

兩階段回溯式門檻接受法求解時窗限制 回程取貨車輛路線問題之研究

A TWO-PHASE BACKTRACKING THRESHOLD ACCEPTING META-HEURISTIC FOR THE VEHICLE ROUTING PROBLEM WITH BACKHAULS AND TIME WINDOWS

卓裕仁 Yuh-Jen Cho¹

朱佑旌 You-Jing Chu²

(96 年 11 月 30 日收稿，97 年 4 月 24 日第一次修改，97 年 7 月 23 日
第二次修改，97 年 11 月 28 日定稿)

摘 要

時窗限制回程取貨車輛路線問題 (vehicle routing problem with backhaul and time windows, VRPBTW) 是車輛路線問題 (vehicle routing problem, VRP) 的延伸，屬於解題複雜度很高的 NP-hard 問題，其在物流配送實務上具有很高的應用價值。本研究結合門檻接受法 (threshold accepting, TA) 與傳統啟發式方法，設計一套可求解 VRPBTW 之兩階段回溯式門檻接受法 (two-phase backtracking threshold accepting, TBTA)，並採用 15 題國際標準例題進行測試，以分析此 TBTA 巨集啟發式解法之解題績效。測試結果發現：車輛數之平均誤差百分比為 3.52%，行駛時間之平均誤差百分比為 2.81%，CPU 之平均運算時間為 33.69 秒；此結果顯示 TBTA 法在求解 VRPBTW 上具有應用潛力。

-
1. 中華大學運輸科技與物流管理學系助理教授 (聯絡地址：30012 新竹市五福路 2 段 707 號 中華大學運輸科技與物流管理學系；電話：03-5186597；E-mail：yjcho@chu.edu.tw)。
 2. 中華大學科技管理研究所運輸科技與物流管理組碩士。

關鍵詞：時窗限制回程取貨車輛路線問題；巨集啟發式解法；兩階段回溯式門檻接受法

ABSTRACT

The vehicle routing problem with backhauls and time windows (VRPBTW), which simultaneously considers the operations of delivery and pickup, is a variant of the classical vehicle routing problem (VRP). Successful application of the VRPBTW in real-world distribution will improve the performance of logistics. Due to the NP-hard complexity of VRPBTW, the most solution methods are heuristics and meta-heuristics. This paper aims to develop a two-phase backtracking threshold accepting (TBTA) meta-heuristic approach, which combines the threshold accepting (TA) meta-strategy and the traditional local search algorithms, to solve the VRPBTW. We adopted fifteen VRPBTW benchmark instances to test and analyze the performance of the proposed TBTA meta-heuristic. The average error percentages of TBTA are 3.52% for the objective of fleet size and 2.81% for the objective of traveling time. Moreover, the average value of CPU running time is merely 33.69 seconds. Computational results implied that the TBTA actually provides an efficient tool for VRPBTW applications.

Key Words: *Vehicle routing problem with backhauls and time windows (VRPBTW); Meta-heuristics; Two-phase backtracking threshold accepting (TBTA)*

一、前言

在注重顧客服務與作業績效的企業環境下，物流與貨運業者必須有效地利用其配送資源來提升服務品質與效率，才能創造出企業的競爭優勢。在顧客服務方面，顧客通常會希望能夠指定商品送達的時間，以配合其時程；對配送業者而言，此項要求雖然會增加車輛排程的困難度，但是卻能提高送達率、減少發車趟次。另一方面，在配送過程結束後，若能繼續收取顧客交運的物件，將可增加車輛的使用率，減少車輛往返時間。

Yano 等人^[1]的研究指出，美國一家高級百貨業者，每天必須把貨品由物流中心送至各銷售點，然後再從鄰近的供應商將貨品取回至物流中心。由於缺乏一套有效的路線規劃方法，造成時間的浪費和車輛利用率不佳的問題。Yano 等人^[1]將該百貨業者的問題轉換成集合涵蓋問題 (set-covering problem)，並結合路線產生法 (route generation procedure) 與分枝定限法 (branch and bound) 求解出最佳路線，結果不但增加車輛回程的利用率，平均每年更為業者節省 45 萬美元。這種利用回程時取貨的物流配送作業型態也常發生在食品雜貨業上；對食品雜貨業來說，超級市場和零售店即屬於送貨需求點，而食品雜貨的供應商則為取貨需求點。Casco 等人^[2]的研究指出，美國自 1982 年以來，食品雜貨業由於有效利用回程空車而增加車輛使用率，估計每年在運輸成本上可節省一億六千萬美元。以國內

某汽車路線貨運公司營業所之作業流程為例，在配送與取件兩項作業中，外務員（兼司機）於上午將到站貨物配送給客戶後，即直接原車（空車）在外至其他客戶處收取交運貨物，然後再回到營業所將貨物卸下與分類^[3]。此外，宅配業者也有類似的做法，不僅採用回程取貨的作業方式，更加入時間窗的服務因素考量^[4]。

