

### 第三章 解題架構與局部解構建模組

#### 3.1 包容性啟發式解法之解題架構設計

本研究採用門檻接受法(Threshold Accepting, TA)與大洪水法(Great Deluge Algorithm, GDA)兩種包容性搜尋啟發式方法(Generic Search Heuristics)，結合傳統鄰域搜尋方法，設計多套解題程序進而構建一套巨集啟發使解法以求解 VRPTW 問題。由於 TA、GDA 皆採用確定性的接受法則，且具執行方式簡單的優點，但在應用這些方法時，仍有許多執行程序必須借重傳統的工具。以下說明包容性啟發式解法應用在 VRPTW 之解題架構。

整套解題架構可以分為起始解構建程序與包容性改善程序。起始解構建程序中又可以分為最初起始解構建模組與鄰域搜尋構建模組，最初起始解構組中以鄰點法變化六種起始解法(NNS1、NNS2、NNS3、NNP1、NNP2、NNP3)構建而成，鄰域搜尋模組則進行 Reduction 模組與交換改善組，交換改善模組中又包括了 1-0 路線間交換、2-1 路線間交換，1-1 路線間交換以及 op3-exch 路線間交換；包容性改善程序針對起始解構建程序的結果，進行包容性搜尋。本研究以 TA 與 GDA 為包容性演算法，搜尋架構採用 G1 包容性搜尋模組，其設計概念在於增強鄰域搜尋的深度。包容性啟發式解法應用 VRPTW 之於解題架構如圖 3.1 所示。



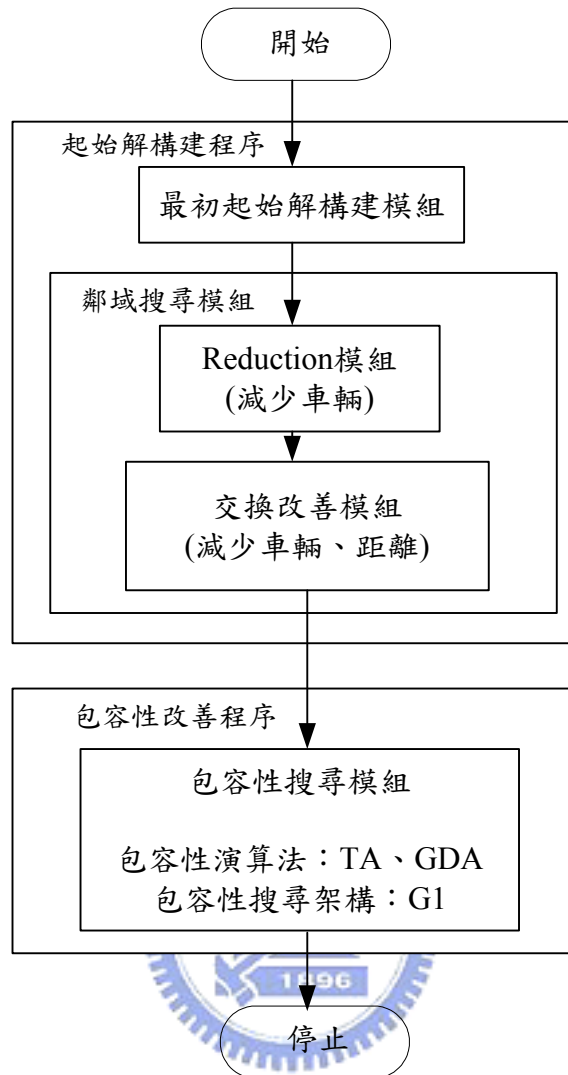


圖 3.1 VRPTW 之解題架構

## 3.2 起始解構建程序之設計

### 3.2.1 最初起始解構建模組

最初起始解構組所構建之每條路線必須滿足車輛容量限制，抵達路線上各點的時間必須滿足時間窗限制，由場站出發再回到場站的時間必須滿足路線最大時限。本研究將 Solomon(1983)[29]提出的最近鄰點法中，時間因子與空間因子單獨提出，分別為距離最短、時間窗下界最早以及等待時間最少三種準則，構建順序又分為循序構建和平行構建兩種，共六種最初起始解作為起始解構建。各最初起始解整理如表 3.1。

