

第一章 緒論

1.1 研究背景與動機

隨著台灣產業自由化、資訊化的快速發展以及產品短暫的生命週期促使更準時、更快速的交貨服務，再加上近年加入 WTO 的壓力下，使得企業間的競爭激烈化且多元化，並且在顧客導向的市場下，意識到物流配送才是企業體保持競爭優勢與永續發展的主要因素之一。根據中華民國物流協會[31]於 2000 年對台灣企業物流成本統計調查顯示，國內大部分企業之物流成本佔其營運總額的 15%~25%，其物流成本佔營業額的比例和物流發展發達之國家(例如，日本、美國)相較之下高出甚多，顯示我國的物流效率仍有很大空間的改善。在物流成本中，運送成本所佔之比例約為 25%~45%，因此若能有效改善運輸配送成本，便能進一步降低物流成本，進而提高企業競爭力。

有鑑於此，車輛途程規劃儼然成為企業重視之方向。然而，由於台灣的生活環境，住宅密集、街道狹小，若以大車在都會區進行配送可能在某些路段會因車輛體積過於龐大造成進出困難，而無法準時送達。但若是利用小車進行配送雖在狹窄之街道進出方便，但卻因可承載容量很小，需經常回物流中心補貨、或額外增添車輛進行配送，易造成往返時間與成本的浪費。為了配合台灣市區街道狹窄，但在大量運輸以節省成本的考量下，本研究延續了 Gerdessen[10]所提出的卡車-拖車配送組合的概念，並調整其限制條件，以期更加符合現實配送環境條件。

所謂「卡車拖車路線問題」(Truck and Trailer Routing Problem, TTRP)屬於「車輛路線問題」(Vehicle Routing Problem, VRP)的衍生性問題。以卡車附掛拖車進行配送，為一種新的物流服務型態，其特點為可將整個服務路線分為整車(卡車附掛拖車)服務之主路線及卡車服務之子路線。當整車服務至主路線的停放點(Parking Place)時將拖車留於該處，以卡車服務較不便到達的顧客(子路線顧客)，譬如位於擁擠市區的顧客，或不易以整車到達的地區；待子路線顧客服務完畢後再回到原停放點與拖車相接，繼續服務主路線的顧客群。此種車輛組合之配送型態，使得配送過程更具彈性，不僅解決大型車行駛市區道路的不便性，亦克服小型車配送的容量限制，對於節省運輸成本有相當之助益。

1.2 研究內容

TTRP 問題屬於新型 VRP 衍生問題，近年來已逐漸受到學者的關注，但就文獻方面，似乎僅有 Gerdessen[10]與 Chao[6]提出相關啟發式解法。表 1.1 為將近年研究 TTRP 相關文獻作一簡單整理：

表 1.1 TTRP 文獻回顧

年代	作者	假設條件	解法
1996	Gerdessen	(1) 每個顧客僅有單位需求量 (2) 每一整車路線中拖車僅能停放一次 (3) 每一顧客點均可當作拖車停放點	以最省插入法構建起始解；以傳統交換法進行路線改善
2002	Chao	(1) 將顧客區分為整車服務顧客與卡車服務顧客 (2) 不限制拖車停放次數 (3) 每個顧客非單位需求量	導入確定型鍛鍊法之深度搜尋觀念於 TS 中，發展出一套求解 TTRP 之巨集啟發式解法

由上述回顧可知，過去研究中對 TTRP 問題著墨不多，且其研究中為簡化問題複雜度提出不少假設條件，然而這些成本與限制之假設忽略現實狀況，降低實務上應用的可行性，如：(1)整車配送與卡車配送之成本均相同；(2)未考慮拖車停放成本(例如：停車費)；(3)未考慮拖車卸下與連結之操作時間成本。

本研究之目的在於放寬上述假設與限制，期使該問題成為「一般化卡車拖車路線問題」(Generalized Truck and Trailer Routing Problem, GTTRP)以更接近現實物流配送問題，降低物流運輸成本。與傳統 TTRP 比較，GTTRP 的一般化考慮包括：

1. 成本構建合理化

- (1)考慮停放點(Parking Place)之成本，如停車費、保管費等。
- (2)考慮拖車卸下及連結之時間成本。
- (3)配送過程考慮車輛以整車型態或卡車型態配送之變動成本。
- (4)考慮整車與卡車使用成本。

2. 限制條件

- (1)考慮整車卸下、連結拖車之地點限制(如現實生活中某些路段不可停車之限制)。
- (2)將顧客區分為整車顧客與卡車顧客。

1.3 研究方法與流程

本研究考慮 GTTRP 問題，將考慮納入各種複雜的限制需求。如此一來，雖然使本研究更能符合於現實應用，但相對的數學模式變得更為複雜。基本上 GTTRP 亦符合「限制滿足問題」(Constraint Satisfaction Problem, CSP)之特質。所謂 CSP 問題係指在給定一組變數及各變數相對應限制式的有限值域下，尋找滿足每一項限制式之解。CSP 問題在本質上不同於最佳化(Optimization)問題，其解題架構中並沒有目標式的存在，而是由一群限制式圍成可行解區域，限制滿足問題之求解目的，在於尋得一組或多組可行解(Feasible Solution)。

所謂「限制規劃」(Constraint Programming, CP)，是指用電腦(程式)來執行求解 CSP 問題的過程(Bradford et. al. [1])。CP 方法對於求解 CSP 問題，具有模式建立方便、可處理多樣限制式、強大空間搜尋能力等優點，因此，本研究擬應用限制規劃方法並配合傳統啟發式構建法於起始解構建模組當中，以求得較佳之起始解。接著，再應用可跳脫局部最佳解的巨集啟發式解法進行起始解的改善，以期提升整體演算法之解題績效。整個研究之求解架構如圖 1.1 所示。

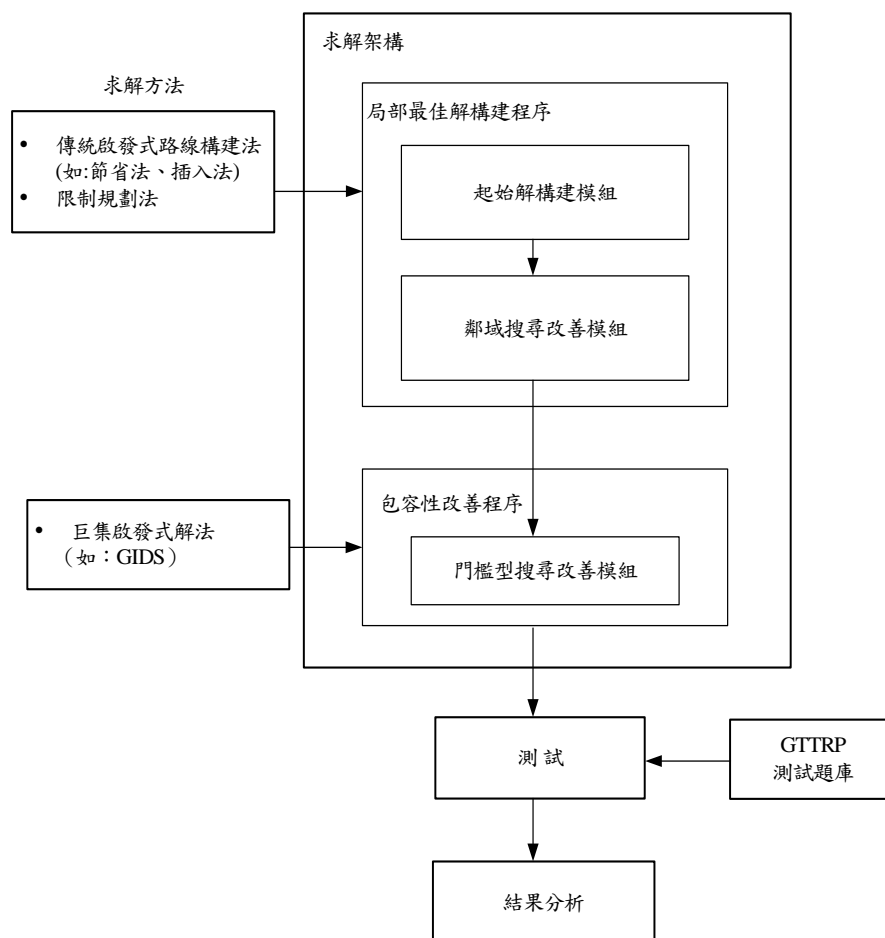


圖 1.1 GTTRP 求解架構

針對 GTTRP 問題之測試，本研究擬參考 Chao[6]之測試例題進行適當的整合、修正與調整，期望設計出合適 GTTRP 問題之測試例題，以驗證本研究所提出之求解架構的解題效率。

本研究之內容與流程(如圖 1.2)分述如下：

1. 相關文獻蒐集與回顧：蒐集、回顧國內外對於車輛路線相關問題(Vehicle Routing Related Problems, VRRPs)及卡車拖車相關問題之解題方法以及新進巨集啟發式方法的發展情況。
2. 測試題庫之蒐集與修正：參考 Chao[6]所提之測試題庫進行適當整合與修正。
3. 起始解建構設計與建立：擬應用限制規劃方法並配合傳統啟發式路線構建法於 GTTRP 問題之起始解構建模組當中，以求得較佳之起始解。
4. 巨集搜尋改善模組之設計與建立：根據 GTTRP 問題的特性，結合數種有效率的鄰域搜尋法構建巨集啟發式方法應用於 GTTRP 之執行架構與模組。由於 GTTRP 之解題難度更甚於 VRP，因此必須特別針對巨集啟發式方法的控制參數與模組設計深入研究，以提升整體解題績效。。
5. 例題測試之實驗設計與參數設定：為瞭解本研究所使用的巨集啟發式方法之解題特性與績效，需對演算法之執行架構與控制參數進行有系統的實驗設計。
6. 程式撰寫與執行：本研究之程式撰寫擬以 ILOG Dispatcher 與 C⁺⁺軟體進行程式撰寫。
7. 測試結果比較與分析：將圖 1.1 巨集啟發式方法之解體架構，以上述實驗設計之方式，對例題進行測試，以探討其應用潛力。
8. 結論與建議：根據綜合分析所得結果，提出具體的結論與建議，並研擬未來後續研究方向與重點項目。

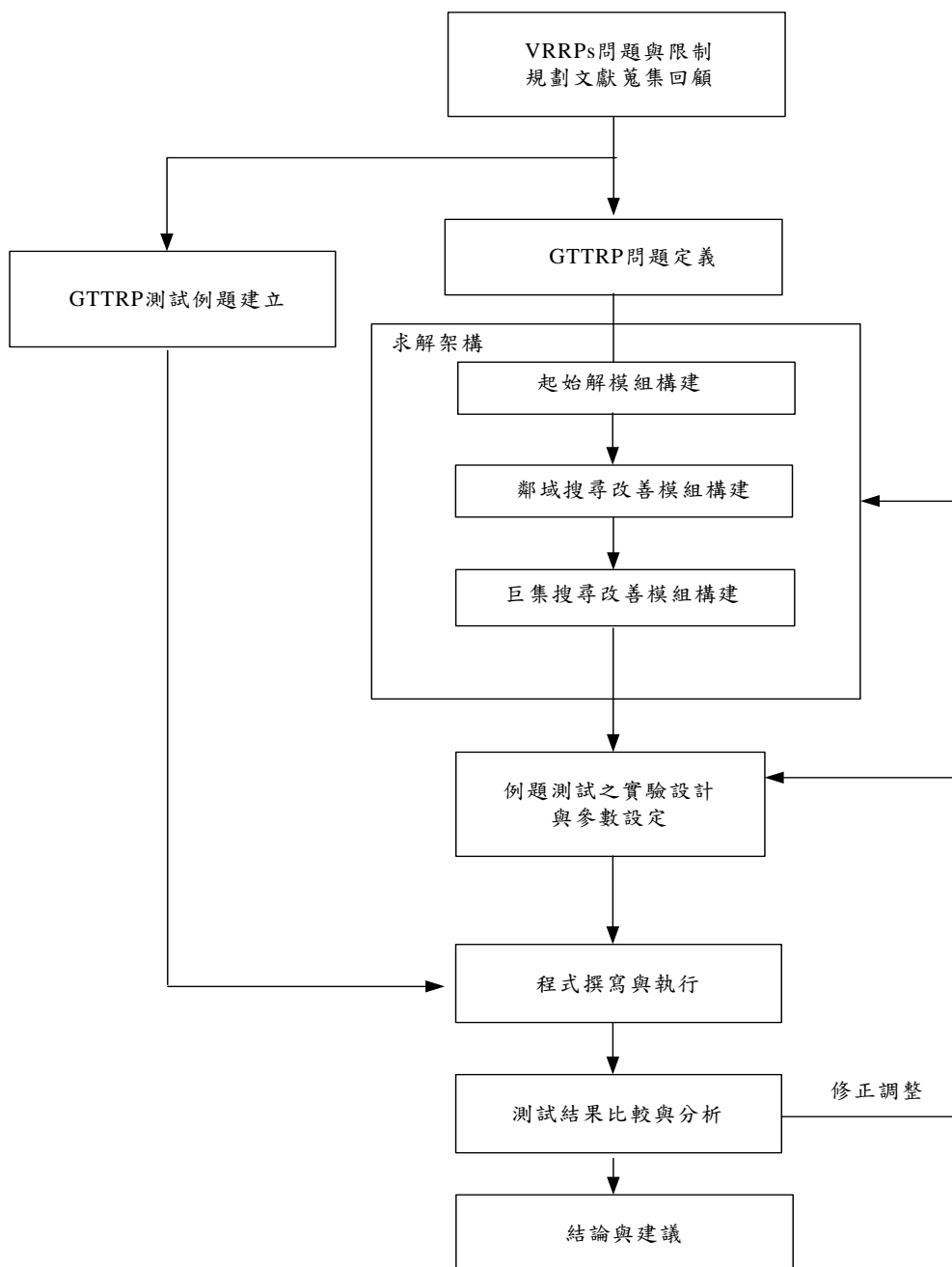


圖 1.2 本研究流程圖

第二章 文獻回顧

2.1 車輛路線相關問題

車輛路線相關問題(Vehicle Routing Related Problems, VRRPs)，屬於網路組合最佳化中節點服務類型之問題型態，其中以旅行推銷員問題(Traveling Salesman Problem, TSP)與車輛路線問題(Vehicle Routing Problem, VRP)最廣為人熟悉亦是節點服務類型中最基本之問題；TSP 問題係在無車輛容量限制下服務單一路線之節點服務問題，VRP 問題則是進一步考慮車輛容量限制，屬於多路線之節點服務問題。圖 2.1 分別表示了兩種問題型態。

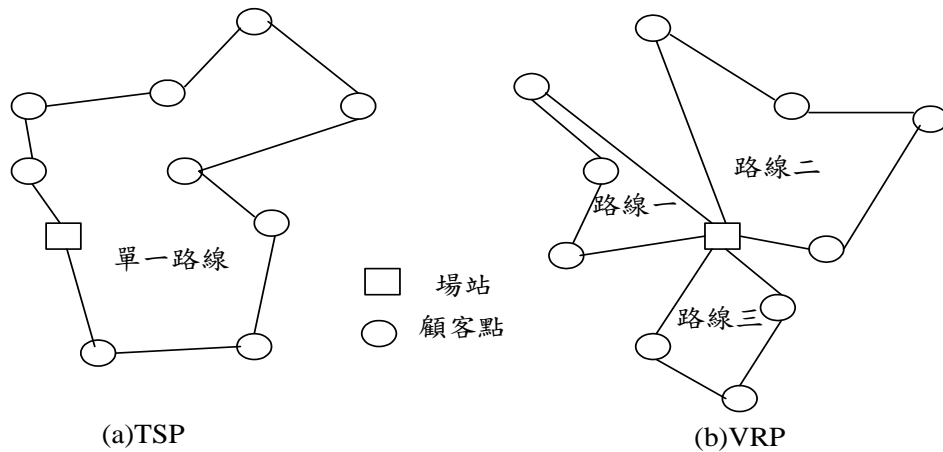


圖 2.1 TSP 與 VRP 之路線型態

VRRPs 在考慮顧客需求、設施資源以及配送條件等限制條件下，追求一般化成本(Generalized Cost)總和最小之目標。所謂一般化成本係指在配送過程中可能發生的種種負面效果並進一步予以量化表示，其中以運輸成本為最常見的負面效果。運輸成本可分為固定成本與變動成本兩種：固定成本包含車輛使用成本(如：折舊、購買成本)與司機薪資；變動成本則與配送路線中之排程相關(如：裝卸貨時間、車輛行駛距離等)。然而除了追求一般化成本最小化外，VRRPs 亦可考慮不同之目標函數；例如，配送公平性、車隊規模最小化以及服務水準極大化等目標。

然而隨著實務狀況之差異以及限制條件之不同，VRRPs 自 TSP 與 VRP 衍生出許多複雜之問題型態，例如：多車種車輛路線問題(Heterogeneous Fleet VRP, HVRP)、多場站車輛路線問題(Multi-Depot VRP, MDVRP)，以及考慮顧客允許服務時段之條件限制下的時間窗車輛路線問題(VRP with Time Windows, VRPTW)等等。VRRPs 問題之間往往只因改變一或二個限制條件而形成不同類型之問題，其間的差異性雖不大，但是其求解方法卻可能有著極不同的設計理念，並且

隨著限制條件的增加造成問題求解亦更加困難。表 2.1 列出部分 VRRPs 典型問題之基本目標與限制條件，實務應用可視問題加以修正、調整。

本研究即以 VRP 問題為基礎，進一步考慮配送方式、限制條件衍生成新的 GTTRP 問題型態。後續幾個小節，除了定義出 VRP 問題外，並對相關啟發式解法、卡車拖車路線問題(Truck and Trailer Routing Problem, TTRP)作一回顧。

表 2.1 傳統 VRRPs 問題類型

問題類型	目標函數	節點服務	顧客服務	設施資源
旅行推銷員問題 (Traveling Salesman Problem, TSP)	路線成本最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 	顧客皆需服務一次，但無需求	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 無容量限制
車輛路線問題 (Vehicle Routing Problem, VRP)	路線成本總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 單一車種 ◆ 有容量限制 ◆ 最大距離/時間
多車種車輛路線問題 (Heterogeneous Fleet VRP, HVRP)	車輛成本與路線成本之總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 多車種 ◆ 有容量限制 ◆ 最大距離/時間
週期性車輛路線問題 (Periodic VRP, PVRP)	多日之路線成本總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 ◆ 配送頻率 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 單一車種 ◆ 有容量限制 ◆ 車輛數限制 ◆ 最大距離/時間
時間窗車輛路線問題 (VRP with Time Windows, VRPTW)	路線成本(時間)總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 ◆ 時間窗限制 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 單一車種 ◆ 有容量限制 ◆ 最大時間
回頭車車輛路線問題 (VRP with Backhauls, VRPB)	路線成本總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 先送後收 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 單一車種 ◆ 有容量限制 ◆ 最大距離/時間
多場站車輛路線問題 (Multi-Depot VRP, MDVRP)	路線成本總和最小	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 固定需求 ◆ 不可分割 	<ul style="list-style-type: none"> ◆ 多場站 ◆ 單一車種 ◆ 有容量限制 ◆ 最大距離/時間
動態車輛路線問題 (Dynamic VRP, DVRP)	路線成本總和最小 (動態路線成本)	<ul style="list-style-type: none"> ◆ 流量守恒 ◆ 避免子迴路 ◆ 只收或只送 	<ul style="list-style-type: none"> ◆ 隨機需求 ◆ 不可分割 	<ul style="list-style-type: none"> ◆ 單一場站 ◆ 單一車種 ◆ 有容量限制 ◆ 最大距離/時間

資料來源：[33]

2.2 車輛路線問題定義與解法回顧

2.2.1 VRP 問題定義

自 1959 年 Dantzig & Ramser 首先提出 VRP 問題以來，VRP 問題成為網路最佳化問題中最基本的問題型態之一。VRP 問題定義：「設有一場站(depot)令其為 0，共有 M 輛貨車，其容量皆為 Q；給定 N 位顧客，每位顧客有其需求量 $D_i(i=1\sim N)$ ； C_{ij} 為貨車從顧客 i 到顧客 j 的成本(含場站)。在不違反車輛容量限制條件下，使用 k 輛貨車自場站出發服務所有顧客後再回到場站，並求所有車輛路線的總成本最小；其中，每位顧客僅能被一輛車服務一次，且每輛車服務的顧客不重複」。傳統上所說的 VRP 問題，若未加指明，係指：單一場站、單一車種、固定節線成本與需求、有容量限制以及最小路線成本為目標之 VRP。一般而言，網路問題可用數學規劃模式表示。以下為 VRP 之數學規劃模式[1][5]：

$$\text{Minimize } \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^M C_{ij} x_{ij}^k \quad (2-1)$$

$$\text{subject to } \sum_{i=0}^N \sum_{k=1}^M x_{ij}^k = 1 \quad (j = 1, \dots, N) \quad (2-2)$$

$$\sum_{j=0}^N \sum_{k=1}^M x_{ij}^k = 1 \quad (i = 1, \dots, N) \quad (2-3)$$

$$\sum_{i=0}^N x_{ih}^k - \sum_{j=0}^N x_{hj}^k = 0 \quad (h = 0, \dots, N; k = 1, \dots, M) \quad (2-4)$$

$$\sum_{i=0}^N D_i \left(\sum_{j=0}^N x_{ij}^k \right) \leq Q \quad (k = 1, \dots, M) \quad (2-5)$$

$$\sum_{j=1}^N x_{0j}^k \leq 1 \quad (k = 1, \dots, M) \quad (2-6)$$

$$\sum_{i=1}^N x_{i0}^k \leq 1 \quad (k = 1, \dots, M) \quad (2-7)$$

$$y_i - y_j + N \sum_{k=1}^M x_{ij}^k \leq N - 1 \quad (i \neq j = 1, \dots, N) \quad (2-8)$$

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

$$y_i \geq 1 \quad \text{for all } i \quad (2-10)$$

C_{ij} ：代表從顧客 i 到顧客 j 的行駛成本

x_{ij}^k ：0-1 整數變數；當車輛 k 從顧客 i 到顧客 j 時其為 1，否則為 0

D_i ：顧客 i 需求量

Q ：表示車輛容量限制

N ：顧客數

M : 車輛數
k : 表第 k 輛車
 y_i : 為任意實數

其中，(2-1)式為求總路線成本最小之目標式；(2-2)式與(2-3)式係限制每一位顧客只能被一輛車服務；(2-4)式為流量守恆限制式；表示到達顧客 i 與離開顧客 i 的車為同一輛車；(2-5)式為車輛所服務顧客的需求量不得超過車輛容量之限制式；(2-6)式與(2-7)式指明並非所有的車輛都要使用；(2-8)式為避免產生子迴路的限制式；(2-9)式則為雙元整數限制式，若決策變數 $x_{ij}^k=1$ 表示使用車輛 k 從顧客 i 到顧客 j，否則 $x_{ij}^k=0$ 。

