七張	G03	板新	Y15
龍山寺	BL10	小碧潭	G03A	幸福	Y19
西門	BL11/G12	大坪林	G04/Y07	新北產業園區	Y20

目前營運中車站分為一般車站、一般轉乘車站與平行轉乘車站，一般車站指非轉乘車站，一般轉乘站指車站內部相同路線之月台在同一樓層，平行轉乘站則指車站內部相同路線之月台在不同樓層，旅客無須上下樓層即可轉乘其他路線，即稱為平行轉乘。目前路網圖中，只有 4 個轉乘站為平行轉乘站，分別為西門站、中正紀念堂站、古亭站及東門站。圖 5 為忠孝新生站 (一般轉乘站) 之車站剖面圖，其中兩條路線 (中和新蘆線與板南線) 分別在不同月台。圖 6 為西門站 (平行轉乘站) 之車站剖面圖，其中兩條路線 (板南線與松山新店線) 相同方向之月台在同一樓層。

doi:10.6342/NTU202504286

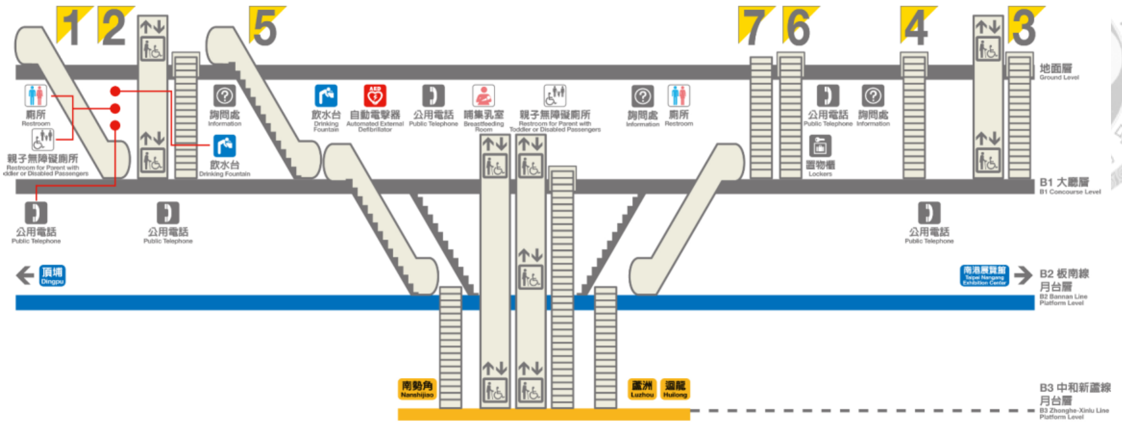


圖 5 忠孝新生站車站剖面圖 (臺北大眾捷運股份有限公司, n.d.-a)

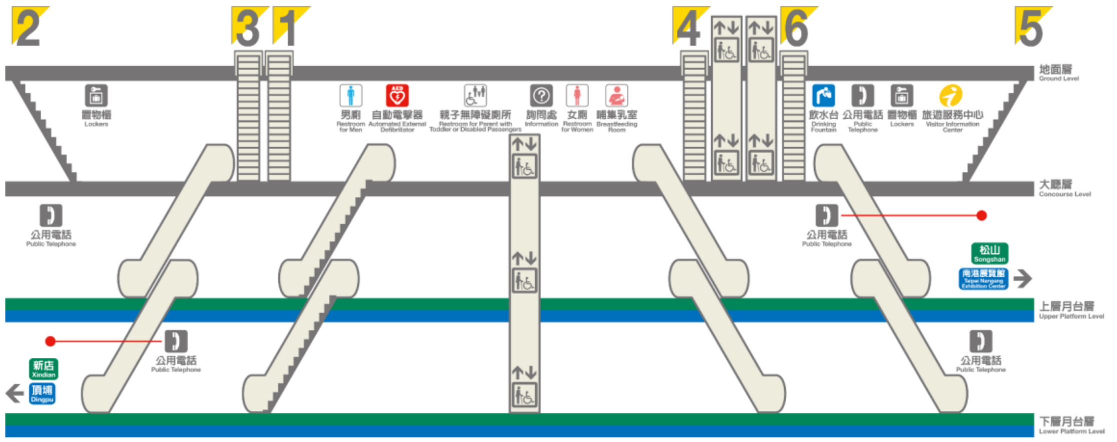


圖 6 西門站車站剖面圖 (臺北大眾捷運股份有限公司, n.d.-b)

高運量路線 (淡水信義線、松山新店線、中和新蘆線、板南線) 上每個車站皆有其時刻表，而中運量路線 (文湖線與環狀線) 中，環狀線有時刻表，而文湖線只有第一班與末班發車時間與其特定時段之班距。北捷營運時間為早上 6 點至晚上 12 點，所有車站第一班車皆為早上 6 點發車，但末班車指晚上 12 點從各路線終點站發車，因此距離終點站較遠的車站末班車時間較晚。每個車站之時刻表有幾個影響要素，分別為平日與假日、尖峰與離峰時段、行車方向與區間車行駛路段。平日 (週一至週五) 與假日 (週六、週日及國定假日) 的班距不同，平日較密集且大部分時段皆有區間車行駛，假日班距較長，且較少時段有行駛區間車。平日尖峰與離峰時段的班距不同，尖峰時段為早上 7 點至 9 點與晚上 5 點至 7 點半，其餘晚上 11 點之前時段為離峰時段，假日則分為早上 6 點至早上 9 點、早上 9 點至晚上 11 點

doi:10.6342/NTU202504286

與晚上 11 點之後三時段，僅早上 9 點至晚上 11 點有區間車行駛，各路線平日班距如表 2，假日班距如表 3。各路線皆有兩個行車方向，路線上各站不同行車方向時刻表不同，行車方向有其代號，1 表示往北行駛，2 表示往南行駛，各路線行駛方向與代號如表 4。部分路線有區間車，即只有行駛該路線中一部份路段，有區間車之路段班距較密，表 4 中有*符號之行車方向為區間車路段。中和新蘆線較其他線不同，有三個終點站，而其中有一路段重疊，如往迴龍與往蘆洲方向之列車，皆會行駛南勢角 (O01) 至大橋頭站 (O12)，此段為重疊區間，重疊區間班距較密集。請注意，2024 年 4 月 3 號，因花蓮地震，導致環狀線部分路段停駛，目前營運路段分為北段與南段，北段為新北產業園區站 (Y20)至板橋站 (Y16)，南段為中和站 (Y12) 至大坪林站 (Y07)，南北段尖峰與離峰班距不同。

表 2 各路線平日班距 (臺北大眾捷運股份有限公司, n.d.-d)

路線	尖峰時段	離峰時段	23:00 之後
淡水信義線	約 6 分	約 8~10 分	約 12 分
	區間路段約 3 分	區間路段約 4~5 分	無區間路段
松山新店線	約 4~6 分	約 6~8 分	約 12 分
	區間路段約 3 分		無區間路段
中和新蘆線	約 6 分	約 9~10 分	約 12 分
	重疊區間約 3 分	重疊區間約 4.5~5 分	重疊區間約 6 分
板南線	約 6 分	約 8~10 分	約 8~12 分
	區間路段約 3 分	區間路段約 4~5 分	無區間路段
文湖線	約 2~4 分	約 4~10 分	約 12 分
環狀線	北段 12 分	北段 15 分	-
	南段 7~8 分	南段 10 分	

doi:10.6342/NTU202504286

表 3 各路線假日班距 (臺北大眾捷運股份有限公司, n.d.-d)