在學術研究方面，Deif 與 Bodin^[5]以及 Casco 等人^[2]，即將上述回程取貨型態的物流配送作業模式轉換成「回程取貨車輛路線問題 (vehicle routing problem with backhauls, VRPB)」來求解，以提升實務業者的作業效率、降低營運成本。Gélinas 等人^[6]更加入了時間窗限制，而衍生成為「時窗限制回程取貨車輛路線問題 (vehicle routing problem with backhauls and time windows, VRPBW)」^[7]。近年來，高複雜度的車輛路線相關問題研究已朝向巨集啟發式方法 (meta-heuristics) 發展，諸如：基因演算法 (genetic algorithm, GA)^[7]、禁制搜尋法 (tabu search, TS)^[8]、螞蟻演算法 (ants colony system, ACS)^[9]等，亦陸續被應用於 VRPBW 之求解上。

根據以往的文獻報導發現「門檻接受法 (threshold accepting, TA)」乃是一套結構簡單、執行容易，且解題績效良好的巨集啟發式方法^[10-12]，應可適合用來發展 VRPBW 的求解方法。因此，本研究有鑒於 VRPBW 在實務應用上的重要性，以及嘗試發展具解題績效的巨集啟發式方法等動機，遂進行 VRPBW 巨集啟發式解法之研究。本研究之目的乃希望以改良式的回溯型門檻接受法為基礎架構，以發展一套求解績效良好且時間合理的 VRPBW 巨集啟發式演算法。

本文後續的章節安排如下：第二節簡要回顧 VRPBW 之定義與文獻已知的求解方法，以及門檻接受法的相關發展；第三節說明本研究提出之兩階段回溯式門檻接受法 (two-phase backtracking threshold accepting, TBTA) 應用於 VRPBW 之解法架構與模組設計；第四節介紹標竿測試例題與相關參數設定內容，並彙整分析例題測試的結果；最後於第五節提出結論與建議。

二、文獻回顧

2.1 VRPBW 問題定義與特性

在各種車輛路線的相關問題中，混合的顧客服務型態近年來逐漸受到重視，例如：路線中的顧客型態可能同時包含送貨點顧客 (delivery 或 line-haul) 與取貨點顧客 (pickup 或 backhaul)。屬於此類混合顧客型態的車輛路線問題有：回程取貨車輛路線問題 (VRPB)、同時收送貨車輛路線問題 (pickup and delivery vehicle routing problem, PDVRP)，及撥召公車問題 (dial-a-ride problem, DARP)。

其中，時窗限制回程取貨車輛路線問題 (VRPBW) 同時考慮了時間窗與回程取貨兩項限制，屬於 VRPB 的一種衍生型態，其複雜度更甚於 VRPB。在 VRPBW 的問題中，

其顧客分為兩種型態：送貨點（將貨物從場站運送至顧客）、取貨點（pickup points，自客戶處收取貨物運回場站），每位顧客還會指定被服務的時間區段（亦即時間窗，time windows）。在不違反車容量與時間窗限制之下，車輛從單一場站出發，必須先服務完該車的送貨點顧客後再服務取貨點顧客，然後才返回場站；其目標在使車輛總數極小化（第一目標）及路線總行駛時間極小化（第二目標）。

一般而言，VRPB 或 VRPBTW 並未限定每條路線都必須同時包含有送貨點顧客與取貨點顧客；亦即，各路線也可以只服務送貨點顧客或只服務取貨點顧客。若是某路線同時服務了送貨點顧客與取貨點顧客，則必須遵守先服務送貨點顧客後再服務取貨點顧客的限制。在實務上，這種服務順序的限制可避免司機員在取貨後還要花費時間整理貨車空間的缺點。

此外，時窗限制則可分為「硬時窗」及「軟時窗」兩種。硬時窗指的是車輛必須在顧客要求之時間區間內開始服務該顧客，但允許車輛可在時間窗下界之前到達該顧客點，惟必須等到時間窗下界開始時才可進行服務，這段時間稱為「等待服務時間（waiting time）」。「軟時窗」指的是車輛可以違反時間窗上、下界的限制，但違反時會給予一懲罰成本值。目前文獻上所探討的 VRPBTW，大多假設為硬時窗限制^[6-9, 13-17]。

2.2 VRPBTW 求解方法回顧

由於 VRPBTW 的高複雜度，目前除了 Gélinas 等人^[6]所提出的分枝定限法是屬於可求出精確解（exact solution）的方法外，其他文獻的方法皆為求取近似解（approximate solution）的啟發式或巨集啟發式方法。以下簡要回顧 VRPBTW 的求解方法，並彙整如表 1 所示。

Gélinas 等人^[6]以集合分割問題（set partitioning problem）模式來處理 VRPBTW，提出一個新的分枝策略（branching strategy）來執行分枝定限法，並配合線性鬆弛法（linear relaxation）求解此集合分割問題；Gélinas 等人^[6]亦產生了 45 個 VRPBTW 例題（顧客點數自 25 至 100）以測試其解法的績效。

