目　錄

圖目錄

第一章 緒論

1.1 研究動機與目的

隨著產品的生命週期縮短與時間競爭的壓力之下，導致供應鏈管理的重要越來越受到重視。也由於市場競爭愈趨激烈，廠商為了能夠迅速回應顧客的需要並滿足顧客所指定的服務，已成為物流配送業者積極欲達成的目標。物流業者要在指定時效內完成服務顧客的需求，尋找出最經濟的配送顧客的順序，可直接降低運輸成本，提昇企業競爭優勢。因此在滿足顧客需求以及不違反車輛最大載量的限制前題下，若能以最少的成本支出來滿足顧客需求及時效限制，必能有效提昇競爭力。而這類相關的研究即學術上經常提及的車輛路線問題(Vehicle Routing Problem, VRP)。

傳統車輛路線問題是延伸組合旅行推銷員的最佳化問題(Traveling Salesman Problem, TSP)，即在一個場站下，對數量眾多的顧客需求點進行純粹送貨或收貨的作業，並尋找最短總路徑距離。在這個問題中，各顧客點之需求量不可超過車容量限制，每輛車均由場站(depot)出發，服務完所指定需求點後再回到場站，且每個需求點只能由一輛車來服務。詳見**圖1-1**。車輛路線問題即在下圖網路中，給定具有容量限制的配送車隊，然後決定出一組具有最小總成本的配送路線（自場站出發，最後回到場站）。其中每條路線由一輛車行駛，各車所服務的顧客需求總和不得超過車輛容量；每位顧客皆須被服務，而且僅能被一輛車服務一次。

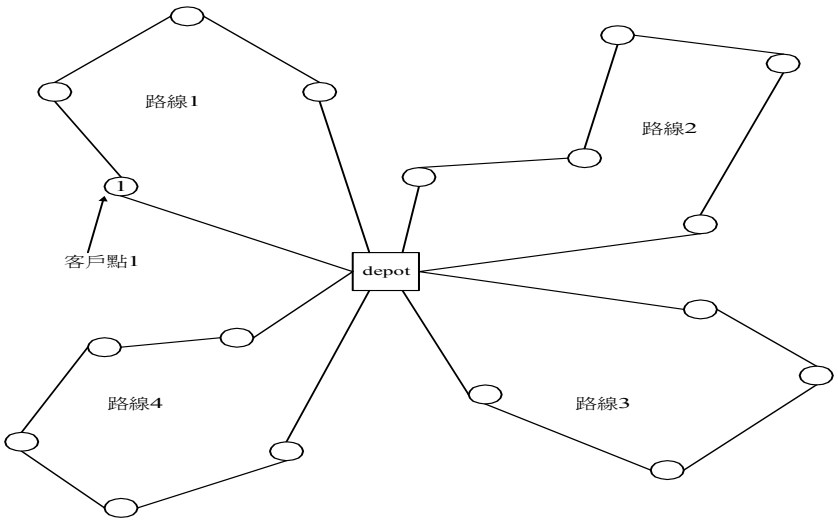

圖 1-1 VRP 示意圖

車輛路線問題相關問題眾多，由於考量到實際情況，後來基於車輛路線問題(VRP)，衍生出不少類型的問題，都是在實務應用上重要的車輛路線問題類型。

以下列出幾種常見的類型：

- 具時間窗限制的車輛路線問題(Vehicle Routing Problem with Time Windows, VRPTW）
- 多場站的車輛路線問題（Multiple Depot Vehicle Routing Problem, MDVRP）
- 多車種的車輛路線問題（Fleet Size and Mix Vehicle Routing Problem, FSMVRP）
- 週期性車輛路線問題（Periodic Vehicle Routing Problem, PVRP）

近半世紀以來，在車輛路線問題的課題上，國內外有許多相關文獻在這個領域裡有豐富的研究成果。此類問題本身具有高度的求解複雜性，是屬於NP-hard的問題類型，隨著顧客點增加，路線的組合數也成指數上升，造成解題的難度提高，所需要之求解演算時間隨問題規模變大而呈指數關係成長，使得問題無法在具效率的時間下求得精確解（exact solution）。

Laporte and Semet(1998)說明未來在車輛路線問題方面的研究，將朝向具有演算策略更簡單、計算量更精簡、可求解更大規模的問題及適用於任何問題結構等特性之演算法來發展，即使需要犧牲某種程度的求解精確度。所以當問題規模擴大時，為了確保在可接受的時間內可以得到精確度高的近似解（approximate solution），發展解題上具效率性的啟發式解法（heuristics）就成為研究車輛路線問題相關課題的重點。故本研究即以此為目的，發展一演算法來求解較大規模車輛路線問題的近似最佳解。

本研究之研究對象為具車容量限制之車輛路線問題(CVRP)—最基礎的車輛路線問題類型，此類問題考慮以下限制：

- 單一場站；
- 單一車種；
- 以最小路線成本為目標；

- 固定節線成本與固定客戶點需求；
- 有車輛容量限制；

1.2 研究方法與流程

首先將過去有關車輛路線問題相關的文獻，進行探討，並作為本研究之參考。本研究方法主要為發展一啟發式解法來求解車輛路線問題，提出一數學規劃模式，其中參考集合涵蓋問題(Set Covering Problem, SCP)的觀念，並以拉氏鬆弛演算法(Lagrangian Relaxation)、變數產生法(Column Generation)、次梯度法(Sub-gradient Optimization)等技巧來分析並求解問題。

主要內容與進行流程簡述如下：

1. 文獻回顧：蒐集國內外文獻中已發表之 VRP 等相關研究，以及近來求解集合涵蓋問題之發展情況。
2. 啟發式解法之解題架構設計：提出數學規劃模式，發展為集合涵蓋問題，並進一步分析並求解問題。
3. 解題程式撰寫：將演算法寫成程式，測試演算法績效之可行性。
4. 測試實驗之設計：以設計之演算法測試 VRP 例題，以了解演算法架構之可行性及效益。
5. 結論與分析：根據綜合分析所得到之結果，提出具體結論與建議，並研擬未來相關領域之研究方向。

第二章 文獻回顧

2.1 車輛路線問題(VRP)之定義

車輛路線問題(VRP)中最基礎的問題即旅行推銷員問題(TSP)，差別在於VRP在於TSP外多考量了多路線與車容量的限制。以下為VRP的數學規劃模式(Golden, 1977)：

目標式：

$$Min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{v=1}^{NV} C_{ij} x_{ij}^{v} \tag{2-1}$$

限制式：

$$\sum_{i=1}^{N}\sum_{k=1}^{NV} x_{ij}^{k} = 1 \quad j = 1,\cdots,N \tag{2-2}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{NV} x_{ij}^{k} = 1 \quad i = 1,\cdots,N \tag{2-3}$$

$$\sum_{i=0}^{N} x_{ih}^{k} - \sum_{j=0}^{N} x_{hi}^{k} = 0 \quad h = 0,\cdots,N;\ \ k = 1,\cdots,NV \tag{2-4}$$

$$\sum_{i=0}^{N} d_i \left(\sum_{j=0}^{N} x_{ij}^{k}\right) \le K \quad k = 1,\cdots,NV \tag{2-5}$$

$$\sum_{j=1}^{N} x_{0j}^{k} \le 1 \quad k = 1,\cdots,NV \tag{2-6}$$

$$\sum_{i=1}^{N} x_{i0}^{k} \le 1 \quad k = 1,\cdots,NV \tag{2-7}$$

$$X = \{x_{ij}\} \in S \quad \left(x_{ij} = \sum_{k=1}^{NV} x_{ij}^{k}\right) \tag{2-8}$$

$$x_{ij}^{k} = \text{binary} \tag{2-9}$$

其中

C_{ij}：顧客i至顧客j的路段成本

x_{ij}^{k}：車輛k由顧客i至顧客j行駛的路段

d_i：顧客i的需求量

K：車容量限制

N：顧客數

NV：車輛數

式(2-1)表示以求取總運輸成本為目標；式(2-2)與式(2-3)表示限制每一位顧客只能被一輛車所服務；式(2-4)表示總流量守恆限制；式(2-5)表示該車輛服務的顧客總需求不得超過車容量限制;式(2-6)與式(2-7)表示並非所有的車輛皆需被使用;式(2-8)為避免子迴路數目之限制；式(2-9)表示x_{ij}^k為二元決策變數。

2.2 以數學規劃為基礎之 VRP 啟發式解法

根據 Fisher(1995)的分析指出，求解車輛路線與規劃問題之演算法可以分成三個階段。第一階段是從 1960 年代到 1970 年代，屬於簡單啟發式方式，包括有各種局部改善之啟發式法和貪婪法則(Greedy Heuristics)，如節省法，交換法，掃瞄法等；第二階段是從 1970 年代到 1980 年代，屬於一種以數學規劃為主的啟發式解法，將車輛路線與規劃問題放鬆(relax)成較為簡單的數學規劃模式加以求解，如分支定限法，拉氏鬆弛法，切割平面法，變數產生法等；第三階段是從 1990 開始至今，屬於較新發展的方法，包括利用嚴謹啟發式方法、人工智慧法及最佳化演算法，如禁忌搜尋法，模擬退法火，基因演算法，大洪水法等，成效頗佳。其他相關 VRP 啟發式解法的發展可參考 Laporte et al.(2000)或 Cordeau et al.(2002)。

第一類簡單啟發式解法，即設計簡單易執行的法則，一般而言，先分群而後建構路線。舉 Clarke and Wright(1964)所提出的節省法為例，即指合併兩節點後所能減少的路線成本。

1. 首先計算出所有節點(i, j)的節省成本值：$S_{ij} = C_{io} + C_{oj} - C_{ij}$，按照節省值由大至小排列，然後選擇序列中節省值最大之節點構建新路線，並依序檢查下一組節點合併到路線端點之可行性（如車輛容量限制），若可行則合

併，否則略過該點。

2. 若序列中尚有節點未檢視完畢，則繼續檢查下一節點。若所有節點皆有車輛服務，則停止；若尚有節點未被服務，則由未被服務的節點中選擇節省值最大者產生新的路線，再檢視節省值序列中之節點合併到新路線之可行性。

3. 重複上述步驟直到所有節點皆被車輛服務為止。此類簡單啟發式解法優點為運算架構簡單，缺點為求解精確度不高且極易陷入局部最佳解的狀況。

至於近期較新發展的第三類人工智慧啟發式解法則是主要在改變傳統的局部搜尋法，建立通用性搜尋（generic search）的演算法，其中這些演算法普遍為利用節省法建立一初始解，再予以改善，且具有可跳脫局部最佳解的機制，但在運算過程中仍需要借助傳統的搜尋方法。舉普遍公認求解品質不錯的禁制搜尋法（Tabu Search, TS）為例，如 Taillard(1993)，其演算法之設計主要有以下幾個觀念與技巧：

1. 首先是以每一個顧客使用一輛車單趟來回的方式建立一初始解，再由改善初始解來找尋鄰近解(neighborhood)來做為搜尋(move)的準則。一般鄰近解的搜尋的方式，是在符合不超過車容量限制的前提下，於初始解內之路線 A 當中找出 μ 個欲作為交換的顧客，與路線 B 當中找出 π 個欲作為交換的顧客，其中 $0 \le \mu \le \min(M,|A|), 0 \le \pi \le \min(P,|B|)$。例如其中參數 (M,P) 設為 $(1,0)$ 意為將路線 A 中的1個顧客交換至 B 路線中，若 $(M,P)=(1,1)$ 意為將路線 A 中的1個顧客與 B 路線中的1個顧客互相交換。

