

第四章 啟發式演算法

針對圖書配送問題所構建之車輛排程問題為混合整數規劃問題(mixed integer programming)，隨著問題大型化(large scale)利用軟體 LINDO 使用的分支界限(branch and bound)求解方法越顯困難，因其限制式將依決策變數數目增加而呈指數增加，尤其是避免子迴路(subtours)產生的限制式，故大型 VRP 不易取得正確解，屬於 NP-hard 之問題。因此，本研究將尋找一啟發式求解方法，以期在可容忍的時間內取得一可行解。

禁忌搜尋演算法(Tabu Search)是一種高階的萬用啟發式方法，(Meta-Heuristic)，專門用來處理組合最佳化的問題。此方法透過彈性記憶體之應用，故常常能跳脫區域最佳解(Local Optimal)，且能在一合理的時間之內求得一近似(或最佳)解。

綜合上述，本研究提出以禁忌搜尋演算法(Tabu Search)來求解圖書配送之車輛排程問題。本章將於 4-1、4-2 節說明禁忌搜尋演算法概念及本研究演算法之程式設計，4-3 節依範例測試演算法之可行性，並於 4-4 節研提出本章之小結。

4-1 禁忌搜尋法則(Tabu Search)

禁忌搜尋法則為 Fred Glover 於 1977 年所提出，目前已經應用的領域[Glover,1990]有排程、通訊、字元辨識、巡迴推銷員問題(TSP)、二次指派問題(QAP)、圖形著色、積體電路設計、時間表設計與類神經網路等組合最佳化之問題，它使用一個非常有彈性的記憶體結構，

因此比起嚴苛的記憶體系統或無記憶體系統，在允許資訊的利用上要來的更具有彈性。

禁忌搜尋法可分為短期與長期二階段。在短期階段，利用禁忌限制式(Tabu Constraints)來限制搜尋的狀態，以避免搜尋的重複與反覆，而免禁準則(Aspiration Level)用來釋放禁忌限制，避免搜尋停滯不前。短期的重點在加速達到區域最佳化；在長期階段則使用加強性(Intensification)與多樣性(Diversification)將搜尋帶入新的區域以求得更佳的解，使得搜尋得以跳脫區域最佳解，進而搜尋總體最佳解。禁忌搜尋法主要由(1)移步；(2)禁忌名單；(3)免禁準則；(4)候選名單；(5)搜尋停止準則，五大模組所組成。以下針對此五大模組加以說明。

(1)移步(Move)

目前禁忌搜尋法最常使用的移步方式為K-OPT法所採用之K邊互換方式(K-edge Exchange)，即刪除原本不相鄰之K條節線，加入新的K條節線。通常K越大則最後所獲得之解品質亦較佳，但因K-OPT法之複雜度為 $O(n^k)$ ，K越大時，其所需耗費的時間也越久。一般而言，K為2或3時最有效率，為一般使用者所採用。

(2)禁忌名單(Tabu List)

用來記錄過去搜尋中每次發生移步(Move)時的屬性，為一個提供禁忌限制的記憶體結構。一般而言，禁忌名單越大則陷入區域最佳解的發生機率將越低，但所需的記憶體空間也將越大，且電腦每次所須的偵測時間將越長，相對之下，可提供移動的空間亦將縮小，這些現象將降低求解的效率。雖然如此，現階段並無一套固定方法來解決禁忌名單尺寸的大小，通常要根據問題本身的特性來決定之。Glover[1990]建議使用魔術數字7作為禁忌名單尺寸。Knox[1994]發

現在 TSP 問題裡，當城市數目小於 100 時，禁忌名單大小與城市數目成線性關係。Taillard[1991]以一隨機變動的禁忌名單尺寸來求解 QAP 問題，其變動的範圍為 $0.9n \sim 1.1n$ (n 為問題的大小)。

(3)免禁準則(Aspiration Level)