2.2.2 傳統啟發式解法

網路組合最佳化問題之解法可分為「最佳解解法」(exact solution)與「近似解解法」(approximation)，相關文獻中最常用的最佳解解法是分枝定限法(Branch-and-Bound)、分枝切面法(Branch-and-Cut)與變數產生法(Column Generation)等，但是由於 VRP 屬於 NP-hard 問題，因此最佳解解法往往無法於有效的時間內求出最佳解；而近似解解法主要依問題的特性設計出合適的啟發式解法，其優點在於能於有效時間內求得近似解。傳統之啟發式解法可歸納分成以下三種[32]：

- (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，常見的方法如，鄰點法(Neighbor Procedure)、插入法[20] (Insertion Method)、節省法[4] (Savings Method)、貪心法(Greedy Algorithm)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，求得更好的解，例如，K-Opt[15]、Or-Opt[18]、 λ -interchange[19]等。
- (3) 綜合型 (Composite / Hybrid): 係將路線構建和路線改善合併執行，或一面構建路線一面改善路線，常見的方法如，「路線構建起始解+2-Opt」、「路線構建起始解+2-Opt+3-Opt」、CCAO 法(Convex Hull + Cheapest Insertion + Largest Angle + Or-Opt)、一般化插入解繫法(GENERalized Insertion / Unstring and String, GENIUS)。

2.2.3 巨集啟發式解法

鑑於傳統之啟發式解法，於鄰域搜尋之機制下所採取的嚴格接受法則，容易陷入「局部最佳解」(Local Optimum)的困境，以致無法提供精確度更好的近似解。為克服此缺點，Glover 在 1986 年揭示了「巨集啟發式方法」(Meta-Heuristic)一新名詞，其意義是指策略性運用各種啟發式方法去搜尋超越傳統啟發式解法僅能找到的局部最佳解。其中著名以及新發展之巨集啟發式解法，包括：禁制搜尋法(Tabu Search, TS)、模擬退火法(Simulation Annealing, SA)、基因演算法(Genetic

Algorithm, GA)、門檻接受法(Threshold Accepting, TA)、大洪水法(Great Deluge Algorithm, GDA)、包容性深廣度搜尋法(Generic Intensification and Diversification Search, GIDS)等。

(一) 禁制搜尋法

此方法最早由Glover[7]在1986年所提出，其觀念是想構建一個智慧型的問題求解程序：在目前解的鄰域進行搜尋，並應用人工智慧的記憶機制，將已經搜尋過的解記錄在「禁制名單(tabu list)」，避免重複性或毫無目標的搜尋；等到整個鄰域都搜尋完畢後，再選擇一個最佳的方向進行移動(move)，以逐漸逼近最佳解。由此可知，TS法的記憶架構為其特色與核心，而TS的記憶機制又可分成長期短期記憶機制，如下所示：

- i. 短期記憶之「禁制結構」(Tabu Structure)：以禁制名單為基礎，記錄最近搜尋過的解或移動之屬性(Attribute)，避免後續搜尋的解重複先前的搜尋途徑。
- ii. 短期記憶之「渴望準則」(aspiration criterion)：打破禁制名單的限制；即當搜尋的新解優於目前最佳解時，雖屬性於禁制名單中，仍允許移動至該解。
- iii. 長期記憶之「深廣策略」(intensification / diversification strategies)：以記錄屬性出現的紀錄為主，配合深度或廣度搜尋策略，擴大TS法的搜尋範圍。所謂的深度搜尋策略係在搜尋過程中將較佳的數個解記錄在精英列(Elite List)內，當短期記憶搜尋無法改善時，再從精英列中選擇一個解做為下階段搜尋的起點，重新開始。至於廣度搜尋策略則需配合長期記憶結構記錄搜尋過程中解或屬性出現的次數，當短期記憶搜尋無法改善時，選擇次數較少之屬性方向重新進行短期記憶搜尋。計算出現次數時，需乘以一懲罰值(Penalty)，以控制其搜尋方向。

(二) 模擬退火法

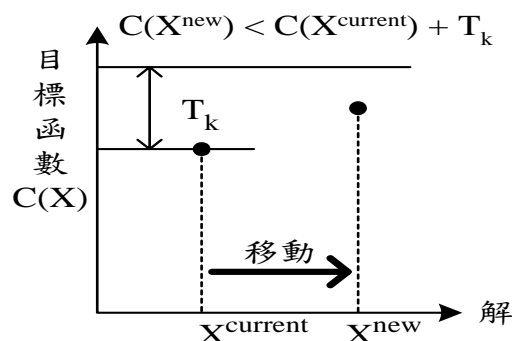
模擬退火法的基本觀念最早由 Metropolis 等人於 1953 年提出，由 Kirkpatrick[16]等人證明該方法可應用到組合最佳化問題上，形成了目前所謂的模擬退火法；SA 可視為一種鄰域搜尋法的變形：對於最小化問題而言，往較低目標值移動的行為可視為物質降溫結晶的過程，而最後的結晶狀態可視為一局部最佳解。與傳統鄰域搜尋法不同的是，SA 可藉由重新加熱的程序使其跳脫目前的局部最佳解，達到另一個更佳解的機會。其 SA 執行的關鍵在於降溫過程(Cooling Process)與接受法則的設計機制。SA 的接受法則為一機率性接受暫劣解：利用一個隨機產生數值與門檻值作比較，該門檻值是鄰解與現有解之目標值差值及溫度的函數，而此處的溫度指的是控制門檻值高低的參數，降溫則是為了能夠逐漸收斂。

(三) 基因演算法

基因演算法最早是由 Holland 所發展[11]，其構想是利用生物染色體基因架構的觀念表示一個複雜系統的結構。例如，一個 0-1 整數規劃問題中，每個決策變數皆可視為一個基因(Genes)，而每個可行解則可視為染色體(Chromosome)；若任意產生一些可行解作為「父代」(Parents)，再經由基因間的複製、交配(Crossover)產生新的可行解「子代」(Offspring)，然後保留最佳的幾個解再產生下一組子代，其過程中亦加入合適的突變組合(Mutation Scheme)，期望得到更佳解。重複執行此過程多次之後，將可找到最佳解或甚佳的近似解。GA 法主要的關鍵在於基因交配的機制設計。

(四) 門檻接受法

門檻接受法最早由 Dueck & Scheuer[7]提出，其解題架構與 SA 相似，皆以門檻數列來決定是否接受新的交換，不同之處在於 TA 使用確定性的接受法則，而 SA 採取的是機率性的接受法。交換過程中，若變動成本在門檻值內其解都予以接受，當達到均衡時便不會大幅變動，而門檻值將逐次遞減，重複新的交換直到門檻數列使用結束為止。以圖 2.2 為例，TA 事先產生一組固定的門檻數列(通常為遞減)，依次使用數列中的門檻值，其接受法則為 $C(X^{new}) < C(X^{current}) + T_k$ 直到門檻數列使用結束。



資料來源：[33]

圖 2.2 門檻接受法

(五) 大洪水法

大洪水法是 1993 年 Dueck[8]根據 TA 的測試經驗所提出的，其與 TA 不同之處在於接受法則的不同，TA 採取逐漸降低至零為止的門檻數列，而 GDA 則是以固定的消退速度下降，其停止機制以無法再找到低於水位之解時自動停止。對於最小問題而言，GDA 之設計是想像一水庫洩洪，其水庫內的地形屬於高低起伏的地形，水位由高往低處下降，以圖 2.3 所示。假設現在的位置為圖中之 A 點，並將水位設於 WL_1 。接著開始找尋「窪地」，假設找到圖中 B 點，便將水位下降 D 的高度至 WL_2 。此時水面 WL_2 下的所有範圍皆為吾人找尋並可接受「窪地」

圖 2.3 大洪水法之解題概念示意圖

Battiti & Tecchiolli[2]提出了求解組合最佳化問題之過程中可能遭遇到之三種障礙，對應到「動態系統理論」中的三種現象：局部最佳解(local optimum)、限制迴圈(limited cycle)及渾沌吸子(chaotic attractor)。針對此三種可能遭遇的現象，韓復華與卓裕仁[12][33]發展了包容性深廣度搜尋法予以一一克服，提供一個不需要記憶結構、容易執行之機制。其流程包括「多起始解構建」(Multiple Initialization Constructor, MIC)、「深度化包容搜尋」(Generic Search for Intensification, GSI)與「廣度化擾動搜尋」(Perturbation Search for Diversification, PSD)三個子系統；整套系統共包含五種模組：在 MIC 群組構建有「加權起始 (Weighted Initialization, WI) 模組」與「鄰域搜尋 (Neighborhood Search, NS)」模組；在 GSI 群組設計有 M1 與 M2 兩種「包容搜尋 (Generic Search)」模組；在 PSD 群組中則構建有「成本擾動 (Cost Perturbation, CP)」模組。其中，M1、M2 與 CP 三個模組皆以 NS 模組為其基本求解工具。整套 GIDS 法運用了接受暫劣解、變換空間、擾動成本與多重起點等巨集策略，有效地導引基本求解工具執行深度搜尋與廣度搜尋，克服搜尋過程中可能遭遇的局部最佳解、限制迴圈及渾沌吸子等障礙。其解題概念如圖 2.4 所示：

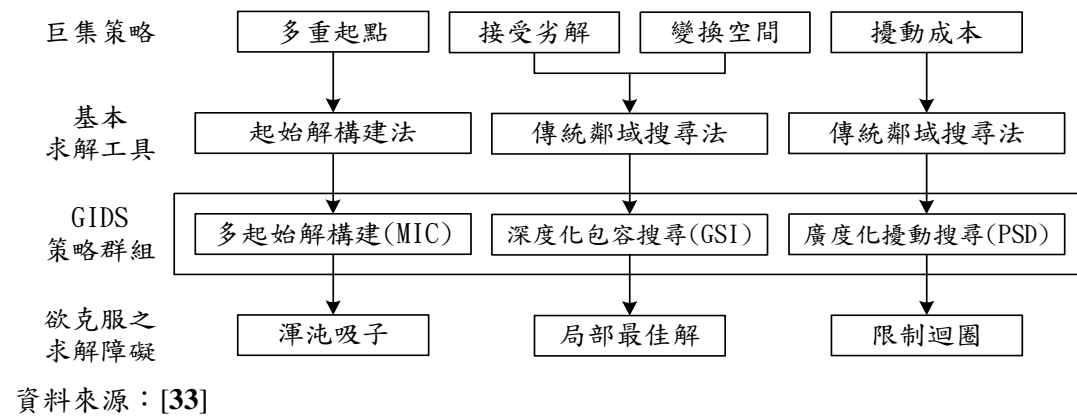


圖 2.4 GIDS 解題概念

2.3 卡車拖車路線(TTRP)問題定義與解法回顧

所謂 TTRP 問題是以卡車附掛拖車進行配送，屬於 VRP 的衍生性問題。其特點為可將整個服務路線分為整車(卡車附掛拖車)服務之主路線及卡車服務之子路線。整車路線服務至停放點(Parking Place)時可將拖車留於該處，以較輕便的卡車服務較不方便到達的顧客(子路線顧客)，譬如位於擁擠市區的顧客，或較不易以整車到達的地區；待子路線顧客服務完畢後再回到原停放點與拖車相接，繼續服務主路線的顧客群。在早期 Gerdessen[10]以車輛連結拖車路線問題(Vehicle Routing Problem with Trailer, VRPT)表現此問題，為簡化整個問題 Gerdessen 事先假設：(1) 每個顧客僅有單位需求量；(2) 每一整車路線中拖車僅能停放一次；(3)每一顧客點均可當作拖車停放點；(4)卡車和整車的行駛時間與距離成正比。此外，以 manoeuvring time 表示顧客點的服務難易度。發展出三套構建 VRPT 路線起始解之啟發式解法：

(1) 啟發式解法 a：

- i. 將其視為標準的 VRP 問題，利用種子啟發式解法(seed heuristic)配合最省插入原則求解 p 條整車路線(p 為所需車輛數)。
- ii. 路線改善：
 - a. 路線間改善—整(卡)車路線改善、整車與卡車路線間改善、2(3)-opt 交換法(以整車路線為主)。
 - b. 某一路線改善--整(卡)車路線路線進行 2(3)-opt 交換法、整車與卡車路線間的改善。
- iii. 以一機制將路線分成主路線與子路線，並找出子路線中一顧客點插入主路線後成本增加最少的點視為拖車停放點。

優點：簡單，從標準的 VRP 延伸出來。

缺點：路線中可能含有過多難服務的顧客點(位於擁擠市區)，而卡車因載貨容量限制下，無法皆以卡車服務，勢必需以整車服務難服務的顧客點，使得該路線因 manoeuvring time cost 過大造成路線總時間成本亦過大。

(2) 啟發式解法 b：

- i. 建構路線前考慮以整車由場站至顧客點所需之時間(包含 manoeuvring time)，將服務的顧客(節點)分成二部分，分別由卡車服務與整車服務。
- ii. 構建卡車路線：指派過程中設了一個門檻值，即當節點被指派至卡車時其達到容量的 70%便停止指派，改由較接近已指派之卡車的節點且到達時間成本較高之未指派節點，繼續指派給卡車，直到滿足該容量為止。

iii. 確定停放點：從未指派之節點中，找出最靠近卡車路線的節點，指派為拖車停放點。

iv. 構建整車路線：在已知場站與節點的情況下，運用種子啟發式解法構建。

v. 路線改善

優點：克服解法 a，路線中可能包含過多難服務的顧客點(節點)。

缺點：太早決定每輛車的拖車停放點，導致在改善過程中，會減少可能較佳的組合解

(3) 啟發式解法 c：與解法 b 類似，但將指派停放點的動作放在最後一步驟執行。

其中以啟發式解法 c 先將顧客分群構建出卡車服務子路線，再決定整車主路線與拖車停放點，會得到較好的結果。

Chao[6]於 2002 年提出 TTRP 問題，其問題本質與 Gerdessen 所提的 VRPT 問題是一致的，然而 Chao 所提的 TTRP 問題除了延續 Gerdessen 對 VRPT 問題的種種假設外，更放寬或增添某些假設條件，其假設條件如下所示：

- (1) 將顧客分類：可被整車服務之顧客(v.c.)，亦可單獨被卡車服務；只能被卡車服務之顧客(t.c.)；
- (2) 組合路線中拖車能停放一次以上；
- (3) 組合路線之主路線中的顧客皆可成為拖車停放點；
- (4) 卡車數量(m_k)與拖車數量(m_l)已知， $m_k \geq m_l$ 等；
- (5) 顧客需求非單位需求。

在路線構建方面，以最省插入法(cheapest insertion)構建三種類型路線如圖 2.5 所示，分別為：

- (1) 卡車路線，如圖 2.5(1)；
- (2) 整車路線(卡車與拖車途中不可分離)，如圖 2.5(2)；
- (3) 組合(卡車與拖車)路線：分為主路線與子路線，藉著從主路線中找出一顧客點停放拖車後，以卡車服務 t.c 顧客，最後再回到拖車停放點連結拖車後，繼續服務 v.c 顧客，如圖 2.5(3)。

對於組合路線之主路線，v.c 顧客被限制構建於其中，子路線則以 t.c.為主插入於已存在之子路線中，或者與主路線之 v.c.顧客連接成為新的子路線節點。由於在構建路線前，Chao 將原本定義之整數規劃指派問題放鬆成線性規劃，因此在其構建路線時，將使得路線載貨量高於車輛之容量，使得所構建出來的起始路線為不合理的，對此，Chao 忽略該不合理之情況以懲罰值來代表其不合理之程度，因此路線除了行駛成本外尚有懲罰值。

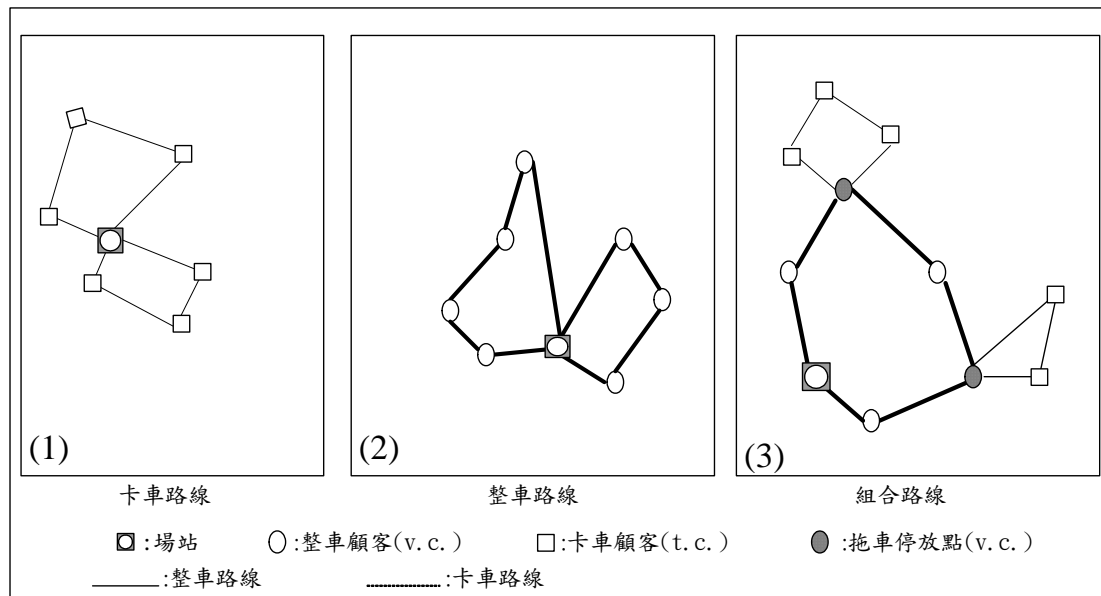


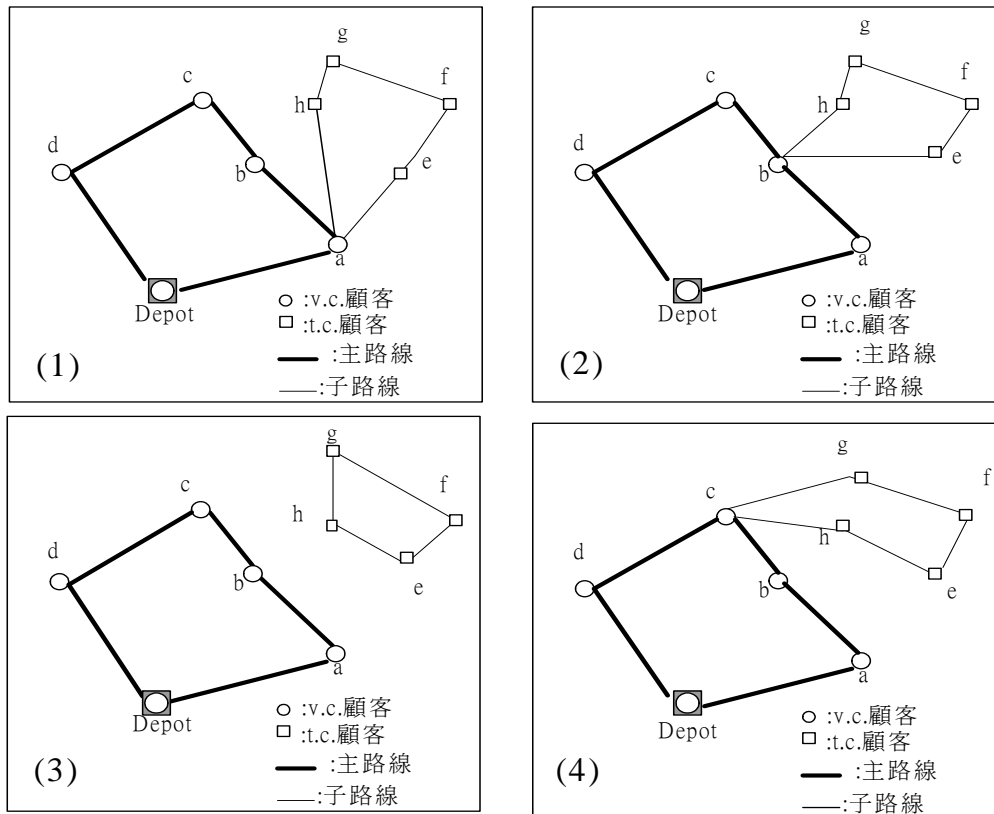
圖 2.5 TTRP 三種類型路線

路線改善方面，提出四個改善啟發式解法，其目的在於將不合理路線改善成合理之路線，分述如下：

- (1) 單節點改善(One-Point Descent movement, OPD)：將某路線一節點移動(move)至另一路線，其改善準則為：移動動作符合減少路線之懲罰值(不論路線之行駛成本是否減少)或減少路線之行駛成本(懲罰值不增加)。
- (2) 雙節點改善(Two-Point Descent movement, TPD)：改善準則與 OPD 一樣，主要是針對兩條不同路線中各一節點交換。

上述這兩種改善法，對於兩種候選移動不可執行：(1)不可將 t.c.顧客移動至組合路線或純整車路線；(2)對於組合路線中被選為停放點之 v.c.顧客不可移動。

- (3) 子路線基點改善(Sub-tour root-refining)：主要是對於子路線之基點重新選擇與重新排序子路線之顧客拜訪順序。以圖 2.6 來表示其該善過程：



資料來源：[6]

圖 2.6 子路線基點改善

圖 2.6(1)為一包含主路線與子路線之組合路線，假如節線 b-h 與 e-b 能改善行駛成本則取代節線 a-h 與 e-a 如圖 2.6(2)；接著暫時移除節線 b-h 與 e-b 如圖 2.6(3)，重新考量整個子路線的拜訪順序，找出對於行駛成本最省的路線，如圖 2.6(4)所示。

(4) 2-Opt：待上述三種改善法獲得一合理起始解後，則執行 2-Opt[14]進行卡車、整車以及組合路線之改善。

經由上述整個步驟構建出 TTRP 之起始解後，Chao 運用禁制搜尋法配合確定性模擬退火法(Deterministic annealing)中誤差(deviation)概念進行路線改善。其中該概念視為一個新的禁制限制(tabu restriction)，藉著改變誤差值進行深度(較小誤差值)與廣度(較大誤差值)搜尋。演算法測試方面 Chao 參考 Christofides et al.[5] 中 7 個基本 VRPs 測試問題，發展出 21 題 TTRP 測試題庫，測試所發展之 TS 演算法在 TTRP 問題有不錯的結果及效率。