2. 構建一記憶機制，在搜尋鄰近解的過程中，將已經搜尋過之解記錄在禁制名單（tabu list）中，以避免重複性或毫無目的的搜尋。禁制搜尋法主要的運算即在於禁制名單的運作。禁制名單是存放之前搜尋鄰近解時，利用交換法或插入法所已經搜尋過的路徑，當再尋找下一個鄰近解時，只利用不在禁制名單中的路徑進行交換或插入，以期能找到比上一解更佳的解。

3. 為了避免鄰近場站的顧客搜尋的頻率過高，而在搜尋顧客的頻率上乘上一懲罰值(penalty)，使搜尋的方式多元化。

4. 由於演算法架構設計的關係，在停止條件方面通常設計成在可接受的時間

內得到求解，停止條件通常會預設運算次數(iteration)、目標值持續未改善次數、允許求解之最長CPU時間、或目標值於容忍誤差內當成終止運算的條件。

由於第一類簡單啟發式解法的求解品質較差，而第三類的人工智慧演算法的求解過程又稍過於繁瑣複雜，為了企圖在這兩類演算法中找尋一個平衡點，故本研究方法則是選擇採用類似第二類以數學規劃為基礎所發展的啟發式解法。以現已有發展良好之集合涵蓋模式與拉氏放鬆法為基礎之演算法，針對基礎的車輛路線問題，提出數學規劃模式，以啟發式解法等求解並分析問題。

數學規劃為基礎的啟發式解法通常是將求解問題轉換成數學模式，放鬆限制後再進一步予以求解。以下介紹三種較為普遍且通常優於簡單式啟發式解法之模式。第一種為一般性指派問題(generalized assignment problem, GAP)，第二種為集合分割問題(set partitioning problem, SPP)，第三種為集合涵蓋問題(set covering problem, SCP)。

A. 一般性指派問題

Fisher and Jaikumar(1981)以一般性指派問題求解車輛路線問題，顧客被分配到對應的車輛，然後再針對顧客作 TSP 問題求解，為了激發一般性指派問題概似最佳解，第一個要注意的是車輛路線問題可用以下「非線性」一般性指派問題表示，定義：

$$y_{ik}=\begin{cases}1, \text{如果節點} i(\text{顧客} i \text{或是depot})\text{被車輛} k \text{服務到}\\ 0, \text{否則為0}\end{cases}$$

及 $\mathbf{y}_k=(y_{0k},...,y_{nk})$，車輛路線問題可定義為

目標式：

$$Min\sum_k f(\mathbf{y}_k) \tag{2-10}$$

限制式：

$$\sum_i a_i y_{ik} \le b,\quad k=1,...,K \tag{2-11}$$

$$\sum_k y_{ik} = \begin{cases} K, & i = 0 \\ 1, & i = 1,...,n \end{cases} \tag{2-12}$$

$$y_{ik} = binary, \quad i = 0,...,n; \quad k = 1,...,K \tag{2-13}$$

$f(\mathbf{y}_k)$是最佳化推銷員問題之旅行成本，定義：

$$x_{ijk} = \begin{cases} 1, \text{如果車輛} k \text{直接由節點} i \text{到達點} j \\ 0, \text{否則為} 0 \end{cases}$$

所以方程式 $f(\mathbf{y}_k)$就可以表示為

其中 $f(\mathbf{y}_k) = Min\sum_{ij} c_{ij} x_{ijk}$ (2-14)

限制式：

$$\sum_i x_{ijk} = y_{jk}, \quad j = 0,...,n; \quad k = 1,...,K \tag{2-15}$$

$$\sum_j x_{ijk} = y_{ik}, \quad i = 0,..,n; \quad k = 1,...,K \tag{2-16}$$

$$\sum_{ij \in S} x_{ijk} \le |S| - 1, \quad S \subseteq N(y_k), \quad 2 \le |S| \le n \tag{2-17}$$

$$x_{ijk} = binary, \quad i = 0,..,n; \quad j = 0,...,n; \quad k = 1,...,K \tag{2-18}$$

當然，這個定義在我們計算上並無幫助，既然我們缺乏式(2-14)之 $f(\mathbf{y}_k)$的封閉表(closed form)，一般性指派問題的啟發式解法通常將 $f(\mathbf{y}_k)$用一個線性的趨近式表示：$\sum_i d_{ik} y_{ik}$ ，加以求解此一般性指派問題而把顧客指派到相對的車輛上。為了得到線性趨近式的解，首先會指派 K 個種子(seed)顧客 $i_1 \cdots i_k$ 到每台車輛上，假設顧客 d_{ik} 是被指派到每台車輛 k ， $k = 1,\cdots,K$ ，接著設定係數 d_{ik} 為插入顧客 i 到車輛 k 由 depot 到顧客 i_k 再回去的路程。值得注意的是，$d_{ik} = c_{oi} + c_{iik} + c_{oiik}$ ，明顯可見種子顧客決定了一開始車輛會走的方向，一般性指派問題一直持續指派顧客到此一開始的路網上。

B. 集合分割問題

集合分割啟發式解法由產生一些候選車輛路線解開始，一個候選車輛路線解可以被定義成一個集合 $S \subseteq \{1,...,n\}$，其中顧客都被相同的車輛運送，把這些候選車輛路線解指標為 j ，並定義以下參數：

c_j = 候選車輛路線 j 的成本

$$a_{ij}=\begin{cases}1,\text{如果顧客}i\text{有被涵蓋在車輛路線}j\text{裡}\\0,\text{否則為零}\end{cases}$$

$J=$所有產生的車輛數目

$$y_j=\begin{cases}1,\text{如果候選車輛路線}j\text{被選用}\\0,\text{否則為零}\end{cases}$$

車輛路線問題就可以概似用以下集合分割問題表示：

目標式：

$$Min\sum_{j=1}^{J}c_j y_j \tag{2-19}$$

限制式：

$$\sum_{j=1}^{J}\sum_{k=1}^{K}a_{ij}y_j=1,\ i=1,...,n \tag{2-20}$$

$$y_j=binary,\quad j=1,...,J \tag{2-21}$$

目標式(2-19)設定極小化總運輸成本之目標，限制式(2-20)為所有顧客皆需要被涵蓋的限制式。限制式(2-21)中所使用的是等號，因此限制顧客只可被涵蓋一次，此即為集合分割問題。限制式(2-21)表示所有 y_i 為 0-1 的整數變數。

C. 集合涵蓋問題

SCP 數學模式如下：

目標式：

$$Min\sum_{j\in N}c_j x_j \tag{2-22}$$

限制式：

$$\sum_{j\in N}a_{ij}x_j\geq 1,\ \ i\in M \tag{2-23}$$

$$x_j\in\{0,1\},\quad j\in N \tag{2-24}$$

SCP 數學模式類似於前述所提之 SPP 數學模式(式(2-19)~式(2-21))，目標式(2-22)類似式(2-19)設定極小化總成本之目標，限制式(2-23)為所有顧客皆需要被涵蓋的次數，值得注意的是式(2-23)中是一個≥不等式，因此顧客可被涵蓋一次以上，是標準

的集合涵蓋問題，如果此不等式為等號，即所謂的集合分割問題(Set Partitioning Problem)之限制式(2-20)，限制顧客只可被涵蓋一次，如果改成≤不等式，則是所謂的集合裝運問題(set packing problem)，限制式(2-24)表示所有 x_i 為 0-1 整數變數。

Agarwal et al.(1989)指出任何一組 SPP 的可行解也為該問題之 SCP 的可行解，任何一組 SPP 的最佳解也為該問題之 SCP 的最佳解。由於限制不同的關係使求解 SCP 比求解 SPP 來的容易，而兩者差別在於 SCP 的可行解相對於該 SPP 解中，有一個以上的顧客點被涵蓋在超過一組路徑內，可透過刪除 SCP 中重覆涵蓋的顧客得到一組成本較低的 SPP 解。本研究則是選擇以求解較為容易的 SCP 模式，進一步修正為 SPP 模式做為本研究求解的架構依據。

2.2 集合涵蓋模式之啟發式解法

SCP 是一個 NP-hard 的問題，而且有許多實務上的應用，很多演算法都已經被建議在尋找此種問題確切解的文獻，這些演算法可以解決多至上百行以及上千列的例子，如 Cullen et al.(1981)。

集合涵蓋問題常使用上下限夾擠的方法來尋找精確解(exact)，目前發展的啟發式解法皆可得到不錯的上限值，而通常問題的下限值是藉由放鬆限制式而求解來的。文獻中常使用分支定限法(Branch and Bound)求解放鬆集合涵蓋模式之整數限制後的整數規劃問題，如 Balas and Carrera(1996)；或是使用拉格蘭式放鬆法(Lagrangian Relaxation)，可行解集合依舊保有二元變數的可行解性質，但是會有其他限制式被放鬆並移至目標式去，再利用次梯度最佳化法(Sub-gradient Optimization)逐步修正拉氏乘數並逼近問題最佳解，如 Beasley(1990)及 Caprara et al.(1999)。

應用拉氏放鬆法配合次梯度法求解集合涵蓋問題可以找到問題的最佳解，但是在求解較大規模問題，由於將車輛路線問題轉成集合涵蓋問題時不可能窮舉出所有的求解集合(如式(2-15)之 $x_j, j \in N$)，因此許多研究遂採用可以隱含考慮所有可能變數集合的變數產生法(Column Generation, CG)來求解。

變數產生法(Column Generation)或是稱為 Dantzig-Wolfe Decomposition，為線性放鬆法，主要是利用線性規劃中的對偶理論(Dual Theory)來產生變數(column)，以避

免浪費不必要的時間去窮舉不可行或對於求解問題沒有貢獻的變數。在應用變數產生法求解集合涵蓋問題時，通常將求解過程分為主問題與子問題兩部份。主問題為原 VRP 轉換成 SCP 模式之後，再放鬆整數限制，如：

目標式：

$$Min\sum_{r\in R} c_r x_r \tag{2-25}$$

限制式：

$$\sum_{r\in R} a_{ir} x_r \geq 1,\ \ i \in N \tag{2-26}$$

$$x_r \geq 0 \quad r \in R \tag{2-27}$$

R 是涵蓋所有顧客集合 N 下的所有可能路線組合，在運算中不可能被窮舉，亦即無法對應 $r \in R$ 考慮所有 a_{ir} ，及計算對應的路線成本 c_r 。

若針對限制式(2-26)定義之對偶變數(dual variables) $\boldsymbol{\pi}$ ，則可獲得此問題之對偶問題(dual problem)如下：

目標式：

$$\text{Max}\sum_{i=1}^{N} \pi_i \tag{2-28}$$

限制式：

$$\sum_{i=1}^{N} a_{ri}\pi_i \leq c_r \quad \forall r \in R \tag{2-29}$$

$$\pi_i \geq 0 \tag{2-30}$$

根據簡捷法(Simplex Method)之對偶可行性(dual feasibility)可知：針對尚未考慮到的變數(remaining columns) $a_{ir}(i = 1,\cdots,N),\ r \in R$ ，若是可以從設計的子問題中發現最小的減少成本(the least reduced cost)，即 $\underset{r}{Minimize}(c_r - \sum_{i=1}^{N} \pi_r \cdot a_{ir})$ 為 ≥ 0 ，代表已找到此問題的最佳解；反之則我們可以從這些尚未考慮的變數中挑選減少成本最小(負)者 a_{ir} * 加回主問題繼續求解。

