第二章　文獻回顧

本研究主要探討通勤交通車路線問題，其屬於車輛路線問題之衍生問題，且與公車路網之規劃有部分相似，故本章第一節先回顧最短路徑問題之各種演算法，以便了解最短路徑之求解方法；第二節先闡述 VRP 問題之定義，並介紹過去專家學者對 VRP 問題之解法；第三節則回顧國內外公車路網設計之相關文獻；第四節則敘述國內外校車與交通車之相關文獻。

2.1 最短路徑問題(Shortest Path Problem)

最短路徑問題(Shortest Path Problem)為路網研究中極重要的一部份，在一個路網當中，給定一起點和迄點，並考慮所有可能影響之因素，求得最低之成本或是最快之時間到達目的。應用於交通運輸方面，如物流運輸和旅客的運輸，另外在緊急救難時是更需要運用最短路徑，提供人員疏散或緊急救難之決策方案[28]。

最短路徑問題根據起迄點及問題特性可分為以下三類：

1. 一對一(one-to-one)：由特定起點至特定迄點之最短路徑。
2. 一對多(one-to-all)：由一個點至其他各點之最短路徑。
3. 多對多(all-to-all)：路網上任意兩點之最短路徑。

前兩類最短路徑問題，主要求解方式為 Label Setting Algorithm 以及 Label Correcting Algorithm。Label Setting Algorithm 只能用來求解節線成本為非負的路網，而 Label Correcting Algorithm 除了可以在非負的節線成本使用外，也可以用於節線成本為負的路網，但不能在有負循環(negative cycle)的路網中使用。至於第三類多對多的最短路徑問題，則可以透過一對一最短路徑的求解方式，將其擴大為每一個節點都是終點來計算。以下將常見的演算法整理介紹如下[18]。

1. 標記設定法(Label Setting Algorithm)

Label Setting Algorithm 中較具代表性的為 1959 年被提出之 Dijkstra's Algorithm，這個方法的基本概念是給節點暫時性標號，表示由起點 S 到此節點路徑之長度上限，這些標號將隨著重複之演算步驟而增大，而且每個循環將會比較所有暫時性標號的距離標籤，選擇最小的距離標籤並將之轉換成永久性標號。而永久性標號代表起點 S 到該節點之最短路徑。以下則是 Dijkstra's Algorithm 的演算步驟：

首先，先針對下面出現之符號進行說明：L(X_i)為節點 X_i 的成本標號值，Γ(P)為與點 P 相連之所有點集合，P 為固定為永久標籤之點，C(i, j)表示為 i 至 j 之成本。

步驟一：令起點S為永久標號，且所有不等於S之 X_i 設為暫時標號，即 $L(S)=0$，$L(X_i)=\infty$，$P=S$。

步驟二：對所有屬於 Γ(P)且為暫時標號之Xi，更新標號值：

$$L(X_i)=\min\left[L(X_i),\ L(P)+C(P,\ X_i)\right]$$

步驟三：於所有暫時性標號中找出成本標號值最小的節點 X_i'，並更新該點之成本標號值，即 $L(X_i')=\min\left\{L(X_i)\right\}$。

步驟四：設定 X_i' 的標號為永久性，即 $P=X_i'$。

步驟五：(1) 若只找起點S到終點T之最短路徑，則可慮以下二點：

i 如果 $P=T$，即停止演算。

ii 如果 $P\neq T$，則重複步驟二。

(2) 若係找起點S到所有點之最短路徑，則到步驟六。

步驟六：當所有點皆為永久標號，則該標號即為該點之最短路徑，停止演算。若標號仍為暫時標號，則重複步驟二。

2. 標記修正法(Label Correcting Altorithm)

Label Correcting Algorithm 的計算方式跟 Label Setting Algorithm 有點類似，不同的是其係將所有節點視為暫時標籤，直到最後才同時將所有節點的距離標籤變成永久標籤。以下為 Label Correcting Algorithm 的演算步驟：

首先，先針對下面出現之符號進行說明：$l^k(i)$ 為從起點 s 經 k 階到達 i 的距離，$L^{(k)}$ 為在 k 階時 $l^k(i)$ 的集合，$S^{(k)}$ 為從起點 s 開始，恰 k 階 SP 到達的(終)點集合，$P^{(k)}$ 為 k 階時所有點的前置點集合，$\Gamma(S)$ 為 S 集合內所有點的下游點的集合，而 $\Gamma^{-1}(S)$ 則為 S 集合內所有點的上游點的集合。

起始條件：當 $i\neq s$ 時，$k=0$，$l(s)=0$，$l(i)=\infty$，且 $S=\left\{s\right\}$，$k=k+1$，進行步驟一。

步驟一：更新所有屬於 $\Gamma(S)$ 的點 j 之 $l(j)$，而 $l(j)=\min\left[l(j),\ \min\limits_{\mathrm{k}\in \mathrm{T_j}}(l(k)+c_{kj})\right]$，其中 T_j 即 S 點集合中可以直接連到 j 點的子集合。