用來釋放一個被列為禁忌之移步，但此禁忌移步若被允許，將可獲得比目前最佳目標函數值還好的目標函數值。

(4)候選名單(Candidate List)

為所有合法移步屬性之集合，即所有可能的移步集合扣除所有禁忌移步集合再加上免禁準則集合所成的集合。

(5)搜尋停止準則(Stopping Criterion)

搜尋停止準則指的是用來終止搜尋進行的條件，較常見的大約有下列四種方式：(1)預設可允許之最大迭代次數；(2)預設目標函數值持續未改善之最大迭代次數；(3)預設 CPU 之最長允許處理時間；(4)預設可接受之目標函數值。一但搜尋達到這些預設值，則停止搜尋，當時最佳解即為最終解。大部份使用禁忌搜尋法的研究者所使用的終止條件為第(1)種，因為使用此種方法保證在一段迭次後會終止搜尋且不因所使用電腦系統不同而有所影響。

禁忌搜尋法之演算法[Glover, 1990,1989]可分為起始與搜尋二階段。起始階段主要目的在輸入禁忌參數、起始解、目標函數係數與搜尋終止條件等項目。而搜尋階段之目的在從一可行的起始解(x_0)開始，針對 x_0 之所有鄰近解(Neighborhood solution) $N(x_0)$ 進行評估，然後挑選出非禁忌(或符合免禁準則)之最佳鄰近解進行移步，此移步不一定要對解的品質有所增進，此為禁忌搜尋法與傳統只允許比目前最

佳解還佳的解進行移動之求解方法最大的不同點之一。例如在圖 4-1 之最小化問題中，我們假設起始解為 x_0 ，最佳鄰域解為 x_{local} ，由於在 x_{local} 之所有鄰域解 $N(x_{local})$ 中無法找出比目標函數值 $f(x_{local})$ 還佳的解，但禁忌搜尋法允許比目前最佳解還差的解進行移動，故有可能跳脫區域最佳解 x_{local} 而繼續搜尋的進行，甚至有可能達到整體最佳解 x_{global} 。又為預防移步的循環(即 $x_0 \rightarrow x_1, x_1 \rightarrow x_0$)，在每次移步之後必須更新禁忌名單之紀錄，其更新的方法為移去最舊的紀錄，加入新的移步資訊。

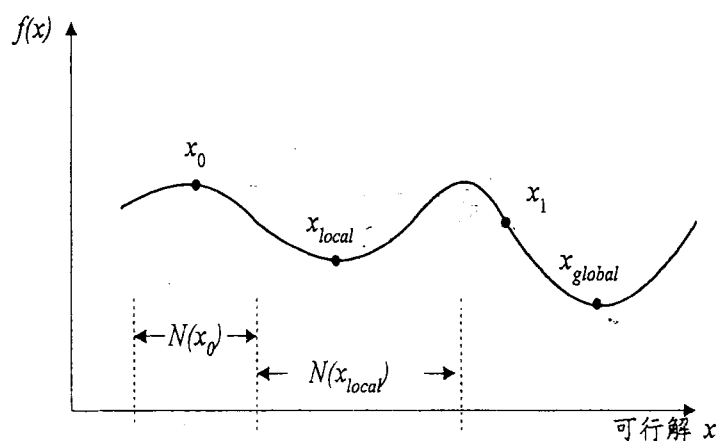


圖 4-1 禁忌搜尋法之基本原理

4-2 VRP 禁忌搜尋演算法

禁忌搜尋法為屬於改善式(Improvement)的啟發式解法，在求解動作開始前，必須先產生一組起始解。本研究所採用的演算法為兩階段法，第一階段為路線建構階段，第二階段為路線改善階段，以下分別就此兩階段演算法進行詳細之說明。4-2-1 節說明路線建構階段之起始解產生法，4-2-2 節則介紹路線改善階段之禁忌搜尋法。