路線	06:00 ~ 09:00	09:00 ~ 23:00	23:00 之後
淡水信義線	約 8~10 分	約 8~10 分	約 12 分
	無區間路段	區間路段約 4~5 分	無區間路段
松山新店線	約 8~10 分	約 6~8 分	約 12 分
	無區間路段		無區間路段
中和新蘆線	約 9~10 分	約 9~10 分	約 12 分
	重疊區間約 4.5~5 分	重疊區間約 4.5~5 分	重疊區間約 6 分
板南線	約 8~10 分	約 8~10 分	約 8~12 分
	無區間路段	區間路段約 4~5 分	無區間路段
文湖線	約 4~10 分	約 4~10 分	約 12 分
環狀線	北段 12 分	北段 15 分	-
	南段 7~8 分	南段 10 分	

表 4 各路線行車方向代號

路線	方向 1	方向 2
淡水信義線	往淡水	往象山
	往新北投*	往大安*
松山新店線	往松山	往新店
		往台電大樓*
中和新蘆線	往蘆洲	往南勢角
	往迴龍	
板南線	往南港展覽館	往頂埔
	往昆陽*	往亞東醫院*
文湖線	往南港展覽館	往動物園
環狀線	往新北產業園區	往大坪林

doi:10.6342/NTU202504286

本研究將捷運路網圖建構為一雙向圖，一般車站在路網上視為一個節點，一般轉乘車站視為兩個節點，分別代表不同路線，如忠孝新生站為板南線與中和新蘆線之交點，將其視為節點 BL14 (板南線) 與 O07 (中和新蘆線)，而平行轉乘站因其根據原本乘車方向與轉乘後乘車方向，會有不同的轉乘時間，因此增設四個虛擬節點 (Dummy Node) 代表平行轉乘站中的四個月台，表 5 為四個平行轉乘站各月台與虛擬節點所代表的行車方向，加上轉乘站原本的兩個節點，平行轉乘站總共有六個節點。行經平行轉乘站時，如果沒有要轉車，經過的節點為該車站對應路線原本的節點，只有需要轉車時，才會經過虛擬節點，如從大安森林公園站 (R06) 到忠孝新生站 (BL14/O07) 的路徑為大安森林公園 (R06) ➔ 東門 (O06_1) ➔ 忠孝新生 (O07)，其中➔代表從一站前往下一站。圖 7 為虛擬節點示意圖，以東門站為例，其中空心節點為虛擬節點，紅色直線與曲線用於連接兩個淡水信義線車站，橘色值線與曲線用於連接兩個中和新蘆線車站。

表 5 平行轉乘站月台與虛擬節點

平行轉乘站	月台	行車方向	虛擬節點
西門 (BL11/G12)	1	往南港展覽館	BL11_1
	2	往松山	G12_2
	3	往頂埔	BL11_3
	4	往新店	G12_4
中正紀念堂 (G10/R08)	1	往淡水	R08_1
	2	往松山	G10_2
	3	往象山	R08_3
	4	往新店	G10_4

doi:10.6342/NTU202504286

古亭 (G09/O05)	1	往松山	G09_1
	2	往蘆洲/迴龍	O05_2
	3	往新店	G09_3
	4	往南勢角	O05_4
東門 (R07/O06)	1	往蘆洲/迴龍	O06_1
	2	往淡水	R07_2
	3	往南勢角	O06_3
	4	往象山	R07_4

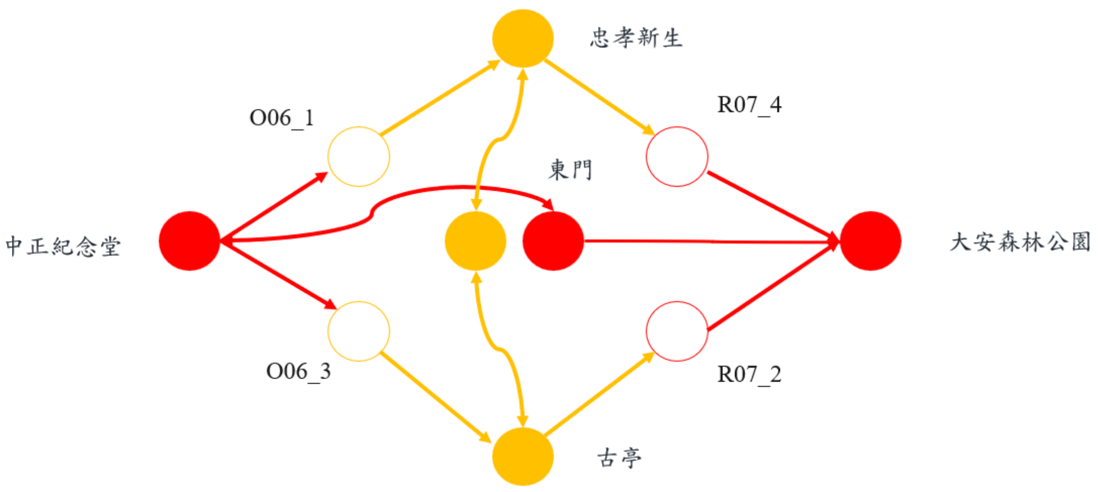


圖 7 東門站節點示意圖

圖 8 為一般轉乘站轉乘時所經過的節點之示意圖，以從板南線至忠孝新生站轉乘至中和新蘆線為例，忠孝新生站無虛擬節點，因此只有板南線 (藍線) 與中和新蘆線 (橘色) 的兩個節點，轉乘即在這兩個節點間移動。圖 9 為平行轉乘站轉乘時所經過的節點之示意圖，以從淡水信義線至東門站轉乘至中和新蘆線為例，假設從中正紀念堂站往象山方向要轉乘至中和新蘆線往南勢角方向，此時要從東門站轉乘，平行轉乘站的轉乘由虛擬節點負責，並且是要轉至往南勢角方向，根據表 5，需要經過虛擬節點 O06_3，方可完成平行轉乘。

doi:10.6342/NTU202504286

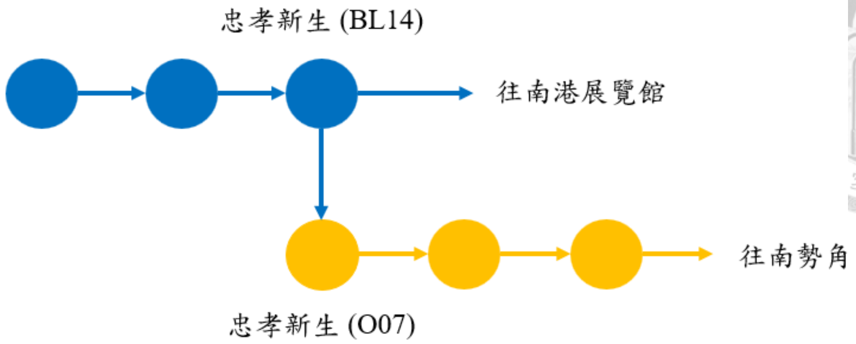


圖 8 一般轉乘站轉乘節點之示意圖

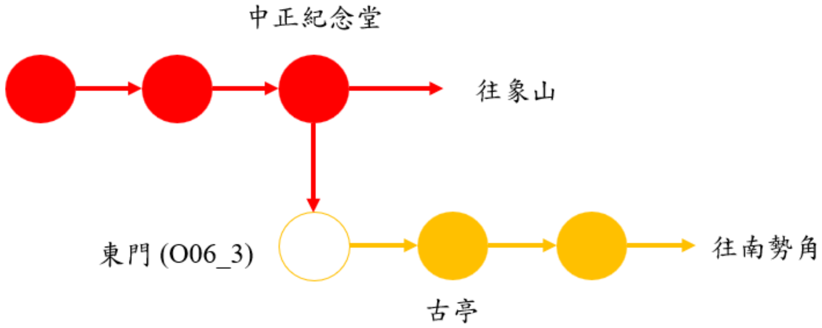


圖 9 平行轉乘站轉乘節點之示意圖

本研究所計算之推薦路線分為最短旅程時間 (包含候車時間) 與最少轉乘次數路徑。當考慮候車時間後，因不同路線之尖峰與離峰時段的班距不同，同樣的起訖站配對 (Origin and Destination Pair, OD Pair) 在一天中不同時段之最短旅程時間路徑將會不同，也使問題從原本的 SPP 變為 TCSPP，計算時間成本方法也有所變化。每個節點 (車站) 所對應之資料，包含時刻表 (及各班車之道站與離站時間)與到附近車站 (Neighbor) 之行駛時間，轉乘站則多一個轉乘時間。圖 10 為本研究計算任意 OD Pair 間時間成本之方法示意圖，詳細計算參見 3.2.2。