步驟二：更新第k階SP到達的點集合。

$S=\left\{j\,|\,在第一步驟更新過的l(j)\right\}$。

步驟三：(1) 若 $\mathrm{k}\leq$(n-1) 且 $S\neq\Phi$，$k=k+1$，則回到步驟一。

(2) 若 $\mathrm{k}\leq$(n-1) 且 $S=\Phi$，則停止演算。

(3) 若 $k=n$ 且 $S\neq\Phi$，則則停止演算。

3.　Floyd-Warshall Algorithm

於 1962 年被提出的 Floyd-Warshall Algorithm 是利用矩陣運算計算出 all-to-all 最短路徑演算法。其演算步驟如下：

起始條件：若節線(i, j)存在，則 $d_0(i, j) = \ell(i, j)$ ；若 $i = j$ ，則 $d_0(i, j) = 0$ ；
若節線(i, j)不存在，則 $d_0(i, j) = \infty$ 。

對所有 $i \neq j$ 時，則前置點矩陣 $p_0(i, j) = i$ ；當 $i = j$ 時，不必追蹤前置點矩陣。

步驟一：令 $k = 1$ 。

步驟二：根據下列式子計算距離矩陣 $D^{(k)}$ ：

$$d_k(i, j) = Min\left[d_{k-1}(i, j), d_{k-1}(i, k) + d_{k-1}(k, j)\right]$$

步驟三：以下列的式子計算前置點矩陣 $P^{(k)}$ ：

$$P_k(i, j) = \begin{cases} P_{k-1}(k, j), & \text{若 } d_k(i, j) \neq d_{k-1}(i, j) \\ P_{k-1}(i, j), & \text{otherwise} \end{cases}$$

步驟四：當k = n，則停止，否則令k = k+1 ，然後回到步驟二。

2.2 車輛路線問題

2.2.1　車輛路線問題(Vehicle Routing Problem, VRP)定義

Dantzig 和 Ramser 二位學者首先於 1959 年提出 VRP 問題，VRP 問題主要是有效率地派遣車輛，使車輛依序到許多個服務地點做收貨和(或)送貨的服務，結束時並回到出發點；其目標是找到一組車輛路線組合可以滿足各種限制並且最小化所有車子之運輸成本[5]。

VRP 問題之定義整理如下：「給定一路網 G（N,L），N 代表節點（Nodes）集合，L 代表節線（Links）的集合，設有一場站（depot）以節點 0 表示，此場站擁有 m 輛車，每輛車之容量皆為 q；給定有 n 位顧客需要服務，每位顧客有 d_i (i=1~N)之需求量，車輛從顧客 i 到顧客 j 的成本(含場站)為 c_{ij} ；其中，每位顧客僅能被一輛車服務一次，且每輛車不可服務相同之顧客。在不違反車輛容量限制之條件下，希望在此路網上找到一組路線組合，使得 k 輛車自場站出發服務完所有顧客並回到場站之成本最小。」一般而言，網路之問題可用數學規劃模式表示，，茲將 VRP 之數學規劃表示如下[5]:

$$\text{Minimize} \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{M} c_{ij} \cdot x_{ij}^{k} \tag{2-1}$$

$$\text{subject to} \sum_{i=0}^{n}\sum_{k=1}^{m} x_{ij}^{k} = 1 \quad (j = 1,\ \ldots,\ n) \tag{2-2}$$

$$\sum_{j=0}^{n}\sum_{k=1}^{m} x_{ij}^{k} = 1 \quad (i = 1,\ \ldots,\ n) \tag{2-3}$$

$$\sum_{i=0}^{n} x_{ip}^{k} - \sum_{j=0}^{n} x_{pi}^{k} = 0 \quad (k = 1,\ \ldots,\ m;\ p = 0,\ \ldots,\ n) \tag{2-4}$$

$$\sum_{i=0}^{n} d_i (\sum_{j=0}^{n} x_{ij}^{k}) \le q \quad (k = 1,\ \ldots,\ m) \tag{2-5}$$

$$\sum_{j=1}^{n} x_{0j}^{k} \le 1 \quad (k = 1,\ \ldots,\ m) \tag{2-6}$$

$$\sum_{i=1}^{n} x_{i0}^{k} \le 1 \quad (k = 1,\ \ldots,\ m) \tag{2-7}$$

$$y_i - y_j + n\sum_{k=1}^{m} x_{ij}^{k} \le n - 1 \quad (2 \le i \ne j \le n;\ y_i\text{為實數}) \tag{2-8}$$

$$x_{ij} = 0 \text{ or } 1 \quad \text{for all } i,\ j,\ k \tag{2-9}$$

c_{ij}:代表顧客i到顧客j的行駛成本

x_{ij}^{k}:0-1變數;若車輛k從顧客i到顧客j則為1,否則為0

d_i:顧客i之需求量

q:車輛之容量

n:節點之數量,也就是顧客數

m:車輛數

其中,(2-1)式為路線本最小之目標式;(2-2)式與(2-3)式係確保每一位客戶只被一輛車服務;(2-4)表示到達顧客 i 與離開顧客 i 的車輛為同一輛車;(2-5)表示車輛所服務之顧客需求量不得超過車輛容量之限制;(2-6)式與(2-7)式表示並非所有車輛皆要使用;(2-8)為避免產生子迴路之限制式;(2-9)式為雙元整數限制式,若決策變數 $x_{ij}^{k}=1$ 表示從顧客 i 到顧客 j 使用車輛 k 服務,否則 $x_{ij}^{k}=0$。