4-2-1 路線建構階段

本研究以節省法(Savings Method)為 VRP 禁忌搜尋法之起始解產生方法，茲針對節省法之演算步驟說明如下：

步驟一：

輸入圖書倉儲中心(depot)與各書店(需求點)的相關資料，如書店數、圖書需求量、退書量、相對距離矩陣、車輛數量及其容量限制。

步驟二：

根據距離資料，求出兩兩節點合併後之成本節省量

$$S_{ij} = C_{oi} + C_{oj} - C_{ij}。$$

步驟三：

將成本節省量 S_{ij} 按大小排序，而後依照 S_{ij} 由大至小排序，根據連結原則與車輛容量限制式滿足與否，逐一構建路線。。

步驟四：

重複步驟三，直到所有 S_{ij} 均已考慮完畢。

步驟五：

將尚未與其他路線連結或合併之書店，直接與圖書倉儲中心連結成單一路線。

步驟六：

分別計算出每條路線之運輸變動成本，並加總算出總運輸成本。

步驟七：

輸出排程路線結果與成本資料。

4-2-2 路線改善階段

路線改善階段是以路線建構階段之起始解為輸入，以禁忌搜尋法為演算法的骨幹，並利用插入法、路線間節點交換法與路線內節點交換法為改善之依據；輸出則為一個較佳之車輛路線。而此禁忌搜尋演算法之運作架構乃參考 Osman[1993]求解典型車輛路線問題所使用的求解架構修改而成。以下介紹本演算法所使用之移步及如何評估一個移步，以作為選取時之依據。

一、移步之方式

移步可表示成一個函數 $M(R, s)$ ，其中 R 表示目前的路線解， s 表示移步屬性，這個函數的意義為：當路線 R 發生移步 s 時，會將路線 R 轉變為 R' ，也就是 $R' = M(R, s)$ 。

本研究使用插入法或節點交換法來作為路線間、路線內的移步，移步之屬性可分為三種：插入法、路線間節點交換法與路線內節點交換法，此三種移步之屬性可一般化成 $s = (d_i, R_i, d_j, R_j)$ ， d_i, d_j 代表需求點， R_i, R_j 代表路線解 R 中車輛 i, j 所行駛的路線，為路線編號，其中 d_i 為路線 R_i 內之一需求點或 0， d_j 為路線 R_j 內之一需求點或 0，而 0 代表一個虛擬之需求點，以下介紹這三種移步：

(1) 插入法移步(屬性 s_i) $(1-0)$ exchange

此移步屬性發生時就是將一個需求點 d_i 由第 i 部車所行駛的路線 R_i 中，重新插入第 j 部車所行駛的路線 R_j ，以 $s_i = (d_i, R_i, 0, R_j)$ 表示之，其中 d_i 為路線 R_i 內之一需求點。使用此法在路線間之改善方面除了降低路線距離之外，同時可以達到改變車輛的使用數。

(2) 路線間節點交換移步(屬性 s_2) (1-1) exchange

此移步為將第 i 部車所行駛的路線 R_i 中的需求點 d_i 與第 j 部車所行駛的路線 R_j 中的需求點 d_j 相互對調所形成之路線之改變，以 $s_2 = (d_i, R_i, d_j, R_j)$ 表示之， d_i 與 d_j 均非虛擬之需求點。

(3) 路線內節點交換移步(屬性 s_3) (swap)

此移步為將第 i 部車所行駛的路線 R_i 中的需求點 d_i 與需求點 d_j 相互對調所形成之路線之改變，以 $s_3 = (d_i, R_i, d_j, R_i)$ 表示之， d_i 與 d_j 均非虛擬之需求點。

不相鄰

Reverse

二、移步之評估

移步評估值為移步選擇的依據，而移步評估值是由下列兩個元素組成：節省值與處罰值，我們將移步之評估值表示成 $m(s) = x(s) + q(s)$ ，其中 $x(s)$ 表示移步之節省部份， $q(s)$ 表示處罰值部份，以下將從這兩方面來探討移步評估值。