由於 CG 的子問題設計因人而異，如 Agawal et al. (1989)將子問題設計如下，其中 $y_i,\ i = 1,\cdots,N$ 是二元決策變數，而 **y** 即所有 y_i 所形成的向量。

目標式：

$$\text{Min} \quad Z = f(\mathbf{y}) - \sum_{i=1}^{N} \pi_i \cdot y_i \tag{2-31}$$

限制式：

$$\sum_{i=1}^{N} a_{ri}\pi_i \le c_r \quad \forall r \in R \tag{2-32}$$

$$y_i \text{ , } Binary \tag{2-33}$$

式(2-31)中 $f(\mathbf{y})$ 是 column $\mathbf{y}$ 下的最佳 TSP 成本，而式(2-32)為車容量限制。由於難以求得 $f(\mathbf{y})$，Agawal et al.(1989)再以線性方程式 $f(\mathbf{y}) = \mathbf{p} \cdot \mathbf{y}$ 近似代換之，其中 $\mathbf{p}$ 為 $p_i \quad (i = 1, \cdots, N)$ 所形成的向量，p_i 是對應顧客 i 成本的線性估計值，進一步將子問題轉變成為背包問題(Knapsack Problem)，再以分支定限法(branch-and-bound)求解，得 $a_{ir}{}^* = \mathbf{y}^*$。以此程序逐步改善原有問題的解，直到子問題無法再產生新的 column 為止。若放鬆整數限制後所得到的最佳解為整數解，即為原集合涵蓋問題的最佳解，若不為整數解，則採用分支定限法繼續求解以獲得整數解。

此法的優點在於求解過程中一次只考慮部份的變數，並利用主、子問題間的訊息傳遞，逐步得到最佳解，如此相對將可增進求解效率，可避免因變數過多而導致無法求解的缺點。

雖然 CG 解決了須以窮舉出所有求解集合來求解問題的缺點，求解品質亦佳，但亦導致相對複雜的求解過程，尤其子問題(如式(2-31)至式(2-33))通常求解亦不容易。另外，求得 SCP 放鬆整數限制後的解通常也是不可行的分數解，之後還是必須依賴 branch-and-bound 求解，尤其針對較大規模問題時依然得花長時間求解，顯得較無效率。故本研究採用了 CG 中以部份求解集合求解問題的觀念，針對集合涵蓋問題選擇放鬆後較為容易處理的涵蓋限制，而非二元整數限制，來設計出一求解方式較為容易的演算法。

第三章 模式建立

3.1 啟發式解法架構

首先將車輛路線問題轉換為集合涵蓋問題，將車輛路線視為一個集合，轉換為類似集合涵蓋問題的數學規劃模式如下：

i：顧客，$i=1\sim n$；I：所有顧客所成的集合。

j：指派之車輛，$j=1\sim m$；J：為所有可指派車輛 j 所成的集合。

r：車輛路線，$r\in R$；R：所有可行的車輛路線組合所成的集合

c_r：第 r 組車輛路線的成本。

a_{ir}：{0,1}；當顧客 i 包含於第 r 組車輛路線之內時，$a_{ir}=1$；否則為 0。

x_r：{0,1}；當第 r 組車輛路線被選取時，$x_r=1$，否則為 0。

目標式：

$$Min\sum_{r\in R} c_r x_r \tag{3-1}$$

限制式：

$$\sum_{r\in R} a_{ir} x_r \geq 1 \quad \forall i \in I \tag{3-2}$$

$$x_r \quad Binary \tag{3-3}$$

目標式(3-1)設定欲使總運輸成本極小化。限制式(3-2)表示為所有顧客皆需要被涵蓋到。限制式(3-2)中所使用的是不等號而非等號，也因此並未限制顧客只可被涵蓋一次。因此當求解發生顧客被涵蓋兩次以上的情形時，便需要進行修正，此即為集合涵蓋問題(SCP)與集合分割問題(SPP)之差別。限制式(3-2)使用不等式而非等式是因為一般而言求解 SCP 會較 SPP 容易。限制式(3-3)表示所有 x_r 為 0-1 整數變數。

本研究將此車輛路線問題轉換成數學規劃基礎的整數規劃模式—「集合涵蓋模式」再求解，優點在於集合涵蓋模式可以藉由該集合內的路線相互比較後挑出相較為佳的車輛路線，易於計算成本，而缺點在於求解較大規模的問題時，會出現如前述無法窮舉出 x_r 所有集合 R 的問題。

故本研究不以 Agawal et al.採用 CG 的方式放鬆整數限制，反而改採 Lagrangian Relaxation 放鬆涵蓋限制，但配合部份集合 R 當成解題空間（solution space）的概念，在每次運算時發展一有效改善解題空間的機制來彌補以部份解題空間為出發點導致求解品質不佳的缺點。

類似概念與應用參考了紀玟豪(2004)有關航空貨運承攬業之貨物併裝問題模式與曾筠予(2005)求解車輛定線問題之模式架構，綜合以上文獻想法來發展本研究之啟發式解法架構。

啟發式解法之架構圖詳見**圖 3-1**，本研究之啟發式解法細節於此後各節所述。

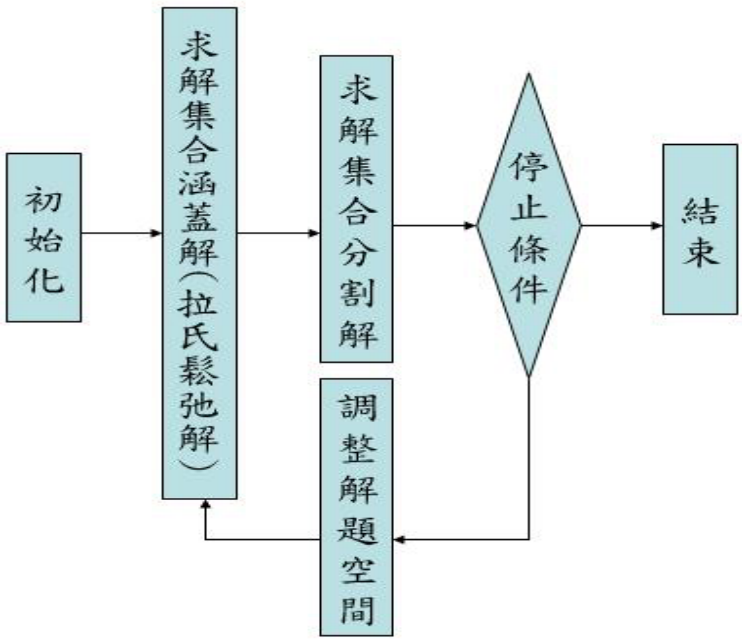

圖 3-1 啟發式解法之架構圖

3.2 拉氏鬆弛模型

3.2.1 放鬆涵蓋限制

本研究主題是對於式(3-1)到式(3-3)之模式，對於式(3-2)使用拉氏鬆弛法放鬆限制，其中以 u_i 代表拉氏乘數(正值)，$\boldsymbol{u}$ 代表所有 u_i 所形成之向量，對於本研究之車輛路線問題而言，可以將其視為該顧客在此解題空間之下的相對成本，故可藉由拉氏乘數來做為搜尋車輛路線的標準。

以下為放鬆限制後的模式：

目標式：

$$L(\boldsymbol{u}) = Min\sum_{r\in R} c_r(\boldsymbol{u})x_r + \sum_{i\in I} u_i \tag{3-4}$$

限制式：

$$x_r, Binary$$

其中 $$c_r(\boldsymbol{u}) = c_r - \sum_{i\in I_r} u_i \quad \forall r \in R \tag{3-5}$$

I_r：對於第 r 組車輛路線，所有被涵蓋的顧客所成之集合。($I_r = \{i \in I : a_{ir} = 1\}$)。

求解上述拉氏問題時，當 $c_r(\boldsymbol{u}) < 0$ 或是 $c_r(\boldsymbol{u}) = 0$ 時則 $x_r = 1$；反之，$c_r(\boldsymbol{u}) > 0$ 則 $x_r = 0$。

為了達到總成本最小化的目的，故求解集合設定在挑選出 $c_r(\boldsymbol{u})$ 為 ≤ 0 的路線。詳見**圖 3-2** 說明如下：

車輛路線組合依 $c_r(\boldsymbol{u})$ 值由小至大排序

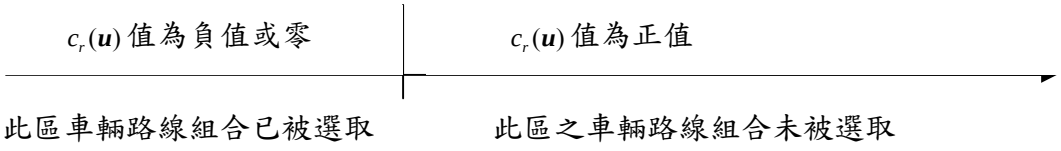

圖 3-2 起始車輛路線組合之挑選方式示意圖

在求得各個較低 $c_r(\boldsymbol{u})$ 值的路線集合後，算出總路線成本可得 $L(\boldsymbol{u})$ 值，因為求解出的 $L(\boldsymbol{u})$ 值是根據放鬆限制式後求得，就該解題空間 R 所對應的問題而言，其值

為下限值。但鬆弛解通常為不可行解，加上此$L(\boldsymbol{u})$值是以該次運算的部份解題空間而得的下限值，嚴格來說，此下限值對於原問題之式(3-1)而言並非為一可靠的下限值。

3.2.2 修正非可行集合涵蓋解為可行集合分割解

因為拉氏問題為放鬆限制式後再進行求解，在以部份求解集合當成解題空間的限制下，極可能求出不可行解(未涵蓋每一個顧客)。欲求得可行解，可透過修正拉氏放鬆解，進而修正成涵蓋每個顧客各只有一次的情況(SPP)。然而該修正程序的設計上必須注意，一方面希望使目標總成本在增加最少的情況下完成，但是又不宜太複雜，以免大幅增加運算的負荷。

本研究之演算法主要係利用$c_r(\boldsymbol{u})$值的觀念來進行修正解題空間，如前述求解拉氏問題時，將已求出各車輛路線組合的$c_r(\boldsymbol{u})$值由小到大進行排序，在本研究所建構之模式下，挑選$c_r(\boldsymbol{u})$值為負值或為零之車輛路線組合做為此次運算的標準，並在此時以(3-6)式檢視在$c_r(\boldsymbol{u})$值為負值或為零之車輛路線組合中各顧客點的覆蓋情況。

$$s_i(\boldsymbol{u}^t) = 1 - \sum_{r \in R_i} x_r(\boldsymbol{u}^t) \quad \forall i \in I \tag{3-6}$$

R_i：針對顧客i，所有涵蓋此顧客之車輛路線組合所成的集合，$\{i \in I : a_{ir} = 1\}$

方程式(3-6)中$s_i(\boldsymbol{u}^t)$值大小之涵意為顧客i被涵蓋之次數，當其值為負代表被涵蓋超過一次，越負代表被涵蓋越多次；其值等於 0 時，表示該項目僅被涵蓋一次；而其值等於 1 時，表示該項目未被涵蓋，詳見**圖 3-3**。