距離最短是將所有顧客點之間彼此的距離成本算出，在符合車容量與時間窗的限制下，挑選距離成本最小者插入到路線最末點；時間窗下界最早是將所有顧客點之時間窗下界，從早到晚的順序，在符合車容量與時間窗的限制之下，依序插入到路線最末點，倘若有兩個以上的顧客點其時間窗下界相等時，則以距離最

表 3.1 最初起始解彙整表

順序 準則	循序	平行
距離最短	距離最短-循序 NNS1	距離最短-平行 NNP1
時間窗下界最早	時間窗下界最早-循序 NNS2	時間窗下界最早-平行 NNP2
等待時間最少	等待時間最少-循序 NNS3	等待時間最少-平行 NNP3

短為第二原則進行路線插入；等待時間最少是計算車輛到達顧客點的時間，與顧客點之時間窗下界之間的時間差，挑選差值最小者插入到路線最末點，等待時間之示意圖如圖 3.2 所示。

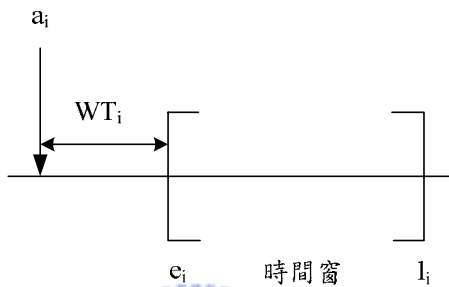


圖 3.2 等待時間之示意圖

其中  $e_i$  與  $l_i$  分別代表第  $i$  點的時間窗下界與上界， $a_i$  代表車輛到達第  $i$  點的時間， $WT_i = e_i - a_i$ ，倘若  $WT_i$  均為 0，則以距離最短為第二原則進行路線插入。

Solomon(1983)[29]提出的最近鄰點法是以循序方式進行構建，即由場站開始，符合車容量限制、時間窗限制下，以成本指標最小者之最近顧客點，加入路線末端以擴大路線，一條路線構建完畢後，再構建下一條路線，直到所有顧客皆被服務為止。而平行式構建法，則是同時計算場站與現有路線最末點，對其他尚未被服務顧客點之成本指標最小者，假如由場站出發之之成本指標最小，則新增一條路線，否則插入到成本指標最小之路線最末點之後，以此法則同時完成多條路線的送貨作業，並直到所有顧客皆被服務為止。循序與平行構建之虛擬碼以表 3.2 表示。

### 3.2.2 鄰域搜尋構建模組

本研究之鄰域搜尋模組架構如圖 3.3 所示，包含兩個部份，首先先執行 Reduction 模組，再執行交換改善模組。Reduction 模組最主要的目的是在於縮減車輛數；而在交換改善模組中，則先同時執行 1-0、2-1 路線間交換模組，再依序執行 1-1 路線間交換模組與保持方向性之 3-exch(orientation-preserving, op3-exch)路線內交換模組。其中，1-0、2-1 路線間交換模組主要的目的在於縮減車輛數與距離成本，而 1-1 路線間交換模組與 op3-exch 路線內交換模組則是與縮減距離成本為主要目的。

在實際執行交換法時，依考慮交換範圍之不同可分為「全套」與「半套」之

交換法，其間的差異分別如表 3.3 之虛擬碼(pseudo codes)所示。由表 3.2 可知，全套交換法的兩個迴圈皆是由第 1 個指標檢查至最後一個指標，而半套交換法的第二個迴圈則是由第 i 個指標檢查至最後一個指標。本研究後續各種交換改善模組執行架構中，交換範圍均考慮「全套」式交換範圍。