doi:10.6342/NTU202504286

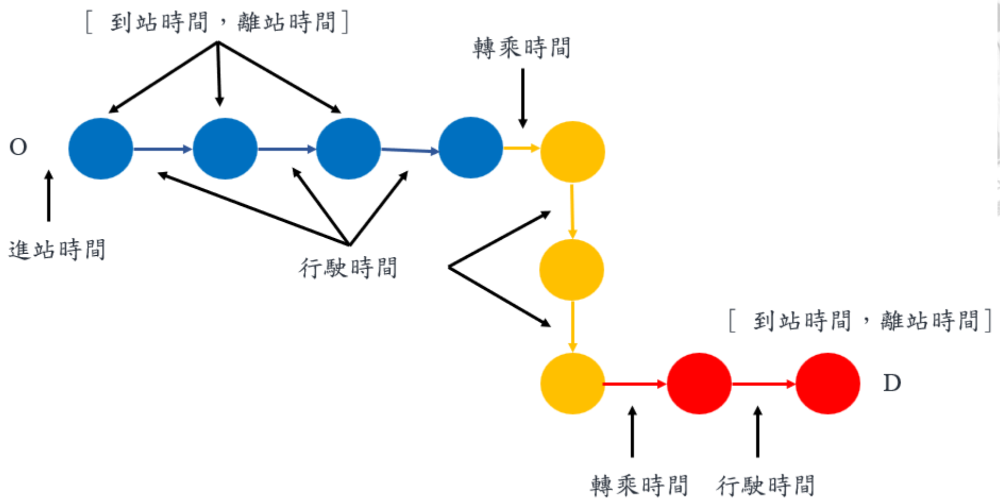


圖 10 本研究時間成本計算示意圖

最少轉乘次數路徑為一般的 SPP，唯一不同的部分為路網圖的成本，成本不同於最短旅程時間路徑，邊上無成本，僅節點 (車站) 上有成本。轉乘站成本為 1，其餘車站成本為 0。圖 11 為轉乘成本之示意圖。

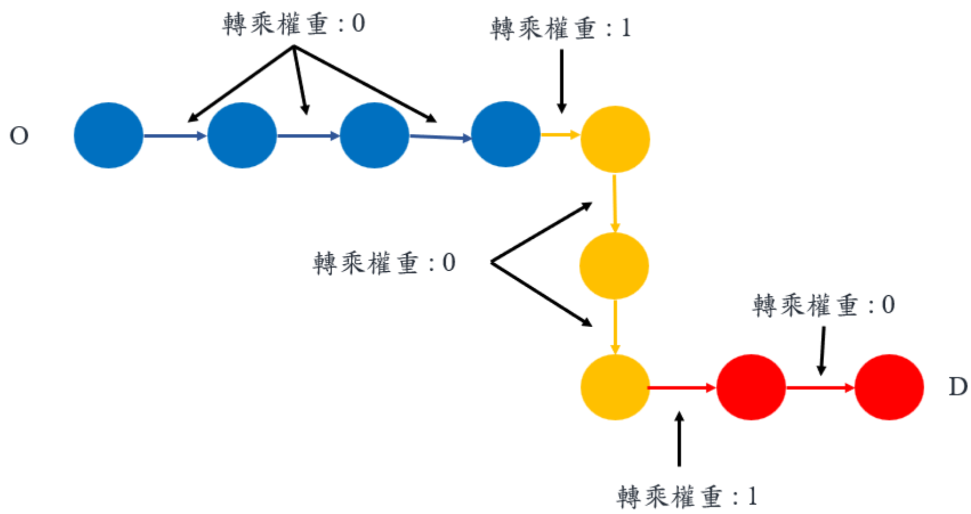


圖 11 轉乘成本示意圖

doi:10.6342/NTU202504286

第四章 研究方法

Dijkstra's Algorithm 被廣為使用於 SPP，可根據不同路網圖與成本來求得最短路徑，但只能求出單一路徑，缺乏替代方案的彈性，而 Yen's Algorithm 能補足此限制，提供多條可行路徑選擇，有助於應對乘客偏好與即時配置變化。

4.1 演算法

本研究將最短旅程時間路徑問題與最少轉乘次數問題分別視為 TCSPP 與 SPP，並使用 Dijkstra's Algorithm 求解。最短旅程時間問題中，使捷運路線的集合為 $L=\{R,BL,O,G,BR,Y\}$，R、BL、O、G、BR 與 Y 分別代表淡水信義線、板南線、中和新蘆線、松山新店線、文湖線與環狀線，路線l上的車站集合為S_l，n表示車站 (如忠孝敦化站為 BL16)，$line(n)$表示車站n所屬的路線 (如$line(BL16)=BL$)，dir 表示行車方向 (1 或 2)，$dest$ 表示該列車的終點，不同訖站能搭乘的列車會不同 (如要去象山站不能搭往大安的區間車)，$A^j_{n,dir,dest}$ 與 $D^j_{n,dir,dest}$ 表示車站n往dir 方向$dest$站的第j班次的抵達時間與離站時間，假設車站n往dir 方向$dest$ 站一天總共有 $J_{n,dir,dest}$ 班車，則$T_{n,dir,dest}=\{(A^j_{n,dir,dest},D^j_{n,dir,dest})|j\in J_{n,dir,dest}\}$ 表示車站n行車方向為dir往$dest$站的完整時刻表，使$T^{tvl}_{n,n'}$為從車站n行駛至車站n'的行駛時間，如果車站n與車站n'互為不同路線上的同一車站，則$T^{tsf}_{n,n'}$表示從車站n轉乘至車站n'的轉乘時間。

假設旅客從O (起站，Origin) 出發，目的地為D (訖站，Destination)，則對照函數一步驟判斷 dir 與 $dest$，其中 $Num(n)$ 表示取得該車站編號 (如$Num(BL16)=16$)，$line(n)$如上述，同樣表示車站n所屬的路線。若查詢時間 (進站時間) 為t，需再加上從進站閘門步行至月台時間w，抵達月台候車或乘車的時間改為$t+w$，本研究將w設為 30 秒固定值。根據函數二步驟，判斷搭乘該站對應方向的第j班車，其中CN與n分別代表現在所在車站與即將經過的下一站，S^{tsf} 表示為所有轉乘站的集合，計算後得到$A^j_{n,dir,dest}$與$D^j_{n,dir,dest}$，表示旅客將在$D^j_{n,dir,dest}$時離開該站，之後每經過一站，需再加上到下一站的行駛時間，時間變為$t+T^{tvl}_{CN,n}$，若即將在下一站轉乘，則需額外加上轉乘時間，時間則變為$t+T^{tvl}_{CN,n}+T^{tsf}_{CN,n}$。