Potvin 等人^[7]利用基因演算法（GA）來求解 VRPBTW。首先利用貪心插入法（greedy insertion）在不同的參數設定下產生一組起始解來作為起始族群（population），然後根據適應度（fitness）函數值隨機選擇兩個父代解進行複製（reproduction），再以 MX1（multi-point crossover 1）、MX2（multi-point crossover 2）、1X（one-point crossover）等三種交配（crossover）運算元進行基因重組（recombination）後產生兩個子代解，並隨機進行 RAR（remove-and-reinsert）、交換（swap）、LwbF（last-will-be-first）等突變（mutation）運算；重複執行複製、重組與突變等運算直到產生一組相同數量的子代族群後，再從父代與子代族群中篩選出較佳的解形成新的族群；重複上述的世代演化程序，直到無法改善後，停止。在績效測試方面，Potvin 等人^[7]採用 Gélinas 等人^[6]的 VRPBTW 題庫來進行解題績效測試。

Thangiah 等人^[13]提出以區域搜尋法（local search）為基礎的啟發式解法。該解法首先

利用後推式插入法 (push-forward insertion) 構建起始解，再使用 λ -交換法 (λ -interchange) 及 2-opt*交換法 (2-opt* exchange) 來加以改善。在例題測試方面，Thangiah 等人^[13]除採用 Gélinas 等人^[6]的 VRPBTW 例題外，還自行產生了 24 個新的測試例題，其規模為 250 與 500 個顧客點。

Duhamel 等人^[8]將禁制搜尋法 (TS) 應用於 VRPBTW 之求解。Duhamel 等人^[8]採用 Potvin 等人^[7]所提出的貪心插入法來構建起始解，然後隨機選擇 2-opt*、Or-opt 或 swap 其中一種交換法進行鄰域搜尋，並在鄰域搜尋過程中持續更新禁制名單與暫優解，直到達到停止條件 (例如：重複 10000 次) 為止。在績效測試方面，Duhamel 等人^[8]不僅採用 Gélinas 等人^[6]的 VRPBTW 標竿例題，也自行產生了 15 個規模為 100 個顧客點的新測試例題。

申生元^[14]提出一個稱為 RNETS (route-neighborhood-based two-stage) 的兩階段巨集啟發式方法。RNETS 法根據路線鄰域結構 (route neighborhood structure)，第一階段使用巢式平行法來構建起始解，然後於第二階段以區域搜尋法配合擾動 (perturbation) 機制來改善起始解。申生元^[14]以 Gélinas 等人^[6]的 VRPBTW 標竿例題來測試解題績效。

Reimann 等人^[9]使用螞蟻演算法 (ant system, AS) 來求解 VRPBTW。不同於大多數 AS 法在構建路線時採用最近鄰點法 (nearest neighbor) 或節省法 (savings) 為基礎的機制，Reimann 等人^[9]提出一個以插入法 (insertion) 為基礎的路線構建方式，並認為較傳統方式更能適用於 VRPTW 或 VRPBTW 等類型的問題。Reimann 等人^[9]亦採用了 Gélinas 等人^[6]的標竿例題來測試解題績效，並與 Thangiah 等人^[13]的最佳結果進行比較。

Cheung 與 Hang^[18]將航空貨運承攬業者的配送問題視為是 VRPBTW 問題，並提出一個考慮標籤配對條件 (label matching conditions) 的數學規劃模式。此外，Cheung 與 Hang^[18]也根據多重屬性標籤 (multi-attribute label) 的架構，提出兩個最佳化基礎 (optimization-based) 的啟發式方法：同步指派 (simultaneous assignment) 法與循序車輛—需求指派 (sequential vehicle-demand assignment) 法。Cheung 與 Hang^[18]除了以物流業者的三個真實例題 (規模皆為 122 個顧客點，取貨比例分別為 5%、10% 與 30%) 進行解題測試外，還隨機產生了 18 個 VRPBTW 測試例題 (規模為 25、50 及 100 個顧客點，取貨比例為 5%、10% 及 30%，時間窗比例為 25% 及 50%)。

Zhong 等人^[15]提出一個導引式區域搜尋法 (guided local search, GLS) 來求解 VRPBTW。此 GLS 法包含兩個階段：第一階段使用適應性掃描法 (adapted sweep) 產生一起始解，然後以 GLS 的懲罰機制導引 2-opt、1-move、1-exchange 等交換法來改善起始解。由於在第一階段獲得的解不一定是可行解，所以在第二階段應用分段規劃 (section planning) 的技巧，將造成不可行的顧客點插入新增的路線中，以達成解的可行性。Zhong 等人^[15]採用 Gélinas 等人^[6]的 VRPBTW 標竿例題來測試 GLS 的解題績效。

Cho 與 Wang^[16]應用門檻接受法 (TA) 來求解 VRPBTW，並提出一個數學規劃模式來描述 VRPBTW 之問題定義。此 TA 法首先以最近鄰點法 (nearest neighbor) 來構建起始解，

再以 2-opt、1-0、1-1 與 S-S 等交換法進行鄰域搜尋改善，最後加上門檻接受機制重複執行多次的鄰域搜尋以改善解的品質。Cho 與 Wang^[16]以自行產生的 81 個例題來進行解題績效測試，其例題規模為 25、50 及 100 個顧客點，取貨比例為 30%、50%與 70%。

