

行政院國家科學委員會專題研究計畫 成果報告

一般化卡車拖車路線問題(TTRP)之研究

計畫類別：個別型計畫

計畫編號：NSC91-2211-E-009-043-

執行期間：91年08月01日至92年07月31日

執行單位：國立交通大學運輸科技與管理學系

計畫主持人：韓復華

共同主持人：卓裕仁

計畫參與人員：吳志仁

報告類型：精簡報告

處理方式：本計畫涉及專利或其他智慧財產權，2年後可公開查詢

中 華 民 國 92 年 8 月 6 日

行政院國家科學委員會專題研究計畫成果報告

一般化卡車拖車路線問題(TTRP)之研究

計畫編號：91-2211-E-009-043

執行期限：91年8月1日至92年7月31日

計畫主持人：韓復華 國立交通大學運輸科技與管理學系 教授

共同主持人：卓裕仁 中華大學交通與管理學系 助理教授

計畫參與人員：吳志仁 國立交通大學運輸科技與管理學系 碩士

中文摘要

卡車與拖車路線問題(Truck and Trailer Routing Problem, TTRP)是傳統車輛路線問題(Vehicle Routing Problem, VRP)的一種衍生型態，藉由卡車或整車(卡車附掛拖車)之路線服務具有不同可及性的顧客。本研究依 TTRP 問題特性，從合理化構建成本與擴充限制條件兩方面著手，使其成為「一般化卡車拖車路線問題」(Generalized TTRP, GTTRP)，更能符合實際應用的需求。GTTRP 考慮的物流配送較 VRP 與 TTRP 更具實用彈性，但另一方面問題的複雜性也更高。

本研究使用「限制規劃」(Constraint Programming, CP)與「包容性深廣度搜尋法」(Generic Intensification and Diversification Search, GIDS)，發展適合 GTTRP 的求解工具。CP 應用於起始解構建部分。GIDS 巨集啟發式方法則搭配門檻接受法(Threshold Accepting, TA)與大洪水法(Great Deluge Algorithm)兩種搜尋法，以及兩極跳躍法(Flip Flop, FF)構建多組演算模組進行路線改善測試。本研究依據 Chao 的 TTRP 題庫，建立一組 18 題的 GTTRP 測試題庫，並以 C⁺⁺ 語言撰寫程式在 AMD PC 上執行測試。結果發現，本研究所設計求解 GTTRP 之模組皆有不錯之解題績效。本研究亦以 GTTRP 之模式求解 TTRP 問題，亦發現有 5 題結果較 Chao 佳，整體平均誤差為 4.18%，平均執行時間為 15.08 秒，比 Chao 60747.57 秒快了 4000 倍以上，此亦印證本研究 GTTRP 求解方法的效果。

關鍵字：車輛路線問題，卡車與拖車路線問題，包容性深廣度搜尋，限制規劃

Abstract

The Truck and Trailer Routing Problem (TTRP) is a variant of traditional Vehicle Routing Problem (VRP). The problem considers the use of trucks as well as the combination of truck-and-

trailers to service customers. TTRP is relative new in literature, e.g., Gerdessen (1996) and Chao (2002). This research extends the TTRP to include more generalized considerations, and defines the generalized Truck and Trailer Problem (GTTRP). As

contrast to TTRP, which ignores the costs associated with the decoupling / coupling of the truck and the trailer as well as that of additional parking lots to park the trailers, our proposed GTTRP includes such costs in general.

We applied Constraint Programming (CP) and the meta-heuristic method of Generalized Intensification and Diversification Search (GIDS) to develop heuristic solution algorithms to solve GTTRP. The initial solution of GTTRP is generated by CP. The GIDS integrates the use of Threshold Accept (TA) and Great Deluge Algorithm (GDA) and Flip Flop (FF) to improve the incumbent solution. For numerical-analysis purposes, we developed a bank of 18 test problems of GTTRP based on Chao's test problems of TTRP. We wrote computer programs in C++ and implemented our models on an AMD personal computer. Results found that our proposed models can provide useful tools for GTTRP applications.