(1) 節省值

節省值是指移步所造成之路線總成本的差值，以 $x(s)$ 表示，且 $x(s) = \text{Max}\{x(s_1), x(s_2), x(s_3)\}$ ， $x(s_1)$ 表示插入法之節省值， $x(s_2)$ 表示路線間節點交換法之節省值， $x(s_3)$ 表示路線內節點交換法之節省值。

$(d_i, R_i, 0, R_j)$ (d_i, R_i, d_j, R_j) (d_i, R_i, d_j, R_i) R 是符號相同

(a) 插入法之節省值，以 $x(s_1)$ 表示，如圖 4-2 所示，此圖說明了一個

$s_1 = (d_i, R_i, 0, R_j)$ 的移步。其意義為選擇第 i 部車所行駛的路線 R_i 上的一個需求點 d_i ，並選擇第 j 部車所行駛的路線 R_j 上的兩個相鄰需求點 d_a, d_b 間進行插入，(4.1) 式為插入法之節省值的計算。

$$x(s_1) = c(d_a, d_b) + c(d_{ip}, d_i) + c(d_i, d_{is}) - c(d_a, d_i) - c(d_i, d_b) - c(d_{ip}, d_{is}) \quad (4.1)$$

$c(d_a, d_i)$ 表示 d_a, d_i 兩點間之直線距離， d_{ip} 表示 d_i 在路線 R_i 上的前一個被服務的需求點， d_{is} 表示 d_i 在路線 R_i 上的後一個被服務的需求點。

(b) 路線間節點交換法之節省值，以 $x(s_2)$ 表示，如圖 4-3 所示，此圖說明了一個 $s_2 = (d_i, R_i, d_j, R_j)$ 的移步。其意義為選擇路線 R_i 上的一個需求點 d_i ，並選擇路線 R_j 上的一個需求點 d_j ，然後進行交換，移步節省值之計算，如(4.2)式。

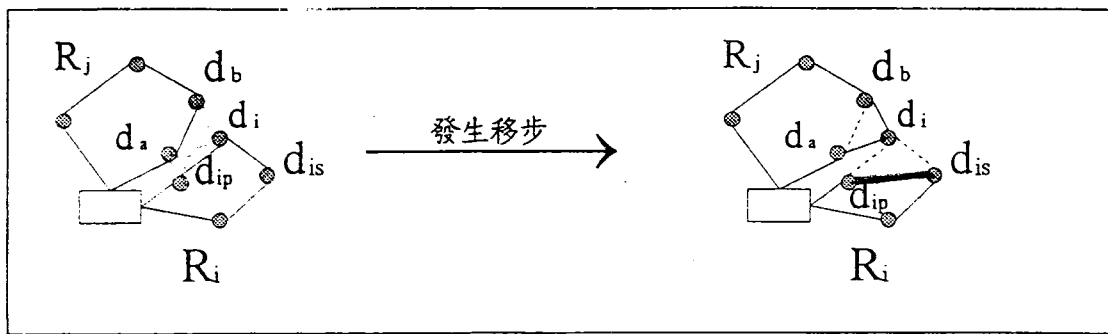
$$x(s_2) = c(g, d_i) + c(d_i, h) + c(e, d_j) + c(d_j, f) - c(g, d_j) - c(d_j, h) - c(e, d_i) - c(d_i, f) \quad (4.2)$$

需求點 g, h 與 d_i 在路線 R_i 之前後順序，與需求點 e, f 與 d_j 之前後順序，如圖 4-3 所示。

(c) 路線內節點交換法之節省值，以 $x(s_3)$ 表示，如圖 4-4 所示，此圖說明了一個 $s_3 = (d_i, R_i, d_j, R_i)$ 的移步。其意義為選擇路線 R_i 上的一個需求點 d_i 與另一個需求點 d_j ，然後進行交換，移步節省值之計算，如(4.3)式。