Ropke 與 Pisinger^[17]提出一個稱之為制式啟發式法 (unified heuristic) 的解題架構，可求解多種 VRPB 相關問題。此架構係以大規模鄰域搜尋法 (large neighborhood search, LNS) 為基礎並加入破壞與重生 (ruin and recreate, R&R) 機制，其中包含了多種移除 (removal) 法則與插入 (insertion) 法則，並採取機率性接受劣解的策略。Ropke 與 Pisinger^[17]採用 Gélinas 等人^[6]與 Thangiah 等人^[13]的大規模例題來測試解題績效。

Reimann 與 Ulrich^[19]延續先前的研究^[9]，採用改良式的螞蟻演算法 (ant colony optimization, ACO)，結合 Solomon^[20]的 I1 插入法來構建路線，並加上區域搜尋法進行改善。Reimann 與 Ulrich^[19]採用 Gélinas 等人^[6]的標竿例題來測試解題績效，並與 Ropke 與 Pisinger^[17]及 Thangiah 等人^[13]的最佳結果進行比較。

由表 1 所彙整之各種 VRPBTW 求解方法可知，大多數文獻的方法皆為求取近似解 (approximate solution) 的啟發式或巨集啟發式方法，且多採用「插入法」來構建起始解，採用「區域搜尋法 (亦可稱為鄰域搜尋法)」來進行改善。此外，本研究亦可發現大多數的文獻皆採用 Gélinas 等人^[6]的 VRPBTW 標竿例題來測試其方法的解題績效。

2.3 門檻接受法之基本概念與演變

門檻接受法 (TA) 是由 Dueck 與 Scheuer^[10]於 1990 年所提出的一個巨集啟發式方法，其執行架構與傳統鄰域搜尋法之架構相似，差異之處僅在於使用的接受法則不同。傳統的鄰域搜尋法僅接受較佳的鄰解，TA 則可接受較劣之鄰解。TA 之基本概念在於：當鄰域搜尋過程陷入局部最佳解 (local optimum) 時，採取較鬆的接受法則來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。TA 預先產生一組固定的門檻值數列 (通常為遞減之正值)，並在鄰域搜尋過程中依次使用數列中的門檻值；對一極小化問題而言，其接受法則為：

$$C(X^{neighbor}) - C(X^{current}) < T_k \quad (1)$$

式 (1) 中， $C(X^{current})$ 及 $C(X^{neighbor})$ 分別表示現解與鄰解的目標函數值， T_k 則為第 k 個門檻值。茲以圖 1 的示意圖說明 TA 接受法則之運作機制：當鄰解的目標值小於現解目標值加上門檻值時，即使鄰解劣於現解，搜尋過程仍可移動至鄰解 (亦即將該鄰解視為是新的現解)。此外，若鄰解的目標值原已低於現解目標值，根據式 (1) 仍可接受該鄰解，並據以更新搜尋過程中的暫優解。

表 1 VRPBTW 求解方法彙整表

年代	作者	求解方法		測試例題
		最佳化方法	(巨集) 啟發式方法	
1995 [6]	Gélinas 等人	<ul style="list-style-type: none"> 集合分割問題 分枝定限法 線性鬆弛技巧 		<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
1996 [7]	Potvin 等人		<ul style="list-style-type: none"> 巨集：基因演算法 起始：貪心插入法 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
1996 [13]	Thangiah 等人		<ul style="list-style-type: none"> 起始：後推式插入法 改善：區域搜尋法 	<ul style="list-style-type: none"> 顧客點數：25、50、100、250、500 取貨比例：10%、30%、50%
1997 [8]	Duhamel 等人		<ul style="list-style-type: none"> 巨集：禁制搜尋法 起始：貪心插入法 改善：區域搜尋法 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
1999 [14]	申生元		<ul style="list-style-type: none"> 起始：巢式平行法 改善：區域搜尋法 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
2002 [9]	Reimann 等人		<ul style="list-style-type: none"> 巨集：螞蟻演算法 起始：插入法 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
2003 [18]	Cheung 與 Hang	<ul style="list-style-type: none"> 標籤配對模式 	(以 MP 為基礎之啟發法) <ul style="list-style-type: none"> 同步指派法 循序車輛-需求指派 	<ul style="list-style-type: none"> 顧客點數：25、50、100、122 取貨比例：5%、10%、30%、50% 時間窗比例：25%、50%
2005 [15]	Zhong 與 Cole		<ul style="list-style-type: none"> 巨集：導引式區域搜尋法 起始：適應性掃瞄法 改善：分段規劃技巧 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：10%、30%、50%
2005 [16]	Cho 與 Wang		<ul style="list-style-type: none"> 巨集：門檻接受法 起始：最近鄰點法 改善：鄰域搜尋法 	<ul style="list-style-type: none"> 顧客點數：25、50、100 取貨比例：30%、50%、70%
2006 [17]	Ropke 與 Pisinger		<ul style="list-style-type: none"> 巨集：制式啟發式法 改善：大規模鄰域搜尋 改善：破壞與重生機制 	<ul style="list-style-type: none"> 顧客點數：100、250、500 取貨比例：10%、30%、50%
2006 [19]	Reimann 與 Ulrich		<ul style="list-style-type: none"> 巨集：改良式螞蟻演算法 起始：I1 插入法 改善：區域搜尋法 	<ul style="list-style-type: none"> 顧客點數：100 取貨比例：10%、30%、50%