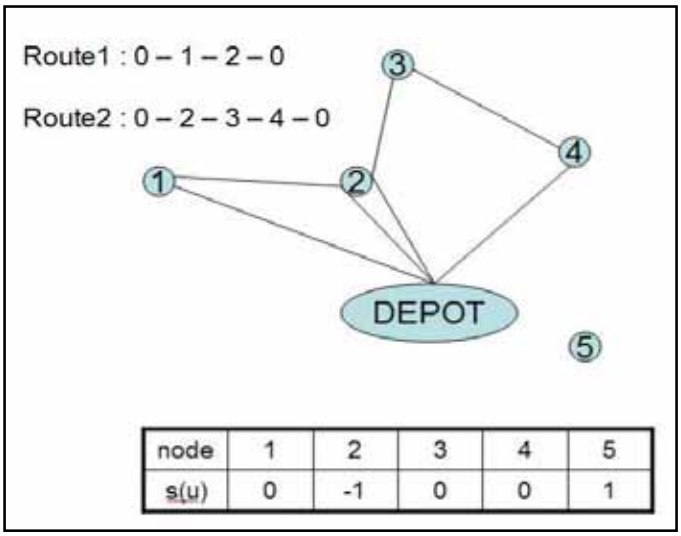

圖 3-3 $s_i(u)$ 值示意圖

求解集合分割解的步驟如下：

1. 假設集合S為空集合ϕ，其代表 SPP 解內已挑選之路線中所涵蓋的顧客。
2. 車輛路線組合由上而下（$c_r(u)$值由小到大）之順序選取一車輛路線組合r，令T為該車輛路線組合r中所涵蓋的顧客。若遇該車輛路線組合r中的顧客T與S無重覆涵蓋，即$S \cap T = \phi$，則更新$S = S \cup T$，並將r列入 SPP 的解之中；反之，若遇該車輛路線組合r中的顧客若有與S中的顧客重覆涵蓋的話，即$S \cap T \neq \phi$，則將該此車輛路線組合r中的該重覆涵蓋之顧客予以刪去，成為一條新路線組合r'並重新計算$c_{r'}(u)$值之後，再依該$c_{r'}(u)$值大小將此路線組合插入已按拉氏成本排序的解題空間中。(註：此時的原始路線r以其他記憶位置予以紀錄，供後續可能之運用。)
3. 重覆步驟 2，直到選取的車輛路線組合中涵蓋的顧客只有一次時，則選取的車輛路線組合即為此次之集合分割解(SPP)。
4. 並計算該選取的集合分割解之總路線成本做為上限值（UB），假設比目前最佳 SPP 解之總路線成本為低則取代為新上限值，並紀錄該次選取之路線組合作為目前遞迴最佳解。下**圖 3-4** 舉以六個顧客點為例，修正為集合分割解的示意圖。

經由修正的不可行解進一步得集合分割解值，此集合分割解值可作為此問題的上限值(UB)，此值為一有效的上限值。關於上下限值，於 3.4 節有更進一步的說明。

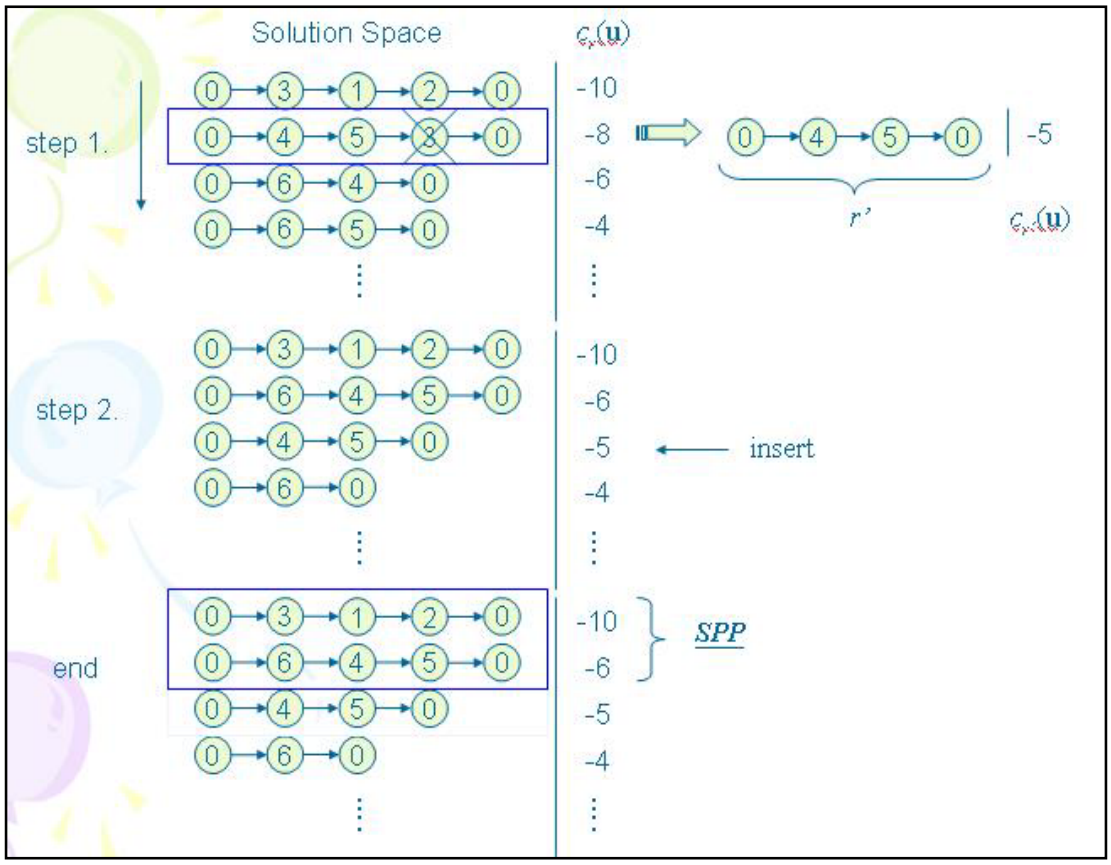

圖 3-4 求解集合分割解之意圖(以 6 個顧客點為例)

3.2.3 更新拉氏乘數

此時為了尋找接近最佳拉氏乘數向量，本研究採用次梯度法。此方法是產生一序列 $u^1, u^2, \cdots$ 等的非負拉氏乘數向量，靠 $s_t(\boldsymbol{u}^t)$ 來調整拉氏乘數向量，對於較大規模的問題，可以得到較快的求解速度。

故有關拉氏鬆弛法遞迴運算過程中，拉氏乘數之修正，可參考(3-7)式之運算 (Held, Karp, 1970)，式中 t 代表第 t 次遞迴運算。

$$u_i^{t+1} = \max\left\{u_i^t + \lambda \frac{UB - L(\mathbf{u}^t)}{\left\|s(\mathbf{u}^t)\right\|^2} s_i(u^t), 0\right\} \quad \forall i \in I \tag{3-7}$$

UB 所代表的是遞迴運算中最佳上限值，由可行解之集合分割解值求得，而 $L(\mathbf{u}^t)$ 為第 t 次所求得的放鬆下限值。在經由 UB 與 $L(\mathbf{u}^t)$ 所夾區間，代表最佳解所在的區間範圍。拉氏乘數 u_i 更新的原則是用於上下限值逼近時，縮小拉氏乘數 u_i 值調整的步幅，慢慢修正逼近最佳解的速度。

a. λ 為更新 $\mathbf{u}$ 值的調整係數(正值)，是一個給定的步幅參數(step-size parameter), 主要在針對 UB 、$L(\boldsymbol{u})$ 和 $s_i(\boldsymbol{u}^t)$ 三項數值做出修正細部調整，用來作為修正幅度的大小。λ 值也可根據求解次數或解的品質來做動態微調，藉由改變其值以加快收斂速度或減緩調整幅度以利於求得較佳的解。對於不同問題類型，也可依自行需求作設定。

b. $s_i(\boldsymbol{u}^t)$ 為顧客被涵蓋的次數，在次梯度法中則代表為下次 $\mathbf{u}$ 值的修正方向。而 $\left\|s_i(\boldsymbol{u}^t)\right\|^2$ 所表示的是向量 $s_i(\boldsymbol{u}^t)$ 的長度平方值，影響每次調整的步幅。透過適度的修正，使得每次拉氏乘數的遞迴求解，逐漸的往最佳拉氏乘數前進，其原則是：當 $\left\|s_i(\boldsymbol{u}^t)\right\|^2$ 變大時(重覆涵蓋顧客次數太多)，代表搜尋的方向誤差仍過大，將 $\left\|s_i(\boldsymbol{u}^t)\right\|^2$ 置於分母，用以降低此次搜尋步幅的大小，避免過度地修正 $\boldsymbol{u}$ 值。

c. 關於 $\mathbf{u}$ 值的部分，本研究將每個顧客 i 之起始 u^0 值，設定為由場站原點出發到顧客 i 之距離，u_i 值的設定影響解收斂的情況，當遞迴次數持續增加此值亦會趨於穩定。

3.3 解題空間的調整機制

為了彌補以解題空間 R 僅包含部份車輛路線組合的缺點，本研究先只建立一個簡單的初始解(3.3.1 節)，藉由刪除相較不佳的路線組合(3.3.2 節)，再新增具有潛力成為較佳的車輛路線組合，逐步運算改善解題空間 R ，求得近似最佳解。產生新的車輛路線方式主要分為二種，第一種是在現有車輛路線組合中刪減掉顧客的方式(3.3.3 節)，第二種是在現有車輛路線中插入新顧客的方式(3.3.4 節)。

3.3.1 解題空間初始化

本研究先產生一個初始車輛路線組合的集合，以挑選兩個顧客來回的路線(產生 C_2^n 個，n 為顧客數)為起始車輛路線組合做為解題空間，因為當車輛路線組合裡只含有一個顧客或是兩個顧客時，對應 TSP 問題相當簡單，亦可輕易求出各車輛路線組合所對應的成本。

3.3.2 刪除不佳的車輛路線組合

當顧客數增加，解題空間劇烈增大，不可能保留所有車輛路線加以運算，故本研究在每次遞迴運算刪除不佳的車輛路線組合，保留固定數目的車輛路線組合數，配合拉氏鬆弛法的遞迴運算，一是為了求解運算上的效率，不能保留太多車輛路線組合，會造成求解時間過長；二是為了求解結果的品質，保留過少的車輛路線組合，則會造成求解品質不佳。

保留車輛路線組合的選取方式為：依照 $c_r(\boldsymbol{u})$ 值由小到大選取固定 m 個車輛路線組合數，其餘路線組合均予以刪去，以此車輛路線組合數的空間為基礎，此後將還有解題空間調整的機制。詳見**圖 3-5** 說明如下：

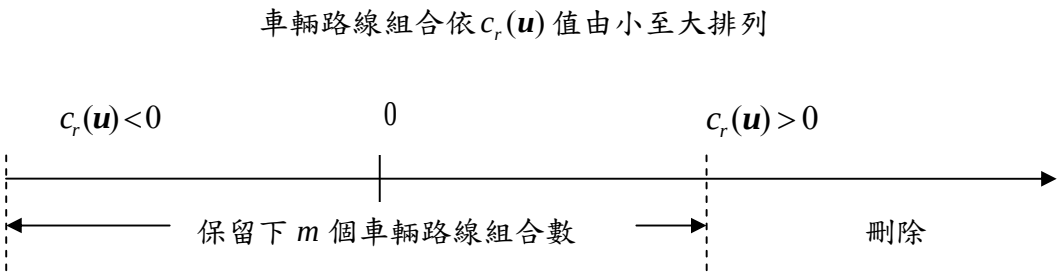

圖 3-5 車輛路線組合空間之調整保留圖

完成刪除的動作之後得到保留的路線，其中若包含先前修正集合分割解時(3.2.2 節步驟 2)所產生之修正後的車輛路線 r'，改以刪除重覆顧客前的原始車輛路線 r 做取代，避免車輛路線過短或變化不足的情況發生。