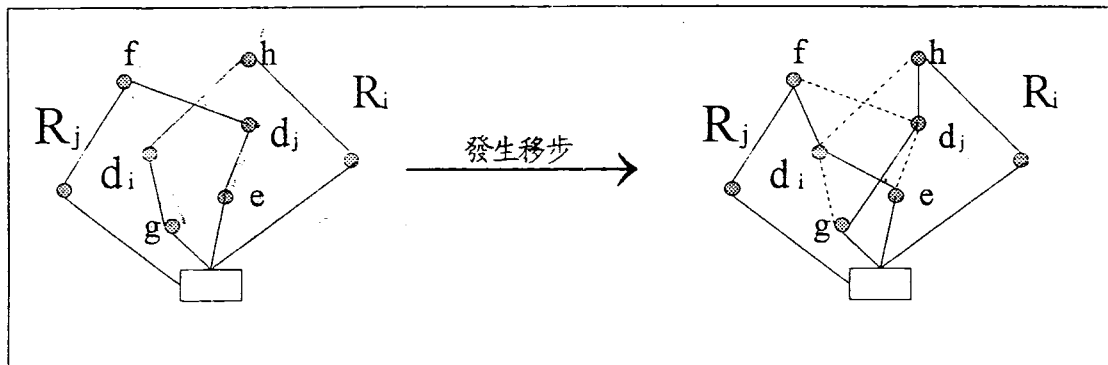
$$x(s_3) = c(g, d_i) + c(d_j, 0) - c(g, d_j) - c(d_i, 0) \quad (4.3)$$

需求點 g, h, d_i 與 d_j 在路線 R_i 之前後順序，如圖 4-4 所示。



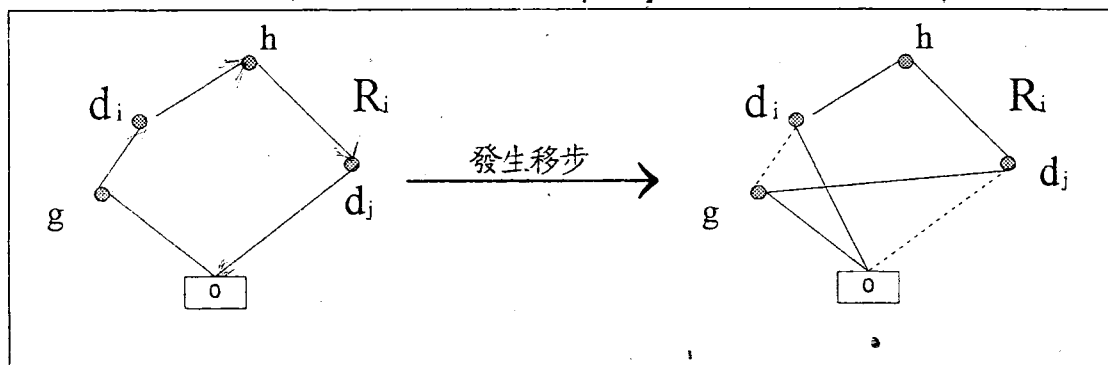
(1-0)

圖 4-2 插入法節省值之計算



(1-1)

圖 4-3 路線間節點交換節省值之計算



(swap)

圖 4-4 路線內節點交換節省值之計算

(2) 處罰值

在建構起始解的階段，為了速度的要求，可能違反車輛容量的限制。因此當路線出現違反容量限制時，加入處罰值可適當地表達在此情況下一個解真正之目標函數值。當違反車輛容量的現象發生變化時，移步中處罰值的差值就可用來驅動非可行解，使之逐漸符合車輛容量限制。