2.2.2 傳統啟發式解法

VRP 問題求解之方式可分為最佳解解法(Exact Solution)和近似解(Approximation)二大類;最佳解解法大多是以數學規劃為基礎,以雙元整數規劃(Binary Integer Programming)的形式來求解最佳解,例如:分枝定限法(Branch and Bound)、變數產生法

(Column Generation)等，而近似解解法大多屬於啟發式解法。因 VRP 屬於 NP-hard 問題，在求解大規模問題時往往無法於有效的時間內求出最佳解，故對於 VRP 問題大多以啟發式解法求解。Bodin 和 Golden[5]等人將 VRP 的啟發式解法分為四類[5][36]：

1. 先分群再設計路線(Cluster-first route-second)：此法分為二階段，第一階段先將需求點分群，第二階段再對各群組進行路線設計。
2. 先設計路線再分群(Route-first cluster-second)：先針對所有的需求點，構建一包含所有點之路線，其次再根據問題之限制條件將此路線分割為數條較小且可行的路線。
3. 節省法或插入法(Savings/insertion)：建構路線時以節省值的大小作為插入之依據，將節省值最大的需求插入路線中，直到所有的需求點都插入路線上，且不能再節省為止。
4. 改善與交換法(Improvement/exchange)：針對起始路徑進行交換改善，以減少路線總成本，直到無法改善為止。

此外，傳統之啟發式解法又可依上述四類再歸納分成以下三種[32][34]:

1. 路線構建型(Tour Construction)：係根據路網距離或成本矩陣直接產生較佳的可行解，常見的方法包括鄰點法(Neighbor procedure)、插入法(Insertion Method)、節省法(Saving Method)等。
2. 路線改善型(Tour Improvement)：係針對任意一個起始可行解，以鄰域搜尋法(Local Search)改善路線成本，以求得更好的解，例如：K-Opt、Or-Opt 等。
3. 綜合型(Composite)：係將路線構建和路線改善合併執行，或一面建構路線一面改善路線，如「路線構建起始解+2-Opt」、「路線構建起始解+2-Opt+3-Opt」等。

上述三種 VRP 傳統啟發式解題架構的績效，以綜合型架構為最佳。而許多績效良好的綜合型架構係以交換型啟發式解法為其主幹，可見交換型啟發式解法確實具有實務應用上之潛力。但是交換型啟發式解法因為只在目標值有改善時才進行交換，加上受起始解組合型態之影響，所以在求解過程中易陷入局部最佳解而無法繼續尋找較佳的解。

2.2.3 鄰域搜尋法

針對任一起始解利用鄰域搜尋法可進行路線改善，以求得較佳之結果，茲將 k-opt、Or-opt 與 Lin & Kernighan 等交換法進行回顧。

1. K-opt 節線交換法[15][35]

此法於 1965 年由 Lin [15]提出，其中 K 表示每次進行交換的節線數，一般最常使用的為 2-opt 及 3-opt。其解題架構之觀念：針對任一起始解，交換同路線上任 K 條不相鄰的節線，然後檢查交換後的解是否優於交換前的解。若是，則更新路線；否則維持原解，

繼續交換其他 K 條節線，直到所有可交換的節線都檢查完畢為止。如圖 2.1 所示：2-opt 換掉（1,2）與（3,4）二條節線後，可連接（1,3）與（2,4）兩條節線形成另一可行的路線，然後檢查其結果是否優於原先的解。

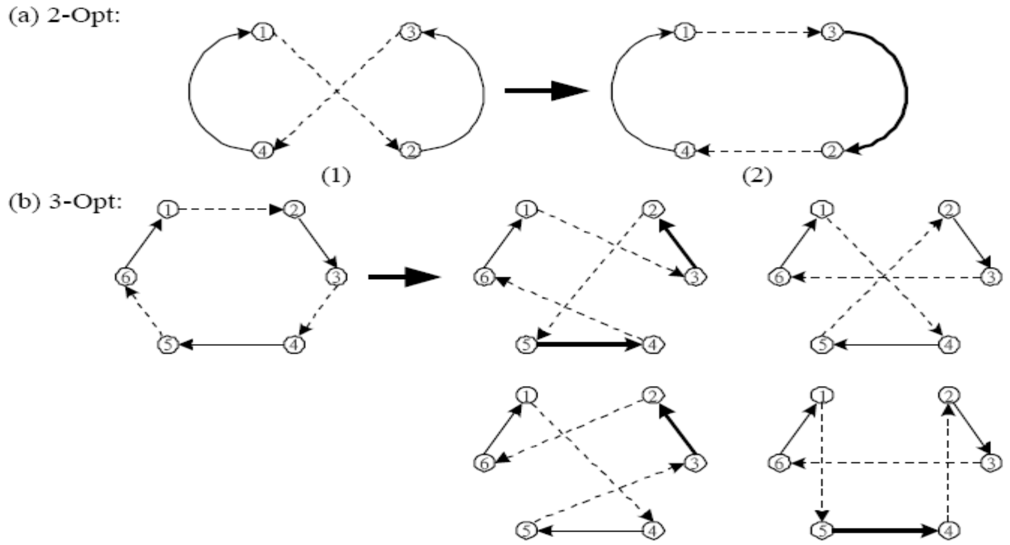

圖 2.1　K-opt 節線交換法之解題觀念

資料來源：[35]