2.4 限制規劃方法

限制規劃由電腦科學(Computer Science)中的人工智慧(Artificial Intelligence, AI)領域所發展出來，不同於數學規劃從傳統的數學理論基礎發展而來。根據 Brailsford et al.[3]對於 CP 的定義：運用電腦程式發展模式化語言，讓使用者能迅速、輕易地描述限制滿足問題，再透過精緻的電腦演算法(sophisticated computer algorithm)有系統地、有效率地求解限制滿足問題。其優點在於對程式語言不熟悉的使用者，能迅速發展其問題模式並進行求解。在早期以限制系統為特色的邏輯規劃語言(logic programming language)(如：PROLOG)，由於搜尋解的時候限制式並沒有被善加利用，導致求解效率上的缺陷。因此，針對這樣的缺陷將邏輯規劃語言作了一番修改，使得未來的邏輯規劃語言能於搜尋過程中充分善用限制式間的關係，提高求解效率。例如 CHIP(Constraint in PROLOG)這類的限制邏輯規劃(Constraint Logic Programming, CLP)。然而，將限制規劃演算法應用在一般的程式語言(programming language)與宣告式語言(declarative language)等亦非不可能的，例如限制規劃軟體 ILOG Solver(C⁺⁺ / Java)的限制規劃程式庫即是以物件導向式(C⁺⁺ / Java)所發展出來。

雖然限制規劃最初的發展是為了解決限制滿足問題，然而其搜尋機制卻也適用於最佳化問題上。例如許多整數規劃(Integer Programming, IP)問題，傳統上仍用分枝定限法(Branch and Bound Algorithm)來求解，若改採限制規劃的空間搜尋技巧，將能有效加快求解速度。整體而言，限制規劃特別適合處理離散變數(Discrete Variables)所組成有限解空間的問題型態。此方法論的優點有：

1. 模式構建的方便性：由於限制規劃語言屬於程式語言中之「宣告式語言」(Declarative Language)，使得使用者可以很容易地建構模式。
2. 可處理多樣之限制式：限制規劃可以處理多種不同之限制式，如：多元限制式(n-ary constraints)、布林限制式(Boolean Constraints)、邏輯限制式(Logical Constraints)、序列限制式(Sequence Constraints)等。
3. 有效率之求解機制：其求解機制結合了許多智慧型的空間搜尋演算法(如：Forward Checking、MAC Algorithm)、一致性(Consistency)技術、限制式推理機制(Constraint Propagation)，可有效增進求解效率。
4. 「模式建構」(Modeling)與「模式求解」(Solving)是互相獨立的：當問題模式中的限制式改變時，只需稍微新增/刪除/修改問題模式之相關限制式，即可運用相同的智慧型空間搜尋演算法重新求解。

本研究所使用之限制規劃軟體為 ILOG Dispatcher 軟體，該軟體為針對車輛路線或維修派遣(maintenance dispatcher)等問題，以 ILOG Solver(C⁺⁺)限制規劃程式庫軟體所延伸出來的軟體。由於該軟體內建有由 ILOG Solver 已構建完成的車輛路線相關之求解演算法物件(如：節省法、插入法、K-Opt 等)以及在宣告式語言的特性下，故可輕易構建求解模式進行求解。

第三章 GTTRP 問題定義與解題架構

3.1 GTTRP 問題定義

根據 Gerdessen[10]、Chao[6]等對於 TTRP 之相關假設條件，以及本研究對於該些文獻之成本構建與假設條件的調整，整個 GTTRP 問題假設條件可歸納成下面幾點：

(1) 成本構建方面：

- i. 考慮拖車停放成本；
- ii. 考慮拖車卸下及連結之時間；
- iii. 考慮車輛以整車型態(卡車附掛拖車)與卡車型態配送的變動成本；

(2) 限制條件方面：

- i. 顧客分類：可被整車服務亦可被卡車單獨服務之整車顧客點(v.c)，以及只能被卡車服務之卡車顧客點(t.c)；
- ii. 組合路線中停放拖車次數可停放一次以上；
- iii. 只有部分整車顧客可成為拖車停放點；
- iv. 顧客非單位需求；

(3) 路線類型方面：

依據 Gerdessen[10]與 Chao[6]之相關 TTRP 文獻，可發現 Gerdessen 對於 TTRP 之路線類型方面皆是由一條整車路線與卡車路線所構成，使得在配送過程中，不管配送成本增加或減少與否，路線皆需硬性搭配一條卡車路線。然而 Chao 則進一步將其分成三種路線類型，使整個配送過程更具彈性，因此本研究擬延續 Chao 對於 TTRP 路線類型之分類，其分類如下所示：

- i. 卡車路線；
- ii. 整車路線：車輛以卡車附掛拖車方式進行配送，配送過程中拖車不能於拖車停放點卸下；
- iii. 組合路線：由卡車路線與整車路線所構成，其中卡車路線為子路線，整車路線為主路線；

本研究將 Gerdessen 與 Chao 對於 TTRP 之假設條件與本研究 GTTRP 之假設條件做一整理比較，以更清楚瞭解彼此間的差異性，其 GTTRP 與 TTRP 之差異整理表如表 3.1 所示，整個 GTTRP 之路線類型示意圖如圖 3.1 所示：

表 3.1 TTRP 與 GTTRP 差異比較表

	項目	TTRP		GTTRP
		Gerdessen	Chao	本研究
成本構建	車輛使用成本	無	無	有
	拖車停放點費用	無	無	有
	拖車裝卸、連結時間成本	無	無	有
限制條件	顧客分群	無	◆ 整車顧客 ◆ 卡車顧客	◆ 整車顧客 ◆ 卡車顧客
	顧客需求	單位需求	非單位需求	非單位需求
	拖車停放次數	一次	不限制	不限制
	可停放拖車點	所有顧客點	整車顧客點	部分整車顧客點
	路線類型	組合路線	◆ 整車路線 ◆ 卡車路線 ◆ 組合路線	◆ 整車路線 ◆ 卡車路線 ◆ 組合路線

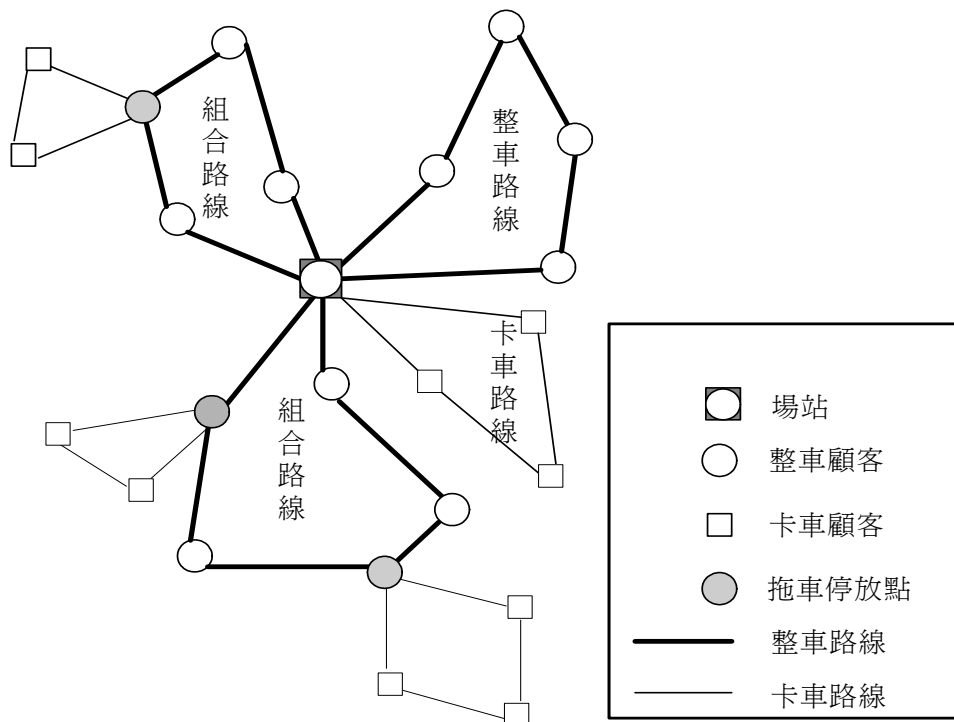


圖 3.1 GTTRP 路線示意圖

3.2 GTTRP 解題架構

本研究整套解題架構可以分為兩個程序，第一個程序為局部最佳解構建程序，第二個程序為包容性改善程序。局部最佳解構建方面，主要是由起始解模組與鄰域搜尋改善模組所構成；在起始解模組構建時，主要是以傳統起始解構建演算法「最省插入法」為主，配合著限制規劃進行起始解構建，接著再以鄰域搜尋改善模組進行改善，以搜尋局部最佳解。包容性改善方面，主要功能是為跳脫傳統鄰域搜尋改善法陷入局部最佳解之缺點，本研究以卓裕仁[33]所提出 GIDS 概念作為本研究包容性改善的解題架構，由於過去所提出的包容性啟發式解法種類相當多，而為了能控制整個求解過程，本研究以確定型接受法則之門檻型演算法(Threshold Algorithms)為主。在 GIDS 之搜尋架構中 M1 與 M2 是兩種執行包容性搜尋之模組，以 TA 與 GDA 等門檻型演算法為核心。整個 GTTRP 之解題架構如圖 3.2 所示。

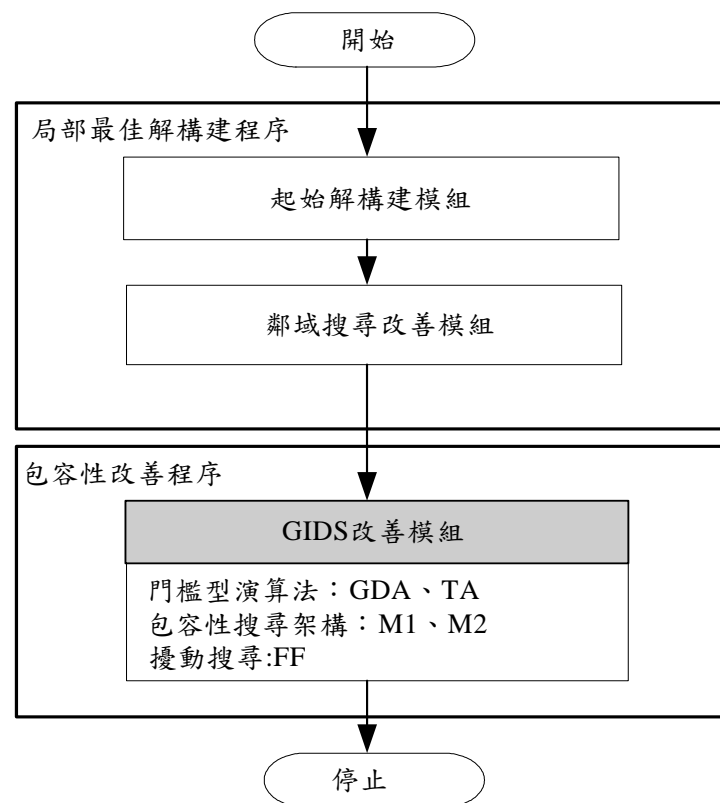


圖 3.2 GTTRP 之解題架構

GIDS 主要是藉由包容性搜尋模組中的 M1 與 M2 兩個模組交叉執行進行「深度化」搜尋，再配合成本擾動模組做「廣度化」搜尋以擾動目標函數方式擴大搜尋空間。其整個 GTTRP_GIDS 之執行架構如圖 3.3 所示。其中的 M1 模組是以強化搜尋為主，採用門檻型演算法(TA 或 GDA)與傳統鄰域搜尋法為核心，以交叉執行方式進行搜尋；而 M2 模組則是加強跳脫局部最佳解，藉著執行完整

門檻型演算法(GDA 或 TA)後，再執行傳統鄰域搜尋法作收尾。成本擾動模組則是採用陳國清[30]提出之兩極跳躍法(Flip Flop, FF)進行成本擾動。

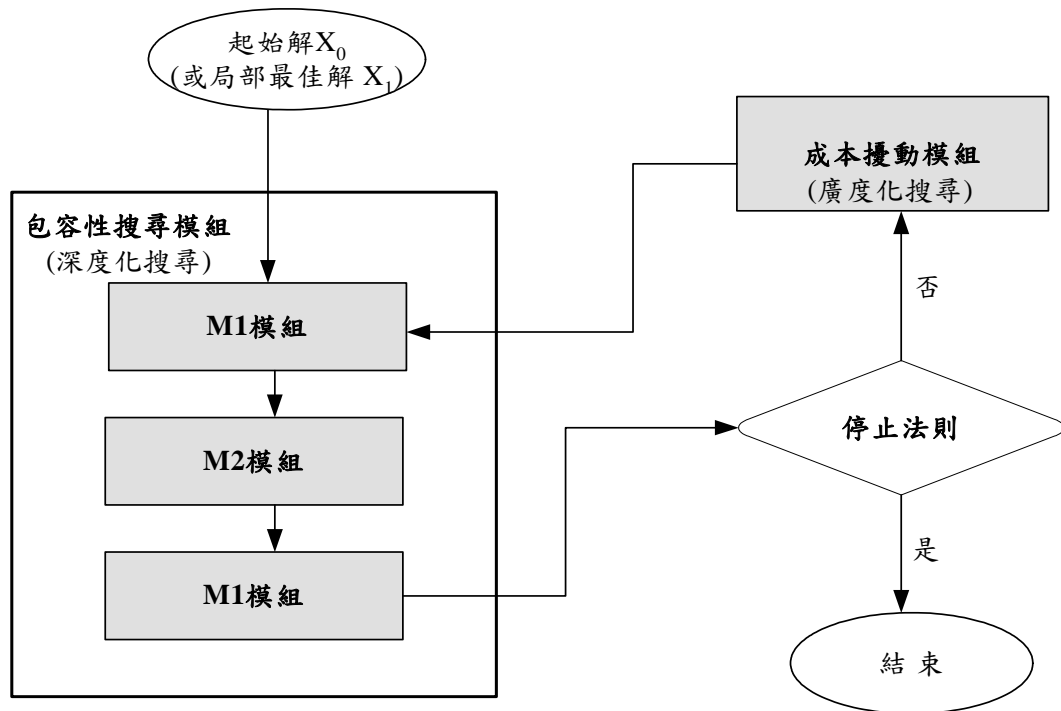
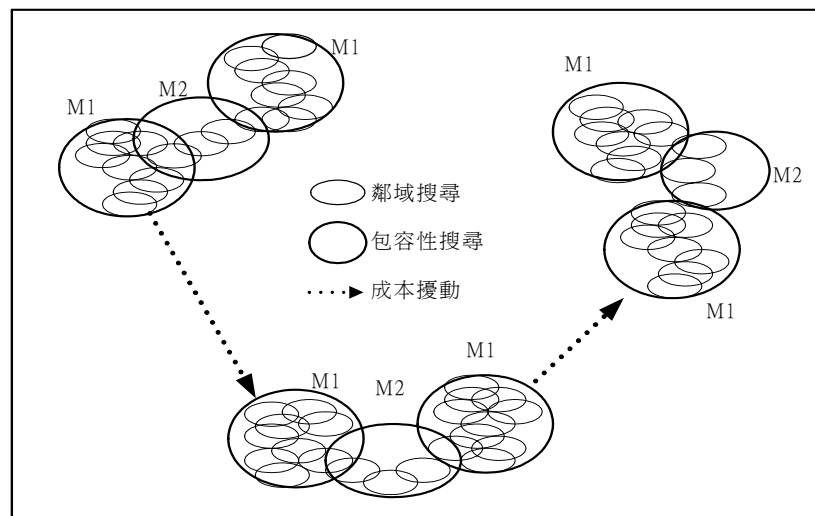


圖 3.3 GTTRP_GIDS 之執行架構

在解集合的空間中 GIDS 的搜尋空間軌跡如圖 3.4 所示。藉著 M1 與 M2 在一區域深度化搜尋，待搜尋結束，以成本擾動方式跳至另一區域重覆搜尋，這樣的搜尋方式對於組合問題龐大的解集合空間而言，比地毯式搜尋方式有效率的找到較佳的解，甚至亦可能跳至具有最佳解的區域短時間內找問題最佳解。



資料來源：[12]

圖 3.4 GTTRP_GIDS 搜尋概念圖

第四章 局部最佳解構建程序之設計

4.1 起始解構建模組

由於 GTTRP 問題最大的特色在於車輛以卡車加掛拖車進行顧客服務，且服務過程中亦可將拖車停放在拖車停放點(整車顧客點)改以卡車服務卡車顧客點，最後再至停放拖車處連結拖車繼續對未服務之顧客進行服務。故 GTTRP 路線可以說是將數條 VRP 路線組合在一起。因此本研究依此特性，對於起始路線構建方面以兩階段方式進行構建，即先針對整車顧客構建出整車路線；接著對於卡車顧客點進行子/卡車路線構建。然而為路線構建方便，本研究於構建過程中限制整車顧客只能被整車服務、卡車顧客只能被卡車服務，其整個起始解模組構建執行架構請參考圖 4.1。

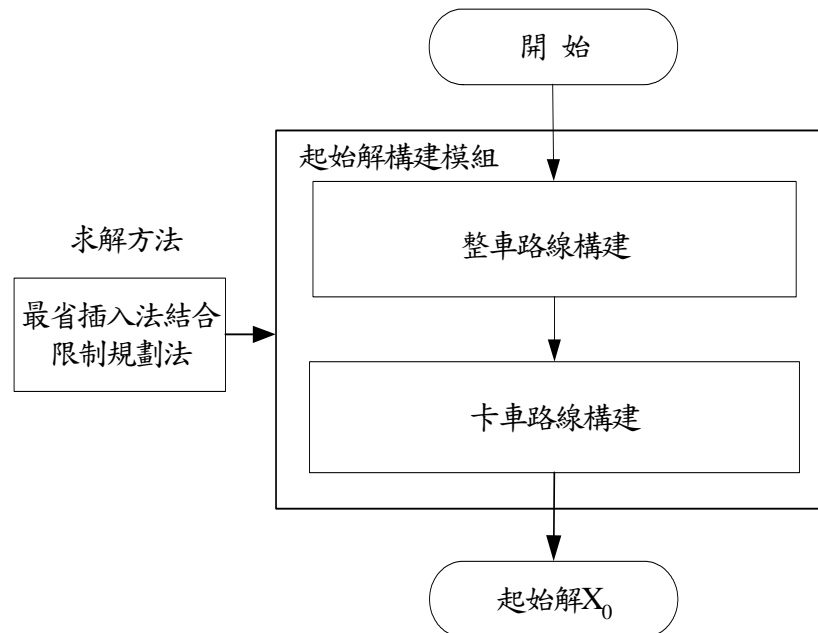


圖 4.1 起始解構建執行架構

(一) 整車路線構建

對於整車路線構建，本研究運用傳統最省插入法準則配合限制規劃進行路線構建，其種子點以距離場站最遠之顧客為選擇依據。由於在構建整車路線時其整車剩餘可裝載之容量會影響到接下來組合路線之子路線與卡車路線構建，亦即若構建整車路線後車輛剩餘可裝載之容量過小時，則於構建子路線時可服務的顧客點則因車輛剩餘容量過小，服務點數相對地減少，而未服務之卡車顧客點勢必需加派卡車進行服務方能滿足需求；若是在整車顧客數遠少於卡車顧客數的情況下，有可能使得整車顧客點皆被一部整車服務完，而卡車顧客點只能被由場站出發之卡車所服務。在卡車容量低於整車容量之條件下，一群可由一部整車服務之顧客群，可能需至少兩部卡車服務方能滿足需求，因而造成過多車輛使用，使得總成本增加；同理，若構建整車路線後車輛剩餘可裝載之容量過大，而卡車顧客

數遠少於整車顧客的情況下，則最後可能造成車輛容量剩餘空間過大，形成不必要的資源浪費。因此，若在構建整車路線時，能限制構建後合適的剩餘容量空間進行子路線構建，將可使車輛使用數減少。故在構建整車路線時，本研究以測試例題中整車顧客之需求量(q_v)佔總需求量(q_T)之比例乘上整車容量(Q_1)，作為構建整車路線時其裝載量之上限(\overline{Q}_1)，以期使能更有彈性地構建路線解。其公式如式(4-1)所示。

$$\overline{Q}_1 = Q_1 \times \frac{q_v}{q_T} \dots\dots\dots(4-1)$$

(二) 子/卡車路線構建

對於子路線構建，則從已構建完成之整車路線中選擇合適之顧客點作為拖車停放點，其選擇的準則為拖車停放成本最小的；第二階段同是採用最省插入法構建子路線，其種子點選擇則是依最靠近該停車點之卡車顧客點為準則。然而在卡車顧客點插入現有之子路線時，若插入成本大於該顧客點成為另一拖車停放點之種子點時之成本，則形成一條新的子路線，依此原則構建子路線，直至所有卡車顧客皆排程為止。若尚有卡車顧客未被排程，而無法再插入子路線時，則從場站指派卡車服務剩餘之顧客，其路線構建亦以最省插入法為主。

(三) ILOG Dispatcher 限制規劃軟體應用

本研究以 ILOG Dispatcher 限制規劃軟體進行 GTTRP 問題起始解求解，由於該程式語言屬於「宣告式語言」，因此使得使用者容易上手，再加上該程式語言已經將一般常用之啟發式解法予以模組化，更節省程式撰寫時間。本文於此部分只針對如何以其內建函式進行兩階段路線構建機制，對於該軟體之詳細使用狀況請參考 ILOG Dispatcher 使用手冊[13]。