Keywords: Vehicle Routing Problem (VRP), Truck and Trailer Routing Problem (TTRP), Generalized Intensification and Diversification Search (GIDS), Constraint Programming (CP)

一、前言

據中華民國物流協會統計[17]，我國的物流成本仍有很大改善空間。在物流成本中，運送成本佔其比例約為 25%~45%，因此若能有效改善運輸配送成本，便能降低物流成本，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。由於台灣的生活環境，住宅密集、街道狹小，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進出困難，造成配送困難。若以小車進行配送雖進出方便，但卻因裝載容量小，需經常往返物流中心補貨。為了克服街道狹窄，且在大量運輸的考量下，本研究延續了 Gerdessen[6]所提出的卡車-拖車配送組合的概念，並調整其限制條件，以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，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。整車服務至主路線的停放點(Parking Place)時，可將拖車留於該處僅用卡車服務若干不便到達的顧客(子路線顧客)；待子路線顧客服務完畢後再回到拖車停放點與拖車相連接，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。

二、研究內容與目的

近年來 TTRP 問題已逐漸受到學者關注，表 1.1 為近年研究 TTRP 相關文獻之整理表。

表 1.1 TTRP 文獻回顧

年代	作者	假設條件	解法
1996	Gerdessen	<ul style="list-style-type: none"> ◆ 每個顧客僅有單位需求量 ◆ 每一整車路線中拖車僅能停放一次 ◆ 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	<ul style="list-style-type: none"> ◆ 將顧客區分為整車服務顧客與卡車服務顧客 ◆ 不限制拖車停放次數 ◆ 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

過去研究中對 TTRP 問題著墨不多，且為簡化問題提出之限制假設忽略現實狀況，降低實務可行性，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；(3)未考慮拖車卸下與連結之時間成本。

本研究之目的在於放寬上述假設與限制，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。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，
- (2)考慮拖車卸下及連結之時間成本。
- (3)考慮車輛以整車型態或卡車型態配送之變動成本。
- (4)考慮車輛使用成本。

2. 限制條件

- (1)考慮整車卸下、連結拖車之地點限制。
- (2)將顧客區分為整車顧客與卡車顧客。

三、文獻探討

3.1 車輛路線問題定義與解法回顧

3.1.1 傳統啟發式解法

由於 VRP 屬於 NP-hard 問題，因此最佳解解法往往無法於有效時間內求出最佳解；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，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[18]：

- (1) 路線構建型(Tour Construction)：根據路網距離或成本矩陣產生較佳的可行解，常見的方法如，插入法[11] (Insertion Method)、節省法[1] (Savings Method) 等。
- (2) 路線改善型(Tour Improvement)：係針對任意一個起始可行解，以鄰域搜尋法 (Local Search)改善路線成本，求得更好的解，例如，K-Opt[9]、Or-Opt[10]。
- (3) 綜合型(Composite / Hybrid)：係將路線構建和路線改善合併執行，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」。

3.1.2 巨集啟發式解法

鑑於傳統鄰域搜尋法，容易陷入「局部最佳解」(Local Optimum)的困境，為克服此缺點，發展出以策略性運用啟發解法去搜尋超越傳統啟發式解法僅能找到的局部最佳解之「巨集啟發式解法」(Meta-Heuristic)。其中著名之巨集啟發式解法如表 3.1 所示：

表 3.1 巨集啟發式解法回顧

作者	巨集搜尋策略	解題基本概念
Glover et al.(1986)	禁制搜尋法 (TS)	TS 法在現有解的鄰域進行搜尋，並應用記憶機制，將搜尋過的解記錄在禁制列，以避免重複性搜尋；待整個鄰域都搜尋完畢後，再選擇一個最佳的方向進行移動，以逐漸逼近最佳解。
Kirkpatrick et al.(1983)	模擬退火法 (SA)	SA、TA 與 GDA 皆屬於門檻型演算法，其基本觀念在於鄰域搜尋陷入局部最佳解時，採取放鬆的接受法則（通常為一門檻值）接受劣於現解之鄰解，以便脫離局部最佳解的束縛而繼續搜尋下去。
Dueck & Scheuer(1990)	門檻接受法 (TA)	
Dueck (1990)	大洪水法 (GDA)	
韓復華、卓裕仁	包容性深廣度搜尋法(GIDS)	運用接受劣解、變換臨域、擾動成本與多重起點等巨集策略，執行深、廣度搜尋，克服可能面臨之局部最佳解、限制迴圈及渾沌吸子等障礙。