doi:10.6342/NTU202504286

函數一：判斷 dir 與 $dest$

輸入: 起站 O 與訖站 D

輸出: dir 與 $dest$

1. 給定起站 O 與訖站 D ，$i \leftarrow Num(O), i^* \leftarrow Num(D), l \leftarrow line(O)$ ；
2. 如果 $i < i^*$: $dir = 1$，否則 $dir = 2$ ；
3. 如果 $l = R$:
4. 如果 $dir = 1$ 與 $i^* > 22$ ： $dest =$ 淡水，否則 $dest =$ 淡水&北投；
5. 如果 $dir = 2$ 與 $i^* < 5$ ： $dest =$ 象山，否則 $dest =$ 象山&大安；
6. 如果 $l = BL$:
7. 如果 $dir = 1$ 與 $i^* > 21$ ： $dest =$ 南港展覽館；
   否則 $dest =$ 南港展覽館&昆陽；
8. 如果 $dir = 2$ 與 $i^* < 5$ ： $dest =$ 頂埔，否則 $dest =$ 頂埔&亞東醫院；
9. 如果 $l = O$:
10. 如果 $dir = 1$ 與 $i^* \geq 50$ ： $dest =$ 蘆洲；
11. 否則如果 $dir = 1$ 與 $i^* \geq 13$ ： $dest =$ 迴龍；
12. 如果 $dir = 2$ ： $dest =$ 南勢角；
13. 如果 $l = G$:
14. 如果 $dir = 1$ ： $dest =$ 松山；
15. 如果 $dir = 2$ 與 $i^* < 8$ ： $dest =$ 新店，否則 $dest =$ 新店&台電大樓；
16. 如果 $l = BR$:
17. 如果 $dir = 1$ ： $dest =$ 南港展覽館，否則 $dest =$ 動物園；
18. 如果 $l = BR$:
19. 如果 $dir = 1$ ： $dest =$ 新北產業園區，否則 $dest =$ 大坪林

doi:10.6342/NTU202504286

函數二：判斷搭乘班次

輸入：現在所在車站、下一站、O、D、t、w、dir與$dest$

輸出：搭乘班次 $(A^{j}_{n,dir,dest}, D^{j}_{n,dir,dest})$

1. $CN \leftarrow$ 現在所在車站，$n \leftarrow$ 下一站；
2. $dir, dest \leftarrow$ 函數一
3. 如果 $CN = O$： $t \leftarrow t + w$；
4. 如果 $CN \in S^{tsf}$： $t \leftarrow t + T^{tvl}_{CN,n}$；
5. 如果 $CN \notin S^{tsf}$： $t \leftarrow t + T^{tvl}_{CN,n} + T^{tsf}_{CN,n}$；
6. 對於所有 $(A^{j}_{n,dir,dest}, D^{j}_{n,dir,dest}) \in T_{n,dir,dest}$：
7. 　　如果 $A^{j}_{n,dir,dest} \le t \le D^{j}_{n,dir,dest}$：搭乘第 j 班次

在本研究所使用路網圖中，文湖線並沒有時刻表，因此不適用函數二，文湖線所擁有的資料只有各站首班車時間、末班車時間、停留時間 t^{wait} 與各時段班距 h，因此每當從其他路線移動到文湖線的車站，因為無法得知確切到站時間，成本將多加一項班距，即假設旅客每次轉乘至文湖線或是起站為文湖線車站時，都須等候 h 時間才會上車，且離站時間為到站時間加上停留時間，即 $A^{j}_{n,dir,dest} = D^{j}_{m,dir,dest} + T^{tsf}_{n,m} + h, m \notin S_{BR}, n \in S_{BR}, j \in J_{n,dir,dest}$，文湖線車站間的抵達時間為 $A^{j}_{m,dir,dest} = D^{j}_{n,dir,dest} + T^{tvl}_{n,m} + h, n, m \in S_{BR}, j \in J_{n,dir,dest}$，文湖線車站間的離站時間為 $D^{j}_{n,dir,dest} = A^{j}_{n,dir,dest} + t^{wait}, n \in S_{BR}, j \in J_{n,dir,dest}$，其中 dir 與 $dest$ 皆來自函數一計算結果。

各站的抵達時間為起站至該站的成本，即 Dijkstra's Algorithm 中的總成本，而演算法目標為尋找最短旅程時間之路徑，即要尋找最早抵達訖站的時間。使用演算法一，其中 N^{n} 為節點 n 的鄰近節點之集合，$V^{visited}$ 為已經過之節點的集合，C^{n} 為所對應節點的成本 (即抵達該站的時間)。經過所有迭代之後，會得到最佳路徑與抵達訖站時間，將訖站抵達時間減掉進站時間，即為該路徑所花費的總時間。至於最少轉乘次數路徑，為一般的 SPP，僅成本有經過調整，使用典型的 Dijkstra's Algorithm 即可求解，如演算法二，其中 $C^{tsf}_{CN,n}$ 為從 CN 到 n 的轉乘次數。

doi:10.6342/NTU202504286

演算法一 TCSPP Dijkstra's Algorithm (修改自 Dijkstra, 1959)

輸入: $G(V,E)$ 、S^{tsf} 、O 、D 、t 與 w

輸出: 最佳路徑

1. 給定路網 $G(V,E)$ 、S^{tsf} 、起站 O 、訖站 D 、t 與 w；
2. 將所有節點成本設為 $C^n=\infty,\forall n$ ，$V^{visited}=\{O\}$ ；
3. 目前節點 $CN\leftarrow O$，目前時間 $t\leftarrow t+w$，目前總成本 $C\leftarrow\infty$ ；
4. $dir,dest\leftarrow$ 函數一；
5. $(A^j_{n,dir,dest},D^j_{n,dir,dest}),j\leftarrow$ 函數二；
6. 當 $CN\neq D$ ，執行以下步驟:
7.   如果 $CN\in V^{visited}$: 返回步驟 7；
8.   對所有鄰近節點 $n\in N^{CN}$:
9. $dir,dest\leftarrow$ 函數一；
10.   如果 CN 不屬於文湖線:
11. $(A^j_{n,dir,dest},D^j_{n,dir,dest}),j\leftarrow$ 函數二；
12. $C^n\leftarrow A^j_{n,dir,dest}$ ；
13.   如果 n 屬於文湖線且 n 為轉乘站:
14. $A^j_{n,dir,dest}=D^j_{CN,dir,dest}+T^{tsf}_{n,CN}+h$ ；
15.   如果 CN 屬於文湖線且 n 屬於文湖線:
16. $D^j_{\mathrm{CN},dir,dest}=\mathrm{C}+t^{wait}$ ；
17. $A^j_{n,dir,dest}=D^j_{CN,dir,dest}+T^{tvl}_{CN,n}+h$ ；
18.   如果 $C^n<C$:
19.   將 n 加到 $V^{visited}$ ，並儲存對應的 C ；
20. $CN\leftarrow$ 當前擁有最小 C^n 的節點；
21. $C\leftarrow C^n$

doi:10.6342/NTU202504286

演算法二 SPP Dijkstra's Algorithm (修改自 Dijkstra, 1959)

輸入: $G(V,E)$ 、O 與 D

輸出: 最佳路徑

1. 給定路網 $G(V,E)$ 、起站 O 與訖站 D ;
2. 將所有節點成本設為 $C^n = \infty, \forall n$ ，$V^{visited} = \{O\}$;
3. 目前節點 $CN \leftarrow O$ ，目前總成本 $C \leftarrow \infty$;
4. 當 $CN \neq D$ ，執行以下步驟:
5. 如果 $CN \in V^{visited}$: 返回步驟 5;
6. 對所有鄰近節點 $n \in N^n$:
7. $C^n = C + C^{tsf}_{CN,n}$;
8. 如果 $C^n < C$:
9. 將 n 加到 $V^{visited}$ ，並儲存對應的 C ;
10. $CN \leftarrow$ 當前擁有最小 C^n 的節點;
11. $C \leftarrow C^n$