由於 Dispatcher 軟體是針對 VRRPs 問題發展而成，不管對於何種型態之 VRRPs 問題有其一套固定之流程，如圖 4.2 所示：

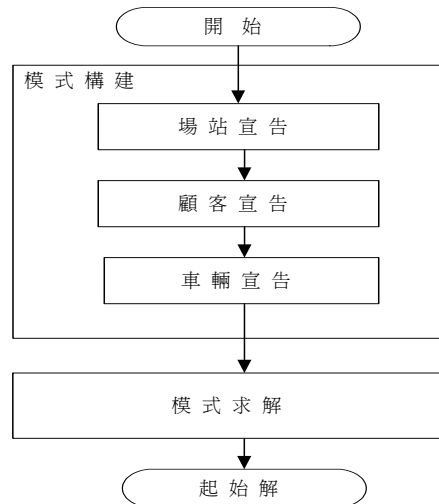


圖 4.2 Dispatcher 建構 VRRPs 問題之流程圖

其中，不論是何種類型之 VRP 問題其場站、車輛宣告部分皆是大同小異，

只有顧客宣告部分則是依不同問題特性進行限制式新增/刪除/修改。本研究在第一階段對於顧客宣告，主要以下列 3 類限制式進行整車路線構建。

C1：整車路線構建之裝載量限制。

C2：整車顧客只能被整車服務之限制。

C3：卡車顧客暫不服務之限制。

待整車路線構建結束，再移除 C3，進行卡車顧客插入子路線或卡車路線之動作。

4.2 鄰域搜尋改善模組

本研究所採用的鄰域改善模組可分成三大類：(一)路線內改善模組；(二)路線間改善模組；以及(三)拖車停放點改善模組。由於改善過程中，需遵循卡車顧客不能被整車服務以及整車顧客可以被卡車服務之限制條件，因此在路線間改善模組又可分成(1)同類路線間改善模組；(2)異類路線間改善模組。所謂同類路線間改善指的是對任兩條皆被同一種車輛型態所服務之路線進行交換改善，如：整車路線對整車路線、卡車/子路線對卡車/子路線間的改善，而異類路線改善則是指任兩條被不同車輛型態所服務之路線間的改善，如：整車路線與子/卡車路線間節點彼此的交換，其整個鄰域搜尋改善模組之執行架構如圖 4.3 所示。

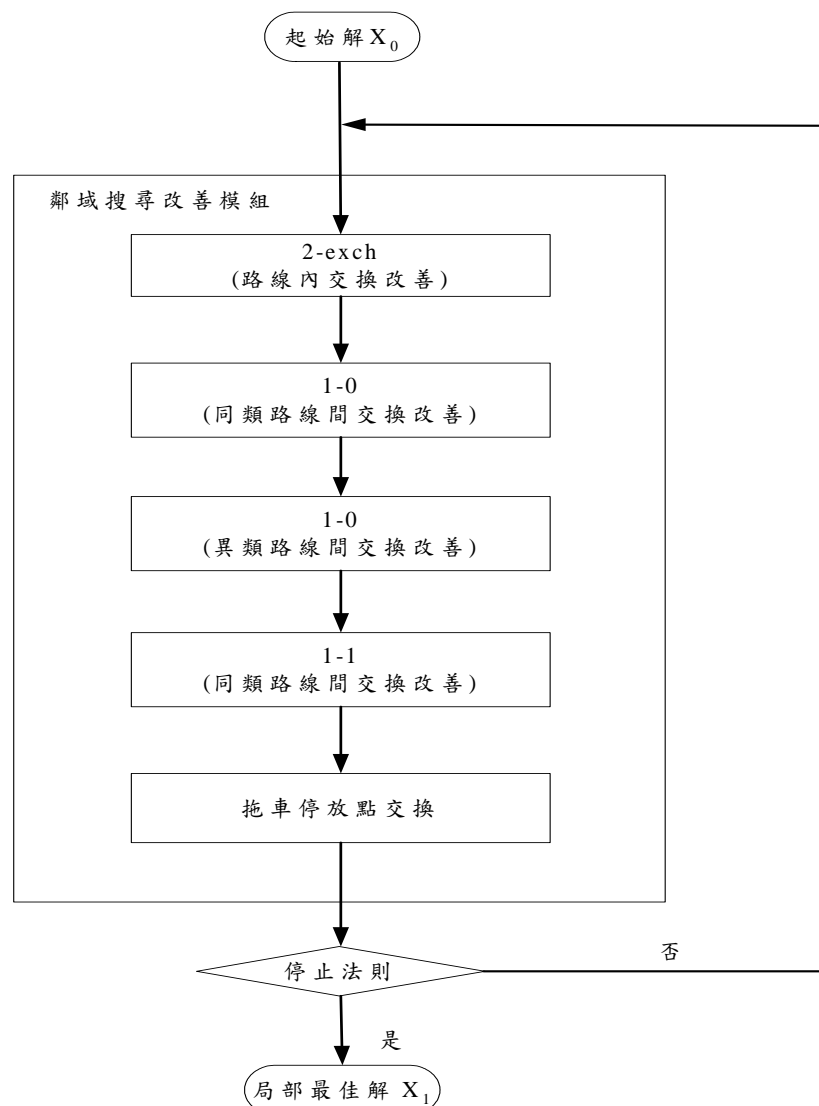


圖 4.3 鄰域改善模組之執行架構

在執行鄰域搜尋，尋找可交換的鄰近解時，決定選擇那一個鄰近解進行交換的準則稱為「選擇策略(selection strategy)」。根據 Osman[19]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；及(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。此兩種策略並無法證明孰優孰劣，但從執行效率來看，使用「首先改善」選擇策略優於「最佳改善」選擇策略，因此在後續各種交換改善模組執行架構中，一律採用「首先改善」選擇策略。

此外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 4.1 之虛擬碼(pseudo codes)所示。由表 4.1 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。

表 4.1 「全套」與「半套」交換法之虛擬碼

全套交換法	半套交換法
<pre> for i = 1 to index_number for j = 1 to index_number if (i ≠ j) then [exchange procedure] next j next i </pre>	<pre> for i = 1 to index_number for j = i to index_number [exchange procedure] next j next i </pre>

資料來源：[29]

(一) 路線內改善模組

路線內改善模組即對同一條路線進行交換改善。本研究擬採用 2-exch 交換法進行改善，然而依據路線類型可分成整車路線 2-exch、卡車路線 2-exch 以及子路線 2-exch，其中由於卡車路線和子路線皆是由卡車進行配送服務，不同之處在於卡車路線是由場站為起始點，子路線是由拖車停放點為起始點，故在交換過程中是無異地，因此在解題示意圖以卡車路線 2-exch 表示。此外，後述成本交換改善計算式中， C^1 表示整車路線成本； C^2 表示卡車路線成本。

i. 整車路線 2-exch

令圖 4.4(a)為一路線解。若換掉其中(m, p)及(k, n)兩條節線，然後連接(m, k)及(p, n)兩條節線而成為圖 4.4(b)的路線，將可改善其解。其中，路徑(p-k)必須反轉順序為(k-p)。檢查交換後的成本改善值： $I(m, k; p, n) = C_{mp}^1 + C_{kn}^1 - C_{mk}^1 - C_{pn}^1$ ，是否大於零。若是，則更新路線；否則維持原解，繼續交換下一組節線，直到可能交換的節線都檢查完畢且沒有改善為止。

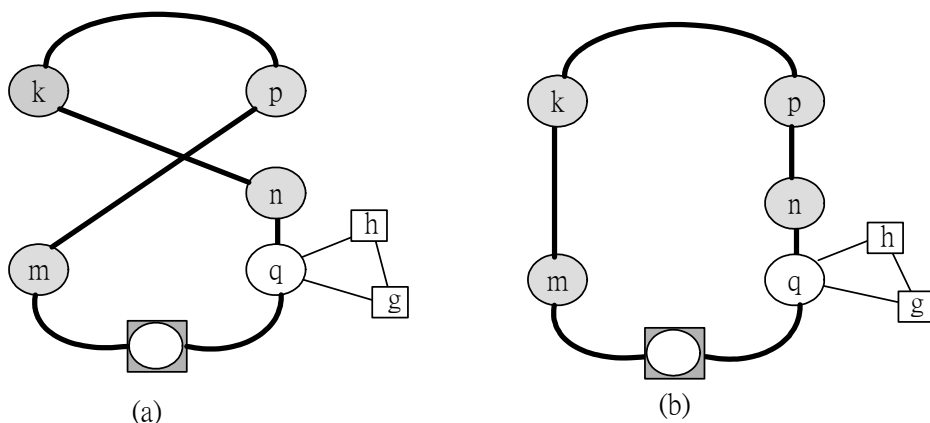


圖 4.4 整車路線 2-exch 節線交換示意圖

ii. 卡車/子路線 2-exch

而對於卡車/子路線改善，其解題示意圖，如圖 4.5 所示，令圖 4.5(a) 為一卡車/子路線解，若換掉其中(m, p)及(n, q)兩條節線，然後連接(m, n)及(p, q)兩條節線而成為圖 4.5(b)的路線，將可改善其解。其中，路徑(p-n)必須反轉順序為(n-p)。檢查交換後的成本改善值： $I(m, n; p, q) = C_{mp}^2 + C_{nq}^2 - C_{mn}^2 - C_{pq}^2$ ，是否大於零。若是，則更新路線；否則維持原解，繼續交換下一組節線，直到可能交換的節線都檢查完畢且沒有改善為止。

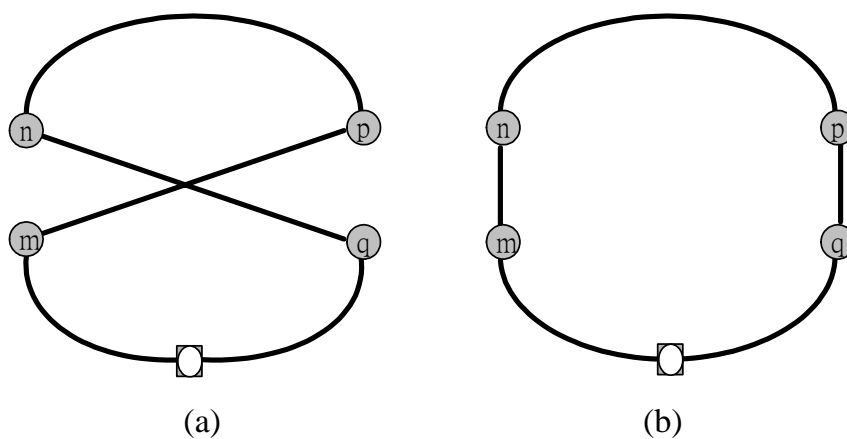


圖 4.5 卡車路線 2-exch 節線交換示意圖

由於 2-exch 法並沒有檢查所有可能交換的節線，而是依序固定每條節線，循序檢查該節線之後的節線至場站或拖車停放點為止，可算是「半套」的節線交換法。

(二) 同類路線間改善模組

Osman[19]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 λ -路線間節點交換法(λ -interchange)。本研究擬以(1-0)及(1-1)節點交換為主，以下將針對(1-0)及(1-1)節點交換做一詳細介紹：

i. 整車路線間(1-0)節點交換

圖 4.6 說明整車路線間(1-0)節點交換法之解題概念。令圖 4.6(a)為 TTRPTW 解之其中兩條路線解 A、B。若將路線 A 的節點 m 插至路線 B 的節點 p 與 q 之間(如圖中虛線所示)，且路線 B 仍滿足車輛容量限制下，成為圖 4.6(b)的兩條新路線，將可改善總路線成本。其節點交換改善值之計算公式為： $I(m; p, q) = (C_{lm}^1 + C_{mn}^1 - C_{ln}^1) + (C_{pq}^1 - C_{pm}^1 - C_{mq}^1)$ ，節點 p 與 q 為連續的點對。此外，對於卡車路線間(1-0)節點交換之解題概念是一樣的，亦即對任兩條卡車路線進行相同步驟之改善。其中，在交換過程中，若交換路線只剩下一節點之情況下，則不必考慮交換成本直接交換之，藉以減少車輛使用數。

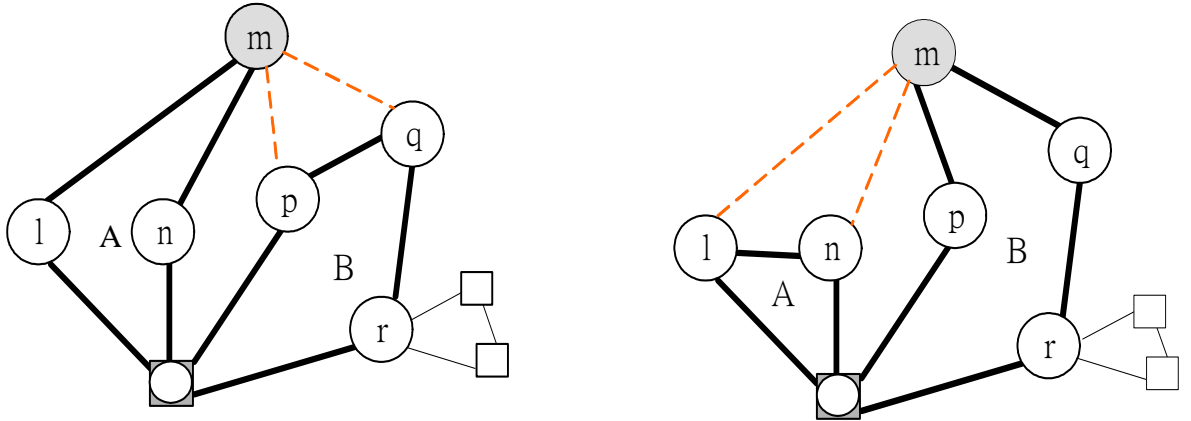


圖 4.6 整車路線間(1-0)節點交換

ii. 子路線間(1-0)節點交換

圖 4.7 說明子路線間(1-0)節點交換法之解題概念。令圖 4.7(a)為 GTTRP 解之其中一條組合路線解，其中 A、B 路線表示組合路線的子路線(由卡車進行服務)。若將路線 A 的節點 m 插至路線 B 的節點 l 與 n 之間(如圖中虛線所示)，在路線 B 之車輛容量限制下，且可改善總路線成本則插入之，成為圖 4.7(b)的兩條新路線。其節點交換改善值之計算公式為： $I(m; l, n) = (C_{qm}^2 + C_{mi}^2 - C_{qi}^2) + (C_{ln}^2 - C_{lm}^2 - C_{mn}^2)$ ，節點 l 與 n 為連續的點對。該組合路線之子路線亦可與其他組合路線之子路線做交換。

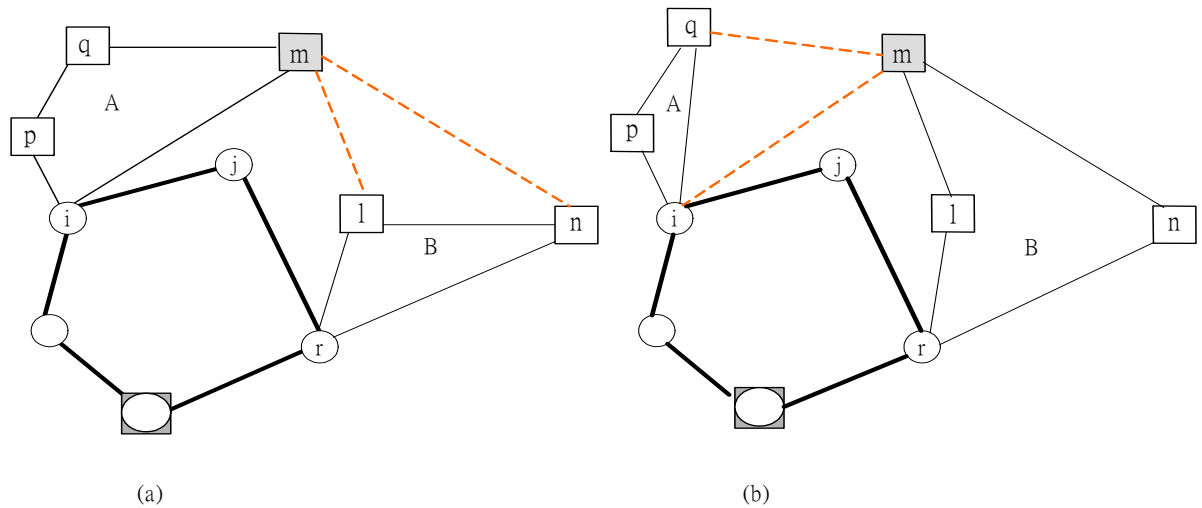


圖 4.7 子路線間(1-0)節點交換

iii. 卡車路線與子路線間(1-0)節點交換

圖 4.8 說明卡車路線與子路線間(1-0)節點法之解題概念。令圖 4.8(a) 為 GTTRP 解之兩條路線解，其中 A 路線表示卡車路線，B 路線表示組合路線的子路線(由卡車進行服務)。若將路線 A 的節點 m 插至路線 B 的節點 l 與 n 之間(如圖中虛線所示)，在路線 B 及組合路線之車輛容量限制下，且可改善總路線成本則插入之，成為圖 4.8(b)的兩條新路線。其節點交換改善值之計算公式為： $I(m; l, n) = (C_{qm}^2 + C_{m0}^2 - C_{q0}^2) + (C_{ln}^2 - C_{lm}^2 - C_{mn}^2)$ ，節點 l 與 n 為連續的點對。然而對於子路線與卡車路線間的(1-0)節點交換，則是一樣的解題步驟，其順序是從圖(b)至圖(a)，節點交換改善值之計算公式為： $I(m; q, 0) = (C_{nm}^2 + C_{ml}^2 - C_{ln}^2) + (C_{q0}^2 - C_{qm}^2 - C_{m0}^2)$ ，節點 q 與 0 為連續的點對。此外，對於卡車路線與子路線間之交換，若交換路線只剩下一節點之情況下，則不必考慮交換成本直接交換之，藉以減少車輛使用數。

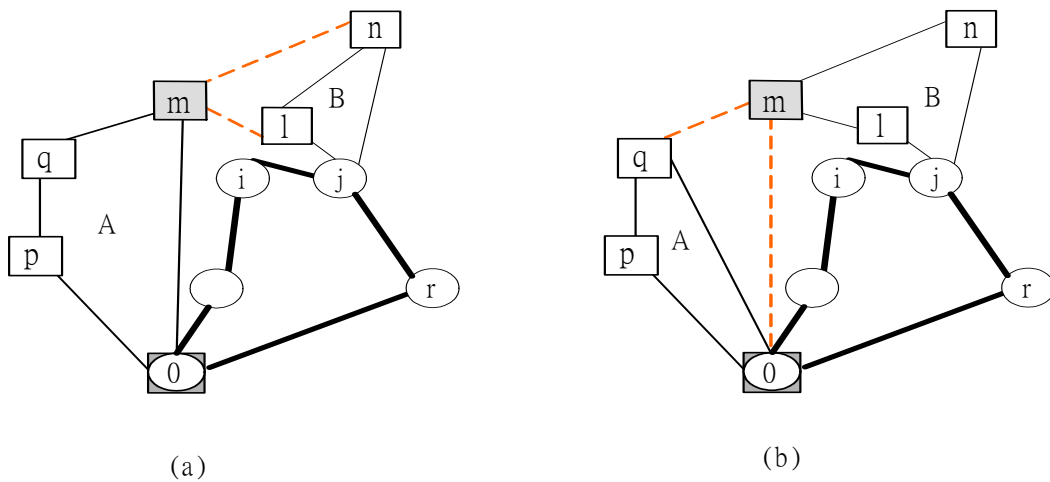


圖 4.8 卡車路線與子路線間(1-0)節點交換

iv. 整車路線間(1-1)節點交換

圖 4.9 說明整車路線(1-1)交換法之解題概念。令圖 4.9(a)為 GTTRP 解之其中二條整車路線 A、B。若將路線 A 的節點 m 與路線 B 的節點 q 交換(如圖中虛線所示)，且路線 A 及路線 B 仍滿足車輛容量限制下，且可改善總路線成本則交換之，成為圖 4.9(b)的二條新路線。節點交換改善值之計算公式為： $I(m; q) = (C^1_{lm} + C^1_{mn} - C^1_{pm} - C^1_{mr}) + (C^1_{pq} + C^1_{qr} - C^1_{lq} - C^1_{qn})$ 。此外，對於卡車路線間(1-1)節點交換之解題概念是一樣的，亦即對任兩條卡車路線進行相同步驟之改善。

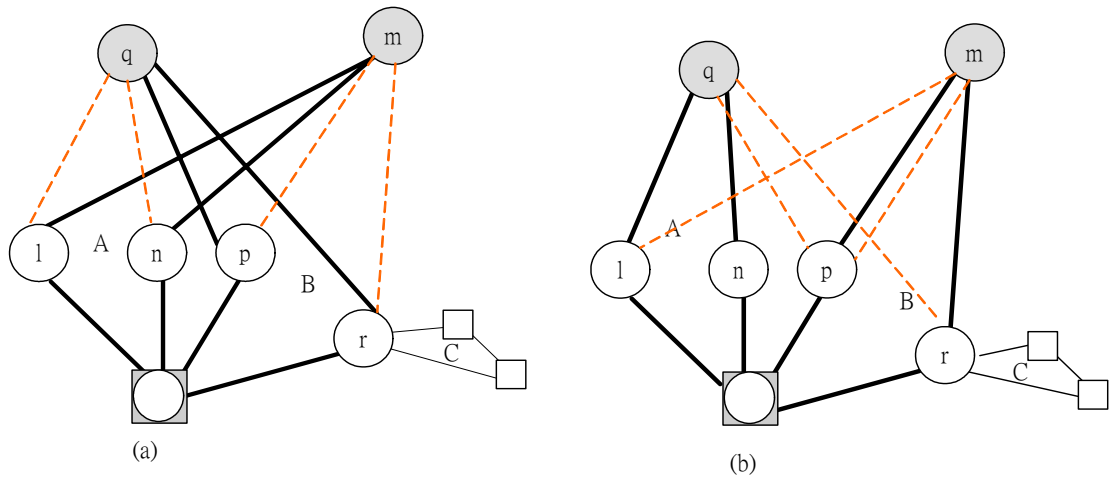


圖 4.9 整車路線間(1-1)節點交換

v. 子路線間(1-1)節點交換

圖 4.10 說明子路線(1-1)交換法之解題概念。令圖 4.10 為 GTTRP 解之其中一條組合路線解，其中 A、B 路線表示組合路線的子路線(由卡車進行服務)。若將路線 A 的節點 m 與路線 B 的節點 q 交換(如圖中虛線所示)，不違反路線 A 及路線 B 之車輛容量限制下，且可改善總路線成本則交換之，成為圖 4.10(b)的二條新路線；其節省值之計算公式為 $I(m; g) = (C^2_{qm} + C^2_{mi} - C^2_{lm} - C^2_{mn}) + (C^2_{lg} + C^2_{gn} - C^2_{qg} - C^2_{gi})$ 。該整車路線之子路線亦可與其他整車路線之子路線做交換。