3.3.3 產生新車輛路線之調整機制—減點的集合

拉氏乘數可視為此次遞迴運算中該路線中此顧客的所擁有的合理成本，故拉氏乘數為搜尋該顧客的一個有效指標，刪點的調整機制也由此產生。

首先從解題空間中隨機挑選a條路線（如為$m/5$）做為刪點的目標路線，再由這些路線當中參考拉氏乘數去檢視哪些顧客在此路線當中是相對不佳的顧客，予以刪去，成為新路線組合。檢視的方法有二階段：

1. 於所挑選的車輛路線組合中就每一顧客i作檢視，進行以下計算：$(C_{i-1,i}+C_{i,i+1}-C_{i-1,i+1})-u_i$，（示意圖如**圖 3-6**，其中$C_{i,j}$為顧客$i$至顧客$j$的路段成本），即欲求出挑選出的車輛路線中因多涵蓋了該顧客i而增加的成本減去該顧客所對應的拉氏乘數的大小。此值最大者代表該顧客為該車輛路線當中成本相對較高者，較不適合在該車輛路線中。將此顧客於該路線中刪去，重新計算新路線之拉氏成本，挑選出因刪減單一顧客造成拉氏成本變成負值的路線者，做為刪點的集合。
2. 類似步驟 1，但以相鄰二個顧客一併作檢視，決定是否因為多涵蓋該此二顧客i與顧客$i+1$而增加的成本減去該顧客所對應的拉氏乘數後相對最大者。故將此二個顧客於該路線中刪去，再將新路線重新計算拉氏成本，挑選出因刪減此二個顧客造成拉氏成本變成負值的路線者，做為刪點的集合。
3. 並紀錄該刪點集合中被刪除的顧客。

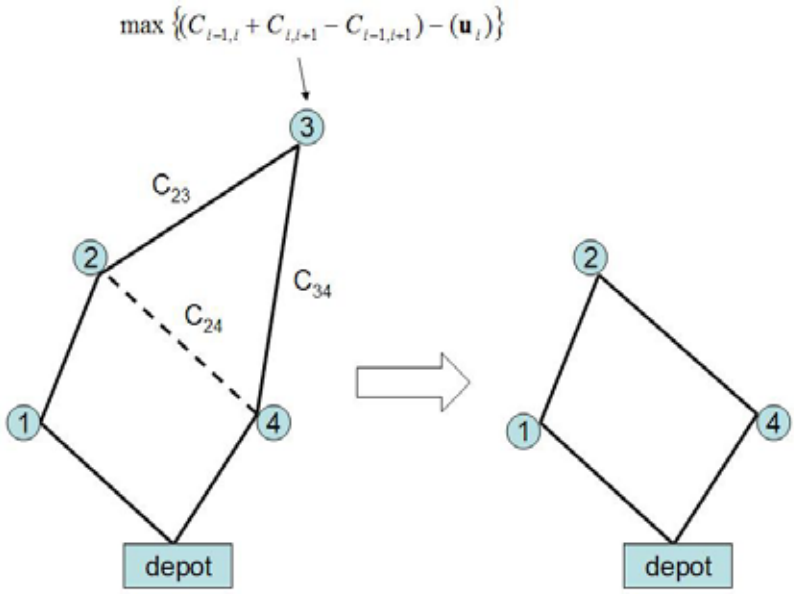

圖 3-6 挑選單一刪除顧客點之示意圖

3.3.4 產生新車輛路線之調整機制—加點的集合

從求解集合中由最負 $c_r(\boldsymbol{u})$ 值的路線（$c_r(\boldsymbol{u})$ 由小到大的路線）開始挑選出 b 條路線（如為 $m/5$）做為插入的目標路線，再將欲插入的顧客插入目標路線內，產生新路線組合。而插入顧客的產生目前有以下二個來源：

1. 於 3.3.3 節中刪除的顧客
2. $s_i(\boldsymbol{u}^t)$ 為 1 的顧客（即該次運算中在挑選的路線內未被涵蓋的顧客）

並設一門檻值 c (如為 $m/10$)，如果來源 1 之顧客數 $< c$，則加入來源 2 之顧客至 c；反之，若來源 1 之顧客數 $> c$，則來源 1 顧客數設為 c，而不再考慮來源 2。

根據 Agarwal et al.(1989)證明，在一最佳 TSP 中欲加入一顧客時，假設插入顧客的位置(顧客與顧客之間)使插入後的路線總成本為最小，則該路線即為插入顧客後的最佳 TSP 走法。因此在目標車輛路線組合裡插入的方式是試著將欲插入的顧客安插於該路線組合裡的顧客點與顧客點之間，尋找出最小成本的路線組合，而不再求插入顧客之後的排列組合最佳解，以減少求解運算時間，並依此為新的車輛路線組合。

例如車輛路線組合為[0→1→2→0]，其中 0 代表場站原點，由原點 0 出發，依序經過了顧客點 1、2，最後回到原點 0。而如果要增加的顧客為顧客點 3，則插入此顧客後將會有三種路線組合：[0→**3**→1→2→0]、[0→1→**3**→2→0]和[0→1→2→**3**→0]，計算此三種路線組合的成本，假設[0→1→**3**→2→0]為其中成本最小路線組合，則挑選路線組合為[0→1→**3**→2→0]插入顧客點 3 後的新車輛路線組合，詳見**圖 3-7**。

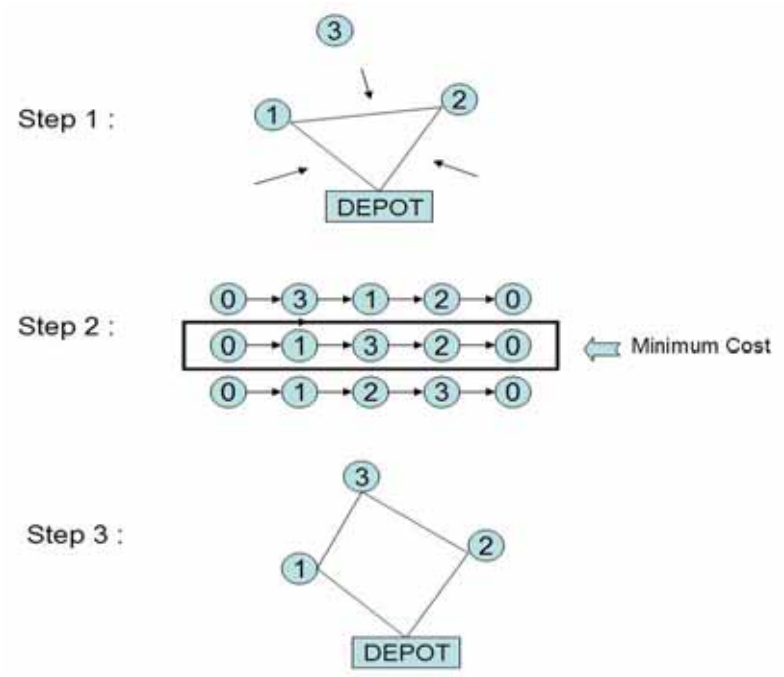

圖 3-7 車輛路線組合中增加顧客之示意圖

從成功完成插入顧客後的 TSP 中重新計算其$c_r(\boldsymbol{u})$值，如果$c_r(\boldsymbol{u})$值小於零，則將此路線視為加點的集合，加回解題空間內。

3.3.5 加入目前最佳解至解題空間內

每次運算的解題空間會加上「反覆遞迴運算裡所紀錄目前最佳解之車輛路線組合」，此舉是由於本研究預期從這些組合裡，可以產生更好的車輛路線解，也使得下次遞迴運算修正需要可行解時，車輛路線解題空間裡一定含有可行解供挑選。這些全部保留下來的車輛路線組合，將提供作為下次遞迴運算之起始車輛路線解題空間。

3.4 上限值、下限值的意義與停止機制

原本以拉氏鬆弛法求解集合涵蓋問題的方法，是產生全部可行之車輛路線的解題集合空間，當上限值UB及下限值$L(\boldsymbol{u})$兩解相等或兩解之差值在容忍範圍內時，即停止求解。但本研究以部分車輛路線組合為解題空間，並在遞迴運算的過程中加以調整解題空間的方式予以求解，就UB值與$L(\boldsymbol{u})$值的意義上，已經與原本拉氏鬆弛法求解集合涵蓋問題略有不同。

以下將說明本研究演算法中，求解拉氏問題、集合涵蓋問題與原本拉氏鬆弛法求解集合涵蓋問題不同意義之處與停止機制。

a. 本研究演算法求解拉氏鬆弛問題所得到的解，在文獻中是設定$L(\boldsymbol{u})$值是在窮舉解題集合R的前題下以$c_r(\boldsymbol{u})$值為負的路線總成本為下限值，雖然在遞迴運算中針對該次之車輛路線之解題空間是一個有效的下限值，但對於本研究以部份路線為解題空間來說並非是一個有效的下限值，已與原來車輛路線問題的下限有所差異。主要是因為本研究僅產生部份部分的車輛路線解題空間，所得之下限僅是針對目前車輛路線的解題空間所求解到的一個下限值，不確保車輛路線問題的最佳解必定高於此$L(\boldsymbol{u})$值。因此下限值$L(\boldsymbol{u})$因為每次遞迴運算出$c_r(\boldsymbol{u})$值的路線組合總成本變動過劇，為了能夠使每次遞迴運算可有效往最佳解逼近，所以目前本研究僅是設定下限值$L(\boldsymbol{u})$為一常數值0。

b. 根據拉氏問題的解如3.2部分，傳統放鬆原問題後的作法是設定以集合涵蓋解為上限解，而以集合涵蓋解為上限值相對於本研究未免過大，導致收斂效率太差，因此本研究設定以集合涵蓋問題的可行解，進一步修正為集合分割解的值取代成為上限值。此上限值是以部份車輛路線解題空間所求得，因此可視為一個區域最佳解。由於最佳解必小於或等於區域最佳解，所以此上限值雖然意義上僅是根據部分車輛路線解題空間所求得，但對於原來車輛路線問題來說仍是一個有效的上限值。

由此可知，雖然採用與原先拉氏鬆弛法同樣方式，但本研究之演算法修正了因為以部份解題空間求解導致每次運算的上下限呈現不穩定的缺點，故更改設定後仍可利用上限值與0夾擠出確切的最佳解所在區間，而不需要根據持續地改進有限的