透過演算法一與演算法二，可得出最少旅程時間路徑與最少轉乘次數路徑，接下來即可使用 Yen's Algorithm 計算出其他可行路徑。Yen's Algorithm 即為執行 $K*(L-1)$ 次 Dijkstra's Algorithm，K 為希望找出多少路徑的數量 (本研究使用 $K=3$，L 為最佳路徑的長度 (節點數)，在每一次執行之間根據上一次的結果調整路網成本 (旅程時間或轉乘次數)，舉例來說，如果最佳路徑為 R10➔R09➔G10_4➔O05_4➔O04➔O03，則在第二次迭代之前須依序將所有邊上的成本設為∞，每調整一次就再執行一次 Dijkstra's Algorithm，執行完後須將重改為原本的值，再進行下一次的調整，完整步驟如演算法三，其中 $P^{optimal}$ 為最佳路徑 (最短旅程時間或最少轉乘次數)，函數 $length(P^{optimal})$ 表示 $P^{optimal}$ 路徑中的車站數量，$p \in \{1,2,...,length(P^{optimal})-1\}$ 代表 $P^{optimal}$ 上的節點位置編號，n_p 表示 $P^{optimal}$ 位置編

doi:10.6342/NTU202504286

號為 p 所對應到的車站，n_1 表示最佳路徑上的第一個車站，n_2 表示最佳路徑上的第二個車站，以此類推，$length(N^{n_p})$ 表示節點 n_p 所擁有的鄰近車站數量。本研究為減少計算時間，不會去調整最佳路徑上所有邊的成本，若該站相鄰車站數量不大於二 (代表該車站為一般車站)，則跳過不進行調整。

演算法三 Yen's Algorithm (修改自 Yen, 1971)

輸入: K

輸出: K 最佳路徑

1. 給定 K ，$k=1$；
2. 使用演算法一 (或演算法二) 求出最佳路徑 $P^{optimal}$ ；
3. 路徑 $R \leftarrow \varnothing$，其成本 $C^R \leftarrow \infty$；
4. 當 $k < K$:
5. 　　對節點 $n_p, n_{p+1} \in P^{optimal}$:
6. 　　　　如果 $length(N^{n_p}) \leq 2$: 返回步驟 5；
7. 　　　　將邊 $n_p n_{p+1}$ 的成本設為∞；
8. 　　　　再計算一次演算法一 (演算法二)，得到路徑 r 與其成本 C^r ；
9. 　　　　如果 $C^r < C^R$:
10. 　　　　　　$R \leftarrow r, C^R \leftarrow C^r$ ；
11. 　　　　將邊 $n_p n_{p+1}$ 的成本改回原本的值；
12. 　　$P^{optimal} \leftarrow R$ ；
13. 　　$k \leftarrow k+1$

4.2 實驗結果

為了呈現演算法的結果，本研究以台北車站 (BL12/R10) 到永安市場站 (O03) 的計算結果為例，圖 12 為現有台北捷運 GO 應用程式所推薦之路徑，目前只有單一路徑，花費時間約 18 分鐘，但不包含候車時間，因其計算過程無考慮時刻表，

doi:10.6342/NTU202504286

無論何時的查詢結果皆相同。

圖 12 台北捷運 GO 推薦路徑

下方圖 13 為本研究所提出之 Dijkstra's Algorithm 在一天中兩個不同時間的計算結果，圖 13 a 為 14:00 出發路徑，圖 13 b 為 18:40 出發路徑，可看出本研究所提方法能夠隨著時間調整推薦路徑，提供更準確的資訊給旅客，表 6 為兩個結果的資訊總覽，其中旅程時間包含候車時間與轉乘時間。14:00 出發時，將從台北車站搭乘淡水信義線至東門站轉乘中和新蘆線至永安市場站，耗時約 20 分；18:40 出發時，將從台北車站搭乘板南線至西門站轉乘松山新店線至古亭站再轉乘中和新蘆線至永安市場站，耗時約 22 分，雖然此路徑需轉乘兩次，但因西門站與古亭站皆為平行轉乘站，且轉乘方向皆相同，因此轉乘時間較短。

(a) (b)

圖 13 本研究 Dijkstra's Algorithm 推薦路徑(演算法一)

doi:10.6342/NTU202504286

表 6 台北車站到永安市場路徑 (演算法一)

	路徑	轉乘次數	旅程時間
14:00 出發	R10➔R09➔R08➔R07/O06➔O05➔O04➔O03	1	20 分
18:40 出發	BL12➔BL11/G12➔G11➔G10➔G09/O05➔O04➔O03	2	22 分

圖 14 為 Yen’s Algorithm 在 14:00 出發時所計算的結果，本研究將 K 設為 3，求出前三佳路徑，圖 14a 為最短旅程時間路徑，圖 14b 為第二短，圖 14c 為第三短的路徑，表 12 為前三佳路徑的資訊總覽。最佳路徑如同 Dijkstra’s Algorithm 的結果，從台北車站搭乘淡水信義線至東門站轉乘中和新蘆線至永安市場站，耗時約 20 分；第二路徑為從台北車站搭乘板南線至忠孝新生站轉乘中和新蘆線至永安市場站，耗時約 24 分；第三路徑為從台北車站搭乘淡水信義線至中正紀念堂站轉乘松山新店線至古亭站轉乘中和新蘆線至永安市場站，耗時約 24 分。

(a) (b) (c)

圖 14 本研究 Yen's Algorithm 推薦路徑 (14:00，演算法三)

doi:10.6342/NTU202504286

表 7 台北車站到永安市場路徑 (14:00，演算法三)

	路徑	轉乘次數	旅程時間
最佳路徑	R10➔R09➔R08➔R07/O06➔O05➔O04➔O03	1	20 分
第二路徑	BL12➔BL13➔BL14/O07➔O06➔O05➔O04➔O03	1	24 分
第三路徑	R10➔R09➔R08/G10➔G09/O05➔O04➔O03	2	24 分

為了更好理解一般轉乘站與平行轉乘站在路網圖上是如何表示，下方以節點示意圖表示圖 14 與表 7 中所提到的三個路徑。圖 15 為最佳路徑之節點示意圖，其中實心圓圈為一般車站或一般轉乘站，空心圓圈是平行轉乘站的虛擬節點，最佳路徑在東門站從淡水信義線轉乘至中和新蘆線，並且是從往象山方向轉乘至往南勢角方向，因此根據表 5，需要至 O06_3 轉乘。圖 16 為第二路徑之節點示意圖，此路徑在忠孝新生站從板南線轉乘至中和新蘆線，忠孝新生站為一般轉乘站，因此只須從兩個不同路線的忠孝新生站間移動，即從 BL14 (板南線) 至 O07 (中和新蘆線)。圖 17 為第三路徑之節點示意圖，此路徑轉乘兩次，分別在中正紀念堂站從淡水信義線轉乘至松山新店線，以及在古亭站從松山新店線轉乘至中和新蘆線。這兩個轉乘站皆為平行轉乘站，根據表 5，在中正紀念堂站從往象山方向轉乘至往新店方向，需在虛擬節點 G10_4 轉乘；在古亭站從往新店方向轉乘至往南勢角方向，需在虛擬節點 O05_4 轉乘。

doi:10.6342/NTU202504286

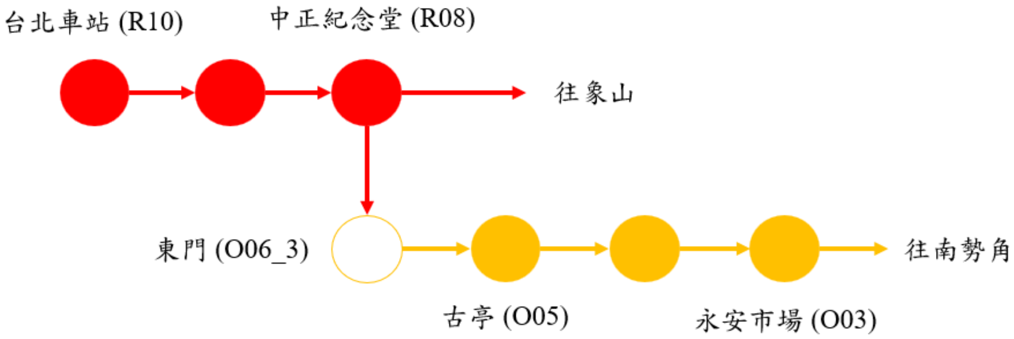


圖 15 最佳路徑節點示意圖 (台北車站➔永安市場，14:00)