資料來源：本研究整理。

Lin 等人^[11]針對門檻接受法與模擬鍛鍊法 (simulated annealing, SA) 各提出了「標準型」與「調適型」兩種版本的演算架構，並將其應用於求解三種複雜的排班問題以比較各演算架構的解題績效。測試結果顯示無論是在標準型或調適型的演算架構方面，TA 皆比 SA 具有較佳的解題績效表現。

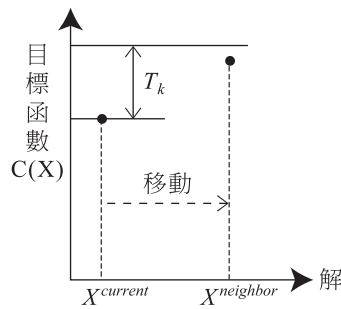


圖 1 TA 接受法則示意圖

近來，Tarantilis 等人^[12]提出一個改良型的門檻接受法，稱之為回溯式門檻接受法 (backtracking adaptive threshold accepting, BATA)，並將其應於多車種固定車隊車輛路線問題 (heterogeneous fixed fleet vehicle routing problem, HFFVRP) 之求解。測試結果顯示 BATA 不僅具有強而有效 (robust and efficient) 的解題績效，在 20 個 HFFVRP 標竿例題中也有 6 題突破了文獻已知最佳解。

BATA^[12]不同於傳統 TA 之處主要有兩點：門檻數列型態與停止條件。門檻數列型態方面，在鄰域搜尋過程中，若可找到滿足式 (1) 的鄰解時，則門檻值按照事先設定的數列順序逐漸遞減；否則，即增高其門檻值（稱之為回溯），但不可超過上一次之門檻值。門檻值之回溯公式如下：

$$T_{back} = T_{now} + r \cdot (T_{old} - T_{now}) \quad (2)$$

式 (2) 中， T_{back} 為回溯後的門檻值、 T_{now} 為目前門檻值（無法找到可接受的鄰解）、 T_{old} 為 T_{now} 之前一次的門檻值（能找到可接受的鄰解）， r 為門檻回溯比例（小於 1 之正值）。

若在回溯後的門檻值下能找到可接受的鄰解，則門檻值就恢復遞減型態，並將此回溯門檻值設為 T_{old} ；否則，門檻值就繼續回溯增高。在連續回溯的過程中， T_{old} 始終維持不變，以避免門檻過高導致搜尋過程無法收斂停止的情形發生；也就是說，無論連續回溯幾次， T_{back} 的值絕對不會超過 T_{old} 的值。由此可知，BATA 的實際門檻數列型態將視搜尋過程的結果而定，是一種可自我調適 (adaptive) 的型態。

茲舉圖 2 之簡單數例說明上述 BATA 的門檻值回溯與門檻數列型態概念。首先假設：門檻數列為等差遞減型態，每次遞減值為 5，門檻回溯比例 $r = 0.8$ ，目前門檻值 $T_{now} = T_k = 100$ ，前一次門檻值 $T_{old} = T_k - 1 = 105$ 。情境一：若在目前門檻值之下能找到可接受的鄰解，則下一個門檻值為 $T_k + 1 = 100 - 5 = 95$ ，並令 $T_{old} = T_k = 100$ ，然後繼續搜尋。情境二：若在 $T_{now} = 100$ 之下無法找到可接受的鄰解，則依式 (2) 計算出回溯後的門檻值 $T_{back} = 100 + 0.8 \times (105 - 100) = 104$ ，然後令 $T_k + 1 = T_{back} = 104$ ， T_{old} 仍為 105；若繼續搜尋後能找到可接受之鄰解，則遞減門檻值 $T_k + 2 = 104 - 5 = 99$ ，並更新 T_{old} 為 104。情境三：若門檻值回

溯後仍無法找到可接受的鄰解，則繼續回溯門檻值 $T_{back} = 104 + 0.8 \times (105 - 104) = 104.8$ ，然後令 $T_k + 2 = T_{back} = 104.8$ ， T_{old} 則仍為 105。

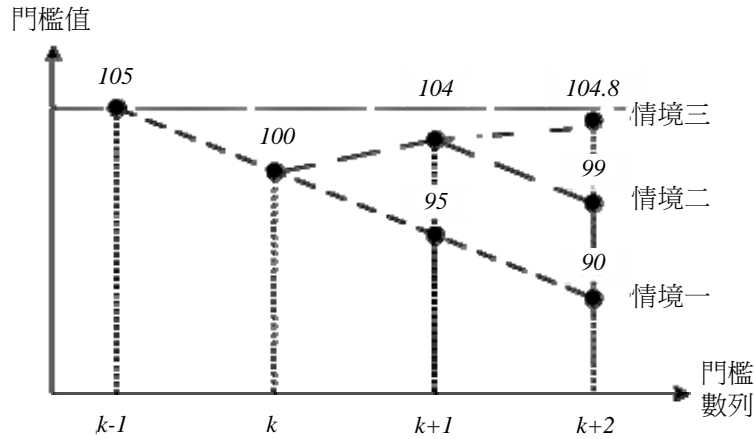


圖 2 BATA 門檻數列型態示意圖

在停止條件方面，除了原本設定的門檻數列次數外，另加上「連續回溯次數達到預設次數」作為停止條件；亦即 BATA 尚須設定另一個參數 L (最大連續回溯次數)，以避免回溯過程無法終止的情形發生。