2.　Or-opt 節線交換法[23][32][35]

Or [23]學者於 1976 年提出一種簡化 3-opt 節線交換法的方法，稱為 Or-opt 節線交換法。此法在每次的鄰域搜尋中，陸續將二段節線 (p=3) 、一段節線 (p=2) 及一個節點 (p=1) 從路線中抽出，再將其插入該路線的其他節線之間，然後檢查交換後的解是否優於交換前的解。若是，則更新路線；否則維持原解，繼續交換其他 K 條節線，直到所有可交換的節線都檢查完畢為止。如圖 2.2（a）為例：若換掉圖 2.2（a-1）的（1,2）、（2,3）及（6,7）三條節線，然後連接（1,3）、（6,2）及（2,7）三條節線而變成圖 2.2（a-2）的路線，將可能改善起始解。值得注意的是，Or-opt 節線交換法在進行交換時，只需要將小部份節線反轉，因而提高程式執行的效率。

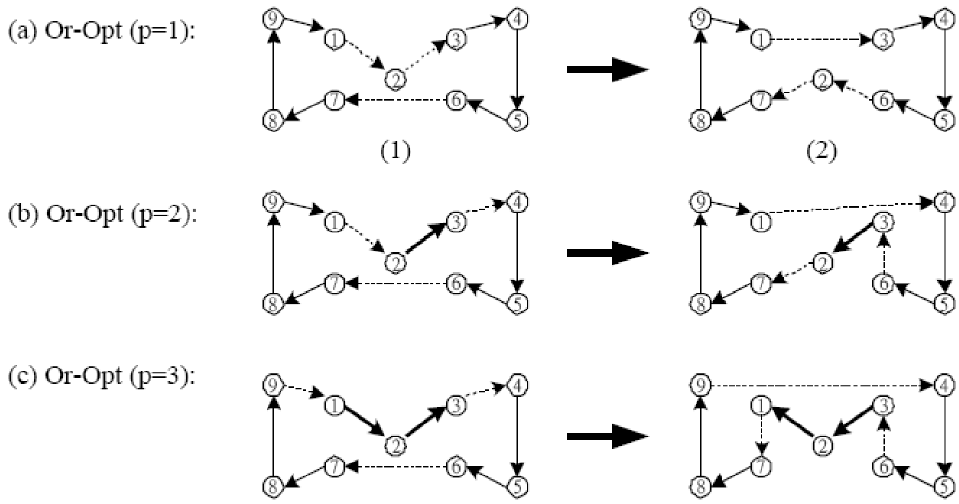

圖 2.2　Or-opt 節線交換法之解題觀念

資料來源：[35]

3.　Lin ＆ Kernighan 交換法[17][32][35]

Lin 和 Kernighan [17]於 1973 年提出一種稱為「變動深度(variable depth)」的節線交換法，每次移動時所交換的節線數並非固定值，使得其產生鄰解的機制更為複雜。Reinelt 曾指出，Lin ＆ Kernighan 交換法的解題精確度明顯優於 K-opt 節線交換法及當時期大多數的啟發式解法（例如：插入法、節省法、掃描法、最小擴張樹等等）。

4.　節點交換法[8][32]

當路線有多條時，可應用路線間的節點交換法來改善路線成本。此法最早由 Christofides 和 Eilon [8]於 1969 年提出，係對任二條路線相互交換其部份節點，然後檢查交換後的結果是否可行且獲得改善，然後決定某種交換形式以獲得較佳的路線。節點交換的形式眾多，例如：1 點對 0 點、1 點對 1 點、1 點對 2 點等等。圖 2.3 顯示 1 點對 0 點與 1 點對 1 點交換型態之示意圖。