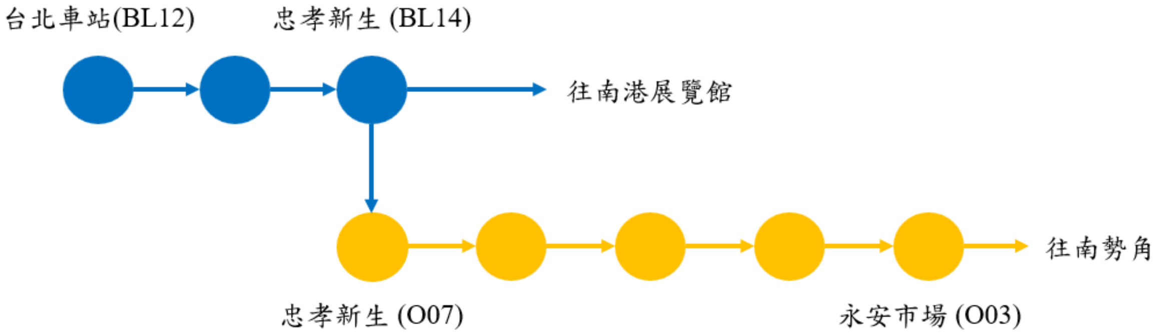


圖 16 第二路徑節點示意圖 (台北車站➔永安市場，14:00)

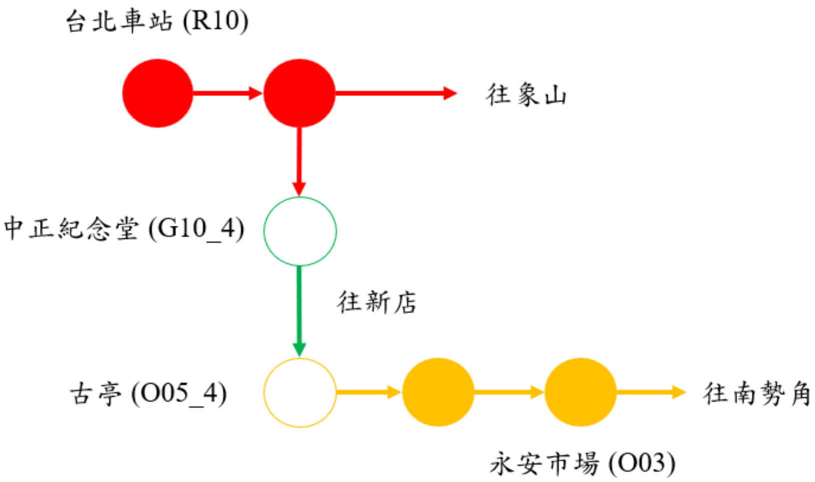


圖 17 第三路徑節點示意圖 (台北車站➔永安市場，14:00)

本研究在計算路徑成本時有考慮時刻表，因此在一天中不同時間，針對相同起訖站，會有不同的計算結果。圖 18 為台北車站到永安市場站 18:40 出發的三條路徑，圖 18 a 為第一條路徑，需轉乘兩次；圖 18b 為第二條路徑，同樣需轉乘兩次；圖 18 c 為第三條路徑，只需轉乘一次，三條路徑的總花費時間皆約為 20 分鐘，表 8 為這三條路徑的資訊總覽，其中第二佳路經又不同於上方所出現的所有路徑。

doi:10.6342/NTU202504286

(a) (b) (c)

圖 18 本研究 Yen's Algorithm 推薦路徑 (18:40，演算法三)

表 8 台北車站到永安市場路徑 (18:40，演算法三)

	路徑	轉乘次數	旅程時間
最佳路徑	R10➔R09➔R08/G10➔G09/O05➔O04➔O03	2	20 分
第二路徑	BL12➔BL11/G12➔G11➔G10/R08➔G09/O05➔O04➔O03	2	20 分
第三路徑	R10➔R09➔R08➔R07/O06➔O05➔O04➔O03	1	20 分

在表 8 中，只有第二路徑不同於上述所有路徑，因此這部分只說明第二路徑之節點示意圖，如圖 19。此路徑轉成兩次，分別在西門站從板南線轉乘至松山新店線，以及在古亭站從松山新店線轉乘至中和新蘆線，這兩個轉乘站皆為平行轉乘站，根據表 5，在西門站從往頂埔方向轉乘至往新店方向，需在虛擬節點 G12_4 轉乘；在古亭站從往新店方向轉乘至往南勢角方向，需在虛擬節點 O05_4 轉乘。

doi:10.6342/NTU202504286

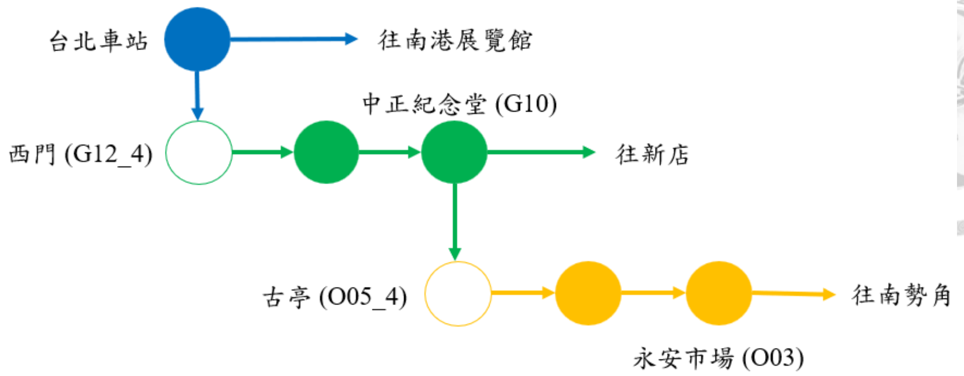


圖 19 第二路徑節點示意圖 (台北車站➔永安市場，18:40)

路徑所花費時間的不同會隨著起訖站間的距離越遠而越明顯，因此也展示永寧 (BL02)➔文德 (BR18) 的實驗結果，同樣使用演算法三計算前三條路徑，出發時間設置為 13:30，路徑資訊總覽如表 9，最佳路徑只有轉乘一次，即在忠孝復興站轉乘，花費時間約 53 分鐘；第二路徑需轉乘三次，分別在台北車站、中山站與南京復興站轉乘，花費時間約 56 分鐘；第三路徑同樣需轉乘三次，分別在忠孝新生、松江南京與南京復興站轉乘，花費時間約 64 分鐘。

doi:10.6342/NTU202504286

表 9 永寧到文德路徑 (13:30，演算法三)

	路徑	轉乘次數	旅程時間
最佳路徑	BL02➔BL03➔BL04➔BL05➔BL06➔BL07➔BL08➔BL09➔BL10➔BL11➔BL12➔BL13➔BL14➔BL15/BR10➔BR11➔BR12➔BR13➔BR14➔BR15➔BR16➔BR17➔BR18	1	53 分
第二路徑	BL02➔BL03➔BL04➔BL05➔BL06➔BL07➔BL08➔BL09➔BL10➔BL11➔BL12/R10➔R11/G14➔G15➔G16/BR11➔BR12➔BR13➔BR14➔BR15➔BR16➔BR17➔BR18	3	56 分
第三路徑	BL02➔BL03➔BL04➔BL05➔BL06➔BL07➔BL08➔BL09➔BL10➔BL11➔BL12➔BL13➔BL14/O07➔O08/G15 G16/BR11➔BR12➔BR13➔BR14➔BR15➔BR16➔BR17➔BR18'	3	64 分