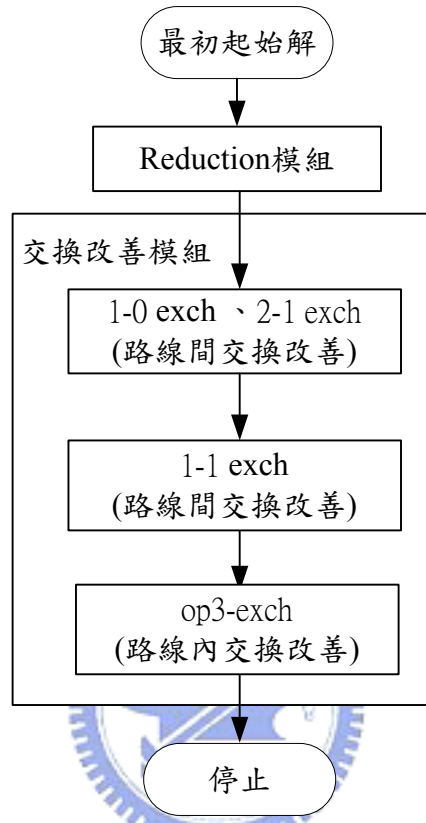


圖 3.3 鄰域搜尋執行架構

表 3.2 「全套」與「半套」交換法之虛擬碼

全套交換法	半套交換法
For( i = 1 ; i <= index_number ; i++) For(j = 1 ; j <= index_number ; j++) if (i≠j) then [exchange procedure]	For( i = 1 ; i <= index_number ; i++) For(j = i ; j <= index_number ; j++) [exchange procedure]

本研究發現，在交換模組中由於 Reduction 模組、1-0、2-1 路線間交換模組，均具有縮減車輛數的特性，因此本研究在上述交換模組中的成本計算公式方面，彈性增加車輛剩餘容量因素加以考量，即複合式成本計算公式  $\Delta C'(x)$  進行交換改善， $\Delta C'(x)$  之複合式成本計算公式如式(3-1)：

$$\Delta C'(x) = \Delta c - w \times \frac{\bar{c}}{(K - TD / F_{\min})} \times SK_i < 0 \dots\dots\dots(3-1)$$

其中： $\Delta c$  = 傳統距離計算公式

$w$  = 車輛剩餘容量權重值

TD(Total Demand)：總需求

$\bar{c}$  = 平均節線成本  
 K = 車容量

$F_{\min}$  = 最少需使用之車輛數  
 $SK_i$  = 第 i 條路線之車輛剩餘容量

複合式成本計算公式  $\Delta C'(x)$  與傳統距離因素的成本計算公式  $\Delta c$  不同的地方在於， $\Delta C'(x)$  不僅可以單獨考量距離因素，更可以彈性的增加考量車輛剩餘容量，藉此突顯車容量使用之效率，此期望可以更有效的達到縮減車輛數的目的與效果。公式中， $\Delta c$  為一般交換模組中距離因素的交換公式，不同交換模組， $\Delta c$  不同； $w$  為控制車輛剩餘容量的權重值，當  $w = 0$  時， $\Delta C'(x)$  與  $\Delta c$  相等，若  $w \neq 0$ ，隨著  $w$  值的增加，車輛剩餘容量影響效果也逐漸增加； $\bar{c}/(K - TD/F_{\min})$  代表距離成本與車輛剩餘容量之間正規化的轉換公式； $SK_i$  則代表第  $i$  條路線之車輛剩餘容量，不同的路線  $SK_i$  值均不同。而 1-1 exch 與 op3-exch 交換模組，由於只有縮減距離成本的特性，因此交換成本公式則採用一般距離因素  $\Delta c$  進行交換改善。

在執行鄰域搜尋尋找可交換的鄰近解時，決定選擇哪一個鄰近解進行交換的準則稱為「選擇策略(selection strategy)」。根據 Osman[22] 的文獻指出，選擇策略一般有兩種：(1) 最佳改善(best-improve)策略，即從所有搜尋範圍內的鄰近解中，選擇一個改善最多的解進行交換；(2) 首先改善(first-improve)策略，則是在搜尋過程中，只要鄰近解有改善就進行交換。本研究在進行鄰域搜尋時，各交換模組均採用「最佳改善」策略進行交換。

### 一、Reduction 減少車輛模組

Salhi & Rand[30]之擾動程序(Perturbation Procedure, SRP)中的 Reduction 模組。以圖 3.4 說明解題概念，將某條路線上的顧客點全部打散，在符合時間窗與車容量限制下，分別插入到其他路線內，若顧客點可以完全插入其他路線時，則縮減車輛數，若否則回復原路線，進行下一條路線的插入動作。由於 Reduction