2.4 小結

由上述的文獻回顧可知：BATA 法係由 TA 法改良而來，不僅具有結構簡單、執行容易、參數不敏感等優點^[10, 11]，且應用在複雜的 HFFVRP 問題上也有很好的績效表現^[12]。雖然 Cho 與 Wang^[16]曾經應用 TA 來求解 VRPBTW，其結果證實 TA 的架構確實可以應用於具時間窗與回程取貨等限制之 VRP 問題上，但是該研究所使用的卻是自行設計之測試例題，無法與其他文獻的結果進行比較。

有鑑於此，本研究即以 BATA 法為基礎，結合傳統鄰域搜尋方法提出一個可適用於求解 VRPBTW 之「兩階段回溯型門檻接受法 (two-phase backtracking threshold accepting algorithm, TBTA)」的解題架構，並將其命名為 TBTA_VRPBTW。下節介紹整個 TBTA_VRPBTW 之解法架構與模組設計細節。

三、TBTA_VRPBTW 之解法架構與模組設計

整套 TBTA_VRPBTW 解題架構共包含三個模組：起始解構建模組、鄰域搜尋模組，以及回溯型門檻接受模組。其中，起始解構建模組係根據傳統的循序插入法 (sequential

insertion heuristic)，提出三種不同的插入準則，以分別構建不同的起始解（將於 3.1 小節中詳述）。鄰域搜尋模組內則包含了「路線內交換」與「路線間交換」兩類改善方法（將於 3.2 小節中詳述）。至於回溯型門檻接受模組方面，由於 TBTA_VRPBTW 採用了兩階段的執行方式，因此又可分為 BTA1 與 BTA2 兩種回溯型門檻接受模組（將於 3.3 小節中詳述）。圖 3 顯示 TBTA_VRPBTW 之解題架構流程，其中，停止法則可設定為 BTA1 與 BTA2 模組之最大回合次數，或是門檻值的最大連續回溯次數。

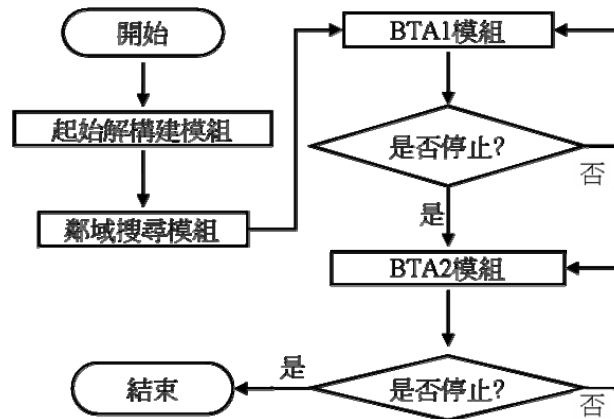


圖 3 TBTA_VRPBTW 之解題架構流程

3.1 起始解構建模組

本研究以循序插入法的概念為基礎來設計起始解構建模組，圖 4 顯示插入法的基本概念。由圖 4 可知，首先選擇一個尚未被服務的顧客點作為種子點（如點 1），連結種子點與場站（點 0）形成初始路線，然後根據所設定的插入準則陸續將尚未被服務的顧客點（如點 5、點 4）插入至路線中，直到該路線的相關限制條件（例如：容量）無法被滿足為止。在插入準則方面，又可分成「插入點選擇」與「插入位置選擇」兩部分：「插入點選擇」係從尚未被服務的顧客點中選擇要插入到路線中的節點；「插入位置選擇」則是在決定該插入節點的插入位置。

至於插入法的執行方式則有循序 (sequential) 與平行 (parallel) 兩種。「循序」意指一次構建一條路線，路線構建完成後若仍有顧客點尚未被服務，則再構建另一條新的路線，重複上述步驟直到所有顧客點皆已被某條路線服務為止。而「平行」則是在路線構建之初，即產生多條初始路線，然後依據前述的插入準則同時構建多條路線。本研究即根據上述的插入法概念，採用循序插入的執行方式來設計起始解構建方法。