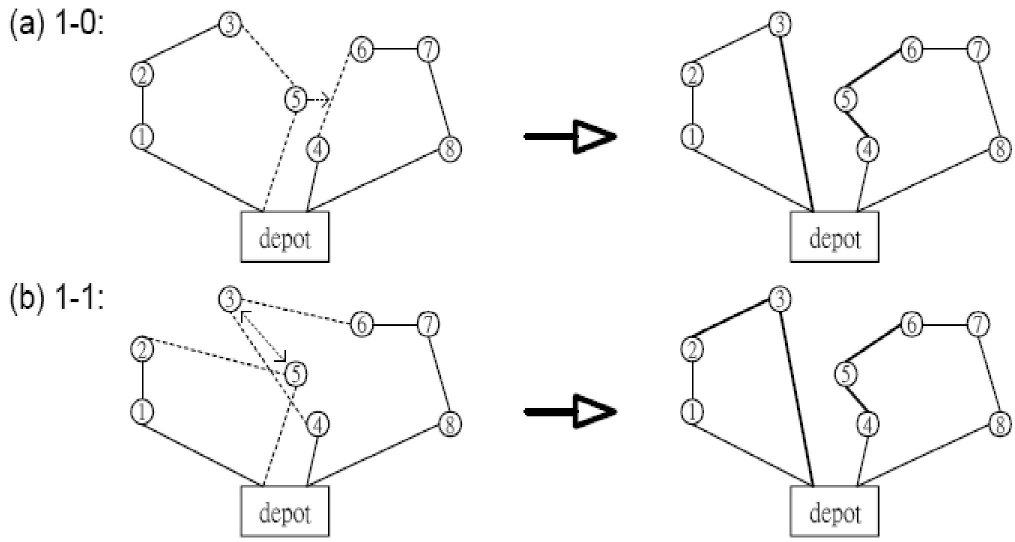

圖 2.3　(1-0)與(1-1)節點交換法的解題觀念

資料來源：[32]

2.2.4　小結

根據上述 VRP 問題之特性與定義，發現 VRP 問題與本研究所探討之通勤交通車路線問題之特性略有不同，本節將針對此二問題進行比較，討論二問題相似處與相異處。

首先，考量單一個場站或單一個工作廠區，二問題之起、迄點型態皆屬於多對一之型態，即多個起點、一個迄點。限制部份二問題皆有車容量限制及路線長度或時間限制，且需求皆不允許分割。而相異之處主要在於每一車輛之路線型態，VRP 問題之路線因從場站出發，最後又回到場站，故屬於一個迴路(cycle)；而通勤交通車路線問題之路線屬於單向之路徑(Hamiltonian path)。

對於只有單一個工作廠區時，可以利用成本函數將通勤交通車路線問題轉換成 VRP 問題，轉換方式係將從工作廠區至其他任何一個點之距離成本設定為 0 即可。若 i、j 代表節點，工作廠區為節點 1，其距離成本之數學列式如下：

$$\begin{cases} d'_{ij} = d_{ij} & if\ \ i \neq 1 \\ d'_{ij} = 0 & if\ \ i = 1 \end{cases} \qquad (2\text{-}10)$$

然而，若考量多個場站或多個工作廠區時，二問題之特性則有較多之差別，對於多場站 VRP 問題而言，每一個場站所提供之服務皆為相同，每個節點上之需求可以由任何一個場站之車輛服務，但對多工作廠區之通勤交通車問題而言，不同工作廠區之上班員工不同，每個起點上之需求有其對應之迄點，不能將其隨意指派。

此外，多場站 VRP 問題其各場站位置通常較為分散，以便可以涵蓋整個服務範圍；

也因各場站位置較為分散，故可以將整個問題劃分成單一場站之子問題，每個場站只需服務其所屬之服務區域即可。而多工作廠區之通勤交通車問題其廠區位置非常相近，如每個工作廠區皆位於科學園區內，若將多工作廠區之通勤交通車問題劃分成單一個工作廠區之子問題時，可能會造成成本之浪費，故對於多工作廠區之通勤交通車問題本研究建議不要將其劃分成單一個工作廠區之子問題，應同時考量多個工作廠區。

2.3 公車路網規劃方法

有關交通車路網之規劃方法並沒有專門的文獻探討，但國內外有關公車路網規劃方法分類的相關文獻甚多，由於交通車之特性與公車具有某些程度之相似性，因此可參考公車路網規劃的方法，加以研究。此節將先說明公車路網規劃方法的分類，以了解目前學者針對公車路網規劃有哪些方法。

2.3.1 公車路網規劃方法分類

公車路網規劃方法的分類以 Chua 與 Silcock[9]的分類最具代表性，他們依據公車路網的規劃程序觀念，將公車路網的規劃方法歸納為規劃手冊法(Manual)、系統分析法(System Analysis)、市場分析計畫法(Market Analysis Project, MAP)、交談式電腦繪圖輔助法(System Analysis with Interactive Graphics, SAIG)、數學尋優法(Mathematical Optimization)和啟發式解法(Heuristic Method)六類，而後 Chua[10] 於 1984 年又將數學尋優法和合理求解法合併為數學法(Mathematical Approach)。茲將各類規劃方法簡述如下[9][10][31]：

1. 規劃手冊法

規劃者依據公車路網之特性及路網設計原則，憑其個人之「專業知識」與「主觀認知」來規劃公車路線。此法之優點為操作簡便、費用低廉，缺點則為較易產生無效率的公車路網型態。

2. 系統分析法

依照一套程序步驟建立公車路網，利用大眾運輸指派模式預估各公車路線之運量，並建立評估模式以做為分析公車路網之依據。此法著重於評估模式之建立，而對於路網設計較缺乏明確之方法。

3. 市場分析計畫法

市場分析計畫法類似於系統分析法的系統化程序。此法僅在乘客起迄及家戶資料之分析應用電腦程式處理，而在公車路網的產生與路線率次運量的預測仍採人工處理，因此對於大部分複雜的都市地區顯得較不符合實際之狀況。

4. 交談式電腦繪圖輔助法

所謂交談式電腦繪圖輔助法係指設計者可經由電腦連線的映像管，利用「對話方式」直接進行公車路網的測試工作，可用於改進傳統的指派與評估模式。其缺點為執行太慢且所產生可測試之路網方案太少。