3.1.3 卡車拖車路線問題(TTRP)回顧

Gerdessen[6]於1996對於VRPT提出了三個構建路線與兩種類別路線改善的啟發式解法。在構建路線啟發式解法方面，以節省插入法為主，並以啟發式解法I為基礎，針對解法I過早決定車輛服務哪些顧客缺乏改善空間的缺點進行修正，進而提出解法II與III提高了改善空間。而在路線改善啟發式解法方面，以路線間的改善與路線內的改善為方向，以k-opt交換法為主。最後，設計了U、W、S三個類別的問題進行測試，以啟發式解法III的解題效果最好，並發現解題績效與問題規模大小有關，與問題類別無關。

Chao[3]於2002年提出TTRP問題，然而所提出的TTRP問題除了延續Gerdessen對VRPT問題的種種假設外，更放寬或增添假設條件。如：(1)顧客分成整車、卡車顧客；(2)拖車可停放一次以上；(3)卡車顧客點不可停放拖車；(4)顧客非單位需求。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。並發展出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。

最後本研究將Gerdessen和Chao對於TTRP問題假設條件與本研究GTTRP之假設條件做一比較整理，其整理如表3.2所示：

表3.2 TTRP與GTTRP之差異

	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

3.1.4 限制規劃方法

「限制規劃(Constraint Programming, CP)」是求解「限制滿足問題(Constraint Satisfaction Problem, CSP)」的一種途徑。所謂 CSP 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。CP 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[1]對於 CP 的定義：CP 是以電腦程式發展模式化語言，讓使用者能簡單、輕易地描述(State)並透過智慧型的空間搜尋演算法，系統地、效率地求解 CSP。CP 之優點有：模式構建的方便性、可處理多樣之限制式、有效率之求解機制以及「模式建構(Modeling)」與「模式求解(Solving)」互相獨立性。

四、研究方法

GTTRP 較 TTRP 考慮更多實務的成本與限制，故較能符合現實之應用，但相對地問題求解亦變得較為複雜。基本上 GTTRP 亦符合 CSP 之特質。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 GIDS 進行起始解的改善，以期提升整體演算法之解題績效。其整個研究之求解架構如圖 4.1 所示：

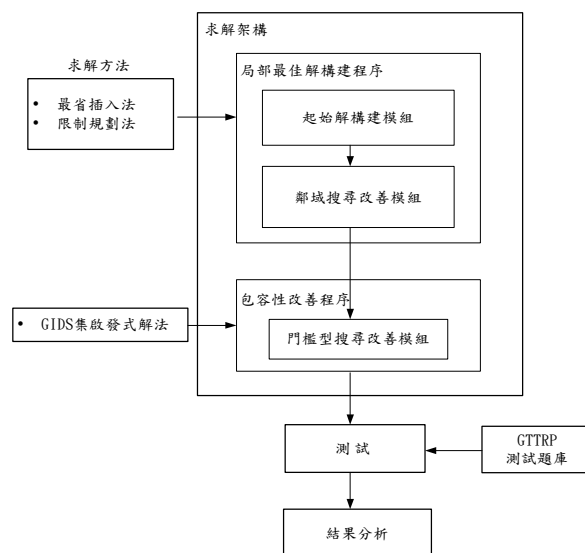


圖 4.1 GTTRP 求解架構

4.1 局部最佳解構建程序

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，以求得局部最佳解。

4.1.1 起始解模組

起始路線構建以兩階段方式進行，先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。且為路線構建方便，構建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。

(1) 整車路線構建

以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。由於構建整車路線時剩餘可裝載之容量會影響到子路線可服務卡車顧客數之多寡；亦即若整車剩餘可裝載量少時，相對地卡車裝載量亦減少，行駛於子路線可服務顧客數相形減少。同理，若整車剩餘可裝載容量過多時，而實際需服務之卡車顧客數過少時，則會留下過多的裝載空間，形成資源浪費。因此本研究以整車顧客之需求量(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(\bar{Q}_1)，以期使能有彈性地構建路線解。其公式如式 4-1 所示：

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

(2) 子/卡車路線構建

子路線構建，從已構建完成之整車路線中選擇拖車停放成本最小的顧客點作為拖車停放點；第二階段亦採最省插入法構建子路線，其種子點選擇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。