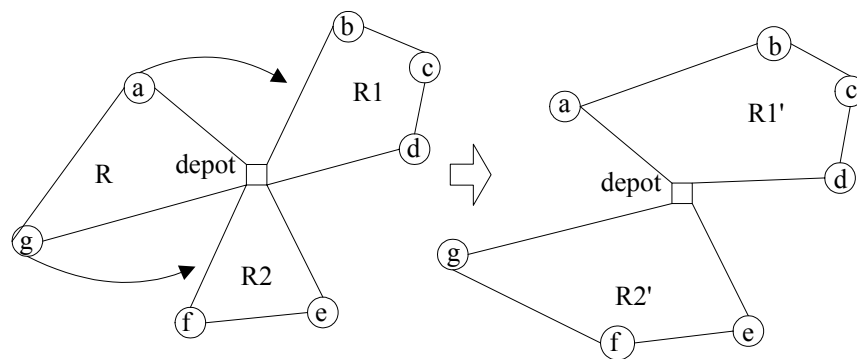


圖 3.4 Reduction 解題概念

模組目的在於縮減車輛數，因此交換成本採用複合式成本計算公式  $\Delta C'(x)$  進行交換改善，其中  $\Delta c$  計算方式以 a 點為例，如式(3-2)所示：

$$\Delta c = (C_{depot,a} + C_{a,b} - C_{b,depot}) \dots \dots \dots (3-2)$$



## 二、交換改善模組

交換改善模組共可分為兩類：(1)路線間改善模組，與(2)路線內改善模組。以下對各交換模組分別作詳細的介紹與說明。

### (1) 路線間改善模組

Osman[22]將(1-0)節點交換法稱為一個轉移過程(shift process)，而將(1-1)及(1-2)節點交換法稱為 $\lambda$ -路線間節點交換法( $\lambda$ -interchange)。假設現在有一個可行解集合  $S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_v\}$ ，其中  $R_p$  表被同一輛車服務的路線  $p$  中之顧客集合。所謂  $R_p$  與  $R_q$  兩個路線間之 $\lambda$ -交換法是指在滿足可行性條件的情況下，將兩個子集合相互交換後得一個包含兩個新路線集合之新的可行解。其中兩子集合分別為： $S_1 \subseteq R_p$ ，子集合大小為  $|S_1| \leq \lambda$  及  $S_2 \subseteq R_q$ ，子集合大小為  $|S_2| \leq \lambda$ ；交換後得到兩個新的路線集合分別為： $R_p' = (R_p - S_1) \cup S_2$  及  $R_q' = (R_q - S_2) \cup S_1$ ；而最後獲得之新可行解集合為  $S' = \{R_1, \dots, R_p', \dots, R_q', \dots, R_v\}$ 。通常  $\lambda = 1$  或  $2$ 。而在(1-0)、(1-1)、(1-2)路線間結點交換法具有減少車輛數以及減少距離成本的功能，(1-1)路線間結點交換法僅僅只有減少距離成本的功能。以下分別對(1-0)、(1-1)、(1-2)路線間節點交換法做詳細的說明。