圖 20 為表 9 中最佳路徑之節點示意圖，節點間三個小點表示省略中間所有練續節點，此路徑僅轉乘一次，在忠孝復興站從板南線轉乘至文湖線，忠孝復興站為一般轉乘站，因此只須從兩個不同路線的忠孝復興站間移動，即 BL15 (板南線) 至 BR10 (文湖線) 間移動。圖 21 為表 9 中第二路徑之節點示意圖，此路徑轉乘三次，分別在台北車站從板南線轉乘至淡水信義線，在中山站從淡水信義線轉乘至松山新店線，以及在南京復興站從松山新店線轉乘至文湖線，這三個轉乘站皆為一般轉乘站，分別從 BL12 (板南線) 至 R10 (淡水信義線)，從 R11 (淡水信義線) 至

doi:10.6342/NTU202504286

G14 (松山新店線)，與從 G16 (松山新店線) 至 BR11 (文湖線)。圖 22 為表 9 中第三路徑之節點示意圖，此路徑同樣轉乘三次，分別在忠孝新生站從板南線轉乘至中和新蘆線，在松江南京從中和新蘆線轉乘至松山新店線，以及在南京復興站從松山新店線轉乘至文湖線，這三個轉乘站同樣皆為一般轉乘站，分別從 BL14 (板南線) 至 O07 (中和新蘆線)，從 O08 (中和新蘆線) 至 G15 (松山新店線)，與從與從 G16 (松山新店線) 至 BR11 (文湖線)。

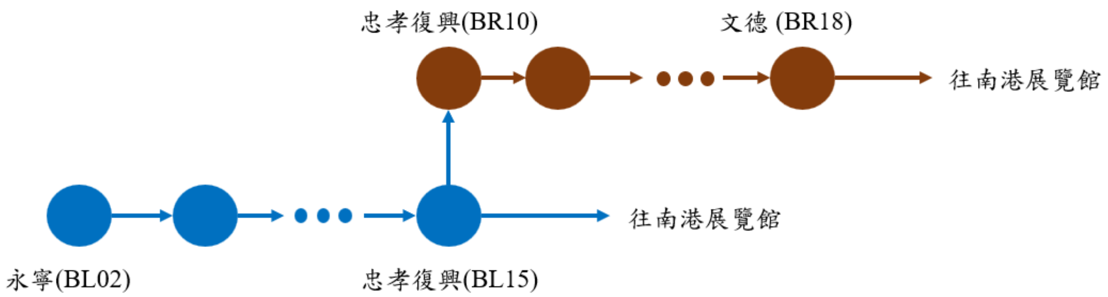


圖 20 最佳路徑節點示意圖 (永寧➔文德，13:30)

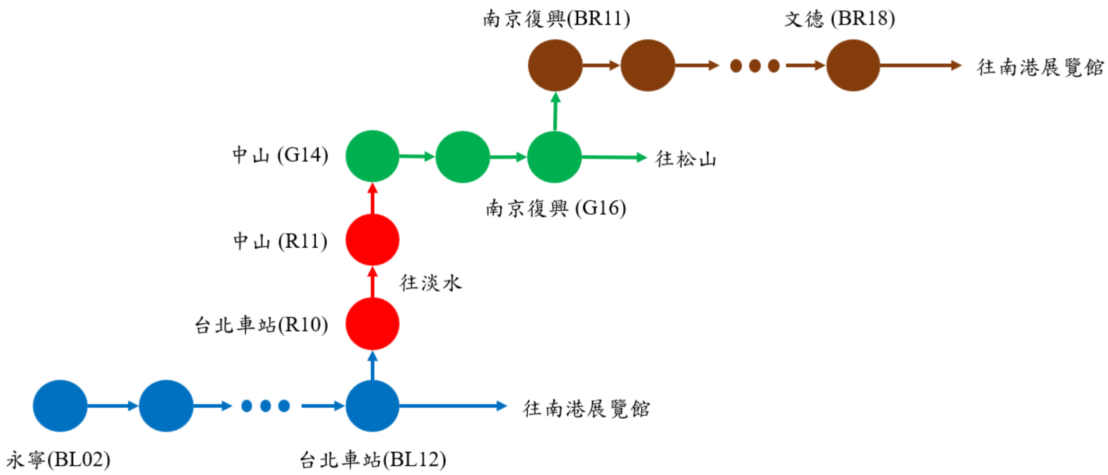


圖 21 第二路徑節點示意圖 (永寧➔文德，13:30)

doi:10.6342/NTU202504286

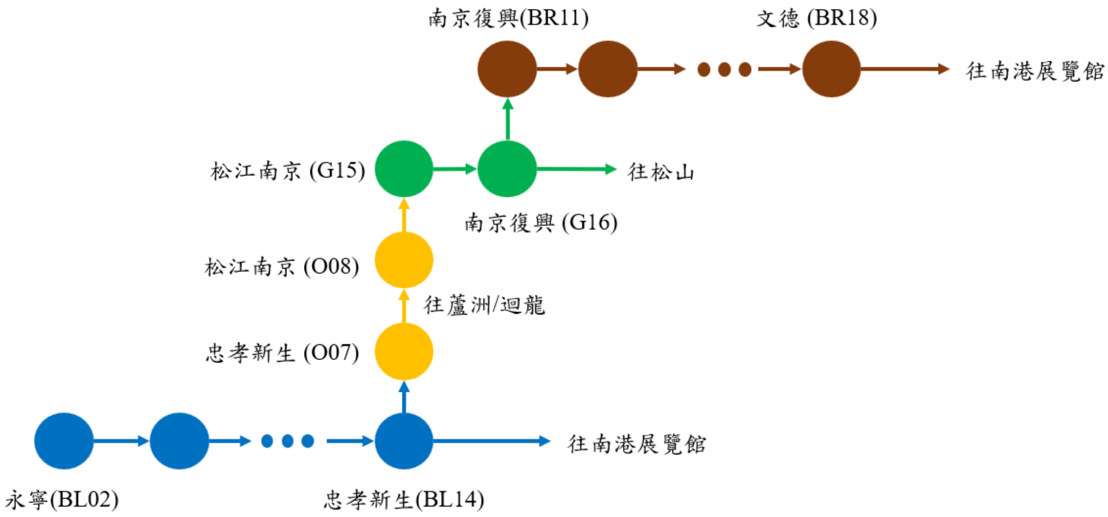
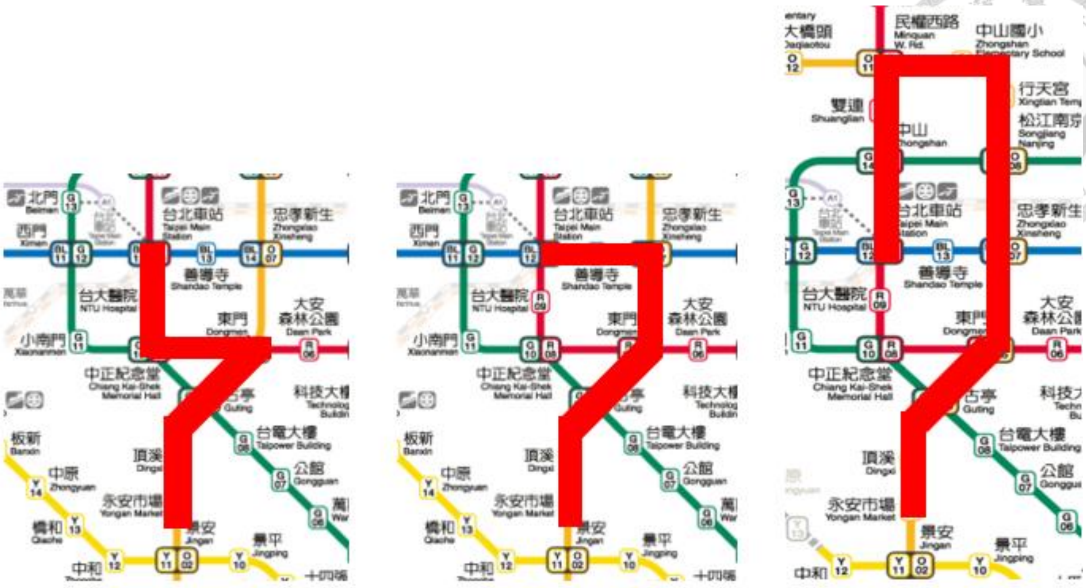


圖 22 第三路徑節點示意圖 (永寧➔文德，13:30)