解題空間所得的上下限值，再反回饋取代之。

停止運算機制的部分，目前是利用「集合分割問題的解趨於穩定」、「當求解次數達到所設定次數」、「總**u**值趨於穩定」，來做為停止繼續搜尋最佳解的條件。

3.5 演算法流程設定

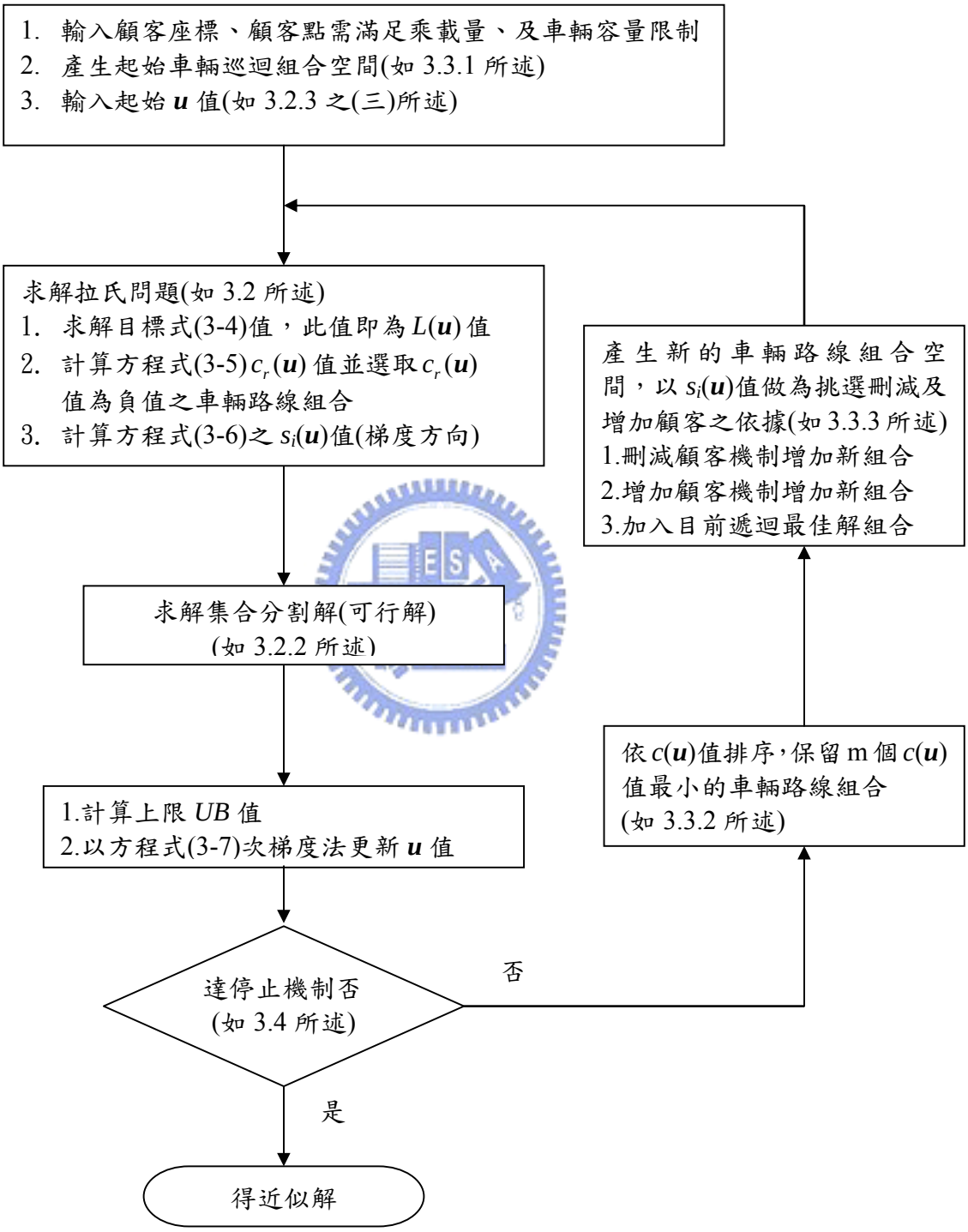

圖 3-8 演算法詳細流程圖

以下針對演算法的一些細部設定做說明。

1. 有關保留車輛路線組合數的數值(m)

目前測試是採以固定數值的方式，但也可採用變動數值方式，隨問題規模增大自由增減。選擇變動數值方式，有利於產生足夠的車輛路線組合空間，但會將降低求解效率；而固定數值的方式，則是選定適合之限制個數，不因問題規模而改變，但有可能會加長求解過程。但因為多數的車輛路線組合即為最佳路線的排列組合，充斥於解題空間對於求解問題並沒有實質貢獻，所以即使問題規模加大，也不需要產生太多的集合，目前以測試經驗來設定為 100，可視問題規模大小作調整。

2. 有關用於調整解題空間中的參數值 $a,\ b,\ c$

目前是以測試經驗將其設定為保留的車輛路線數 m 的某個固定比例，設定 $a=\frac{m}{5},\ b=\frac{m}{5},\ c=\frac{m}{10}$，亦可視問題規模大小與求解品質作動態調整。

3. 有關調整係數 λ 值方面

於 $\mathbf{u}$ 值更新方程式中的調整係數 λ 值，在 Carprara et al. (1999)中，另外提及以上下限值的差距對 λ 值做小幅度調整的方式，本研究在實例測試方面，對於調整係數 λ 值與向量長度平方影響等議題將不再深入，將步幅大小視為固定，採參考過往文獻的經驗法則來設定為 0.05。

4. 有關 $s_i(\boldsymbol{u}^t)$ 的細部設定

由於本演算法以每次運算改善的部份路線做為解題空間，而非傳統以窮舉所有可能路線集合做為求解集合，再逐步經由 $s_i(\boldsymbol{u}^t)$ 改善 $\mathbf{u}$ 來挑選適宜的路線組合。故為了針對此點，本研究將 $s_i(\boldsymbol{u}^t)$ 限制在-2 到 1 之間（限制覆蓋情況在 3 次到 0 次），避免因下次改善解題空間造成 $s_i(\boldsymbol{u}^t)$ 變動過劇造成收斂過慢的情況，而非傳統不做任何限制，以該次覆蓋情形做為下次改善的準則。

總合第三章內容所述，本研究之演算法採用拉氏放鬆法後，往最佳解修改之方向是使用一般貪婪法則、次梯度法，以此啟發式解法挑選及產生路線集合反覆求解，依然以部分解題空間為起始解求解問題，不去產生所有可能的集合，遞迴

過程中也同樣是以部分解題空間求解，每個求解循環之中，核心機制在著重於如何產生較佳且有限的車輛路線解題空間，進一步利用此有限車輛路線解題空間，來求得近似解。因此在產生新車輛路線的解題空間方面，設定了兩項改善機制，可以增加新車輛路線組合來調整車輛路線解題空間，避免車輛路線的解題空間變化不足，導致求解品質不佳。

本研究架構參考紀玟豪(2004)與曾筠予(2005)之碩士論文，以下分別說明與其相關架構的主要差異：

1. 紀玟豪(2004)在集合涵蓋模式由於求解不同類型問題，所以在成本結構與限制皆有所不同，例如紀玟豪限定每個貨物集合只能上一個航班，此限制在本研究並不存在，另外「修改為集合分割解」與「調整新增集合的方式」則是最主要之差異。

2. 與曾筠予(2005)在模式架構上目前主要差別：

   I. 在於可行解的調整機制，如曾筠予在可行解調整過程中是在$c_r(\boldsymbol{u})$值為正的車輛路線解題空間中找尋在$c_r(\boldsymbol{u})$值為負或零的車輛路線組合未涵蓋該顧客的車輛路線組合，而本研究是從$c_r(\boldsymbol{u})$值為最負的車輛路線解題空間裡依序尋找未重覆涵蓋顧客的路線，直到挑選的路線恰涵蓋所有顧客為止。

   II. 上下限的設定，由於曾筠予將上限假設為集合涵蓋解，下限則假設為解題空間中$c_r(\boldsymbol{u})$值為負值的總路線成本。由於部份解題空間的原故會導致每次運算上下限的範圍變得不穩定，非一有效之夾擠區間範圍，使得曾筠予只能以集合分割解求解趨穩定作為求解的停止條件，且因難以收斂導致求解品質不佳。本研究將上限值設為集合分割解之總成本，而下限值則設為 0。

   III. 在調整解題空間的機制中，由於本研究修改了可行解的調整機制，而在加點的集合裡為了使有效改善解題空間，所以在加點的集合中從$c_r(\boldsymbol{u})$值最小的幾個路線選擇來做路線變化，而不同於曾筠予在所有保留的解題空間組合作調整集合的運算。

第四章 數值測試結果

4.1 CVRP 題庫測試

本研究數值測試對象方面則是挑選了 Solomon 網站(如參考文獻)上 CVRP Instance 中的 Christofides and Eilon 與 Augerat et al.之 Set A、Set B 車輛路線例題題庫測試本演算法成效。

測試環境為 CPU AMD Athlon XP 2000+ 1.67GHz，1.5GB RAM，運算時間以 CPU 時間(秒)為衡量基準，僅視為一參考值。經初步數值測試後，求解品質尚稱穩定，與已知最佳解平均誤差約在 3%左右。

其中 Christofides and Eilon 題庫則是挑選出九題作為測試，題型規模由約 20 個顧客到 100 個顧客，顧客位置與需求皆為隨機產生，且車容量限制不一，除例題 n76-k8、n101-k8、n101-k14 外，其餘文獻均列有最佳解(exact solution)，以 '*' 表示，如**表 4-1** 第二行所示。

表 4-1 Christofides & Eilon 之 VRP 例題之結果比較整理表

Instance	Best Known	Result	Truck	Gap (%)	CPU-Time
n22-k4	375*	375*	4	0	74
n23-k3	569*	569*	3	0	104
n30-k3	534*	504	4		124
n33-k4	835*	837	4	0.24	155
n51-k5	521*	531	5	1.92	185
n76-k7	682*	700	7	2.64	291
n76-k8	735	742	8	0.95	285
n101-k8	815	848	8	4.05	540
n101-k14	1071	1111	14	3.73	403

除例題 n30-k3 比目前最佳解多一車輛數，但成本降至 504 外，其他例題均達與目前最佳解相同車輛數，與目前最佳解平均誤差約在 2.3%左右，求解的 CPU 時間平均約為 240 秒左右，如**圖 4-1**。求解時間與問題規模大小幾乎呈線性，當然主要是以保留固定 *m* 個數目的車輛路線為前題下，保留越多求解時間越長，求解品質亦佳，詳細求解結果列於附錄供參考。

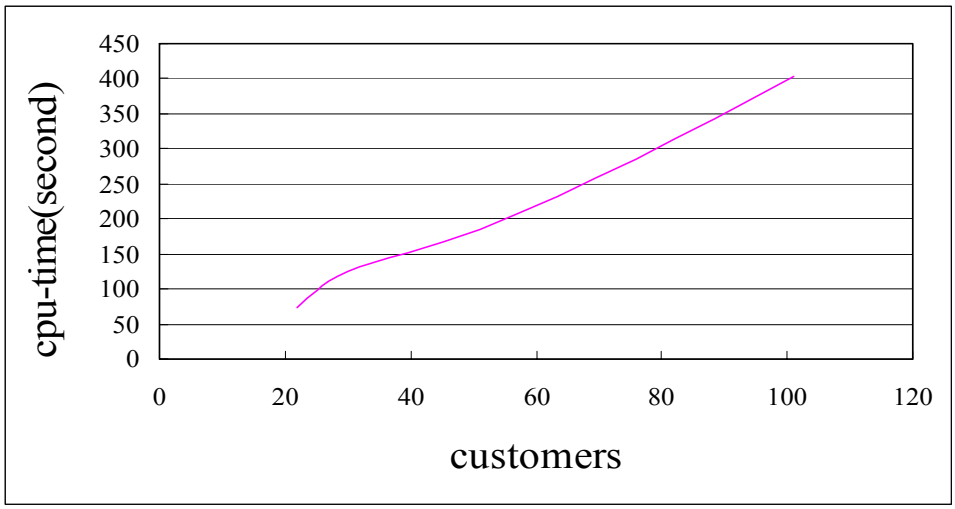

圖 4-1 Christofides & Eilon 之 VRP 例題求解時間圖

為了進一步測試求解品質，本研究繼續選擇 Augerat et al.的題庫做測試。其中 Set A 題庫的特性是顧客的位置與需求皆為隨機產生，而 Set B 題庫則是顧客具有群集分佈(clustered)的特性，問題規模由約 30 個顧客到 80 個顧客，車容量限制均為 100，目前文獻均已解出最佳解，如**表 4-2**、**表 4-4** 第二行所示。