#### (a) 1-0、2-1 路線間節點交換模組

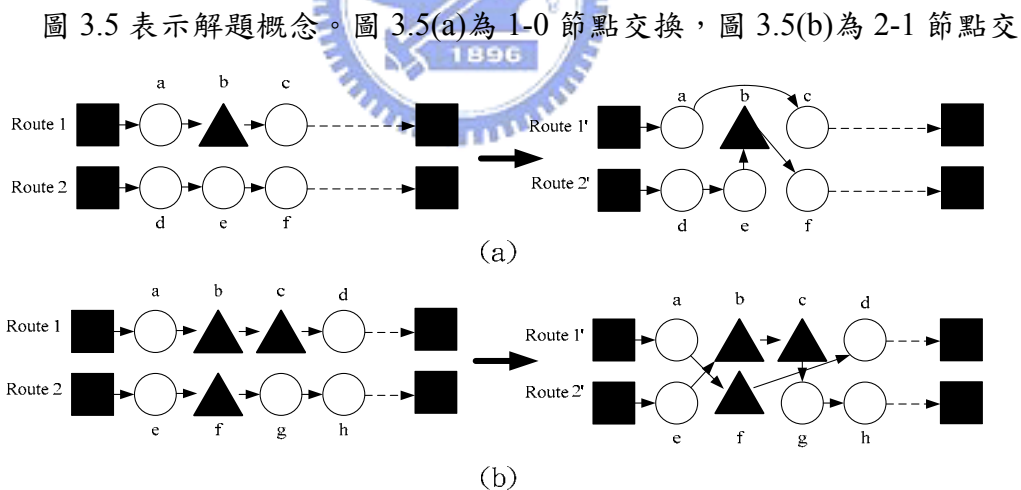


圖 3.5 1-0、2-1 路線間節點交換解題概念

換。主要的解題概念在於符合時間窗與車容量的限制下，1-0 路線間節點交換為將一條路線中一個顧客點插入另一條路線內之任兩連續顧客之間；2-1 路線間節點交換則是將一條路線中其中任兩個連續顧客點與另一條路線中任一顧客點做交換。由於兩種交換法彼此具有相似的交換行為，避免重複執行，本研究同時執行兩種交換法，挑選改善成本較大者進行改善；且兩種交換法又都有縮減車輛數與距離成本的特性，因此採用複合式成本計算公式  $\Delta C'(x)$  進行交換改善，交換公式如式(3-1)所示。其中  $\Delta c$  則根據不同的交換法，進行不

同的距離成本計算公式，以圖 3.5 為例，1-0 節點交換中  $\Delta c$  如式(3-3)所示：

$$\Delta c = (C_{e,b} + C_{b,f} + C_{a,c} - C_{a,b} - C_{b,c} - C_{e,f}) < 0 \dots\dots\dots(3-3)$$

2-1 節點交換中  $\Delta c$  如式(3-4)所示：

$$\Delta c = (C_{a,f} + C_{f,d} + C_{e,b} + C_{c,g} - C_{a,b} - C_{c,d} - C_{e,f} - C_{f,g}) < 0 \dots\dots\dots(3-4)$$

(b) 1-1 路線間節點交換模組

1-1 節點交換法以下圖 3.6 表示解題概念。在符合時間窗與車容量限制之

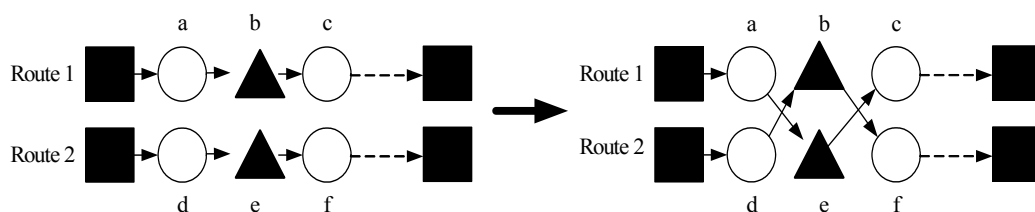


圖 3.6 1-1 路線間節點交換解題概念

下，將一條路線的一個顧客點與另一條路線的一個顧客點進行交換。1-1 節點交換並無縮減車輛數之特性，只有縮短距離成本之功用，因此交換成本採用一般距離因素  $\Delta c$  進行交換，計算公式如式(3-5)所示：

$$\Delta c = C_{a,e} + C_{e,c} + C_{d,b} + C_{b,f} - C_{a,b} - C_{b,c} - C_{d,e} - C_{e,f} < 0 \dots\dots\dots(3-5)$$

(2) 路線內改善模組：

本研究路線內改善模組採用 op3-exch 交換法，由 Pesant& Gendreau (1999)[15]提出。一般路線內節點交換均面臨交換路線內交換路段之路徑必須反轉的現象，對於時間窗車輛路線問題，此現象較會違反時間窗的限制，解集合較小。因此本研究在路線內交換模組採用「保持方向性的三-三節線交換」(orientation-preserving 3-exch)的 op3-exch。特色在於路線內交換路段之路徑不必反轉，對於具有時間窗特性的問題，可以找到較多的解集合，解題概念以圖 3.7 所示。打斷路線內任意三段不同節線  $\overline{ab}$ 、 $\overline{cd}$  與  $\overline{ef}$ ，使路線分成四部分，在符合時間窗條件下，保持  $\overline{bc}$ 、 $\overline{de}$  內顧客的服務順序，並交換  $\overline{bc}$ 、 $\overline{de}$  兩部份顧客群的服務順序，達到縮減距離的目的。其中顧客點 b、c 與顧客點 d、e 均可為相同或相異點。由於 op3-exch 路線內節點交換僅只有減少距離成本的特性，因此交換成本採用一般距離因素  $\Delta c$  進行交換，計算公式如式(3-6)所示：

$$\Delta c = C_{a,d} + C_{e,b} + C_{c,f} - C_{a,b} - C_{c,d} - C_{e,f} < 0 \dots\dots\dots(3-6)$$