(3) ILOG Dispatcher 限制規劃軟體應用

本研究以 ILOG Dispatcher 限制規劃軟體[8]進行 GTTRP 問題起始解求解。Dispatcher 軟體對於 VRP 問題在模式構建上主要是對場站、車輛以及顧客三方面進行限制條件宣告，然而該軟體對場站、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，主要以下列三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。

4.1.2 鄰域搜尋改善模組

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。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；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。本研究所使用的交換法如表 4.1 所示，各交換法之細節請參閱吳志仁[21]。

表 4.1 本研究使用交換法之彙整表

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"> ◆ 停放點 1-1 節點交換

4.2 包容性改善程序

本節以最小化(Minimize)問題為例，介紹本研究使用的包容性(generic)演算法。

4.2.1 門檻型演算法(Threshold Accepting, TA)

TA 法[4]主要是先產生一組固定門檻數列，依次使用數列中的門檻值，藉著接受法則： $C(X^{new}) < C(X^{current}) + T_k$ ，接受暫劣解脫離局部最佳解之束縛。其門檻型數列型態因尚無法證明何種型態最好，本研究以梯狀型門檻數列為主，其起使門檻值為： $T_1 = \overline{C(X_0)} \cdot T_0 / N$ 。其中 $\overline{C(X_0)}$ 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。此外亦考慮到問題規模(N)，其原因為在相同的空間範圍

內， N 與 $\overline{C(X_0)}$ 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。

4.2.2 大洪水法(Great Deluge Algorithm, GDA)

GDA 法[5]是設定一起使水位(WL_0)，只要有改善就降低水位，接受法則為 $C(X^{new}) < WL_i$ ，停止法則為 $C(X^{new}) \geq WL_i$ 時停止。其起始水位、下降速度(D_i)如式(4-2)、(4-3)以及式(4-4)所示：

$$WL_0 = C(X_0) \cdot (1 + T_0 / N) \dots\dots\dots(4-2)$$

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

$$D_1 = C(X_0) \cdot 10 \cdot (T_0 / N) \dots\dots\dots(4-4)$$

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，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。

4.2.3 GIDS 模組

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 4.2 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 Flip Flop[16]進行成本擾動。

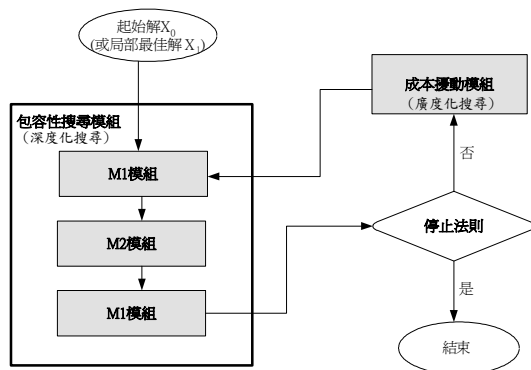
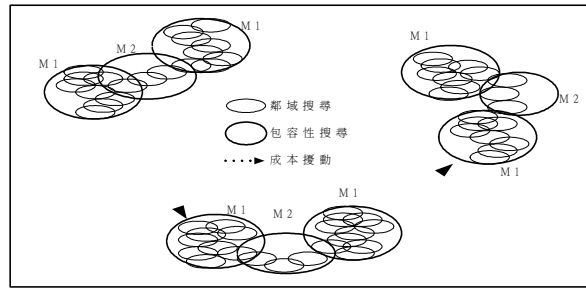


圖 4.2 GTTRP_GIDS 執行架構

在解集合的空間中 GIDS 的搜尋空間軌跡如圖 4.3 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，以其獲得較佳解。



資料來源[7]

圖 4.3 GTTRP_GIDS 搜尋概念圖

五、結果與討論

5.1 GTTRP 測試例題建立

本研究以 Chao 之 TTRP 題庫[3]為基礎，建立一組共 18 題的 GTTRP 測試題庫，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。其 18 題 GTTRP 測試題庫定義如表 5.1 所示。

表 5.1 GTTRP 測試題庫定義

GTTRP 測試例題								Chao[3] 測試例題
題號	規模	整車顧客	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

成本構建合理化方面，根據曾國雄[13]對車輛耗油特性分析，小貨車每行駛一公里需花費 2 元的燃油費，整車方面則為 2.67 元/公里。拖車操作之時間價值，依 Small[12]對於工作旅次時間價值分析為工資率的 50%，運輸倉儲業工資為 236