5. 數學尋優法

係利用數學規劃的方法求得最佳公車路網之解答。此法將問題公式化時，難免有過多的假設使得與實際狀況脫節，而且在實際應用時受到電腦容量之限制。

6. 啟發式解法

係利用符合邏輯、有條理的程序來產生和改進路網，並藉著改變路網設計變數，以增加求得「最佳解」之機會。規劃者可視實際的環境背景，以數學化的方式求解，為都市實際網路設計較為理想可行的方法。

2.3.2 公車路網規劃方法之數學法

數學法乃最常用來探討路網設計的規劃方法，其如上節所述分成數學尋優法與啟發式解法，茲將此二法簡述如下：

數學尋優法有較多之假設，較適用於簡單路網，因此，此類方法之研究對象大都侷限於平行公車路線或棋盤型路網[30]。Holroyd [13]學者於 1967 年推導旅次分布均勻之棋盤路網的最佳路線間距及班次，研究結果顯示有效步行時間成本、公車營運成本與等待時間成本皆相等時，可使系統總成本最小，並得最適之路線間距。Byrne 和 Vuchic [3]於 1972 年推導棋盤路網之平行路線的最佳間距及班次，結果顯示公車路線兩側之旅次需求相等時，可得最適之路線位置。Byrne [4]於 1976 年同樣針對平行路線進行探討，主要為推導速率不同之平行路線的最適位置、長度及班次，以使得系統總成本最小。然而，因公車營運具有目標多元化的特性，所以多年來一直無所謂的最佳公車路網設計方法。

啟發式解法利用符合邏輯、有條理的程序來產生和改進路網，雖無法保證產生最佳解，但較接近公車路網之實務應用，且可用合理之運算時間獲得良好解。Lampkin 與 Saalmans 認為良好的公車路網應具備以下幾點特性：(1)大多數的公車乘客皆不需轉車；(2)公車路線應儘量直捷，避免過度彎繞；(3)路線之設計應善於配合環境之改變；(4)路線總數不宜過多，並根據上述之目標提出一種公車路線設計方法，亦即先產生一條包含四個節點且路線二端皆有場站可用的基本路線，然後再依增點準則將點插入路線中，直至無法插入為止，即產生一條新路線。若待服務旅次仍很大時，則重複上述動作，產生另一新路線[16][30]。

Rea 學者於 1972 年提出樣板網路(Template Network)的觀念，將公車可能行駛之道

路構築為基本路網，然後依下列步驟進行路徑之取捨[25] [30]：

1. 對於樣板路網的各路段皆給予最高的公車服務水準(如最高之班次密度)。
2. 決定樣板路網上任意二旅次起訖點間之最短路徑(以候車時間與行車時間之和表示)。
3. 以容量限制指派法進行公車旅次量指派。
4. 檢視各路段之供給與需求（旅次指派結果）是否一致。
5. 若某路段之供需平衡，則維持原服務水準，否則降低其班次密度至剛好滿足現有需求者，然後回到步驟 2。當所有路段皆達供需平衡時，路線設計工作即停止。

Rea 學者的方法較適用於簡單的大眾運輸系統，對於複雜的公車路線問題，此種方法可能無法適用。

Hsu 和 Surti 於 1977 年建立一個路線選擇的架構，先將公車路網中的節點區分為活動型節點、社區型節點與轉運點；然後，將路網設計的程序分為數個發展階段，每一個階段皆包含擴線、運量預測及路網評估三個部份，且每階段僅發展一類公車路線，依序為活動路線、社區路線、走廊路線與運轉路線等[14] [30]。

藍武王與施孅孅[40]以路線所經地區須服務最多旅客為原則，並以旅次發生量最大之交通分區作為起點向鄰近各區作連續性之樹狀發展，提出掃描法、簡捷法與直接法三種方法分別針對二種循環與非循環路線進行探討。然而，此種方法以服務最多旅客數為原則設計路線，並未考量路線之直捷與否，易產生路線彎繞現象。

Marwah 等人於 1984 年針對 Ahmedabad 之公車路網設計方法，結合了啟發式解法和數學規劃模式，以預估之旅客期望行駛路線取代最短路徑為選線原則，透過最小營運成本與旅行時間成本之目標式，先尋找符合之公車路線，然後再藉由啟發式求解之方法，透過系統最大轉車次數節省數量之目標式，同時選出公車路線與班次數[21][39]。

周義華與林祥生於 1985 年利用啟發式解法發展出三類公車路線設計方法，其以基本路線設計法與直捷路線設計法來設計直線型公車路線，以環狀路線設計法發展環狀公車路線。此研究雖已將公車營運與路網特性納入考量，但對於轉車旅次之處理與到路容量之限制並未列入考慮，故實際應用上仍受限制[30][31]。

近幾年來，隨著巨集啟發式解法之發展，有許多學者都採用基因演算法來設計公車路網。1998 年 Pattnaik 、Mohan 和 Tom 三位學者[24]對市區公車路線進行設計，其設計方式分成二個階段進行，第一階段產生候選的路線組合，第二階段利用基因演算法選擇最佳的路線組合。2003 年 Tom 和 Mohan[26]二人同樣利用基因演算法針對大眾運輸路線的網路提出一個設計方法，不同的地方在於他們將路線的頻率視為一個變數納入考