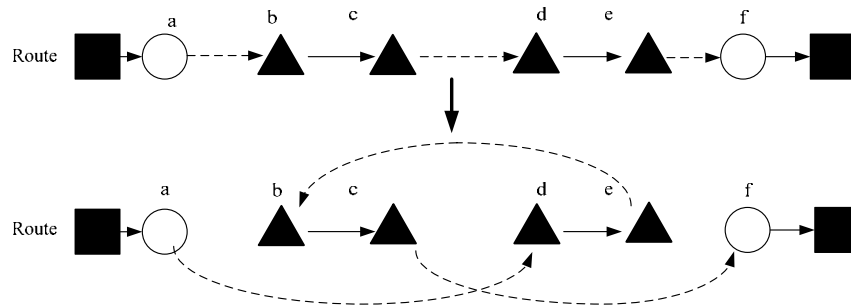


圖 3.7 保持方向性之路線內 3-exch 路線交換解題概念

### 3.3 起始解測試結果分析

本節針對前述之起始解構建程序加以測試與分析。測試題庫如 2.4 節所述，以 Solomon(1983)[29]發表 56 題國際標準題庫為測試例題。解題績效分析包括兩個項目：(1)解題精度：第一目標為車輛數，第二目標為路線成本；(2)解題速度：用以評估啟發式演算法之執行效率，一般使用 CPU 時間來表示。由於各種電腦系統的速度水準並不一致，且影響演算法之執行速度快慢之因素太多，在此 CPU 時間僅提供為評估時之參考指標。在起始解構建程序之測試中，CPU 運算時間則均為 0。

本研究之所有執行架構以 C++ 語言撰寫程式，並在 Pentium IV 3G Hz 個人電腦建立 Windows XP 系統執行測試所有例題。

在起始解構建程序中，共包含「最初起始解構建」與「鄰域搜尋」兩種模組。本研究首先測試各種起始解之解題績效，再配合鄰域搜尋法測試其改善效果。

#### 3.3.1 最初起始解構建模組之測試

本研究首先對最初起始解法進行初步測試。起始解法如 3.2.1 節所述包括距離最短-循序(NNS1)、時間窗下界最早-循序(NNS2)、等待時間最少-循序(NNS3)、距離最短-平行(NNP1)、時間窗下界最早-平行(NNP2)、等待時間最少-平行(NNP3)。以下分別就六種起始解法對 56 題例題測試結果加以說明。

表 3.3 六種最初起始解測試結果

最初起始解類別	NNS1	NNS2	<b>NNS3</b>	NNP1	NNP2	<b>NNP3</b>
車輛數( $Z(X_0)$ )	879	939	<b>526</b>	1053	801	<b>557</b>
距離成本( $C(X_0)$ )	101876.1	156930.9	<b>98494.9</b>	95028.59	83543.43	<b>87881.3</b>

最初起始解測試結果整理如表 3.3，並以圖 3.8 表示結果分佈範圍。績效指標包括車輛數、距離成本。在表 3.3 中黑體字部分表示，起始解表現最佳的結果為 NNS3，車輛數為 526 輛，距離成本為 98494.9。整體來看，車輛數方面以循序方式構建起始解所得的結果較佳，其中又以 NNS3 的表現最好，車輛數為 526 輛、距離成本為 98494.9，而距離成本方面以平行方式構建起始解所得的結果較