表 4-2 Augerat et al. Set A 例題之結果比較整理表

Augerat et al. Set A				
Instance	Best Known	Result	Truck	Gap (%)
n32-k5	784*	787	5	0.38
n33-k5	661*	668	5	1.06
n33-k6	742*	744	6	0.27
n34-k5	778*	790	5	1.54
n36-k5	799*	809	5	1.25
n37-k5	669*	682	5	1.94
n37-k6	949*	950	6	0.11
n38-k5	730*	737	5	0.96
n39-k5	822*	830	5	0.97
n39-k6	831*	854	6	2.77
n44-k6	937*	949	6	1.28
n45-k6	944*	957	7	1.38

n45-k7	1146*	1181	7	3.05
n46-k7	914*	936	7	2.41
n48-k8	1073*	1096	7	2.14
n53-k7	1010*	1033	7	2.28
n54-k7	1167*	1209	7	3.60
n55-k9	1073*	1098	9	2.33
n60-k9	1354*	1375	9	1.55
n61-k9	1034*	1055	9	2.03
n62-k8	1288*	1328	8	3.11
n63-k9	1616*	1685	9	4.27
n63-k10	1314*	1360	10	3.50
n64-k9	1401*	1458	9	4.07
n65-k9	1174*	1218	9	3.75
n69-k9	1159*	1189	9	2.59
n80-k10	1763*	1828	10	3.69

除例題 n45-k6 比目前最佳解多一車輛數外，其他均達與目前最佳解相同車輛數。其中例題 n48-k8 的車輛數為 7，比目前最佳解少一車輛數，下**表 4-3** 則列出 n48-k8 的求解結果。與目前最佳解誤差差距最大者為例題 n63-k9，達 4.27%。求解之總平均誤差為 2.16%，標準差為 0.0119，求解的 CPU 時間平均約為 220 秒左右。

表 4-3 Augerat et al. Set A – n48-k8 測試結果

Instance：n48-k8	Capacity：100	Truck：7	Cost：1096
Route 1： 1 29 22 31 47 14 5 12 43 46 1			
Route 2： 1 42 3 11 34 30 25 10 35 1			
Route 3： 1 8 4 21 27 40 9 16 28 1			
Route 4： 1 41 37 38 20 26 39 33 1			
Route 5： 1 17 48 18 15 1			
Route 6： 1 44 32 6 2 7 23 36 1			
Route 7： 1 13 24 45 19 1			

表 4-4 Augerat et al. Set B 例題之結果比較整理表

Augerat et al. Set B				
Instance	Best Known	Result	Truck	Gap (%)
n31-k5	672*	688	5	2.38
n34-k5	788*	791	5	0.38
n35-k5	955*	975	5	2.09
n38-k6	805*	809	6	0.50
n39-k5	549*	555	5	1.09
n41-k6	829*	855	6	3.14
n43-k6	742*	759	6	2.29
n44-k7	909*	945	7	3.96
n45-k5	751*	766	6	2.00
n45-k6	678*	709	6	4.57
n50-k7	741*	759	7	2.43
n50-k8	1312*	1336	8	1.83
n51-k7	1032*	1039	8	0.68
n52-k7	747*	760	7	1.74
n56-k7	707*	726	7	2.69
n57-k7	1153*	1175	8	1.91
n57-k9	1598*	1628	9	1.88
n63-k10	1496*	1542	10	3.07
n64-k9	861*	876	9	1.74
n66-k9	1316*	1356	9	3.04
n67-k10	1032*	1053	10	2.03
n68-k9	1272*	1320	9	3.77
n78-k10	1221*	1260	10	3.19

除例題 n51-k7 與例題 57-k7 比目前最佳解多一車輛數外，其他均達與目前最佳解相同車輛數。與目前最佳解差距最大者為例題 n45-k6，達 4.57%，其他例題誤差維持在 3%左右。求解之總平均誤差為 2.28%，標準差為 0.0107，求解的 CPU 時間平均約為 200 秒左右

4.2 參數測試

目前本研究是採用以保留固定數目的車輛路線數於解題空間做運算，然而求解的品質會與求解時間長短與此參數之設定有密切的關係。保留車輛路線越多、遞迴次數越多，則求解時間越長，求解品質亦佳。為了在可接受的時間內找到一個不錯的近似解，有關保留車輛路線數(參數 m)及相關參數(a，b，c)的設定，較佳的做法是在運算過程中依當時情況(如問題規模大小、該解題空間情況、…)做動態微調的方式。以下採用 Christofides and Eilon 題庫，各挑選小、中、大三個例題(n22-k4、n51-k5、n101-k14)做為測試，以參數 $a = m/5$，$b = m/5$，$c = m/10$ 固定的方式，說明保留 m 值的大小造成求解品質好壞的差異，如**圖 4-2**、**圖 4-3**、**圖 4-4**。由測試的結果發現，確實大致上保留路線越多(m 越大)，求解品質越理想，但是隨者問題規模的加大，所需增加的路線保留數目比起前在的可能路線數目，相對非常地少，表示本啟發演算法應該是有利於大型問題之適用。

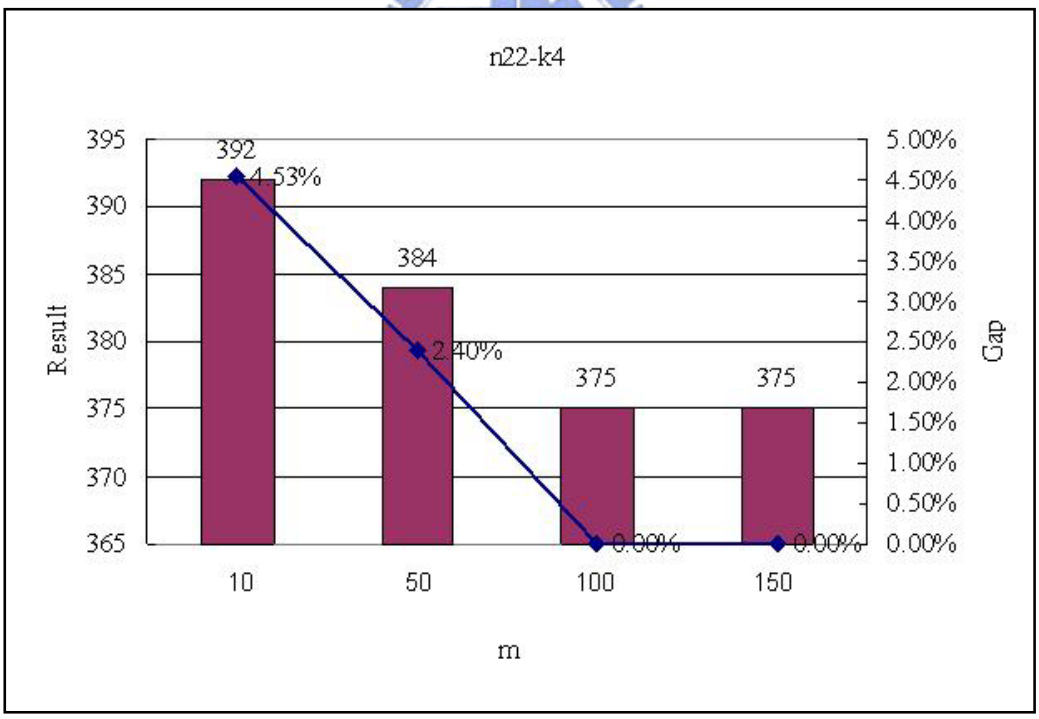

圖 4-2 例題 n22-k4 依 m 大小求解之示意圖

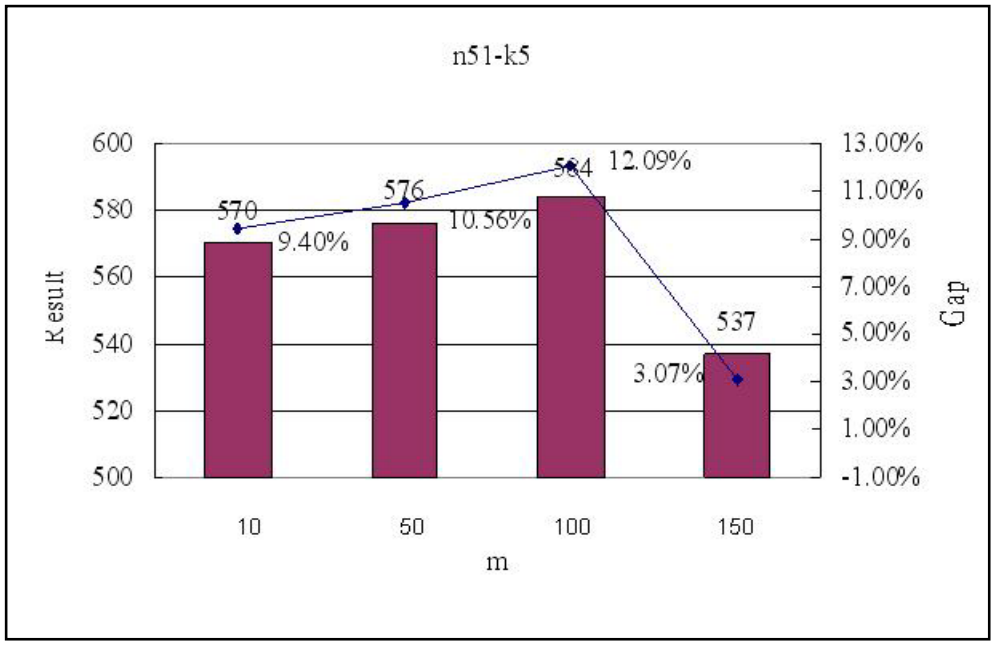

圖 4-3 例題 n51-k5 依 m 大小求解之示意圖

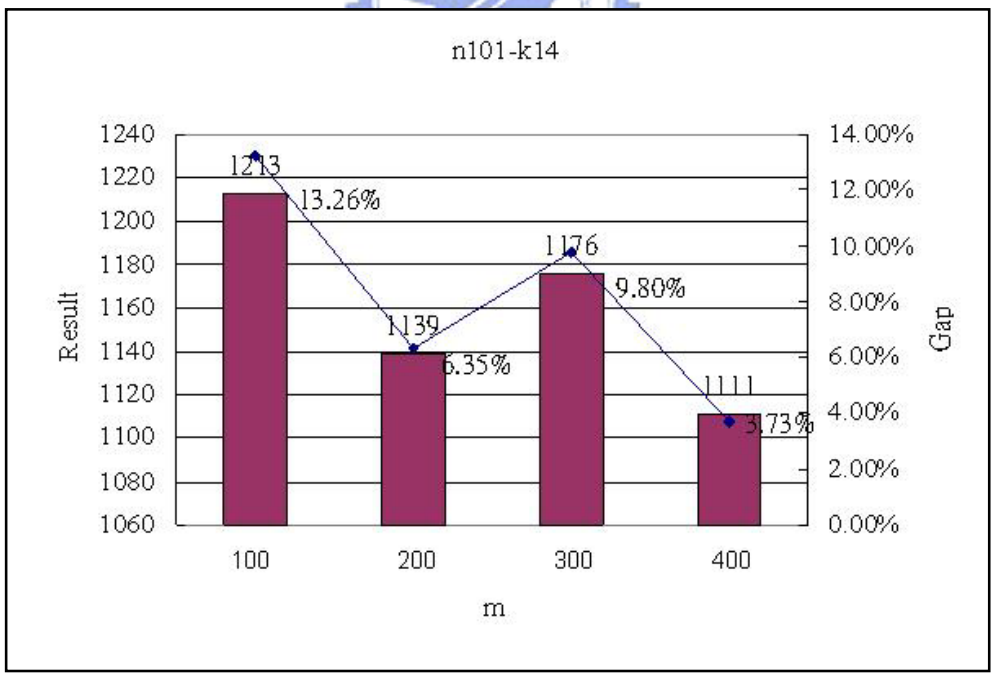

圖 4-4 例題 n101-k14 依 m 大小求解之示意圖

第五章 結論與建議

由於以集合涵蓋模式求解大規模問題時會因求解集合過大而使得求解時間過長，顯得比較沒有效率。但若用部份子集合去求解又會使得求解品質變差，故過去文獻中鮮少有以集合涵蓋模式求解較大規模的車輛路線問題。而本研究即針對此一缺點，欲發展一適宜的求解演算法，於演算法架構模式中加入一能有效改善求解品質的機制，設法補償在求解過程中以部份子集合去求解問題的缺點。