我們令 $s = (d_i, R_i, d_j, R_j)$ ，目前的解為 R ， $R' = M(R, s)$ ， R' 為發

生移步後的解， $R' \in \tilde{R}$ ， R 與 R' 的差異為 R_i 變為 R'_i 、 R_j 變為 R'_j ，因此，處罰值的變化只會發生在 R'_i 與 R'_j 上，(4.4)式表示處罰值的設定。

$$q(s) = \alpha [\max(0, q(R_i) - w(R_i)) + \max(0, q(R_j) - w(R_j))] - \alpha [\max(0, q(R'_i) - w(R'_i)) + \max(0, q(R'_j) - w(R'_j))] \quad (4.4)$$

因此當一個移步造成一條路線的總需求違反車輛裝載容量限制發生變化時， $q(s)$ 的加入才可以代表移步真正的評估值。其中 $w(R_i)$ 表示服務路線 R_i 之車輛容量，而 $q(R_i)$ 表示路線 R_i 內之總需求， α 為一處罰值係數，以避免所獲得之解出現非可行解現象。

三、移步之記錄

令移步屬性 $s = (d_i, R_i, d_j, R_j)$ ，當此移步發生時，需求點 d_i 會從 R_i 重新分派至 R_j ，需求點 d_j 會從 R_j 重新分派至 R_i 。因此，若希望在未來的一段時間，避免 d_i 回到 R_i 、 d_j 回到 R_j ，而這段時間也就是禁忌名單大小(Tabu List Size)或是保留期限。

在禁忌名單之資料結構設計上，可以一 $n \times n$ 的二維陣列來儲存相關資訊， n 表示書店總數，故共有 $n \times n$ 個儲存格(cell)。每個儲存格的記憶體位置可以 $Tabu(d_i, d_j)$ 表示，意指加入節線 (d_i, d_j) 為一禁忌的移步，而每個記憶體位置存有一個元素 t_size ， t_size 記錄著節線 (d_i, d_j) 仍為禁忌的期間長度。

例如當插入法移步發生時，即刪除節線 $(d_a, d_b), (d_{ip}, d_i), (d_i, d_{is})$ ，加入節線 $(d_a, d_i), (d_i, d_b), (d_{ip}, d_{is})$ ，記錄為

$$Tabu(d_a, d_b) \cdot t_size = tls$$

SWAP

$$Tabu(d_{ip}, d_i) \cdot t_size = tls$$

$$Tabu(d_i, d_{is}) \cdot t_size = tls$$

其中 tls 為禁忌名單大小，因此在未來移步期間內，加入節線
 $(d_a, d_b), (d_{ip}, d_i), (d_i, d_{is})$ 是一個禁忌的移步。

四、免禁準則

(4.5)式將符合免禁準則之禁忌移步以集合的方式來表示，

$$A(R) = \{s \mid s \in T(R), z(M(R, s)) < Z_{best}, R \& M(R, s) \in \tilde{R}\} \quad (4.5)$$

$A(R)$ 表示本迭次路線解 R 符合免禁準則的禁忌移步集合， R 為本迭次的路線解， s 為移步屬性， $T(R)$ 代表由 R 轉移至鄰近解之禁忌移步的集合， Z_{best} 為到目前迭次為止所搜尋到的最好的目標函數值， $M(R, s)$ 表示路線解 R 發生移步 s 後的解， $z(M(R, s))$ 表示路線解 R 發生移步後解的目標函數值， $\tilde{R}$ 表示所有可行解的集合。

五、候選名單之設計

所有候選名單的集合可以(4.6)式來表示：

$$Cset(R) = S(R) - T(R) + A(R) \quad (4.6)$$

其中 R 表示本迭次的路線解， $S(R)$ 表示本迭次可由 R 轉移至鄰近解 $N(R)$ 之所有集合， $A(R)$ 為符合禁忌準則的集合。

六、搜尋停止準則

由於希望本研究所提出之演算法在不同的系統所求得的解，不會在解的品質上有所差異，因此採用預設可允許之最大迭代次數為搜尋停止準則並以 $Maxi$ 來表示。