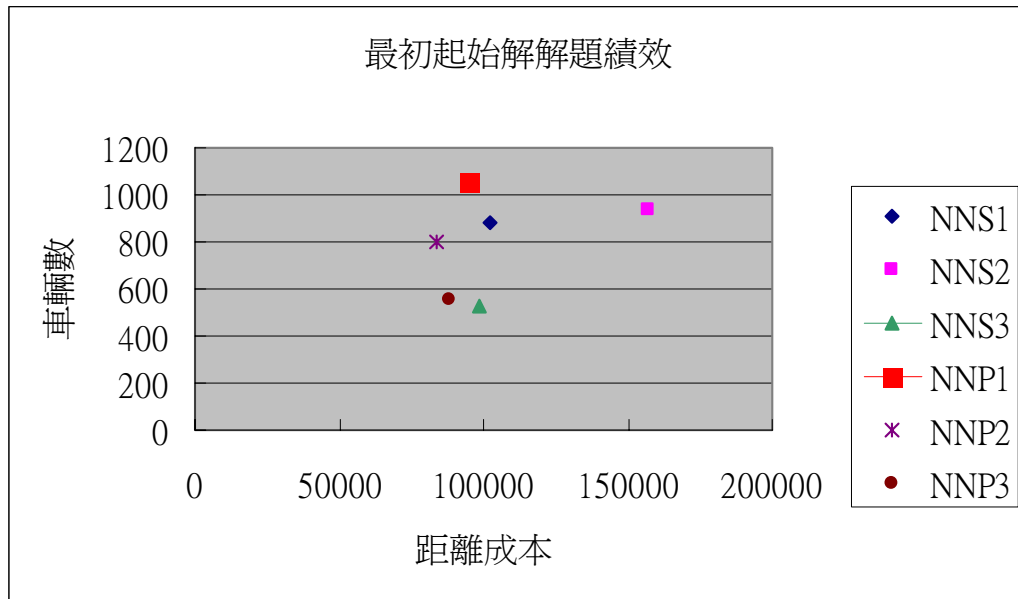


圖 3.8 最初起始解解題精確度比較

佳，其中又以 NNP3 的表現較好，車輛數為 557 輛、距離成本為 87881.3。無論循序或平行構建法方面，均以等待時間最少的策略表現最好，但在循序方面，以距離最短的策略次之，時間窗下界最早的策略最末；而在平行方面則以時間窗下界最早之策略次之，距離最短的策略最末。起始解以 NNS3 與 NNP3 兩組的表現結果較佳，因此本研究以 NNS3 與 NNP3 配合鄰域搜尋模組進行下一階段之測試。

### 3.3.2 鄰域模組之測試

如 3.2.2 節所述，鄰域搜尋模組包括減少車輛(Reduction)模組、路線間 1-0、2-1 節點交換模組、1-1 節點交換模組、路線內 op3-exch 節線交換模組，其中 Reduction 模組、1-0、2-1 節點交換模組中，成本改善公式採用複合式成本計算公式  $\Delta C'(x)$  進行交換改善，在  $\Delta C'(x)$  中則具有車輛剩餘容量權重值  $w$  之考量因素。因此根據 3.2.2 節之執行架構，配合起始解-NNS3、NNP3，本節針對 56 題測試例題，在鄰域搜尋模組中進行不同車輛剩餘容量權重值  $w$  之測試與分析。

在鄰域搜尋中，測試之權重值  $w = 0 \sim 4.5$ 。當  $w = 0$  時，成本交換僅考慮距離因素，即傳統交換模組所採用的  $\Delta c$ ；而當  $w \neq 0$  時，成本交換多增加車輛剩餘容量因素，即本研究之複合式成本計算公式  $\Delta C'(x)$ 。測試結果以表 3.4 表示，並以圖 3.9、圖 3.10 表示車輛數與距離成本在不同權重值下之表現。圖中可以發現到，無論起始解為 NNS3 或是 NNP3，當鄰域搜尋模組之 Reduction、1-0、2-1 交換模組的交換成本中， $w \neq 0$  之車輛數與距離成本均較  $w = 0$  時要來的低，此代表在交換改善時適時的增加車輛剩餘容量之因素，較僅單純考量距離因素可使車輛數與距離成本更有效的下降。所有測試結果中，其中以 NNS3 在權重值  $w = 0.5$  時，車輛數與距離成本表現最佳，車輛數為 472、距離成本為 64575.58。因此，本研究以 NNS3 為起始解，配合鄰域搜尋中車輛剩餘容量之權重值  $w = 0.5$  的結果，以兩種權重值  $w = 0$  與 0.5，進行下一階段-包容性改善模組之測試與分析。

表 3.4 鄰域搜尋模組之權重值  $w$  測試結果

最初起始解-NNS3 鄰域搜尋之結果										
權重值	w = 0	w = 0.5	w = 1	w = 1.5	w = 2	w = 2.5	w = 3	w = 3.5	w = 4	w = 4.5
車輛數	477	472	473	472	474	476	475	475	475	475
距離成本	65198.57	64575.58	64782.33	64670.59	64502.42	64694.43	64702.16	64666.87	64718.33	64661.69
最初起始解-NNP3 鄰域搜尋之結果										
權重值	w = 0	w = 0.5	w = 1	w = 1.5	w = 2	w = 2.5	w = 3	w = 3.5	w = 4	w = 4.5
車輛數	478	477	478	478	476	475	476	477	477	476
距離成本	65203.83	64602.57	64147.55	64263.79	64473.94	64446.42	64598.4	64623.05	64615.65	64721.07