而以往文獻多以集合涵蓋問題求解車輛路線問題是以變數產生法放鬆二元決策變數限制的方式，再以分支定限法解決放鬆主問題後子問題之非整數解，雖可獲得較佳的求解品質，但相對複雜的求解過程換取較長的求解時間。

本研究發展的求解架構在求解複雜度上相對明顯簡單許多，從原車輛路線問題轉成集合涵蓋問題後以拉氏鬆弛法放鬆顧客涵蓋限制，直接從逐步改善的解題空間內挑選出最佳路線組合。實例測試證明本研究其求解品質尚可接受，也擺脫為人垢病的求解時間，而且由於在每次運算的過程中皆能有效調整求解集合，產生相對較佳路線，並不會陷入所謂局部最佳解的情況。本研究架構著重在求解品質上，在相關參數部份，目前參數設定以穩定求解品質，以經驗法則暫訂之，未作相關深入探討。

目前有以下幾種可持續探討的議題，期許在未來達成整體演算法之完整性：

1. 能夠在運算過程中以一機制來動態保留較佳的車輛路線組合，而非以目前保留固定路線數來作為求解空間，冀希能縮短有效運算時間。
2. 嘗試利用每次運算出的上下限差值來動態調整 λ 值，或是以顧客點 $\mathbf{u}$ 值的變化配合 $\left\|s_i(\boldsymbol{u}^t)\right\|^2$ 來調整，使 λ 值能夠有效率的變化來增快收斂的速度。
3. 能夠找到更有效率的調整機制去產生較佳的車輛路線組合，增加路線變化的組合；刪除多餘不佳的的路線組合，減少程式運算負擔。
4. 以更有效的下限值來取代目前的常數設定，以有效加快求解效率與求解品質，如以解目標式的線性規劃解為下限值取代目前常數的假設。
5. 以原演算法模式繼續延伸發展，如加入時間窗限制去求解具時間窗限制的

車輛路線問題，測試 Solomon 的標竿題庫試題，用來檢視本演算法整體的求解效益，期許能夠獲得不錯的結果。

6. 將 code 進行最佳化再進行測試，可以以更短的求解時間突顯本研究的成效。

參考文獻

Agarwal, Y., Mathur, K., and Salkin H.M. (1989) " A Set-Partition-Based Exact Algorithm for the Vehicle Routing Problem". ***Networks, 19***, 731-749.

Balas E. , Ho. (1980) "Set Covering Algorithms Using Cutting Planes, Heuristics, and Subgradient Optimization: A Computational Study". ***Mathematical Programming, 12***, 37-60.

Balas E. , Carrera M.C. (1996) "A Dynamic Subgradient-based Branch-and-Bound Procedure for Set Covering". ***Oper. Res., 44***, 875-890.

Beasley, J.E. (1990) "A Lagrangian Heuristic for Set-Covering Problems". ***Naval Research Logistics, 37***, 151-164.

Bramel, J. , Simchi-levi, D. (1997) "On the effectiveness of set covering formulations for the vehicle routing problem with time windows". ***Oper. Res., 45***, 295-301.

Caprara, A., Fischetti, M. and Toth, P. (1999) "A heuristic method for the set covering problems", ***Oper. Res., 47***, 730-743

Clarke, G., , Wright, J. (1964) "Scheduling of vehicles from a central depot to a number of delivery points", ***Oper. Res., 12***, 568-581.

Cullen, F., Jarvis J. and Ratliff D., (1981) "Set partitioning based heuristics for interactive routing". ***Networks, 11***, 125-144.

Desrochers, M., Desrosiers, J. and Soumis, M. (1984) "Routing with time windows by column generation". ***Networks, 14***, 545-565.

Desrochers, M., Desrosiers, J. and Solomon, M. (1992) "A new optimization algorithm for the vehicle routing problem with time windows". ***Oper. Res., 40***, 342-354.

Fisher, M. (1995), "Vehicle routing", Chapter 1 in M. Ball, Magnanti, C., Monma, G., Nemhauser (eds.), ***Handbooks in Operations Research and Management Science, 8***, 1-33.

Fisher, M. L., Jaikumar, R., (1981) "A generalized assignment heuristics for vehicle routing", ***Networks, 11***, 109-124.

Gilbert Laporte, Michel Gendreau, Jean-Yves Potvin, and Frederic Semet, (2000)

"Classical and Modern Heurisic for the Vehicle Routing Problem", ***Intl. Trans. In Op. Res. 7,*** 295-300

Held M. , Karp R. M.,(1970) "The traveling salesman problem and minimum spanning trees.", ***Operations Research, 18***, 1138-1162,1970

Laporte G. , Semet F. (1998) , Classical Heuristics for the Vehicle Routing Problem. Les Cahiers du GERAD, G98-54, Group for Research in Decision Analysis, Montreal,

Michel Gendreau, Alain Hertz, and Gilbert Laporte, (1994)"A Tabu Search Heuristic for the Vehicle Routing Problem", ***Management Science, 40***. No10

Cordeau J.F., Gendreau M., Laporte G., Potvin J. Y. and Semet F., (2002), "A guide to vehicle routing heuristic", ***Journal of the Operational Research Society, 53***, 512-522 ,

E. Taillard, (1993) "Parallel Iterative Search Methods for Vehicle Routing Problems", ***Networks, 23***, 661-673

紀玟豪，以拉格蘭式鬆弛法求解航空貨運承攬業之並裝決策問題，國立交通大學，碩士論文，民國 93 年

曾筠予，以拉格蘭式鬆弛法求解車輛定線問題，國立交通大學，碩士論文，民國 94 年

Solomon website:
http://neo.lcc.uma.es/radi-aeb/webvrp/index.html?/resultsSolom.htm

附錄

Christofides & Eilon 之 VRP 例題求解結果

Instance：n22-k4	Capacity：6000	Truck：4	Cost：375
Route 1： 1 17 20 18 15 12 1			
Route 2： 1 16 19 21 14 1			
Route 3： 1 13 11 4 3 8 10 1			
Route 4： 1 9 7 5 2 16			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n23-k3	Capacity：4500	Truck：4	Cost：569
Route 1： 1 13 12 7 2 3 4 17 16 15 18 23 21 20 19 1			
Route 2： 1 11 14 1			
Route 3： 1 22 5 6 9 10 8 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n30-k3	Capacity：4500	Truck：4	Cost：507
Route 1：1 21 4 5 6 3 23 1			
Route 2：1 20 7 2 25 26 30 28 29 27 1			
Route 3：1 19 11 12 13 16 17 14 8 18 10 9 15 24 1			
Route 4：1 22 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n33-k4	Capacity：8000	Truck：4	Cost：837
Route 1： 1 13 19 20 22 21 23 24 25 26 18 15 1			
Route 2： 1 5 8 9 10 11 33 7 6 4 1			
Route 3： 1 2 16 27 28 17 29 30 1			
Route 4： 1 31 32 14 12 3 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n51-k5	Capacity：160	Truck：4	Cost：531
Route 1： 1 12 17 51 22 35 31 10 39 13 1			
Route 2： 1 28 49 9 27 8 24 44 25 15 7 1			
Route 3： 1 2 23 32 29 4 37 36 21 30 3 33 1			
Route 4： 1 47 6 50 11 40 34 46 16 45 38 18 48 1			
Route 5： 1 5 43 20 41 42 14 26 19 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n76-k7	Capacity：220	Truck：4	Cost：700
Route 1： 1 69 3 63 23 65 43 42 44 2 74 34 7 1			
Route 2： 1 68 35 47 53 28 5 76 1			
Route 3： 1 52 17 64 24 57 50 25 4 45 33 41 18 1			
Route 4： 1 9 36 20 55 14 58 16 38 21 71 61 72 70 37 49 31 1			
Route 5： 1 13 73 40 10 51 19 56 26 32 11 59 1			
Route 6： 1 75 29 62 22 48 6 30 46 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n76-k8	Capacity：180	Truck：8	Cost：742
Route 1： 1 27 8 36 15 60 20 55 9 47 1			
Route 2： 1 76 31 75 22 62 29 63 3 69 1			
Route 3： 1 34 64 24 57 42 44 43 65 23 2 74 1			
Route 4： 1 49 48 37 70 72 61 71 21 38 6 16 58 14 1			
Route 5： 1 18 41 10 26 56 32 40 73 13 1			
Route 6： 1 54 12 67 66 39 11 59 1			
Route 7： 1 68 35 53 28 30 46 5 1			
Route 8： 1 7 17 50 25 19 51 33 45 4 52 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n101-k8	Capacity：200	Truck：8	Cost：848
Route 1： 1 13 55 25 30 79 35 36 72 66 67 21 31 71 2 70 1			
Route 2： 1 51 52 10 82 34 80 4 78 69 81 77 29 1			
Route 3： 1 95 94 86 92 101 43 44 16 58 42 23 75 74 54 1			
Route 4： 1 7 97 100 62 17 45 15 39 87 18 85 6 61 90 1			
Route 5： 1 53 8 83 49 37 50 65 64 91 33 11 32 1			
Route 6： 1 19 84 9 46 47 48 20 12 63 89 28 1			
Route 7： 1 14 96 60 99 38 93 98 88 3 59 1			
Route 8： 1 27 5 56 26 40 68 24 57 76 73 22 41 1			

Christofides & Eilon 之 VRP 例題求解結果

Instance：n101-k14	Capacity：112	Truck：14	Cost：1111
Route 1： 1 93 38 101 92 17 87 62 97 7 1			
Route 2： 1 74 42 23 76 24 68 26 56 1			
Route 3： 1 5 40 57 75 73 22 41 1			
Route 4： 1 51 80 34 82 10 52 2 1			
Route 5： 1 60 99 86 94 100 1			
Route 6： 1 21 67 66 72 36 35 79 30 25 81 1			
Route 7： 1 63 12 65 50 37 48 47 9 1			
Route 8： 1 32 11 64 91 33 31 71 70 1			
Route 9： 1 28 89 20 49 83 8 53 1			
Route 10： 1 54 27 55 13 29 1			
Route 11： 1 14 88 98 96 95 1			
Route 12： 1 59 3 58 16 44 15 39 45 43 1			
Route 13： 1 19 84 46 18 85 6 61 90 1			
Route 14： 1 77 78 4 69 1			