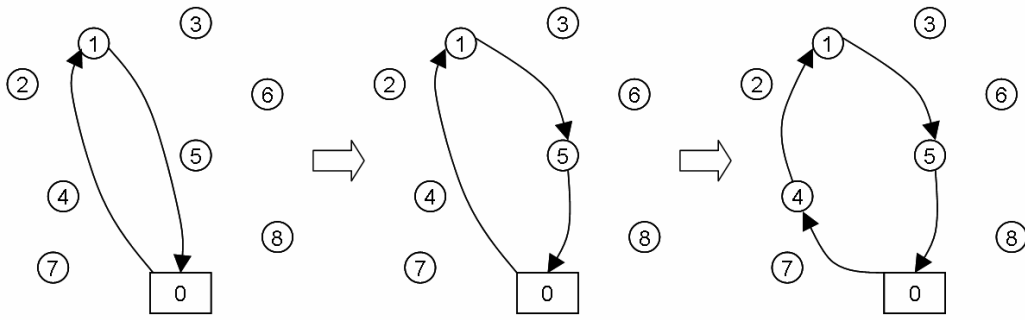


圖 4 插入法的概念示意圖

此外，在起始路線方面，本研究則提出雙種子點的路線型態。首先，從所有尚未被服務的顧客點中選取時間窗起點最早之顧客點作為路線的第一點（種子點 1）。若該點屬於送貨點顧客，則再從尚未被服務的顧客點中選取時間窗起點最晚之顧客點（可以是送貨點或取貨點顧客），作為路線的最後一點（種子點 2）以構成初始路線。反之，若種子點 1 屬於取貨點顧客，則只能從尚未被服務的取貨點顧客中選取時間窗起點最晚之顧客點作為路線的最後一點（種子點 2）。

在路線構建與節點插入的過程中，皆須符合所有限制條件（容量限制、時窗限制、先送貨後取貨限制），直到所有顧客點皆被服務為止。其中，插入位置之選擇即須符合「時間窗」與「先送貨後取貨」兩項限制，而無法完全以插入成本來考量。因此，本研究在插入準則方面，除了傳統的最省插入成本（cheapest insertion cost, CIC）外，另提出兩種針對 VRPBTW 問題特性的插入準則：最窄時間窗（narrowest time window, NTW）與最小等待時間比例（minimal waiting-time ratio, MWR）。NTW 準則係希望先將較難插入的節點（時窗較窄）建構於路線中；而 MWR 準則係希望插入某節點後，該節點與後一節點之間有較寬鬆的時間，以便能插入更多的節點。茲說明三種插入準則如下。

（一）最省插入成本（CIC）準則

CIC 準則係以插入後所增加的路線成本最小為標準，同時決定插入節點及其插入位置。假設要將顧客點 k 插入至目前路線的節線 (i, j) 當中，則其插入成本之計算方式為： $c_{ik} + c_{kj} - c_{ij}$ ，其中 c_{ik} 、 c_{kj} 與 c_{ij} 分別為節線 (i, k) 、 (k, j) 與 (i, j) 的成本。

（二）最窄時間窗（NTW）準則

NTW 準則係選擇時間窗寬度最窄且尚未被服務的顧客點作為插入節點，而其插入位置則仍以插入成本最小為原則。

（三）最小等待時間比例（MWR）準則

由於 VRPBTW 為硬時窗限制，因此當車輛在某節點的時窗開啟時間之前即已抵達該節點時，就須等待時窗開啟後才能服務該顧客，這段時間即稱為等待時間。而等待時間比例則指：將顧客點 k 插入至路線的節線 (i, j) 時，車輛在節點 k 的等待時間除以在節點 j 的等待時間之比值。因此，所謂 MWR 準則即是選擇等待時間比例最小的顧客點及其插入位置。

茲以圖 5 之簡例說明上述三種插入準則的概念與差異。圖 5 顯示某初始路線 $\text{depot} \rightarrow A \rightarrow B \rightarrow \text{depot}$ ，另有三個顧客點 (C、D、E) 尚未被服務。假設 t 為兩點間之行車時間，各節點旁中括號內的兩個數字分別為該節點之時窗開啟與關閉時間，且各節點的服務時間為 10 單位。表 2 列出節點 C、D、E 分別以 CIC、NTW、MWR 等準則進行計算之結果。由表 2 可知：在 CIC 準則下，以 C 點的插入成本 5 為最小，故選擇將 C 點插入至節線 (depot, A) 之間；在 NTW 準則下，以 D 點時間窗 [160, 170] 的寬度 10 為最窄，故選擇將 D 點插入至節線 (B, depot) 之間；而在 MWR 準則下，則以 E 點之等待時間比例為 0 最小，故選擇將 E 點插入至節線 (A, B) 之間。茲進一步解釋 E 點等待時間比例之計算：當插入 E 點至節線 (A, B) 之間後，從 A 點至 E 點的抵達時間為 $40 + 10 + 15 = 65$ ，因此 E 點的等待時間為 $65 - 65 = 0$ ，從 E 點至 B 點的抵達時間為 $65 + 10 + 10 = 85$ ，B 點的等待時間為 $120 - 85 = 35$ ，故其等待時間比例為 $0 / 35 = 0$ 。

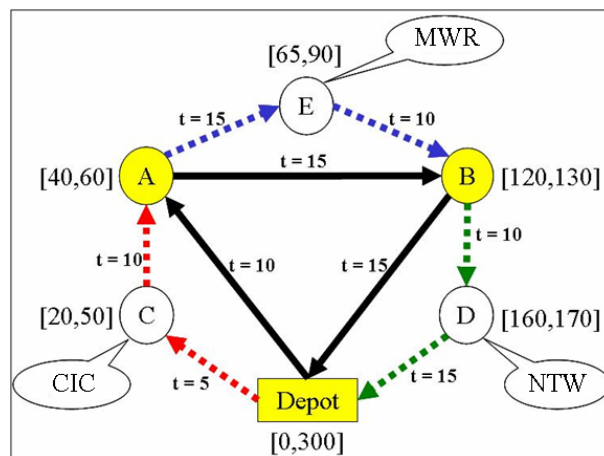


圖 5 三種插入準則之簡例說明圖

表 2 三種插入準則簡例之計算結果

插入準則	C 點	D 點	E 點	選擇
最省插入成本 (CIC)	5	10	10	C 點
最窄時間窗 (NTW)	30	10	25	D 點
最小等待時間比例 (MWR)	∞	0.17	0	E 點