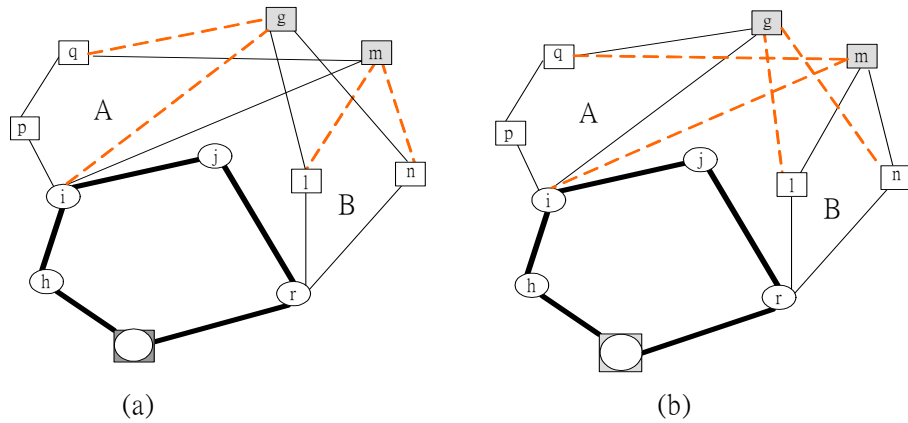


圖 4.10 子路線間(1-1)節點交換

(三) 異類路線改善模組

異類路線改善指的是對於由不同車輛型態所服務之路線間的改善。在 GTTRP 之限制下，可分成整車路線對子路線間的改善以及整車對卡車路線間的改善。

i. 整車路線對子路線間(1-0)節點交換

圖 4.11(a)為一組合路線可行解，其路線 A 為其主路線由整車服務，路線 B 為其子路線由卡車所服務。若將路線 A 的節點 m 插入路線 B 節點 l 與節點 h 之間(如圖中虛線所示)，在不違反路線 B 之車輛容量限制下，且可改善總路線成本則交換之，成為圖 4.11(b)的二條新路線。此外，對於其交換過程並不止侷限於同一組合路線之主路線與子路線間的交換，亦可對於不同組合路線間之主路線與子路線進行交換。而對於整車與卡車路線間的(1-0)節點交換亦是同道理。在交換過程中，若交換路線只剩下一節點之情況下，則不必考慮交換成本直接交換之，藉以減少車輛使用數。

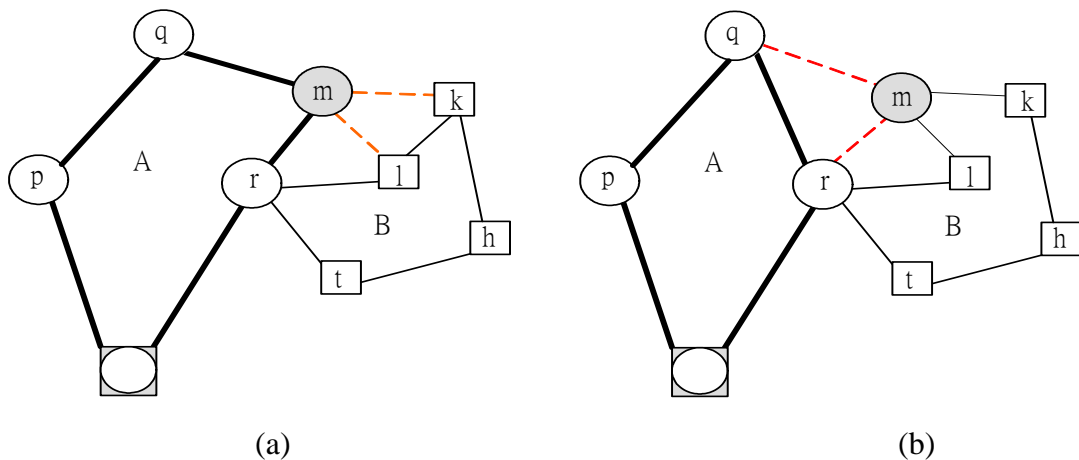


圖 4.11 整車路線對子路線(1-0)節點交換

(四) 拖車停放點交換

圖 4.12(a)，為一組合路線可行解，其路線 A 為其主路線由整車服務，路線 B 為其子路線由卡車所服務。先將路線 b 與拖車停放點 r 連接 \overline{rl} 之 \overline{tr} 與節線打斷如圖(b)所示，接著從路線 A 中找一顧客點作為拖車停放點如圖(c)所示，若成本有所改善時則交換之；否則維持原解。

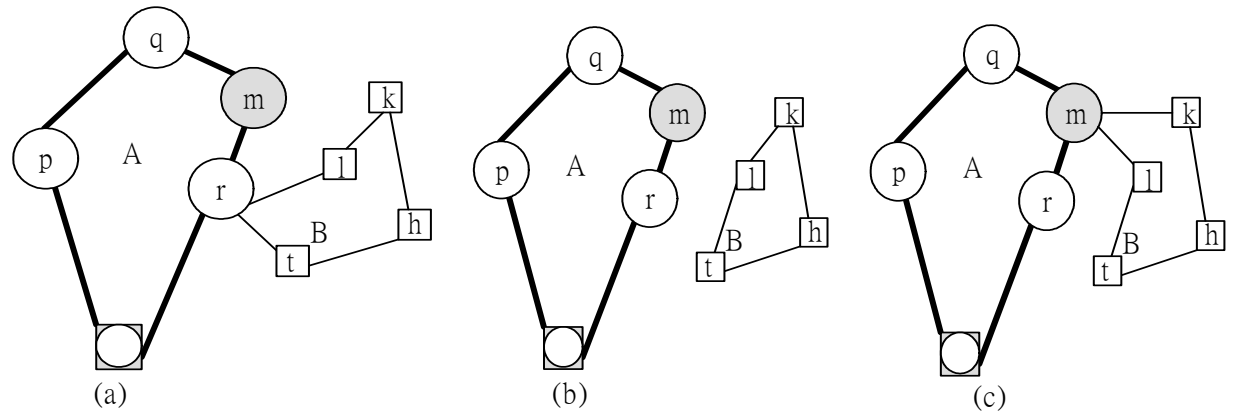


圖 4.12 拖車停放點交換示意圖

在此，本研究將所運用的交換法整理成表 4.2，以更清楚瞭解本研究所使用之交換法：

表 4.2 本研究使用之所有交換法彙整

鄰域改善模組	交換法
路線內	<ul style="list-style-type: none"> ◆ 整車 2-exch ◆ 卡車 2-exch ◆ 子路線 2-exch
同類路線間	<ul style="list-style-type: none"> ◆ 整車-整車 1-0 節點交換 ◆ 卡車-卡車 1-0 節點交換 ◆ 子路線-子路線 1-0 節點交換 ◆ 卡車-子路線 1-0 節點交換 ◆ 子路線-卡車 1-0 節點交換 ◆ 整車-整車 1-1 節點交換 ◆ 卡車-卡車 1-1 節點交換 ◆ 子路線-子路線 1-1 節點交換
異類路線間	<ul style="list-style-type: none"> ◆ 整車-子路線 1-0 節點交換 ◆ 整車-卡車 1-0 節點交換
拖車停放點	<ul style="list-style-type: none"> ◆ 拖車點 1-1 節點交換

第五章 包容性改善程序之 GIDS 模組設計

5.1 門檻型演算法：TA 與 GDA

(一) TA 法

TA 法之控制參數方面，主要為起使門檻值(T_0)與門檻數列長度(K)，而對於門檻收斂型態，雖然有直線型門檻收斂型態、等比遞減型門檻收斂型態以及梯狀遞減型門檻收斂型態等等，但目前在文獻上並沒有提及哪一種收斂型態改善績效比較佳，因此本研究擬以梯狀遞減型門檻收斂型態為主。其整體 TA 法架構如圖 5.1 所示：

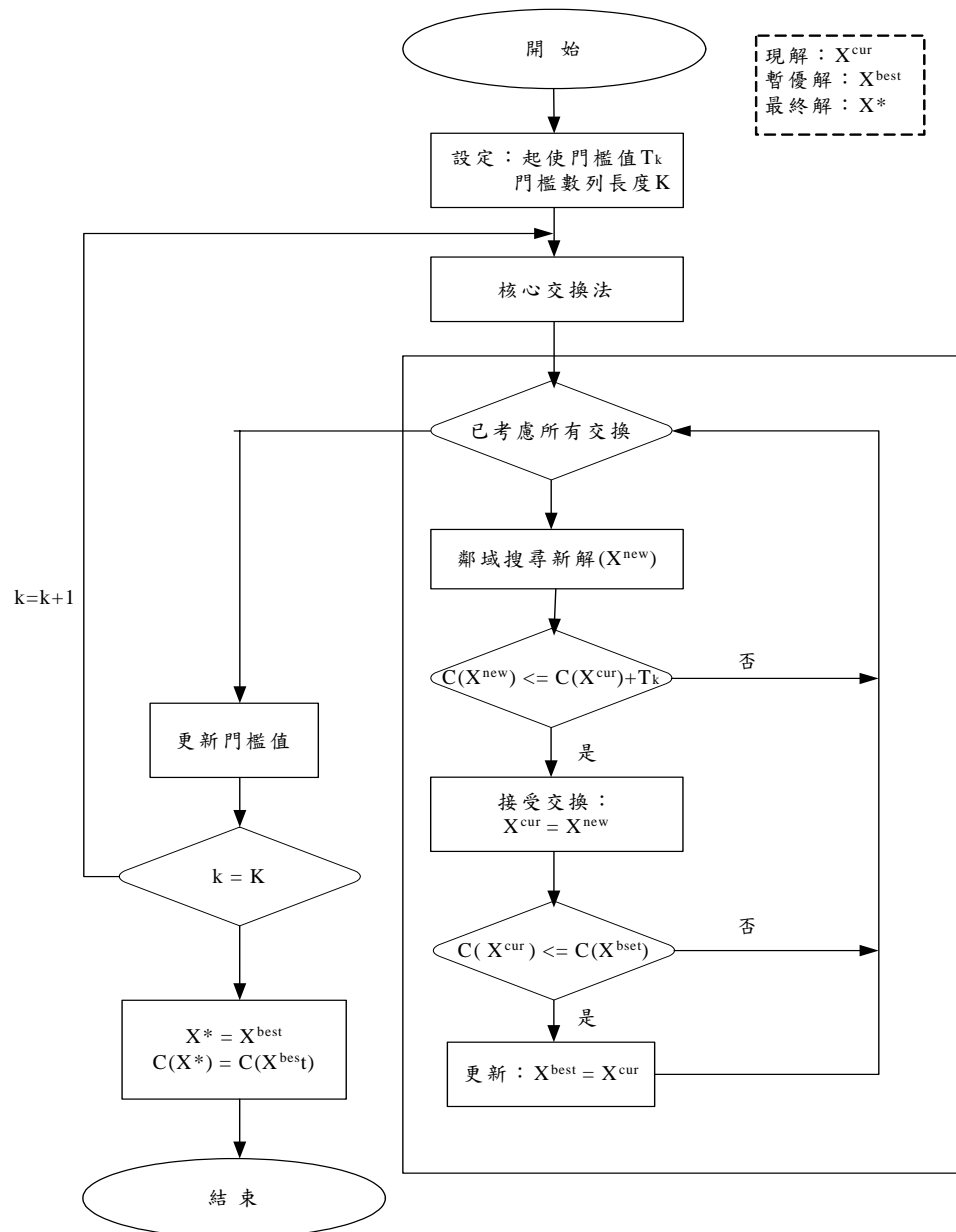


圖 5.1 門檻接受法之執行架構

(二) GDA 法

GDA 法主要是藉由水庫洩洪的概念進行解的改善。由於 GDA 控制參數只有一個，使得該演算法更容易操作，且最終解為整個搜尋過程的最佳紀錄。其執行架構如圖 5.2 所示。

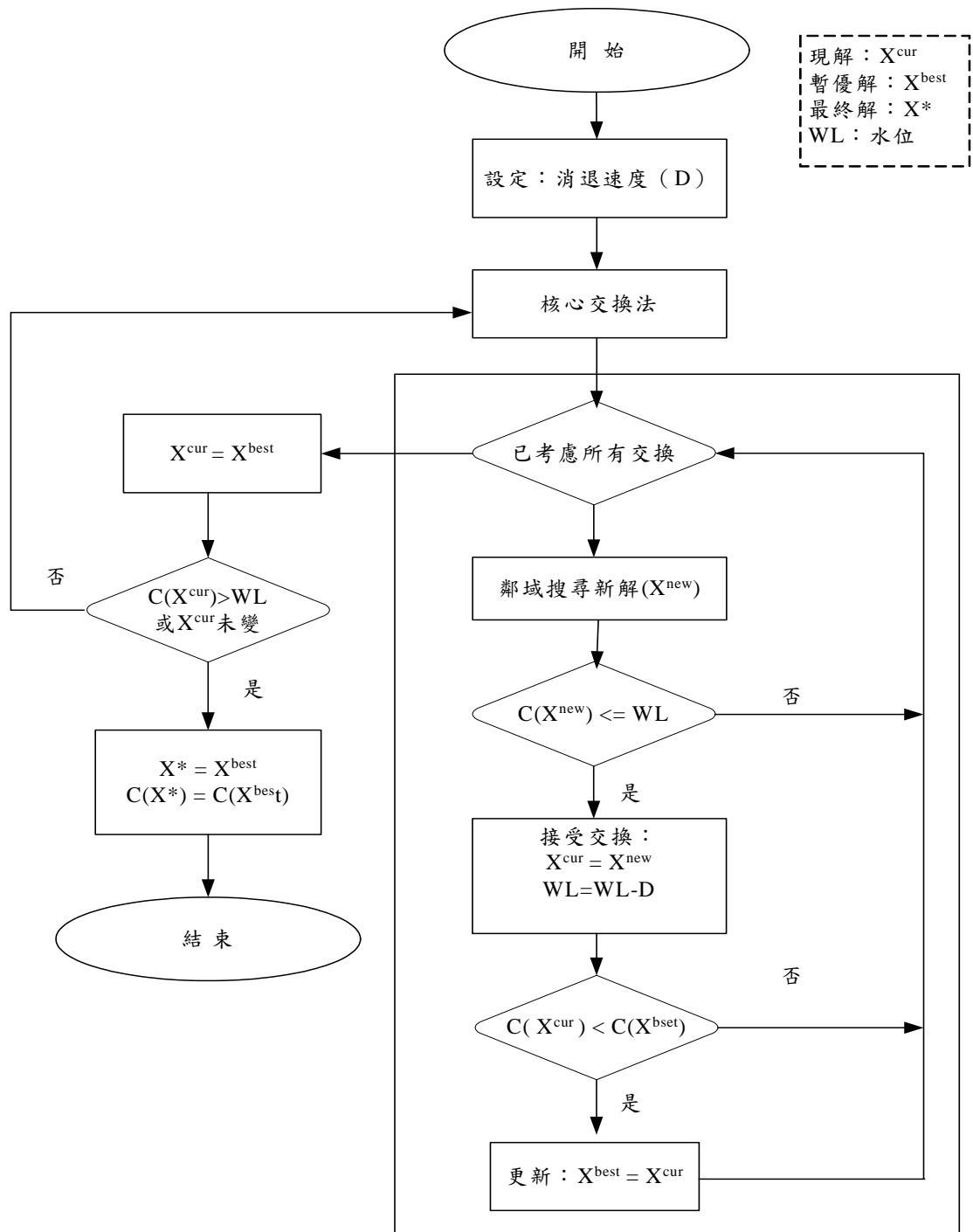


圖 5.2 大洪水法執行架構

5.2 包容性搜尋模組：M1 與 M2 模組

根據卓裕仁[33]對於 M1 與 M2 模組之建議，認為 M1 與 M2 模組所採用的包容性演算法不宜一致，以免陷入搜尋盲點。亦即若本研究於 M1 模組採用 GDA 法時，則 M2 模組需採用 TA 法。而對於其中的鄰域搜尋建議以「最先改善」之選擇法則，即遇到符合接受法則之鄰解立即移動過去，且以「半套」為執行機制，以避免搜尋過程走回頭路。

(一) M1 模組

M1 模組之主要功能在於強化深度化搜尋，其解題程序為包容性搜尋與傳統鄰域搜尋交互執行，當符合停止法則時即停止。GDA 法之停止法則為無法找到低於水位之鄰解即停止；而 TA 法則是滿足執行次數時即停止。然而本研究在 M1 模組設計中，主要是由內、外兩迴圈控制執行，等迴圈執行完畢後再以鄰域搜尋最佳改善接受法則作一結束。其內迴圈主要是當暫優解獲得改善時，即將門檻值減半；而外迴圈主要是當改善模組未獲得改善時，則依照 GDA 或 TA 的收斂法則進行參數更新。圖 5.3 為 M1 模組之執行架構。

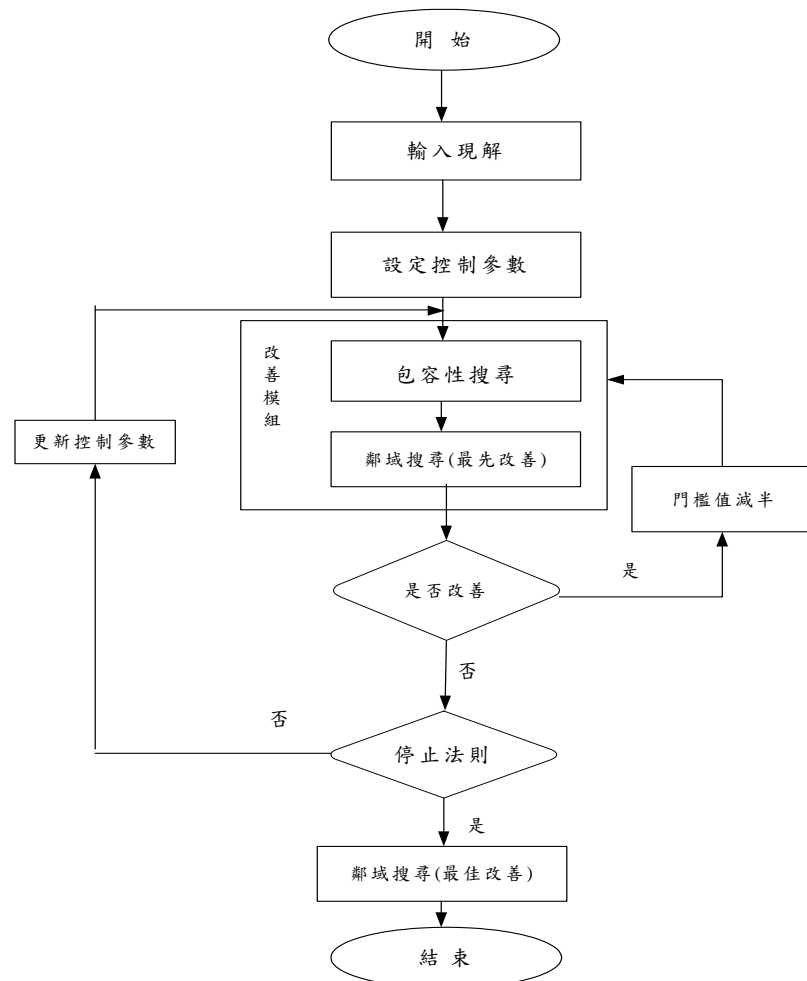


圖 5.3 M1 模組執行架構圖

(二) M2 模組

M2 模組在於跳脫局部最佳解之束縛，與 M1 模組不同之處在於其執行程序為先執行完一完整之包容性搜尋後方再執行傳統鄰域搜尋，其執行架構如圖 5.4 所示：

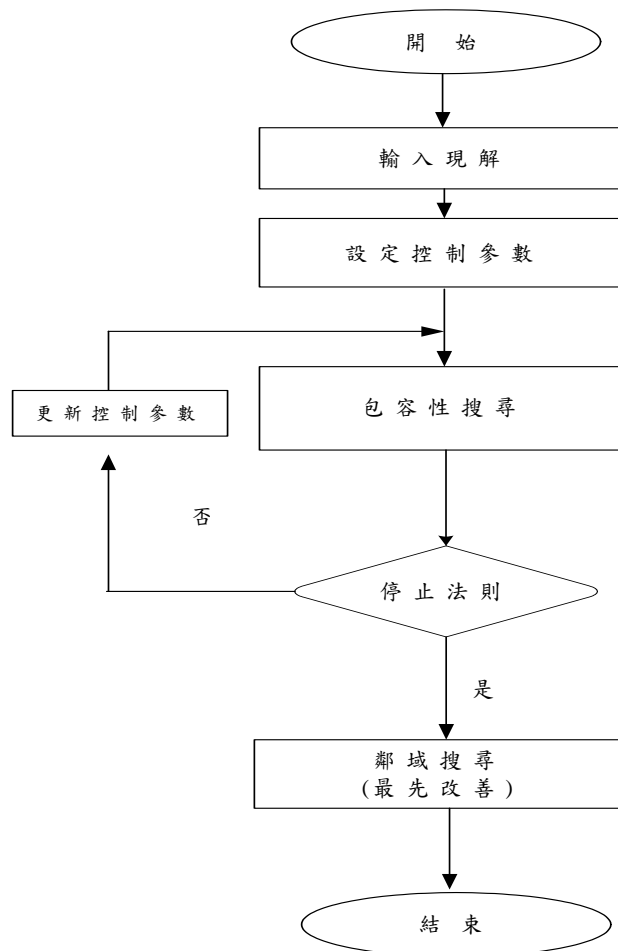
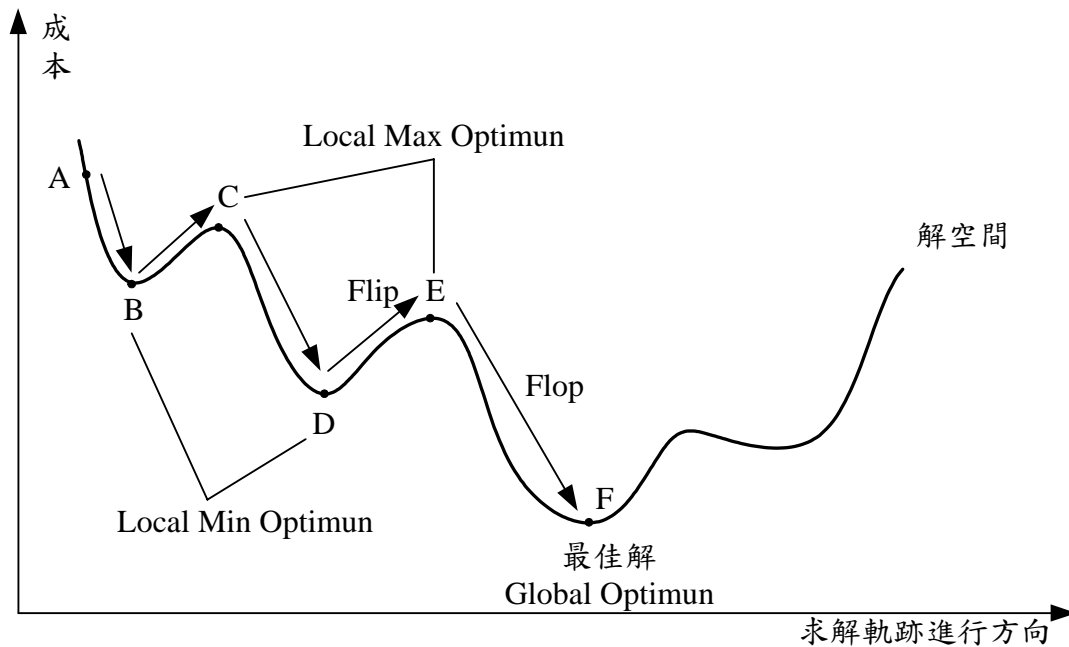