元/每小時[14]，故工作旅行時間價值為 168 元/小時。車輛使用成本，據陳俊豪[20]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，因使用拖車不需額外增加人員，故其使用成本以折舊費為主，依平均折舊法進行計算，其折舊費為 107 元/天，因此整車之使用成本為 1557 元/天。

5.2 起始解與局部最佳解結果分析

本研究以 C⁺⁺ 撰寫電腦程式，在 AMD 1.6GHz CPU 與 1GHz RAM 之 Microsoft Windows 2000 作業系統下的 PC 進行求解。以 ILOG Dispatcher 產生起始解如表 5.2 所示，再以表 4.1 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 5.3 所示：

表 5.2 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 5.3 GTTRP 局部最佳解

GTTRP 題號	目標成本	改善率 (%)	CPU (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 5.3 可看出，以傳統鄰域改善法進行起始解改善，平均執行時間只有 0.88 秒，改善率從 2.51~17.23%，平均改善率為 6.45%。

5.3 GIDS 模組測試分析

由於模組之控制參數和組合方式等因素會影響演算法解題績效，因此本研究針對這兩項因素進行測試。由於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，因此要先找出 TA 與 GDA 法應用於第一階段合適的參數範圍。接著執行 M1 與 M2 模組搭配組合測試，使其為第二階段；最後將第二階段測試進行「成本擾動」，並適當重覆整個程序，使其為第三階段測試。為方便組合模組撰寫，以 M1 模組執行的 GDA 與 TA 法表示成 G1 與 T1；同理，以 M2 模組執行的 GDA 與 TA 法以 G2 與 T2 表示。

(一) 第一階段測試

第一階段為測試 TA 與 GDA 法應用於模組 M1 與 M2 時之合適參數範圍。在參數 $k=30$ 的條件下，TA 與 GDA 法之 T_0 參數測試範圍以 0.05~0.29 進行測試，發現 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。再各取十個參數進行測試，其結果發現 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 14.34 秒；G1 組合平均改善率 8.75% 雖略遜 T1，但其平均執行時間卻比 T1 少的許多。以執行之時間效率來考量，G1 應為四組比較中表現最好的。四組測試平均改善率皆有 8% 以上比傳統鄰域改善法 6.45% 來的佳。

(二) 第二階段測試

本階段採用 T1+G2+T1、G1+T2+G1、T2+G1+T2 與 G2+T1+G2 四種組合進行測試，為測試各組合之穩定性，從第一階段各組合之測試範圍挑選出三個 T₀ 值，挑選依據是將參數範圍以三等分的方式從中各選出一個 T₀ 值。其各模組參數分別為：T1_T₀=0.25、0.3、0.33；G1_T₀=0.02、0.05、0.09；T2_T₀=0.15、0.2、0.24 以及 G2_T₀=0.12、0.16、0.2。將各組第二個參數與其搭配模組之三個參數進行搭配測試，其測試結果如表 5.4 至 5.7 所示：

表 5.4 T1+G2+T1 組合測試結果

模組組合		T1+G2+T1		
T1_To 值	G2_To 值	成本	改善率(%)	CPU(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 5.5 G1+T2+G1 組合測試結果

模組組合		G1+T2+G1		
G1_To 值	T2_To 值	成本	改善率(%)	CPU(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 5.6 T2+G1+T2 組合測試結果

模組組合		T2+G1+T2		
T2_To 值	G1_To 值	成本	改善率(%)	CPU(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 5.7 G2+T1+G2 組合測試結果

模組組合		G2+T1+G2		
G2_To 值	T1_To 值	成本	改善率(%)	CPU(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

每一種組合在不同 T_0 值搭配下，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，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

第三階段測試是承襲第二階段測試後，加上「成本擾動」執行廣度化搜尋，並適當重覆整個程序。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 之平均執行時間較低(約 13 秒)，因此設定重覆擾動以十次為主；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。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 各組合之測試結果如表 5.8 所示。

表 5.8 第三階段 FF 擾動測試結果

模組組合	(m,n)	平均成本	平均改善率 (%)	平均 CPU (秒)
T1+G2+T1	(2,5)	19346.83	11.06	135.12
G1+T2+G1	(2,10)	19476.19	10.27	64.73
T2+G1+T2	(2,10)	19258.58	11.33	80.38
G2+T1+G2	(2,5)	19556.19	9.87	112.99