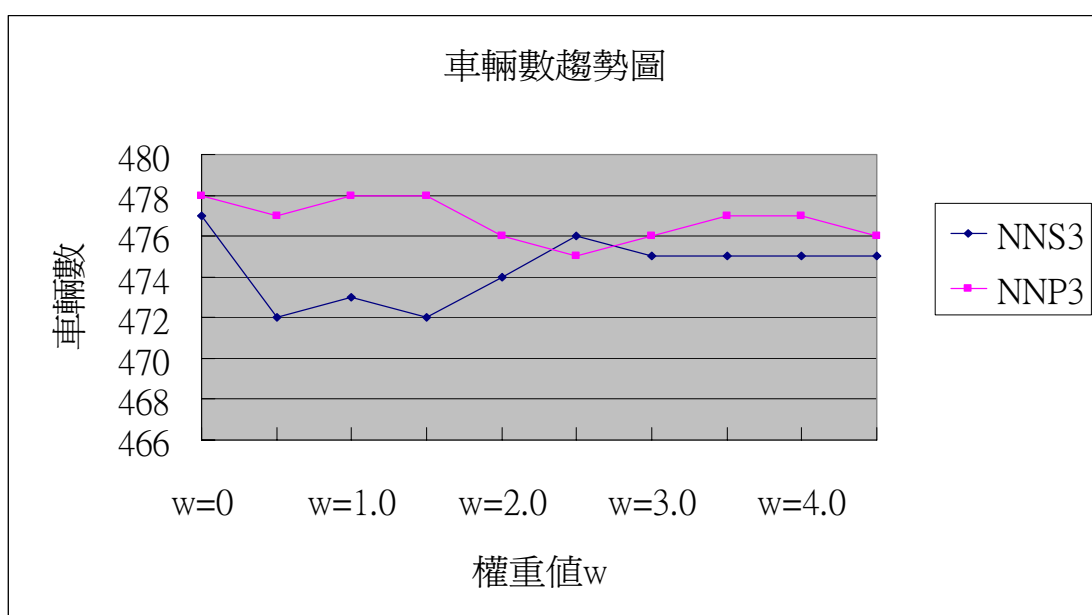


圖 3.9 起始解 NNS3 與 NNP3 之車輛數趨勢圖

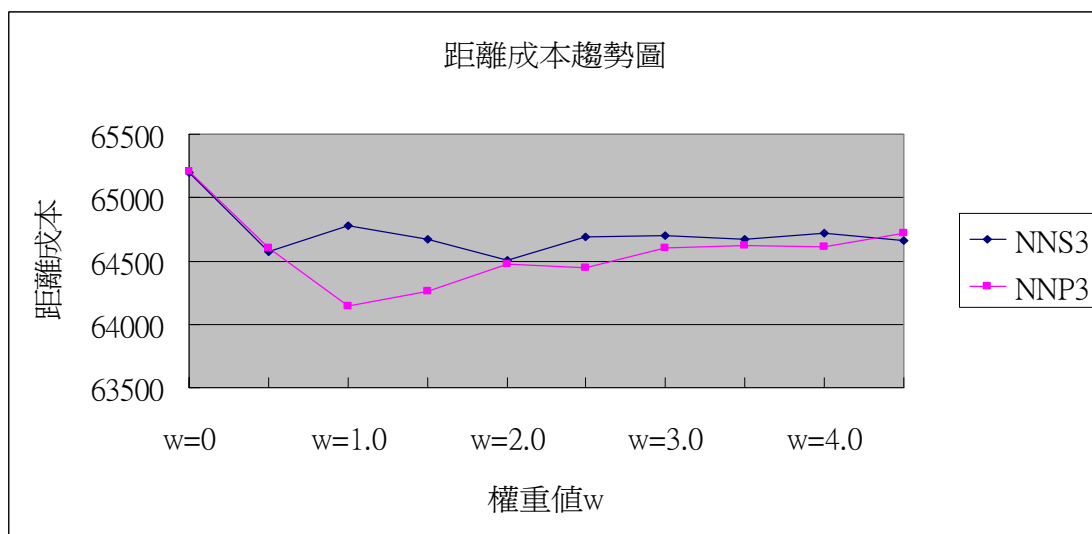


圖 3.10 起始解 NNS3 與 NNP35 之距離成本趨勢圖