圖 5.4 M2 模組執行架構圖

5.3 成本擾動模組

在成本擾動模組方面，本研究擬採用陳國清[30]所提出的確定型之擾動型演算法—兩極跳躍法(FF)為主。其搜尋原理即當鄰域搜尋陷入一局部最佳解時，將所有成本乘上-1，使最小化問題轉變成求最大化問題進行鄰域搜尋；最後再將成本乘上-1 轉變回求最小化問題，藉此機制擴大搜尋範圍，其示意圖如圖 5.5 所示。為避免搜尋過程重複先前之路徑，於成本擾動前後宜採用不同之鄰域搜尋。本研究擬在成本擾動前採用「最先改善」接受法則，擾動後採用「最佳改善」接受法則。



資料來源：[30]

圖 5.5 兩極跳躍法解題示意圖

第六章 GTTRP 測試與結果分析

6.1 GTTRP 測試例題建立

本研究以 Chao 之 TTRP 題庫[6]為基礎，建立一組共 18 題的 GTTRP 測試題庫，資料型態均屬於「平面(x,y)座標」資料型態，題目規模從 50 位顧客至 199 顧客，其中每一種問題規模之卡車顧客數隨著佔總顧客數之比例 25%、50% 以及 75% 分成三種不同例題。18 題 GTTRP 題庫定義如表 6.1 所示。其中對於 GTTRP 問題中的不可停放拖車之整車顧客點以隨機方式挑選出，拖車停放費用收費範圍為\$0~\$200，至於拖車點之拖車卸下與連結時間，皆為 15 分鐘。此外，對於卡車容量由於以同一車型配送，故其容量皆設為 100。其 GTTRP 測試例題之基本資料請參考附錄一。

表 6.1 GTTRP 測試題庫定義

GTTRP 測試例題								Chao[6] 測試例題
題號	規模	整車顧客數	拖車停放點數	卡車容量	拖車容量	拖車停放費用	拖車裝卸時間	題號
1	50	38	20	100	100	[0,200]	15 分	1
2	50	25	15			[0,200]	15 分	2
3	50	13	7			[0,200]	15 分	3
4	75	57	36	100	100	[0,200]	15 分	4
5	75	38	26			[0,200]	15 分	5
6	75	23	16			[0,200]	15 分	6
7	100	75	45	100	100	[0,200]	15 分	7
8	100	50	33			[0,200]	15 分	8
9	100	25	16			[0,200]	15 分	9
10	120	90	57	100	100	[0,200]	15 分	16
11	120	60	38			[0,200]	15 分	17
12	120	30	20			[0,200]	15 分	18
13	150	113	68	100	100	[0,200]	15 分	10
14	150	75	41			[0,200]	15 分	11
15	150	38	24			[0,200]	15 分	12
16	199	150	95	100	100	[0,200]	15 分	13
17	199	100	66			[0,200]	15 分	14
18	199	50	21			[0,200]	15 分	15

對於整個成本構建合理化方面，本研究以 3.5 噸之小貨車來考量，而小貨車每公里行駛之成本，根據曾國雄[27]對於車輛耗油特性之分析，小貨車每公升約可行駛 9 公里左右，而汽油單價約為 20 元/公升，可換算小貨車每行駛一公里則需花費 2 元的燃油費，至於整車方面則為每公升可行駛 6 公里，其燃油費為 2.67 元；拖車卸下與連結之時間價值方面，則參考 Small[23]對於工作旅次時間價值

之分析約為工資率的 50%，其工資則參考張晉芬[28]對於國內運輸倉儲業之每小時之薪資統計調查為 236 元/每小時，因此其工作旅行時間價值為 168 元/小時。至於車輛固定使用成本方面，其成本主要由司機薪資與折舊費所構成，根據陳俊豪[34]調查 3.5 噸小貨車之固定成本為 1450 元/天。對於拖車之固定使用成本方面，由於使用拖車不需額外增加人員，因此其固定使用成本以折舊費為主，以平均折舊法進行計算，其折舊費為 107 元/天，因此整車之固定使用成本為 1557 元/天。

6.2 起始解與局部最佳解結果分析

GTTRP 在局部構建解程序中，共包含「起始解構建模組」與「鄰域搜尋模組(NS)」部分。對於起始解構建模組方面係以 ILOG Dispatcher 軟體進程式撰寫，而鄰域搜尋模組則是以 C⁺⁺ 語言撰寫而成。在績效評估準則方面，由於 GTTRP 問題並非如傳統 VRP、VRPTW 或者 HVRP 等問題有已有標竿題庫及文獻已知最佳解，因此，本研究以 GTTRP 問題起始解為基準，針對不同模組組合進行改善效率比較。本研究以 AMD 1.6GHz CPU 及 1GHz RAM 在 Microsoft Windows 2000 作業系統下之 PC 進程式測試，其後續之電腦程式運算之執行效率以 CPU 時間(秒)為衡量基準，而鑑於各種方法使用之電腦計算平台速度多有不同，CPU 時間僅視為一參考值。表 6.2 為以 ILOG Dispatcher 求解 GTTRP 之起始解結果，再以表 4.2 所示之傳統鄰域改善模組：路線內、同類路線間、異類路線間以及拖車停放點等改善模組進行改善產生局部最佳解，如表 6.3 所示：

表 6.2 GTTRP 起始解

GTTRP 題號	起始解成本	組合路線	整車路線	卡車路線
1	8769.65	4	0	0
2	9316.80	4	0	0
3	13198.90	4	1	2
4	15786.20	6	1	1
5	17457.20	5	2	2
6	21346.30	7	1	3
7	16207.60	7	1	0
8	18824.20	7	1	1
9	23032.90	7	1	4
10	16419.00	7	0	0
11	16468.10	7	0	0
12	18106.40	8	0	0
13	24514.10	12	0	0
14	25922.80	12	0	0
15	31809.20	11	1	4
16	35187.50	16	0	1
17	36222.70	17	0	1
18	43504.90	15	2	5
平均	21783.03	8.67	0.61	1.33

表 6.3 傳統鄰域改善結果

GTTRP 題號	目標成本	改善率 (%)	執行時間 (秒)	整車節省數 (輛)	卡車節省數 (輛)
1	8367.35	4.59	0.03	--	--
2	8762.09	5.95	0.03	--	--
3	11285.67	14.50	0.05	1	--
4	15217.18	3.60	0.09	--	--
5	17019.88	2.51	0.06	--	--
6	20472.01	4.10	1.66	--	--
7	15643.84	3.48	0.08	--	--
8	17667.76	6.14	0.45	--	--
9	19065.03	17.23	1.02	2	--
10	15402.53	6.19	0.28	--	--
11	15760.09	4.30	0.17	--	--
12	17348.68	4.18	0.13	--	--
13	23274.94	5.05	0.21	--	--
14	24633.45	4.97	0.17	--	--
15	28565.82	10.20	5.28	1	--
16	32647.97	7.22	0.45	--	--
17	34458.51	4.87	0.39	--	--
18	40425.25	7.08	5.21	1	--
平均	20334.34	6.45	0.88		

從表 6.3 可發現以傳統鄰域搜尋法進行起始解改善，改善率從 2.51% 至 17.23%，平均改善率為 6.45%，CPU 執行時間從 0.03 秒至 5.28 秒，平均執行時間為 0.88 秒。

6.3 GIDS 模組測試分析

由於模組之控制參數設定以及組合方式等因素會影響演算法解題績效，因此本研究於包容性搜尋模組測試分析中擬針對控制參數之設定與模組之組合方式兩項因素進行測試。

6.3.1 控制參數之設定

在包容性搜尋法的控制參數方面，TA 法有兩個控制參數：起使門檻值比率 (T_0)與執行次數(K)，並據以構成一門檻值數列 $\{T_k\}$ 。門檻數列收斂型態，本研究採用如圖 6.1 梯狀遞減型門檻數列型態為主。其中， a 為門檻數列長度之 30%處，此時門檻值降至起始門檻的 1/2； b 為門檻數列長度之 70%處，此時的門檻值約降至起使門檻值的 1/3； b 以後之 K 則佔門檻數列之 30%，最後門檻降至零。其起使門檻值 T_1 公式如式(6-1)所示：

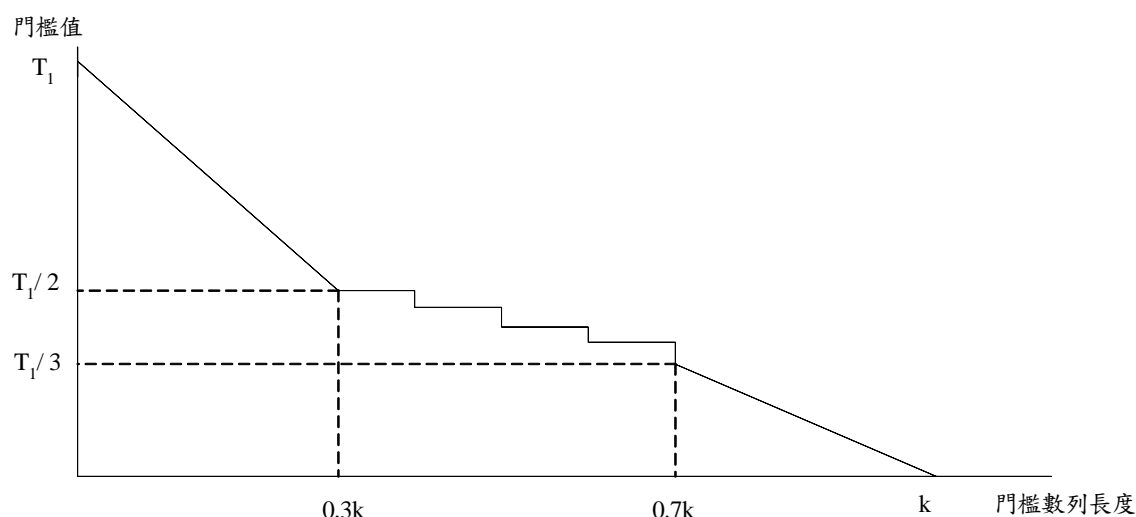


圖 6.1 梯狀遞減型門檻數列型態

$$T_1 = C(\overline{X_0}) \cdot \frac{T_0}{N} \dots\dots\dots(6-1)$$

其中 $C(\overline{X_0})$ 表示不包含車輛使用成本、停車費與拖車卸下、連結時間成本之起始解目標函數值。由於在成本改善方面主要是改善車輛路線成本，若於門檻值設定以 $C(X_0)$ 為主可能造成門檻值過大因而擾動過大，無法做深度搜尋。此外，門檻值亦考慮到問題規模(N)，其原因為在相同的空間範圍內， N 與 $C(\overline{X_0})$ 成正比，而交換型演算法是以節線或節點之交換為基礎，因此取平均之節線成本作為接受門檻值應較為合理。至於 GDA 之控制參數方面，其起始水位(WL_0)與消退速度(D_i)如式(6-2)、式(6-3)與式(6-4)所示，其中 $C(X_0)$ 為起始解之目標函

數值，而在消退速度方面不同以往文獻以固定的消退速度消退，改以等比速度消退，其理由在於 $C(X_0)$ 不只包含路線成本，亦包含車輛使用成本、停車成本與拖車卸下、連結之成本，故若以固定速度消退，若消退速度過小則因擾動過大無法進行深度搜尋，若消退速度過大，則下降速度過大無法找到較佳的解。因此本研究以等比下降速度作為水位下降的策略，其一開始水位下降速度如式(6-3)所示，若有改善時則水位速度乘上 0.5 降低下降速度如式(6-4)所示。

$$WL_0 = C(X_0) \cdot (1 + \frac{T_0}{N}) \dots\dots\dots(6-2)$$

$$D_0 = C(X_0) \cdot \frac{10 \cdot T_0}{N} \dots\dots\dots(6-3)$$

$$D_{i+1} = D_i \cdot 0.5 \quad i=0,1,2,3\dots n \dots\dots\dots(6-4)$$

6.3.2 模組之組合

由於包容性搜尋模組之組件有 M1 與 M2 模組兩種，且模組所搭配的方法有 TA 法與 GDA 法，因此可分成四種組合：M1(TA)、M1(GDA)、M2(TA)以及 M2(GDA)。然而藉著不同組合搭配執行如 M1(TA)+M2(GDA)+M1(TA)之模組執行順序寫法過於冗長，因此本研究針對 M1 與 M2 模組所搭配之門檻型演算法進行寫法的精簡化，以利研究報告的撰寫；對於以 M1 模組執行的 GDA 與 TA 法分別以 G1 組合與 T1 組合表示，以 M2 模組執行的 GDA 與 TA 法則分別以 G2 組合與 T2 組合表示。由於各門檻型演算法在各模組中所適用的 T_0 值參數範圍不盡相同，因此本研究先試著找出 TA 與 GDA 法應用於各模組合適的參數範圍，使其為第一階段測試。接著將 M1 與 M2 模組搭配組合測試，使其為第二階段測試；最後延續第二階段測試結果進行成本擾動(CP)後，適當地重覆整個程序，使其為第三階段測試。

(一) 第一階段測試：

第一階段測試主要是將所求得的起始解 $C(X_0)$ ，分別以 M1 和 M2 模組所搭配的門檻型演算法進行改善測試，以求得各模組合適的參數範圍。由於有 TA 與 GDA 兩種門檻型演算法，因此有 T1、G1、T2 與 G2 四種組合測試。其參數測試範圍如表 6.4 所示進行測試，得到 T1 組合合適之參數範圍為 0.24 之後；G1 組合為 0.1 之前；T2 組合為 0.15~0.24 左右以及 G2 組合為 0.11~0.2 左右。

表 6.4 GTTRP 之 TA 法與 GDA 法參數設定範圍

方法	參數設定範圍
TA	$T_0=0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23, 0.25, 0.27, 0.29$ $K=30$
GDA	$T_0=0.05, 0.07, 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23, 0.25, 0.27, 0.29$

根據求得 M1、M2 模組分別搭配 TA 與 GDA 法所求得之合適參數範圍，本研究各取 10 個參數再進行求解，可得到表 6.5 與表 6.6 之結果。從表中可發現以 T1 組合表現最好，平均改善率為 9.6%，然而執行時間卻為四組測試中最長，平均需花費 14.34 秒；而 G1 組合其平均改善率 8.75% 雖略遜 T1 組合，但其平均執行時間卻比 T1 組合執行時間少的許多。故若以執行之時間效率來考量，G1 組合應為四組比較中表現最好的。不過四組測試結果其平均改善率皆有 8% 以上比傳統鄰域改善法 6.45% 來的佳。

表 6.5 M1 模組測試

模組組合 (模組縮寫)	T1 (TA)				G1 (GDA)			
	To 值	成本	改善率 (%)	執行時間 (秒)	To 值	成本	改善率 (%)	執行時間 (秒)
	0.24	19729.13	8.84	16.50	0.01	19674.60	9.13	3.32
	0.25	19724.65	9.04	11.57	0.02	19675.11	9.13	2.38
	0.26	19607.13	9.79	15.90	0.03	19770.36	8.57	1.91
	0.27	19631.33	9.65	13.58	0.04	19850.68	8.41	1.64
	0.28	19622.33	9.68	13.78	0.05	19818.27	8.57	1.15
	0.29	19622.25	9.66	12.60	0.06	19797.42	8.67	1.31
	0.30	19551.68	10.05	16.97	0.07	19722.51	9.05	1.08
	0.31	19548.09	10.02	16.43	0.08	19819.44	8.60	1.09
	0.32	19620.66	9.71	12.06	0.09	19734.42	9.02	1.17
	0.33	19645.78	9.53	14.10	0.10	19872.68	8.33	1.25
平均		19630.30	9.60	14.34		19773.55	8.75	1.63

表 6.6 M2 模組測試

模組組合 (模組縮寫)	T2 (TA)				G2 (GDA)			
	To 值	成本	改善率 (%)	執行時間 (秒)	To 值	成本	改善率 (%)	執行時間(秒)
	0.15	19673.25	9.20	7.22	0.11	19949.34	7.91	1.27
	0.16	19764.50	8.75	7.30	0.12	19851.48	8.34	0.88
	0.17	19847.76	8.38	5.52	0.13	19949.09	7.96	0.84
	0.18	19770.75	8.77	4.76	0.14	19910.42	8.08	0.94
	0.19	19781.18	8.80	6.76	0.15	19925.54	7.98	0.87
	0.20	19753.74	8.93	7.25	0.16	19891.22	8.23	0.76
	0.21	19745.52	9.02	7.10	0.17	19963.61	7.87	0.84
	0.22	19845.21	8.45	4.91	0.18	19972.95	7.84	0.88
	0.23	19833.97	8.52	4.94	0.19	19890.92	8.22	0.84
	0.24	19835.90	8.48	4.94	0.20	20021.20	7.58	0.94
平均		19785.18	8.73	6.07		19932.58	8.00	0.90

(二) 第二階段測試

第二階段測試主要是承襲第一階段測試，從每種模組與搜尋方法搭配組合中挑選出三個參數 T_0 值，本研究將參數範圍以三等分的方式從中各選出一個 T_0 值，並和與本身不同模組及搭配方法進行組合測試，以測試各種組合在不同 T_0 值搭配執行下之穩定性。其組合有 T1+G2+T1、G1+T2+G1、T2+G1+T2 與 G2+T1+G2 等四種。以 T1+G2+T1 為例，從 T1 組合找出的 T_0 值為 0.25、0.3 以及 0.33，G2 組合則為 0.12、0.16 與 0.2，將 T1 組合之中間 T_0 值 0.3 分別與 G2 三個 T_0 值進行搭配執行，藉以測試其穩定性。其測試結果如表 6.7、6.8、6.9 與 6.10 所示：

表 6.7 T1+G2+T1 組合測試

模組組合方式		T1+G2+T1		
T1_ T_0 值	G2_ T_0 值	成本	改善率(%)	執行時間(秒)
0.03	0.12	19506.89	10.25	32.50
0.03	0.16	19508.11	10.24	22.07
0.03	0.20	19526.53	10.18	30.81
平均		19513.84	10.22	28.46

表 6.8 G1+T2+G1 組合測試

模組組合方式		G1+T2+G1		
G1_To 值	T2_To 值	成本	改善率(%)	執行時間(秒)
0.05	0.15	19603.66	9.40	7.55
0.05	0.20	19620.45	9.55	7.62
0.05	0.24	19705.68	9.17	7.29
平均		19643.27	9.37	7.49

表 6.9 T2+G1+T2 組合測試

模組組合方式		T2+G1+T2		
T2_To 值	G1_To 值	成本	改善率(%)	執行時間(秒)
0.20	0.02	19475.71	10.04	13.88
0.20	0.05	19585.14	9.64	13.19
0.20	0.09	19611.79	9.68	12.56
平均		19557.55	9.79	13.21

表 6.10 G2+T1+G2 組合測試

模組組合方式		G2+T1+G2		
G2_To 值	T1_To 值	成本	改善率(%)	執行時間(秒)
0.16	0.25	19731.49	8.98	24.38
0.16	0.30	19719.66	9.03	24.88
0.16	0.33	19703.64	9.10	24.16
平均		19718.26	9.04	24.47

從上述結果可發現每一種組合在不同 T_0 值搭配下，其改善率差異並沒有很大，說明了各模組組合之穩定性。而測試結果中以 T1+G2+T1 表現最好，其平均改善率為 10.22%，平均執行時間為 28.46 秒，比其在第一階段平均執行時間增加一倍左右。而其次表現較好的為 T2+G1+T2，其平均改善率為 9.79，平均執行時間為 13.21 秒，亦是比其第一階段之執行時間多出一倍左右，然而與 T1+G2+T1 比較，其改善率差異大約為 0.23%，差異不大，而彼此之執行時間差異卻相差一倍左右，故就其執行效率而言應是 T2+G1+T2 表現來的好。

(三) 第三階段測試

第三階段測試主要是於第二階段測試後，進行成本擾動增加廣度化搜尋，並適當地重覆整個程序。前述第二階段測試的執行時間，提供本階段重覆不同模組次數的依據。前階段發現 G1+T2+G1 與 T2+G1+T2 之平均時間較低(約 13 秒)，因此設定重覆擾動十次為主；而 T1+G2+T1 與 G2+T1+G2 費時較長(約 28 秒)因此以重複擾動 5 次為主。本研究在成本擾動的部分，採用兩極跳躍法(FF)為核心模組。符號上，以 FF[·]代表 GIDS 的模式組合，其中·表示包容性搜尋模組組合如 T1+G2+T1；以(m,n)代表 FF 執行的兩個參數，m 代表爬升高度、n 代表擾動次數。FF 各組合之測試結果分別如表 6.11、6.12、6.13 與 6.14 所示。

表 6.11 FF[(T1+G2+T1),(2,5)]組合測試

模組組合方式	FF[T1+G2+T1]				
T1_To 值	G2_To 值	(m,n)	成本	改善率 (%)	執行時間 (秒)
0.30	0.12	(2, 5)	19346.36	10.93	136.26
0.30	0.16	(2, 5)	19364.57	11.06	134.45
0.30	0.20	(2, 5)	19329.57	11.19	134.65
平均			19346.83	11.06	135.12

表 6.12 FF[(G1+T2+G1),(2,10)]組合測試

模組組合方式	FF[G1+T2+G1]				
G1_To 值	T2_To 值	(m,n)	成本	改善率 (%)	執行時間 (秒)
0.05	0.15	(2, 10)	19410.52	10.52	66.92
0.05	0.20	(2, 10)	19473.94	10.43	66.75
0.05	0.24	(2, 10)	19544.11	9.86	60.50
平均			19476.19	10.27	64.73