表 5.8 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，於執行完 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

GTTRP 題號	規模	目標成本	CPU (秒)	FF[模組組合]	控制參數	
					(GDA_ T_0 , TA_ T_0)	(m, n)
1	50	8137.88	67.306	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			

現最佳為選擇依據 $G1_T_0$ 值為 0.02、 $T1_T_0$ 值為 0.3。其測試結果如表 5.9 所示：

表 5.9 FF[G1]、FF[T1]執行結果

(m,n)		(2,5)		(4,5)		(8,5)	
模組組合	T_0 值	平均成本	平均 CPU (秒)	平均成本	平均 CPU (秒)	平均成本	平均 CPU (秒)
G1	$T_0=0.02$	19482.62	14.18	19478.42	21.60	19543.00	13.48
T1	$T_0=0.3$	19424.53	122.64	19439.62	105.79	19443.96	104.45

為更清楚觀察出經過每次擾動後其成本改善趨勢，將以上測試於每次擾動後改善結果記錄起來，繪出各方法的成本改善趨勢如圖 5.1 所示，以比較其求解效率。

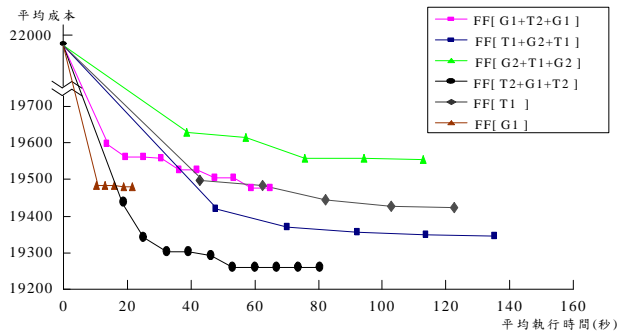


圖 5.1 各方法求解效率比較圖

從圖 5.1 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，能跳脫原本之搜尋區域，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。

5.4 測試結果

表 5.10 為測試過程中 GTTRP 最佳結果整理表。從表中可觀察出，每一題大約可在 2 分鐘左右求得結果，在方法上以 FF[T2+G1+T2]組合較佳。值得一提的是規模為 199 點的問題以 FF[G1]所求的解最佳，執行時間大約為 30 秒，極具潛力，後續研究大規模 GTTRP 問題時值得進一步探討。

表 5.10 18 題 GTTRP 測試最佳結果整理表

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。本研究將以 5.3 節中第一階段測試結果於扣除車輛使用成本下表現最好的 G1 與次佳 T2 組合進行搭配測試，測試所得之結果如表 5.11。表中可發現本研究與 Chao[3]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，進一步表示本研究可提供有效求解 GTTRP 之模式工具。

表 5.11 本研究 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	
	車輛 (V,T)*	目標 成本	CPU (秒)	車輛 (V,T)*	目標 成本	誤 差 (%)	CPU (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		1012.46	60747.57		1057.41	4.18	15.08			

* V：整車數；T：卡車數

5.6 結論與建議

整體而言，本研究延續 TTRP 問題，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。在求解模式方面，本研究所設計求解 GTTRP 之模組皆有不錯之解題績效。

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 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。

3. 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。

參考文獻

1. 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.
2. 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.
3. Chao, I. M. (2002), "A Tabu Search Method for the Truck and Trailer Routing Problem," *Computers and Operations Research*, Vol.29, pp.31-51.
4. 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.

5. 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.
6. Gerdessen, J.C. (1996), "Vehicle Routing Problem with Trailers," *European Journal of Operation Research*, Vol.93, pp.135-147.
7. 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.
8. *ILOG Dispatcher 3.0*, 2000.
9. Lin, S. (1965), "Computer Solutions of the Traveling Salesman Problem," *The Bell System Technical Journal*, Dec., pp.2245-2269.
10. 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.
11. 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.
12. Small, K.A. (1992), *Urban Transportation Economics*, Harwood Academic Publishers.
13. 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。
14. 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。
15. 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。
16. 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。
17. 中華民國物流協會(2000)，<<2000 年中華民國物流年鑑>>。
18. 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。
19. 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。
20. 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。
21. 吳志仁(2003)，(指導教授：韓復華)，「一般化卡車拖車路線問題」，交通大學運輸科技與管理學系碩士論文。