量。Ngamchai 等人[22]於 2003 年考量轉運站的轉運需求，用基因演算法設計巴士的運輸網路。而 Chakroborty 和 Dwivedi[12]於 2002 年根則據基因演算法的原則發展一套程序去求解運輸系統路線的最佳解。

2.4 校車與交通車路線設計方法探討

2.4.1 國外校車路線規劃與排班

國外對於校車網路的設計大都是探討某一學區內各級學校校車行駛路線的規劃與班次之設計。問題之背景可以統整如下：在一個學區內有許多學校，每個學校上學與放學之時間是固定的，且每個學校都有一組所屬站牌。校車服務為政府提供，由政府統籌管理，政府於每個時段內指派車輛服務該時段內上學或放學之學校；因此，國外之校車路網設計主要是對每個時段整體路線進行規劃與排班，使得車輛可達到最大利用率，以減少總運送成本。

在校車網路設計中，資料之處理往往非常費時，1979 年 Bodin 與 Berman 二位學者[2]針對這個煩人的問題提出 ministop 的觀念，將每一個學生指派到離家最近的 mini-stop 上，以便快速確定每個學校所屬的站牌與指派到每個站牌之學生數。校車路線設計上，Bodin 與 Berman 以巴士總旅行時間最小為目標，採用先設計路線再分群的方法，利用 Lin[15]學者於 1965 年提出之 3-Opt 節線交換法，針對研究區域內之學校，先找到一條經過所有站牌之路線，再將此路線依據容量及時間限制條件分割成數條可行之路線。

Bowerman 等人[6]考量多目標之市區校車路線問題，將此問題分成三個子問題：校車停靠站位置之決定、指派學生到停靠站之問題，以及路線產生問題。他們用 allocation-routing-location (ARL)策略來求解這三個子問題，這個策略主要是先將學生分派到每個群組，使得每個群組可以被一條校車路線服務；接著停靠站之位置被選定且產生經過這些停靠站的校車路線。Bowerman 等人利用多目標的 Districting 演算法將學生指派至每個群組，然後結合集合涵蓋演算法和插入法選定停靠站位置和產生路線。

Braca [7]等人發展一套電腦計算系統 CATS (Computer Assisted Transportation System)來幫助校車的路線規劃；這套系統中，主要使用 Location Based Heuristic 方法同時處理行駛路線的規劃與班次之設計，且允許一輛車搭載不同學校之學生。其以 1992~1993 紐約 Manhattan 學區的學校為例，從 838 個停靠站服務 4619 個學生至 73 個學校，結果顯示使用的車輛比原本的車輛少。

Corberan 等學者[11]著重在發展一個方法可以同時考量運輸成本與運輸時間這二個相互衝突的目標，而不是以權重之方式將多目標轉換成單一目標。其採用分散搜尋法(Scatter Search)之架構，先以二種啟發式解法(H1 和 H2)產生起始解，第二個步驟以 SWAP 交換程序去找到每一條路線長度之局部最佳解，接著以 INSERT 交換程序去改善最大的旅行時間，最後結合上面三個步驟所產生的解集合，進而產生一條新的路線。雖然

Corberan 等人所提出的這個方法在執行上仍無法非常有效率，但這個方法可以同時考量運輸成本與運輸時間，產生不同的車輛數與最大旅行時間組合，決策者可以決定成本與服務品質如何折衷，然後再選擇最適合的路線。

Li 和 Fu 二位學者[20]建構了多目標之數學模式，其考量目標之優先順序為最小化總車輛數、最小化所有學生所花的總旅行時間、最小化車輛總旅行時間、平衡車輛間的裝載量和旅行時間。其解法結合了啟發式解法和最佳化方法，包括 Lawler 的第 K 條最短路徑演算法、Dijkstra's 最短路徑演算法、Hungarian 演算法，共分五個步驟求解，第一個步驟先找到滿足第一個目標（服務所有停靠點所需的最小車輛數）之最佳解，第二步驟是建構起始解，第三步驟改善起始解，第四步驟確認所有路線為最短路徑，最後用 Hungarian 演算法進行車輛指派，以最小化車輛空載的旅行時間。

2.4.2 國內校車與交通車路線規劃

國內的校車或交通車服務大多為各學校(機關)自行負責，有些是學校自營，有些則是委外經營。因此，國內之校車多屬於只服務單一學校，上學時只要將分散在不同地區之學生全部載往同一個目的地下車，放學時從同一個學校將學生載回上學時上車之停靠站即可，與上節所述之國外校車服務特性不同。

國內對於校車或交通車路線規劃之相關文獻甚少，陳文瑞[36]首先於 1989 年對交通車的網路設計進行研究，其以台大之交通車做為實例研究，跟據地理區位分成三群組，每個群組利用 VRP 模式以 LINDO 套裝程式求解各群組最佳路線組合。