表 6.13 FF[(T2+G1+T2),(2,10)]組合測試

模組組合方式	FF[T2+G1+T2]				
T2_To 值	G1_To 值	(m,n)	成本	改善率 (%)	執行時間 (秒)
0.20	0.02	(2, 10)	19262.65	11.19	84.70
0.20	0.05	(2, 10)	19215.98	11.72	80.51
0.20	0.09	(2, 10)	19297.10	11.09	75.94
平均			19258.58	11.33	80.38

表 6.14 FF[(G2+T1+G2),(2,5)]組合測試

模組組合方式	FF[G2+T1+G2]				
G2_To 值	T1_To 值	(m,n)	成本	改善率 (%)	執行時間 (秒)
0.16	0.25	(2, 5)	19654.34	9.41	106.46
0.16	0.30	(2, 5)	19442.64	10.33	106.18
0.16	0.33	(2, 5)	19571.61	9.86	126.32
平均			19556.19	9.87	112.99

從測試結果可發現以 FF[T2+G1+T2]表現最好，平均改善率為 11.33%，其次為 FF[T1+G2+T1]，平均改善率為 11.06%，然而 FF[T2+G1+T2]之平均執行時間與 FF[T1+G2+T1]相差 55 秒左右，說明 FF[T2+G1+T2]不僅於精確度比其他組合來得好，其執行效率表現亦不錯。此外，將四種組合測試結果與其在第二階段之測試結果比較，亦是以 FF[T2+G1+T2]改善幅度最大有 1.54%。

從第一階段與第二階段測試結果比較發現，在執行完 M1 或 M2 模組後接續著搭配執行 M2 或 M1 模組，並沒有非常明顯的改善，因此本研究將以深度化搜尋為主之 M1 模組為例，於執行完 M1 模組後直接進行 FF 擾動接著繼續執行 M1 模組，重複該順序 5 次。此外，亦針對 FF 之爬升高度分別以 2、4 及 8 倍之爬升限制進行測試。在所應用門檻型演算法方面，其 T_0 值之選擇將以第一階段表現最佳為選擇依據，因此 G1 組合將以 T_0 值 0.02 進行測試，T1 組合將以 T_0 值 0.3 進行測試，其測試結果如表 6.15 與 6.16 所示：

表 6.15 FF[G1]組合測試

(m,n)	(2,5)		(4,5)		(8,5)	
模組執行順序	成本	執行時間 (秒)	成本	執行時間 (秒)	成本	執行時間 (秒)
G1	19675.24	7.07	19675.24	7.24	19675.24	7.36
FF	20277.91	7.34	21496.54	7.25	21733.48	7.36
G1	19564.80	8.62	19485.38	10.31	19555.96	8.71
FF	20441.19	8.63	21592.84	10.31	21778.71	8.72
G1	19503.03	10.00	19484.39	13.02	19545.17	10.00
FF	20416.80	10.01	21514.98	13.02	21760.49	10.01
G1	19495.56	11.30	19484.39	15.95	19543.00	11.15
FF	20388.37	11.30	21607.08	15.95	21819.99	11.16
G1	19482.62	12.61	19478.42	18.86	19543.00	12.35
FF	20475.79	12.52	21634.11	18.87	21868.13	12.35
G1	19482.62	14.18	19478.42	21.60	19543.00	13.48

表 6.16 FF[T1]組合測試

(m,n)	(2,5)		(4,5)		(8,5)	
模組執行順序	成本	執行時間 (秒)	成本	執行時間 (秒)	成本	執行時間 (秒)
T1	19551.01	22.64	19551.01	20.01	19551.01	19.99
FF	20304.26	22.64	21615.86	20.01	21940.52	20.00
T1	19498.65	42.76	19527.37	37.98	19532.01	38.53
FF	20359.81	42.76	21632.36	37.98	21827.67	38.54
T1	19482.29	62.48	19521.46	54.88	19526.91	55.02
FF	20350.60	62.48	21515.11	54.88	21835.58	55.03
T1	19444.07	82.23	19449.94	71.96	19446.07	71.72
FF	20284.60	82.23	21583.18	71.97	21744.16	71.73
T1	19425.80	102.94	19444.81	89.03	19446.07	88.41
FF	20323.74	102.94	21504.31	89.04	21683.24	88.42
G1	19424.53	122.64	19439.62	105.79	19443.96	104.45

從表 6.15、6.16 可發現，在經過每一次擾動後繼續執行 M1 模組雖有成本上之遞減，但是並沒有很明顯的改善，且隨著不同模組之執行，其 FF 擾動爬升高度限制，並沒有侷限以哪一種爬升高度限制來的最好，並且在三種不同爬升高度限制下測試所得結果，彼此間精確度差異皆低於 0.3% 左右。最後本研究將以上測試於每一次擾動後經過 M1+M2+M1、M2+M1+M2 或者 M1 模組改善結果記錄起來，繪出各方法的成本改善趨勢如圖 6.2 所示，以比較其求解效率。從表中可觀察出除了 FF[G1]遞減趨勢不明顯外，其餘模組經擾動後其平均成本皆有遞減之趨勢，說明了成本擾動模組確有廣度化搜尋之功能，以進一步找到較佳的解。就結果方面以 FF[T2+G1+T2]的表現最好，其平均執行時間只有 80 秒左右，因此不論在時間執行效率或改善率 FF[T2+G1+T2]皆有良好的表現。

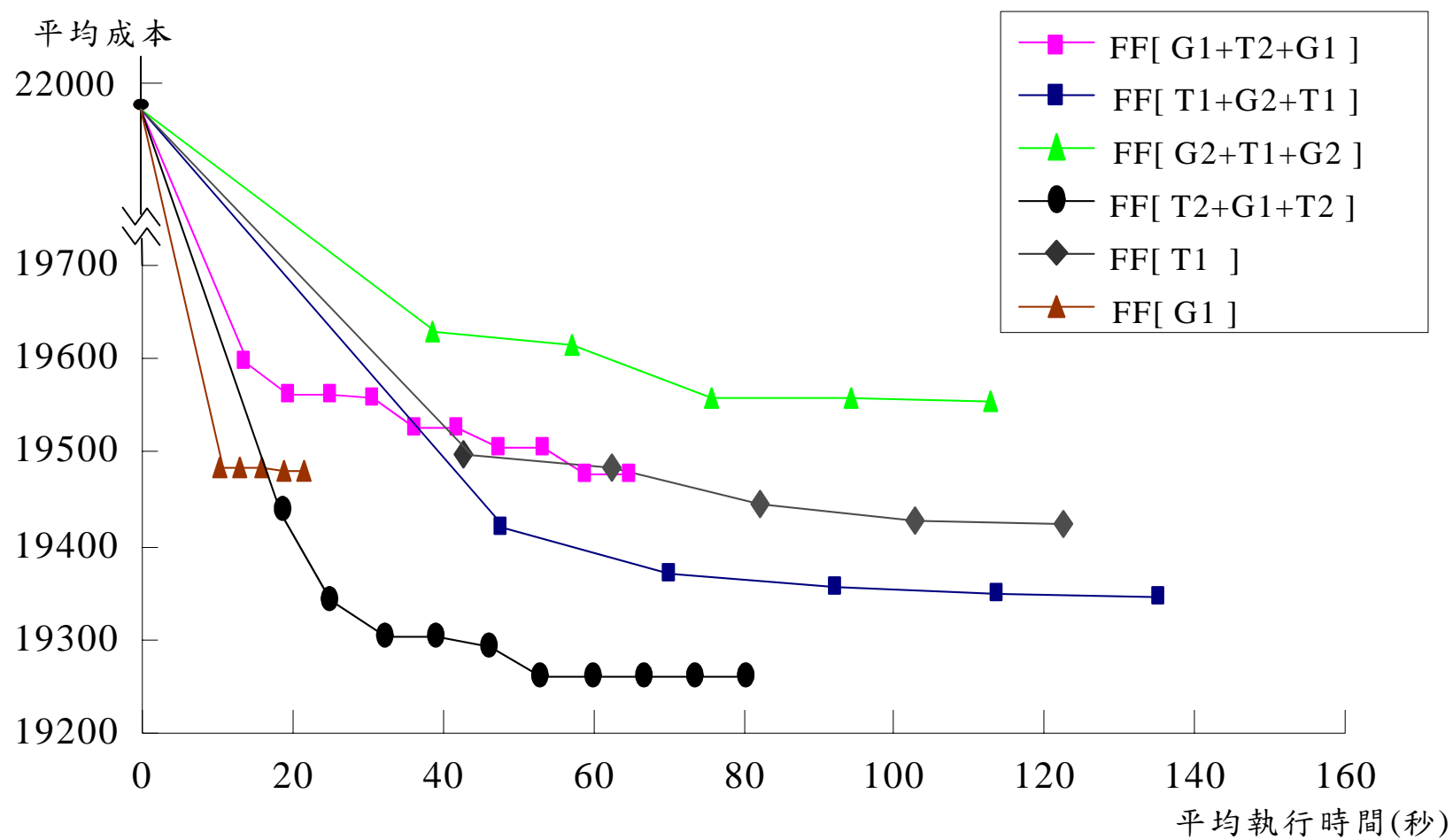


圖 6.2 各方法求解效率比較圖

6.4 GTTRP 最佳結果分析

從 6.3.2 節對於 GTTRP 所測試之結果，本研究將每一題所測得之最佳結果整理成表 6.17。其路線圖請參考附錄一。

表 6.17 18 題 GTTRP 最佳結果

GTTRP 題號	規模	目標成本	執行時間 (秒)	FF[模組組合]	控制參數	
					(GDA_To, TA_To)	(m, n)
1	50	8137.88	67.31	FF[G1+T2+G1]	(0.05,0.24)	(2,10)
2	50	8350.93	63.73	FF[G1+T2+G1]	(0.05,0.2)	(2,10)
3	50	9914.96	119.87	FF[T1]	(X, 0.3)	(4,5)
4	75	14741.8	99.10	FF[T2+G1+T2]	(0.02,0.2)	(2,10)
5	75	15138.7	92.47	FF[T2+G1+T2]	(0.09,0.2)	(2,10)
6	75	16894.5	94.18	FF[T2+G1+T2]	(0.09,0.2)	(2,10)
7	100	15323.2	68.14	FF[G1+T2+G1]	(0.05,0.24)	(2,10)
8	100	15717.3	72.66	FF[G1+T2+G1]	(0.16,0.2)	(2,10)
9	100	17590.2	105.21	FF[G2+T1+G2]	(0.16,0.3)	(2,5)
10	120	14434.7	101.46	FF[T2+G1+T2]	(0.09,0.2)	(2,10)
11	120	14689.4	107.57	FF[G2+T1+G2]	(0.16,0.33)	(2,5)
12	120	15580.6	142.40	FF[T1+G2+T1]	(0.12, 0.3)	(2,5)
13	150	22403.80	12.02	FF[G1]	(0.02, X)	(2,5)
14	150	23520.3	112.53	FF[G2+T1+G2]	(0.16, 0.3)	(2,5)
15	150	26668.1	73.53	FF[T2+G1+T2]	(0.09,0.2)	(2,10)
16	199	30964.80	28.90	FF[G1]	(0.02, X)	(4,5)
17	199	31854.90	23.27	FF[G1]	(0.02, X)	(2,5)
18	199	38709.60	31.17	FF[G1]	(0.02, X)	(2,5)
平均		18830.38	80.54			

從表中可觀察出，每一題大約可在 2 分鐘左右求得結果，方法上以 FF[T2+G1+T2]組合較佳。然而，值得一提的是，規模為 199 點的問題以 FF[G1]組合所得的結果最好，執行時間大約為 30 秒左右，極具潛力，對於後續大規模之 GTTRP 問題之研究可朝該組合進一步探討。

6.5 小結

由上述測試結果可歸納出以下幾點結論：

1. 測試過程中，可發現 GIDS_GTTRP 各模組之解題績效呈現逐漸改善的情形，說明各模組及其組合方法確實具有改善效果。
2. 雖然 M1 模組具有強化搜尋功能，M2 模組為偏重在加強跳離局部最佳解之效果，然而過去經驗往往先執行 M1 模組進行細緻搜尋動作，接著執行 M2 模組以延展搜尋範圍，最後再執行 M1 模組對新的範圍進行強化搜尋。但是，從本研究所求得之結果可發現，包容性搜尋模組之 M1、M2 模組並沒有特定之執行順序，不同之執行順序亦會有較佳之結果。

第七章 TTRP 測試結果分析

7.1 TTRP 結果分析

本研究將所設計之 GTTRP 程式模組亦應用於測試 TTRP，藉以判斷本研究所設計模組之解題績效。由於本研究所設計之模組並非專為求解 Chao 之 TTRP 例題，因此，在測試過程並不計畫仔細尋找適合 TTRP 之參數值。此外，在比較結果方面，因不同測試平台會造成執行效率上不同，因此，後續之執行時間比較只作一參考；Chao 之測試平台是在 350MHZ Pentium II 的 PC 進行測試；本研究測試平台在 AMD 1.6G CPU 之 PC 進行測試。在此本研究以 6.3.2 節針對 GTTRP 第一階段求解結果於扣除車輛固定使用成本情況下整理出表 7.1 與 7.2，找出應用於 M1 與 M2 模組合適的門檻型演算法，以及合適的參數 T_0 值。

表 7.1 GTTRP 扣除車輛固定使用成本之 M1 模組測試

模組組合 (模組縮寫)	T1 (TA)					G1 (GDA)				
	T_0	改善率 (%)	執行時間 (秒)	整車數 (輛)	卡車數 (輛)	T_0	改善率 (%)	執行時間 (秒)	整車數 (輛)	卡車數 (輛)
	0.24	23.19	16.50	160	24	0.01	24.47	3.32	159	24
	0.25	23.40	11.57	159	24	0.02	25.72	2.38	160	24
	0.26	23.76	15.90	159	23	0.03	24.10	1.91	161	24
	0.27	23.24	13.58	159	23	0.04	24.22	1.64	161	24
	0.28	23.39	13.78	159	23	0.05	23.37	1.15	160	24
	0.29	23.46	12.60	159	23	0.06	23.54	1.31	161	23
	0.30	23.17	16.97	159	22	0.07	23.38	1.08	160	23
	0.31	23.03	16.43	159	22	0.08	23.20	1.09	161	23
	0.32	23.33	12.06	159	23	0.09	23.30	1.17	160	23
	0.33	22.84	14.10	159	23	0.10	22.29	1.25	161	23
平均		23.28	14.35	159.10	23.00		23.76	1.63	160.40	23.50

表 7.2 GTTRP 扣除車輛固定使用成本之 M2 模組測試

模組組合 (模組縮寫)	T2 (TA)					G2 (GDA)				
	To	改善率 (%)	執行時間 (秒)	整車數 (輛)	卡車數 (輛)	To	改善率 (%)	執行時間 (秒)	整車數 (輛)	卡車數 (輛)
	0.15	24.33	7.22	160	23	0.11	20.98	1.27	160	24
	0.16	24.17	7.3	161	23	0.12	22.32	0.88	160	24
	0.17	24.16	5.52	161	24	0.13	22.37	0.84	161	24
	0.18	24.18	4.76	160	24	0.14	22.87	0.94	161	24
	0.19	24.17	6.76	161	23	0.15	22.51	0.87	161	24
	0.20	24.57	7.25	160	24	0.16	22.17	0.76	161	23
	0.21	24.60	7.1	160	24	0.17	22.00	0.84	161	24
	0.22	24.29	4.91	161	24	0.18	21.99	0.88	161	24
	0.23	24.57	4.94	161	24	0.19	22.11	0.84	161	23
	0.24	24.38	4.94	161	24	0.2	21.26	0.94	161	24
平均		24.34	6.07	160.60	23.70		22.06	0.91	160.80	23.80

從表 7.1、7.2 可觀察出 M1 模組以搭配 GDA 法表現較好，平均改善率為 23.76%，然而該組合對於 T_0 值比較敏感，表現較好的 T_0 值為 0.01 與 0.02；M2 模組以搭配 TA 法表現較好，平均改善率為 24.34%，然而從表 7.2 可發現該組合對於 T_0 值比較不敏感，因此在測試時 T2 將分別以 $T_0=0.15$ 、 0.16 進行搭配測試。故在 G1 與 T2 的搭配下有 FF[G1+T2+G1]及 FF[T2+G1+T2]之測試組合。其測試結果如表 7.3 至 7.6 所示：

表 7.3 FF[(G1+T2+G1),(2,1)]測試結果 I

TTRP 題號	Chao 最佳解			本研究 FF[(G1+T2+G1),(2,1)]			
				G1_T0=0.01,T2_T0=0.15			
	總車數(整車數,卡車數)	目標 成本	執行時間 (秒)	總車數(整車數,卡車數)	目標 成本	誤差 (%)	執行時間 (秒)
1	5(3,2)	565.02	33901.2	5(4,1)	615.18	8.88	17.4
2	5(3,2)	658.07	39484.2	5(4,1)	684.05	3.95	12.6
3	5(3,2)	648.74	38924.4	5(4,1)	713.77	10.02	12
4	9(5,4)	856.2	51372	8(7,1)	857.32	0.13	16.2
5	9(5,4)	949.98	56998.8	9(7,2)	980.98	3.26	16.2
6	9(5,4)	1053.23	63193.8	9(6,3)	1069.79	1.57	14.4
7	8(4,4)	832.56	49953.6	7(6,1)	849.60	2.05	22.8
8	8(4,4)	900.54	54032.4	7(6,1)	965.03	7.16	16.8
9	8(4,4)	971.62	58297.2	9(6,3)	1103.48	13.57	17.4
10	12(6,6)	1073.5	64410	10(9,1)	1136.38	5.86	22.8
11	12(6,6)	1170.17	70210.2	10(10,0)	1271.54	8.66	22.8
12	12(6,6)	1217.01	73020.6	10(9,1)	1419.21	16.61	17.4
13	18(9,8)	1364.5	81870	14(13,1)	1385.71	1.55	27
14	18(9,8)	1464.2	87852	14(13,1)	1501.87	2.57	25.2
15	18(9,8)	1540.25	92415	16(15,1)	1810.38	17.54	21
16	7(4,3)	1041.36	62481.6	6(6,0)	1117.41	7.30	21.6
17	7(4,3)	1090.46	65427.6	6(6,0)	1104.36	1.27	19.8
18	7(4,3)	1141.36	68481.6	7(6,1)	1245.08	9.09	16.8
19	10(5,5)	854.02	51241.2	9(8,1)	844.97	(1.06)	14.4
20	10(5,5)	942.39	56543.4	9(7,2)	970.41	2.97	10.8
21	10(5,5)	926.47	55588.2	9(8,1)	1014.31	9.48	13.2
平 均		1063.08	60747.57		1133.04	6.62	19.2

表 7.4 FF[(G1+T2+G1),(2,1)]測試結果 II

TTRP 題號	Chao 最佳解			本研究 FF[(G1+T2+G1),(2,1)]			
				G1_T ₀ =0.02,T2_T ₀ =0.16			
	總車數(整車數,卡車數)	目標 成本	執行時間 (秒)	總車數(整車數,卡車數)	目標 成本	誤差 (%)	執行時間 (秒)
1	5(3,2)	565.02	33901.2	5(4,1)	610.052	7.97	13.8
2	5(3,2)	658.07	39484.2	5(4,1)	693.605	5.40	10.8
3	5(3,2)	648.74	38924.4	5(4,1)	699.516	7.83	10.8
4	9(5,4)	856.2	51372	8(7,1)	840.221	(1.87)	12.6
5	9(5,4)	949.98	56998.8	8(6,2)	1061.94	11.79	13.2
6	9(5,4)	1053.23	63193.8	9(6,3)	1015.86	(3.55)	13.8
7	8(4,4)	832.56	49953.6	7(6,1)	847.77	1.83	19.8
8	8(4,4)	900.54	54032.4	7(6,1)	941.96	4.60	15.6
9	8(4,4)	971.62	58297.2	9(5,4)	1077.43	10.89	16.8
10	12(6,6)	1073.5	64410	10(9,1)	1142.87	6.46	18
11	12(6,6)	1170.17	70210.2	10(10,0)	1293.57	10.55	17.4
12	12(6,6)	1217.01	73020.6	10(9,1)	1432.40	17.70	18.6
13	18(9,8)	1364.5	81870	14(13,1)	1381.85	1.27	23.4
14	18(9,8)	1464.2	87852	14(13,1)	1461.37	(0.19)	19.8
15	18(9,8)	1540.25	92415	17(16,1)	1756.71	14.05	17.4
16	7(4,3)	1041.36	62481.6	6(6,0)	1120.05	7.56	15
17	7(4,3)	1090.46	65427.6	6(6,0)	1102.55	1.11	14.4
18	7(4,3)	1141.36	68481.6	7(6,1)	1271.76	11.42	14.4
19	10(5,5)	854.02	51241.2	8(7,1)	874.18	2.36	13.8
20	10(5,5)	942.39	56543.4	9(7,2)	962.65	2.15	10.2
21	10(5,5)	926.47	55588.2	8(7,1)	1226.89	32.43	10.8
平 均		1063.08	60747.57		1140.76	7.59	16.2

表 7.5 FF[(T2+G1+T2),(2,1)]測試結果 I

TTRP 題號	Chao 最佳解			本研究 FF[(T2+G1+T2),(2,1)]			
				T2_T0=0.15,G1_T0=0.01			
	總車數(整車數,卡車數)	目標 成本	執行時間 (秒)	總車數(整車數,卡車數)	目標 成本	誤差 (%)	執行時間 (秒)
1	5(3,2)	565.02	33901.2	5(4,1)	610.052	7.97	13.8
2	5(3,2)	658.07	39484.2	5(4,1)	694.05	5.47	17.4
3	5(3,2)	648.74	38924.4	5(4,1)	733.43	13.05	16.8
4	9(5,4)	856.2	51372	8(7,1)	841.74	(1.69)	19.8
5	9(5,4)	949.98	56998.8	9(7,2)	1003.74	5.66	21.6
6	9(5,4)	1053.23	63193.8	9(6,3)	1014.18	(3.71)	21
7	8(4,4)	832.56	49953.6	7(6,1)	855.74	2.78	25.8
8	8(4,4)	900.54	54032.4	7(6,1)	969.43	7.65	22.2
9	8(4,4)	971.62	58297.2	8(5,3)	1083.25	11.49	25.2
10	12(6,6)	1073.5	64410	10(9,1)	1129.43	5.21	27.6
11	12(6,6)	1170.17	70210.2	10(10,0)	1268.35	8.39	28.2
12	12(6,6)	1217.01	73020.6	10(10,0)	1456.82	19.70	28.2
13	18(9,8)	1364.5	81870	14(13,1)	1394.26	2.18	32.4
14	18(9,8)	1464.2	87852	14(13,1)	1582.51	8.08	30
15	18(9,8)	1540.25	92415	16(15,1)	1799.04	16.80	28.8
16	7(4,3)	1041.36	62481.6	6(6,0)	1138.60	9.34	22.2
17	7(4,3)	1090.46	65427.6	6(6,0)	1119.86	2.70	22.8
18	7(4,3)	1141.36	68481.6	7(6,1)	1262.85	10.64	22.2
19	10(5,5)	854.02	51241.2	8(7,1)	854.91	0.10	14.4
20	10(5,5)	942.39	56543.4	9(7,2)	988.04	4.84	10.2
21	10(5,5)	926.47	55588.2	9(8,1)	1028.65	11.03	19.8
平 均		1063.08	60747.57		1141.45	7.38	23.4