除了前三短旅程時間路徑，本研究也使用相同的 Yen’s Algorithm 計算前三少轉乘次數路徑，圖 23 為其計算結果，圖 23a 為最少轉乘次數路徑，圖 23b 為第二少，圖 23c 為第三少的路徑，以此例子來看，三個路徑的轉乘次數皆為一。最佳路徑與第二路徑皆與較少旅程時間的結果相同，而第三路徑從台北車站搭乘淡水信義線至民權西路站轉乘中和新蘆線至永安市場站，表 10 為前三佳路徑的資訊總覽。

比較兩個由 Yen’s Algorithm 計算的結果，可看出較少轉乘次數路徑與較短旅程時間路徑相似，僅第三路徑不同，第三路徑反而先往反方向搭乘再轉乘至正確的方向，所以即使旅程時間較短不代表轉乘次數會較少。第三路徑雖然轉乘次數較少，但整體所需花費時間較長，不過可減少旅客在轉乘時所需的轉乘步行時間與可能的候車時間。根據本研究所計算的多種路徑，旅客可以根據各自的偏好進行選擇。

doi:10.6342/NTU202504286



(a) (b) (c)

圖 23 本研究 Yen's Algorithm 推薦路徑 (演算法二)

表 10 台北車站到永安市場路徑 (演算法三)

	路徑	轉乘次數
最佳路徑	R10➔R09➔R08➔R07/O06➔O05➔O04➔O03	1
第二路徑	BL12➔BL13➔BL14/O07➔O06➔O05➔O04➔O03	1
第三路徑	R10➔R11➔R12➔R13/O11➔O10➔O09➔O08➔ O07➔O06➔O05➔O04➔O03	1

doi:10.6342/NTU202504286

第五章 結論

本研究成功結合 Dijkstra's Algorithm 與 Yen's Algorithm，發展出一套能同時考慮轉乘次數與搭乘時間的動態路徑搜尋機制。透過實際捷運路網與時刻表資料的整合，使演算法能反映不同時間出發下的列車班距與轉乘限制，有效模擬乘客實際搭乘情境。針對實際營運情況中的平行轉乘問題，本研究在每個平行轉乘站新增四個虛擬節點，分別處理四個行車方向，額外的節點即能有效處理同站轉乘成本不同的問題。將路網資料與時刻表結合後，即可使用 Dijkstra's Algorithm 求出成本最小的路徑。

Yen's Algorithm 進一步提供多條可行替代路徑，有助於乘客依自身偏好或即時營運狀況彈性調整選擇，相較現有方法只能提供一條路徑，本研究能夠提供三條不同路徑。本研究選擇台北車站➔永安市場這對 OD pair 進行實驗，實驗結果顯示，考慮時刻表與轉乘成本能顯著提升推薦路徑的實用性，尤其是在不同時段能彈性更改路徑，較現有方法有更高的彈性與選擇性，例如，台北車站➔永安市場，14:00 出發與 18:40 出發的推薦路徑不盡相同，所需花費時間也不同，旅客可以根據當下需求自行選擇搭乘哪種路徑。

整體而言，本研究證實結合演算法與時刻資訊可提升捷運路徑推薦的精準度與彈性，但能然有許多尚未考量到的因素，例如車廂擁擠度、不同車廂至電梯或手扶梯的距離、車站內部複雜度與旅客不同的走路速度，未來可以進一步將上述因素納入成本運算中，並將模型隨機化，以更加符合實際旅客搭乘情況，或是進一步調整演算法，計算中途經過特定車站路徑，目的不外乎就是要提升旅客搭乘體驗，同時也能提升內部營運效率。

doi:10.6342/NTU202504286

參考文獻

維基媒體專案貢獻者. (2025, July 12). 臺北捷運. 維基百科，自由的百科全書. https://zh.wikipedia.org/zh-tw/%E8%87%BA%E5%8C%97%E6%8D%B7%E9%81%8B

臺北大眾捷運股份有限公司. (n.d.). 路網圖、各站資訊及時刻表. 臺北大眾捷運股份有限公司. https://www.metro.taipei/cp.aspx?n=91974F2B13D997F

臺北大眾捷運股份有限公司. (n.d.-a). 臺北大眾捷運股份有限公司. 臺北大眾捷運股份有限公司. https://web.metro.taipei/img/ALL/stationprofile/089.jpg

臺北大眾捷運股份有限公司. (n.d.-b). 臺北大眾捷運股份有限公司. 臺北大眾捷運股份有限公司. https://web.metro.taipei/img/ALL/stationprofile/086.jpg

臺北大眾捷運股份有限公司. (n.d.-c). 願景、使命與核心價值. 臺北大眾捷運股份有限公司. https://www.metro.taipei/cp.aspx?n=AE0EC5DA7D0804D4

臺北大眾捷運股份有限公司. (n.d.-d). 路線及班距. 臺北大眾捷運股份有限公司. https://www.metro.taipei/cp.aspx?n=EAD981369A065968&s=1CC297E1D0B4C7B1

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network flows: Theory, Algorithms, and applications. Prentice Hall.

Auer, H., & Perger, T. (2020). Energy efficient route planning for electric vehicles with special consideration of the topography and battery lifetime. *Energy Efficiency*, 13, 1705–1726. https://doi.org/10.1007/s12053-020-09900-5

Bertolini, L., le Clercq, F., & Kapoen, L. (2005). Sustainable accessibility: A conceptual framework to integrate transport and land use plan-making. *Transport Policy*, 12(3), 207–220. https://doi.org/10.1016/j.tranpol.2005.01.006

Bozyigit, A., Nasiboglu, E., & Alankus, G. (2017). Public transport route planning: Modified Dijkstra's Algorithm. In *2017 International Conference on Computer Science* and Engineering (UBMK) (pp. 502–505). https://doi.org/10.1109/UBMK.2017.8093444

doi:10.6342/NTU202504286

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. https://doi.org/10.1007/BF01386390

Farhan, M. (2019). Traffic Routing Algorithm for Road Network. *Scientific Bulletin*, 24, 131–138. https://doi.org/10.2478/bsaft-2019-0015

Guo, J., Guo, B., Song, G., & Liu, T. (2024). Solving the Robust Shortest Path Problem with Multimodal Transportation. *Mathematics*, 12(19), 2978. https://doi.org/10.3390/math12192978

Hadi, H., & Ibrahim, I. (2025). A comprehensive review of shortest path Algorithms for network routing. *Asian Journal of Research in Computer Science*, 18(3), 584. https://doi.org/10.9734/ajrcos/2025/v18i3584

Idri, A., Oukarfi, M., Boulmakoul, A., Masri, A., & Zeitouni, K. (2017). A new time-dependent shortest path Algorithm for multimodal transportation network. *Procedia Computer Science*, 113, 692–697. https://doi.org/10.1016/j.procs.2017.05.379

Ulum, U., Chulkamdi, M., & Santi, I. (2025). Implementation of Dijkstra Algorithm in determining the fastest route for goods delivery. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*. https://doi.org/10.59934/jaiea.v4i2.923

Yen, J. Y. (1971). Finding the K shortest loopless paths in a network. *Management Science*, 17(11), 712–716. https://doi.org/10.1287/mnsc.17.11.712

Yu, X., Cao, H., Cao, K., Zhu, L., & Zou, L. (2024). Considering the optimization design of urban bus network scheduling. *Applied Sciences*, 14(14), 6337. https://doi.org/10.3390/app14146337

Zhan, F. B., & Noon, C. E. (1998). Shortest path Algorithms: An evaluation using real road networks. *Transportation Science*, 32(1), 65–73. https://doi.org/10.1287/trsc.32.1.65

Zhang, X., Liu, M., Li, P., Wang, G., Hu, J., Chan, K., & Qiu, J. (2019). Yen's Algorithm-based charging facility planning considering congestion in coupled transportation and

doi:10.6342/NTU202504286

power systems. *IEEE Transactions on Transportation Electrification*, 5(4), 1134–1144. https://doi.org/10.1109/TTE.2019.2959716