陳建都[34]於 1996 年對校車路線指派問題進行研究，以中部某職業學校做為案例，利用修正之掃描法建構路線起始解，再以路線間節點交換來改善，且利用四個例題對掃描法和節省法進行比較，結果顯示平行掃描法與循序掃描法在執行績效上均較循序節省法佳。此外，陳建都亦探討車輛容量限制的放寬對車輛營運成本與乘客車上旅行時間成本的影響。

張靖、卓裕仁等人[38]於 2001 年將地理資訊系統(Geographic Information System, GIS)應用在校車路線與班次排程上，提出一個在 GIS 環境下規劃校車路線的方法，並應用在中華大學的校車路線與班次排程上。其採用集合涵蓋及最短路徑的觀念，將分區中人數最多的點，依序加入路線中，並限制路線長度與最大服務人數，產生路線停靠站組合，再以 ArcView Avenue 程式撰寫旅行推銷員(TSP)之程式，規劃出四條較佳的路線。最後，利用路線服務人數、成本效益與行駛時間三個指標來評估四條路線之優劣。

張容瑄[37]於 2001 年利用模擬退火法規劃一個校車指派系統，其問題定義為軟性時間窗車輛路線問題(VRPSTW)，每個需求點都有其時窗限制，提早到達或超過最晚到達時間都加上懲罰項，讓問題與實務上的情況相符，使業者在最適合的運送路線下，總運送距離最小，並達到準時的要求。

2.4.3 小結

國內外對於校車路線設計與規劃之研究所考量之目標與求解之方法各有不同，茲將國內外對於校車路線之設計與規劃彙整如表 2.1。

表 2.1　國內外校車路線之設計與規劃彙整

學者(年份)	考量目標	求解方法	敘述與結果分析
Bodin、Berman (1979)	車輛數最少	3-opt 交換法	Ministop 的觀念對於這類問題的資料庫建立非常有效率。
陳文瑞 (1989)	路線時間最短	啟發式解法	跟據地理區位分成三群組，每個群組利用 VRP 模式以 LINDO 套裝程式求解各群組最佳路線組合。
Bowerman 等人 (1995)	(1)車輛數最少 (2)路線長度最小 (3)承載量與路線長度平衡 (4)學生步行距離最少	Districting 演算法集合涵蓋演算法插入法	利用不同目標之權重產生不同的結果，結果顯示只要權重較高之目標，其最後該目標之結果較佳。
陳建都 (1996)	車輛總距離最小	掃描法、路線間節點交換	結果顯示平行掃描法與循序掃描法在執行績效上均較循序節省法佳。
Braca 等人 (1997)	車輛數最少	Location Based Heuristic	以紐約 Manhattan 學區為例，從 838 個停靠站服務 4619 個學生至 73 個學校，結果顯示使用的車輛比原本的車輛少。
張容瑄 (2001)	車輛總距離最小	模擬退火法	問題定義為軟性時間窗車輛路線問題，每個需求點都有其時窗限制。其結果顯示模擬退火法有機會接受較劣解，使得最後結果有時會失去最佳解。
Corberan 等人 (2002)	(1)車輛數最少 (2)載客路線時間最小	分散搜尋法 (Scatter Search)	執行上仍無法非常有效率，但可以同時考量運輸成本與運輸時間二個相互衝突的目標。
Li 和 Fu (2002)	(1)車輛數最少 (2)載客路線時間最小 (3)路線時間最小 (4)承載量與旅行時間平衡	結合啟發式解法和最佳化方法	以香港某幼稚園為例，54 個停靠站，86 個學生，結果顯示巴士總旅行時間減少 29%。

以往國、內外文獻多以學校交通車路線之研究為主，對企業員工交通車路線作研究並不多見，從文獻中得知員工交通車與學校交通車之性質雖相近，但仍有些微之不同，茲將員工通勤交通車與學校交通車之特性整理與比較如表 2.2。

表 2.2 通勤交通車與校車服務特性比較表

	通勤交通車	**校車**
服務對象	機關團體 / 企業員工	學生
起迄點數目	多對一 / 多對多	多對一/多對多
起迄型態	固定的起迄點	固定的起迄點
路線型態	單向（上班）	單向(上學)
發班頻率	單趟	單趟
時間限制	於上班前需抵達迄點	於學校規定時間前需抵達迄點
目標	多目標(服務/成本)	多目標(服務/成本)
需求分佈	分散	集中
車種	多車種	單車種/多車種
分割乘載 (split load)	不可	不可

通勤交通車與校車服務之起迄點數目雖皆屬多對一或多對多，但隨著我國工業區與科學園區之發展，往往在一個區域範圍內有多個工作廠區，故通勤交通車較偏向多個起點至多個迄點之情形；而文獻中國、內外校車服務之研究大多皆以一條路線經過一個學校為主(多對一起迄點)，只有 Braca [7]等人針對一條路線經過多個學校進行研究(多對多起迄點)。此外，校車服務之車種雖可為單車種或多車種，但國內、外校車服務之相關文獻大多只針對單車種校車路線進行研究。由上述可知，以往文獻中較少有研究針對多對多起迄點及多車種之特性進行研究，故本研究將針對通勤交通車多對多起迄點及多車種之特性，探討通勤交通車路線設計問題。