表 7.6 FF[(T2+G1+T2),(2,1)]測試結果 II

TTRP 題號	Chao 最佳解			本研究 FF[(T2+G1+T2),(2,1)]			
				T2_T0=0.16,G1_T0=0.02			
	總車數(整車數,卡車數)	目標 成本	執行時間 (秒)	總車數(整車數,卡車數)	目標 成本	誤差 (%)	執行時間 (秒)
1	5(3,2)	565.02	33901.2	5(4,1)	613.64	8.60	21
2	5(3,2)	658.07	39484.2	5(4,1)	687.86	4.53	16.2
3	5(3,2)	648.74	38924.4	5(4,1)	702.50	8.29	16.2
4	9(5,4)	856.2	51372	8(7,1)	836.39	(2.31)	18
5	9(5,4)	949.98	56998.8	9(7,2)	956.51	0.69	21.6
6	9(5,4)	1053.23	63193.8	9(6,3)	1022.73	(2.90)	19.8
7	8(4,4)	832.56	49953.6	7(6,1)	890.72	6.99	24
8	8(4,4)	900.54	54032.4	7(6,1)	934.65	3.79	21
9	8(4,4)	971.62	58297.2	9(6,3)	1134.87	16.80	21.6
10	12(6,6)	1073.5	64410	10(9,1)	1143.68	6.54	27.6
11	12(6,6)	1170.17	70210.2	10(10,0)	1306.33	11.64	26.4
12	12(6,6)	1217.01	73020.6	11(10,1)	1409.37	15.81	30
13	18(9,8)	1364.5	81870	14(13,1)	1390.47	1.90	29.4
14	18(9,8)	1464.2	87852	14(13,1)	1569.40	7.18	27.6
15	18(9,8)	1540.25	92415	16(15,1)	1777.14	15.38	27.6
16	7(4,3)	1041.36	62481.6	6(6,0)	1136.10	9.10	24.6
17	7(4,3)	1090.46	65427.6	6(6,0)	1139.17	4.47	20.4
18	7(4,3)	1141.36	68481.6	7(6,1)	1249.75	9.50	22.2
19	10(5,5)	854.02	51241.2	9(8,1)	867.666	1.60	21
20	10(5,5)	942.39	56543.4	9(7,2)	971.515	3.09	17.4
21	10(5,5)	926.47	55588.2	9(8,1)	1029	11.07	18
平 均		1063.08	60747.57		1138.47	7.09	23.4

從表 7.3 至表 7.6 之結果整理中，以 FF[G1+T2+G1]在 G1_T0=0.01、T2_T0=0.15 時和 Chao 之 TTRP 測試結果比較之平均誤差率最低為 6.62%，且第 19 題測試結果比 Chao 之結果還要好。然而其他三個測試結果之平均誤差雖不比 FF[G1+T2+G1]來的好，但是皆至少有 2 題結果比 Chao 的結果來的好。因此，本研究將繼續從表 7.1、7.2 找出改善率最佳的模組與搭配參數進行多次擾動測試，從表中可發現以 M1 模組中的 GDA 法在 T0=0.02 時表現最佳，因此將以 FF[G1]擾動 5 次，且在不同爬升高度限制 1.25、1.5、1.75 以及 2 倍的條件下繼續進行測試，其測試結果如表 7.7 所示：

表 7.7 FF[G1]擾動測試 TTRP 之結果整理表

(m,n)	(1.25,5)		(1.5,5)		(1.75,5)		(2,5)	
執行模組	平均 成本	執行時間 (秒)	平均 成本	執行時間 (秒)	平均 成本	執行時間 (秒)	平均 成本	執行時間 (秒)
G1	1094.69	6.83	1094.69	6.77	1094.69	6.79	1094.69	6.64
FF	1127.99	6.84	1154.27	6.79	1226.39	7.12	1297.60	6.66
G1	1086.95	7.66	1082.02	7.72	1103.80	7.97	1102.12	8.10
FF	1116.26	7.67	1143.05	7.74	1197.01	8.33	1294.77	8.12
G1	1085.01	8.33	1080.59	8.58	1097.36	9.16	1105.12	9.48
FF	1109.05	8.34	1132.74	8.61	1199.61	9.50	1306.42	9.51
G1	1082.41	8.93	1081.21	9.45	1092.09	10.35	1104.33	10.80
FF	1109.94	8.94	1135.54	9.47	1186.82	10.67	1292.73	10.83
G1	1078.01	9.58	1080.04	10.27	1090.22	11.38	1102.87	12.04
FF	1101.66	9.59	1133.45	10.29	1189.99	11.68	1281.49	12.07
G1	1075.49	10.20	1079.41	11.09	1088.89	12.45	1100.14	13.30

從表 7.7 可觀察出爬升高度在 1.25 倍的限制下其平均成本最低為 1075.49。然而本研究是以 GTTRP 之程式模組對於 TTRP 例題進行測試，從表中可觀察出在每一次執行完 G1 後，理應其平均成本需是漸漸遞減或不變，而表中確有增加之現象，其理由為 GTTRP 之程式模組是以減少車輛使用優先執行的概念設計，因此應用於 TTRP 例題時程式執行仍是以減少車輛使用為主，故在車輛減少的情況下，方才出現路線成本增加的現象。最後本研究將以表現最佳之爬升高度限制為 1.25 倍之改善資料，繪出如圖 7.1 多次擾動下之平均成本趨勢圖，以更明白表現出擾動後平均成本增減趨勢。

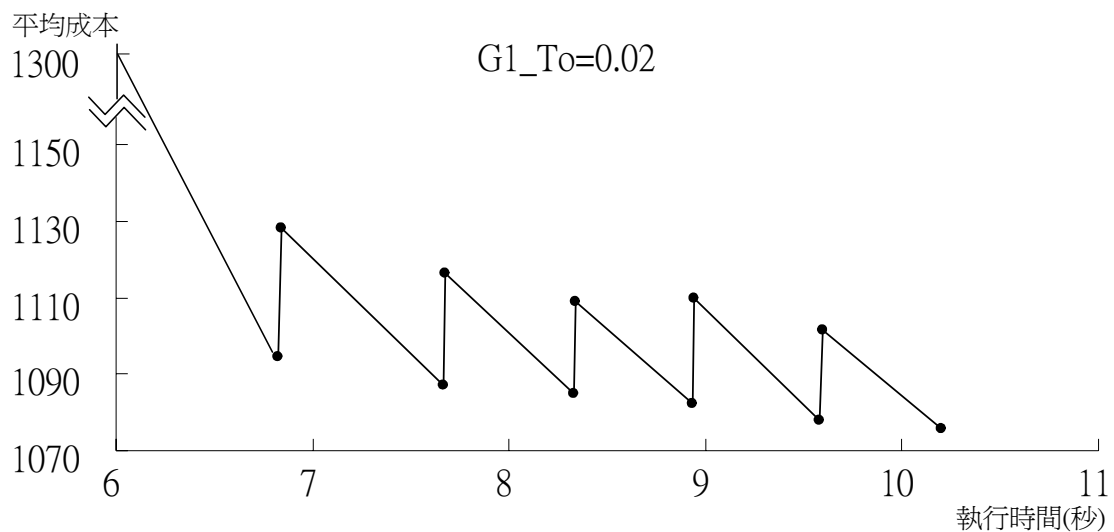


圖 7.1 FF[G1,(1.25,5)]擾動測試 TTRP 之平均成本趨勢圖

本研究將一連串對 TTRP 例題測試之結果，從中將每一題最佳結果彙整成表 7.8，其路線圖請參考附錄二。

表 7.8 本研究求得 TTRP 最佳結果

TTRP 題號	Chao 最佳解			本研究最佳結果				FF[模組組合]	控制參數	
	總車數(V,T)*	目標 成本	Time (秒)	總車數(V,T)*	目標 成本	誤差 (%)	Time (秒)		(GDA_To,TA_To)	(m,n)
1	5(3,2)	565.02	33901.2	5(4,1)	598.56	5.94	6.80	FF[G1]	(0.02 , X)	(1.25,5)
2	5(3,2)	658.07	39484.2	5(4,1)	684.05	3.95	12.88	FF[G1+T2+G1]	(0.15,0.01)	(2,1)
3	5(3,2)	648.74	38924.4	5(4,1)	689.29	6.25	10.72	FF[G1]	(0.02 , X)	(1.75,5)
4	9(5,4)	856.2	51372	8(7,1)	836.39	(2.31)	17.72	FF[T2+G1+T2]	(0.16,0.02)	(2,1)
5	9(5,4)	949.98	56998.8	9(7,2)	956.51	0.69	21.35	FF[T2+G1+T2]	(0.16,0.02)	(2,1)
6	9(5,4)	1053.23	63193.8	9(6,3)	1014.18	(3.71)	21.06	FF[T2+G1+T2]	(0.01 , 0.15)	(2,1)
7	8(4,4)	832.56	49953.6	7(6,1)	847.77	1.83	19.87	FF[G1+T2+G1]	(0.02,0.16)	(2,1)
8	8(4,4)	900.54	54032.4	7(6,1)	934.65	3.79	20.70	FF[T2+G1+T2]	(0.16,0.02)	(2,1)
9	8(4,4)	971.62	58297.2	9(5,4)	1060.11	9.11	12.29	FF[G1]	(0.02 , X)	(1.25,5)
10	12(6,6)	1073.5	64410	10(9,1)	1105.59	2.99	10.19	FF[G1]	(0.02 , X)	(2,5)
11	12(6,6)	1170.17	70210.2	10(10,0)	1268.35	8.39	28.05	FF[T2+G1+T2]	(0.15,0.01)	(2,1)
12	12(6,6)	1217.01	73020.6	11(10,1)	1402.34	15.23	13.46	FF[G1]	(0.02 , X)	(1.25,5)
13	18(9,8)	1364.5	81870	14(13,1)	1376.94	0.91	16.61	FF[G1]	(0.02 , X)	(2,5)
14	18(9,8)	1464.2	87852	14(13,1)	1450.03	(0.97)	13.89	FF[G1]	(0.02 , X)	(1.25,5)
15	18(9,8)	1540.25	92415	17(16,1)	1756.71	14.05	17.40	FF[G1+T2+G1]	(0.02,0.16)	(2,1)
16	7(4,3)	1041.36	62481.6	6(6,0)	1085.23	4.21	12.75	FF[G1]	(0.02 , X)	(1.5,5)
17	7(4,3)	1090.46	65427.6	6(6,0)	1102.55	1.11	14.27	FF[G1+T2+G1]	(0.02,0.16)	(2,1)
18	7(4,3)	1141.36	68481.6	7(6,1)	1245.08	9.09	16.90	FF[G1+T2+G1]	(0.01,0.15)	(2,1)
19	10(5,5)	854.02	51241.2	9(8,1)	844.54	(1.11)	8.36	FF[G1]	(0.02 , X)	(1.25,5)
20	10(5,5)	942.39	56543.4	9(7,2)	933.75	(0.92)	11.37	FF[G1]	(0.02 , X)	(1.75,5)
21	10(5,5)	926.47	55588.2	9(8,1)	1013.01	9.34	10.02	FF[G1]	(0.02 , X)	(1.25,5)
平 均	9.6	1012.46	60747.57	8.8	1057.41	4.18	15.08			

*V：整車數，T：卡車數

表中可發現本研究與 Chao[6]比較，有 5 題優於 Chao 之結果，其平均誤差只有 4.18%，說明本研究所設計之模組在求解 TTRP 問題上亦有良好的解題績效。由於本研究所設計的模組是以減少車輛使用為優先執行，因此，在車輛減少路線成本增加情況下，某些例題之誤差率可能會特別大，不過亦有些例題如例題 7、10、13 以及 17 雖然因車輛減少造成路線成本增加，但是在車輛數低於 Chao 所使用之車輛數情況下，其誤差率皆低於 3%，證明 GIDS 解題績效能力。在執

行時間方面，Chao 之平均執行時間為 60747 秒，而本研究只有 15.08 秒，效率上較其平均快了 4000 倍以上。

7.2 小結

本研究以 GTTRP 之解題模組對於 Chao 之 21 題 TTRP 例題進行測試，雖然該些模組是以減少車輛使用為優先執行，於求解過程可能會因車輛減少，提高路線成本，但是就測試結果發現，有 5 題結果比 Chao 佳，其與 Chao 之平均誤差率只有 4.18%。平均執行時間比 Chao 快了 4000 倍以上，進一步表示本研究可提供有效求解 GTTRP 之模式工具。

第八章 結論與建議

8.1 結論

GTTRP 為一新型之 VRRP 問題，主要是延續 Gerdessen[10]與 Chao[6]等人對於 TTRP 所做之研究，並進一步對其限制條件做適當調整與成本構建合理化而成的。本研究在求解 GTTRP 問題時，初步嘗試以 ILOG Dispatcher 限制規劃軟體進行起始解構建，在改善方面，則以卓裕仁[33]所提出之 GIDS 巨集啟發式解法進行解的改善。其中更對包容性搜尋模組中的 M1 與 M2 模組執行順序進行測試，測試結果亦證明 GIDS 並不只侷限於 M1 模組優先執行，亦可以 M2 模組優先執行，其結果可能比以 M1 模組優先執行來的佳。然而，雖然 GTTRP 問題並不像傳統之 VRRP 問題(如：VRP、VRPTW 等)具有國際標竿題庫以及文獻已知最佳解供測試比較，以瞭解本研究所發展之演算法架構之解題績效，但從測試過程中，可發現 GIDS 對於 GTTRP 問題在每一階段測試過程中皆有逐漸改善之效果，說明各模組及其組合方法確實具有改善效果。並且經由限制條件放鬆將 GTTRP 問題恢復成 TTRP 問題時，其測試結果與 Chao 之最佳結果比較，平均誤差只有 4.18%，而且更有 5 題例題結果比 Chao 來的佳，說明本研究所設計之 GIDS 模組有不錯的解題績效，亦呼應了卓裕仁對於 GIDS 適用於各種 VRRP 問題之說法，說明 GIDS 之應用潛力。

就整體而言，本研究之具體貢獻可歸納出下列幾點：

1. 本研究延續了 TTRP 問題，進一步發展成更接近現實物流狀況之 GTTRP 問題，並發展出 GTTRP 問題之測試題庫以供後續研究參考比較。
2. 由各階段測試過程中，發現 GIDS 對於 GTTRP 問題具有不錯的解題績效，且經由求解 TTRP 問題所得之結果，亦有 5 題結果比 Chao 來得好，進一步驗證了本研究提供了有效求解 GTTRP 的模式工具。

8.2 建議

針對未來 GTTRP 問題之研究，本研究提出下列幾點建議：

1. 未來可考慮把時間窗限制或週期性使用限制納入 GTTRP 問題；
2. 由於本研究只初步應用限制規劃進行起始解構建，且並未深入進行探討，故建議未來可針對限制規劃方法進行求解。然而本研究在使用 ILOG Dispatcher 限制規劃軟體求 GTTRP 之起始解時，發現該軟體所提供之啟發式解法有限，在求解 VRRP 相關問題之改善績效有限，且由於該軟體所提供之解法多已模組化，並不能視問題特性加以調整，故在程式撰寫上較缺乏彈性，建議未來可以 ILOG Solver 進程式撰寫，雖比 Dispatcher 稍為複雜，但是卻更有彈性，在解題方面應可獲得較大之改善空間。
3. 由於 GIDS 模組是由 M1、M2 以及 CP 模組所構成，且本研究所測試模組之執行順序可說是只有基本組合部分，未來可朝多樣化的組合進行測試，例如，M1+FF+M2 執行順序。
4. 從 GTTRP 之測試結果可發現 FF[G1]對於規模較大的問題有較佳的解題績效，未來可朝該模組進一步深入探討。
5. 由於時間因素，本研究在起始解方面並沒有考慮以多重起始點進行測試，因此可能在改善過程中會陷入所謂渾沌吸子，因此建議未來可以多重起點方式進行改善。

參考文獻

1. Bodin, L., B.L. Golden, A. Assad & M. Ball (1983), "Routing and Schedule of Vehicle and Crew: The State of Art", *Special Issue of Computers and Operations Research*, Vol.10, No.2, pp.63-211
2. Battiti, R. & G. Tecchiolli (1994), "The Reactive Tabu Search," *ORSA Journal on Computation*, Vol.6, pp.126-140.
3. Brailsford, S.C., Potts, C. N. and Smith, B. M. (1999), "Constraint Satisfaction Problem: Algorithms and Applications," *European Journal of Operation Research*, Vol.119, pp.557-581.
4. Clarke, G. & J. Wright (1964), "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, Vol.12, pp.568-589.
5. Christofides, N., A. Mingozzi, P. Toth (1979), *Combinatorial Optimization*, John Willey & Sons, Inc..
6. Chao, I. M. (2002), "A Tabu Search Method for the Truck and Trailer Routing Problem," *Computers and Operations Research*, Vol.29, pp.31-51.
7. Dueck, G. & T. Scheuer (1990), "Threshold Accepting: a General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," *Journal of Computational Physics*, Vol.90, pp.161-175.
8. Dueck, G. (1993), "New Optimization Heuristics: the Great Deluge Algorithm and the Record-to-Record Travel," *Journal of Computational Physics*, Vol.104, pp.86-92.
9. Glover, F. & C. McMillan (1986), "The General Employee Scheduling Problem : An Integration of MS and AI," *Computers and Operations Research*, Vol.13, pp.563-573.
10. Gerdessen, J.C. (1996), "Vehicle Routing Problem with Trailers," *European Journal of Operation Research*, Vol.93, pp.135-147.
11. Holland, J.H. (1975), *Adaptation in Natural & Artificial System*, University of Michigan Press, Ann Arbor.
12. Han, A.F. & Y.J. Cho (2001), "A GIDS Metaheuristic Approach to the Fleet Size and Mix Vehicle Routing Problem," Celso Ribeiro & Pierre Hansen (eds.), *Essays and Survey in Metaheuristics*, Chapter 18, pp.399-414, Kluwer

13. *ILOG Dispatcher 3.0*, 2000.
14. Lin, S. & B.W. Kernighan (1973), "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, Vol.21, pp.498-516.
15. Lin, S. (1965), "Computer Solutions of the Traveling Salesman Problem," *The Bell System Technical Journal*, Dec., pp.2245-2269.
16. Kirkpatrick, S., C.D. Gelatt & M.P. Vecchi (1983), "Optimization by Simulated Annealing", *Science*, Vol.220, pp.671-680.
17. Kohl, N., J. Desrosiers, B. G. Madsen, M. M. Solomon & F. Soumis (1999), "2-Path Cuts for the Vehicle Routing Problem with Time Windows," *Transportation Science*, Vol.33., pp.101-116.
18. Or, I. (1976), "Traveling Salesman-type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking," Ph.D. Dissertation, Dept. of Industrial Engineering and Management Sciences, Northwestern University.
19. Osman, I.H. (1993), "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem," *Annals of Operations Research*, Vol. 41, pp.421-451.
20. Rosenkrantz, D., R. Sterns, & P. Lewis (1977), "An Analysis of Several Heuristics for the Traveling Salesman Problem," *SIAM Journal of Computing*, Vol.6, pp.563-581.
21. Russell, R.A. (1995), "Hybrid Heuristics for the Vehicle Routing Problem with Time Windows," *Transportation Science*, Vol. 29, No. 2, pp.156-166.
22. Salhi, S. and Rand, G.K., "Incorporating Vehicle Routing into the Vehicle Fleet Composition Problem," *European Journal of Operational Research*, Vol. 66, pp. 313-330.
23. Small, K.A. (1992), *Urban Transportation Economics*, Harwood Academic Publishers.
24. Solomon, M.M. (1983), "Vehicle routing and Scheduling with Time Window Constraints: Models and Algorithms," Ph.D. Dissertation, Dept. of Decision Sciences, University of Pennsylvania.
25. Solomon, M.M. (1987), "Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Conststraints," *Operations Research*, Vol.33, pp.254-265.

26. Tan, K. C., L. H. Lee, Q. L. Zhu & K. Ou (2001), "Heuristic Methods for Vehicle Routing Problem with Time Windows," *Artificial Intelligence in Engineering*, vol. 15, pp 281-295.
27. 曾國雄(1985)，車輛乘載與耗油特性及其策略之研究，經濟部能源委員會。
28. 張晉芬(1998)，「職場中的性別歧視與就業歧視之認定」，<<第三屆全國婦女國是會議論文集>>，台北市政府。
29. 林修竹(1999)，(指導教授：韓復華)，「包容性啟發式解法在 VRPTW 問題上之應用」，交通大學運輸工程與管理學系碩士論文。
30. 陳國清(1996)，(指導教授：韓復華)，「成本擾動法 (NM) 與兩極跳躍法 (FF) 在 TSP 問題應用之研究」，交通大學運輸工程與管理學系畢業專題報告。
31. 中華民國物流協會(2000)，<<2000 年中華民國物流年鑑>>。
32. 陳建緯(2001)，(指導教授：韓復華)，「大規模旅行推銷員問題之研究：鄰域搜尋法與巨集啟發式解法之應用」，交通大學運輸工程與管理學系碩士論文。
33. 卓裕仁(2001)，(指導教授：韓復華)，「以巨集啟發式解法求解多車種與週期性車輛路線問題之研究」，交通大學運輸工程與管理學系博士論文。
34. 陳俊豪(2002)，(指導教授：葉名山)，「即時資訊下軸輻路網效率評估」，逢甲大學交通工程與管理學系碩士論文。