起始解構建模組在執行時，係先構建出雙種子點起始路線，然後從上述的 CIC、NTW 與 MWR 三種插入準則中選擇一種準則來執行。在後續的例題測試過程中，本研究將分別以這三種插入準則來構建起始解與進行改善。此外，在某一準則下，若有多點的準則值相同時，係選擇顧客點之順序最小者來進行插入。

3.2 鄰域搜尋模組

鄰域搜尋模組係循序執行五種交換法：路線內之 Or-opt ($p = 1$) 節線交換法、路線間之 2-opt* 節線交換法、(1-0) 節點交換法、(1-1) 節點交換法，及 (1-1-0) 節點交換法。亦即，先執行完畢某種方法後，再接著執行另一種方法。其中，(1-0) 節點交換法與 (1-1-0) 節點交換法皆同時具備減少車輛數和路線成本的功能；其餘交換法的主要功能則在於降低路線成本。至於鄰域搜尋方法的執行順序，則是根據其交換節線數或節點數之多寡，由少至多、由路線內至路線間，依序執行。茲簡要說明五種交換法如下。

(一) Or-opt ($p = 1$) 節線交換法

Or-opt 節線交換法乃是將某路線內的一點 ($p = 1$)、連續兩點 ($p = 2$) 或連續三點 ($p = 3$) 自該路線移除後，再重新插入到原路線的其他位置上。本研究僅使用 $p = 1$ 情形的 Or-opt 節線交換法，此法亦可稱之為路線內 (1-0) 節點交換法；為避免與後續之路線間 (1-0) 節點交換法產生混淆，本研究採以 Or-opt ($p = 1$) 法稱之。如圖 6 顯示，將節點 5 自路線移除後，再插入至節線 (6, 8) 中間以形成新路線。對於有時間窗限制的車輛路線問題而言，此方法可局部調整路線內的節點服務順序，以減少路線行駛距離或時間。

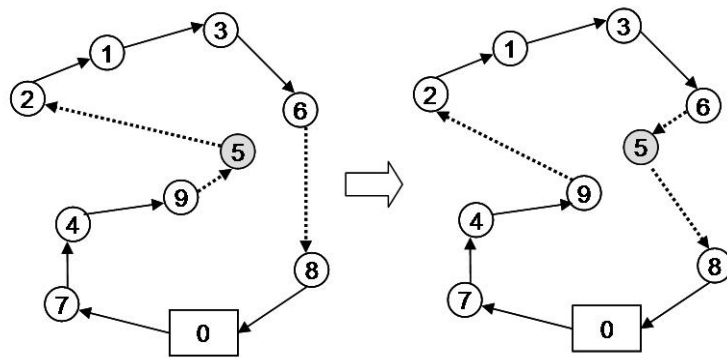


圖 6 Or-opt ($p = 1$) 節線交換法之觀念示意圖

(二) 2-opt* 節線交換法

乃是從路線內 2-opt 節線交換法演變而來，係將某兩條路線的各一條節線進行交換。

如圖 7 顯示，將左路線之節線 (2, 5) 與右路線之節線 (1, 0) 移除，然後重新連接節線 (2, 0) 與節線 (1, 5) 以形成新路線。對於有時間窗限制的車輛路線問題而言，此法可大幅度調整路線間的節點服務順序，以減少路線行駛距離或時間。

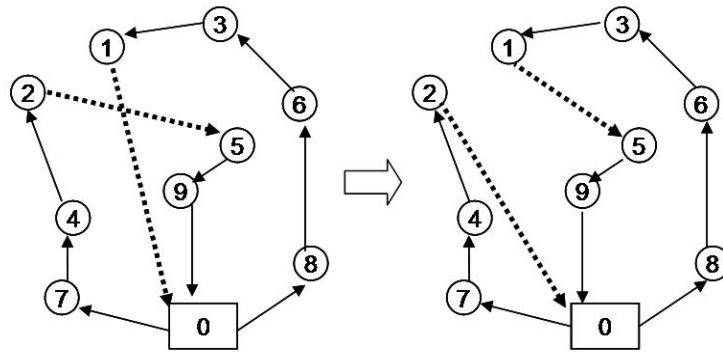


圖 7 2-opt* 節線交換法之觀念示意圖

(三)(1-0) 節點交換法

乃是將某路線中的一點自該路線移除後，再重新插入到其他路線當中。如圖 8 顯示，將右路線中的節點 9 自該路線移除後，再插入至左路線的節線 (1, 0) 中間以形成新路線。此方法不僅可減少路線行駛距離或時間，也有助於減少使用的車輛數。

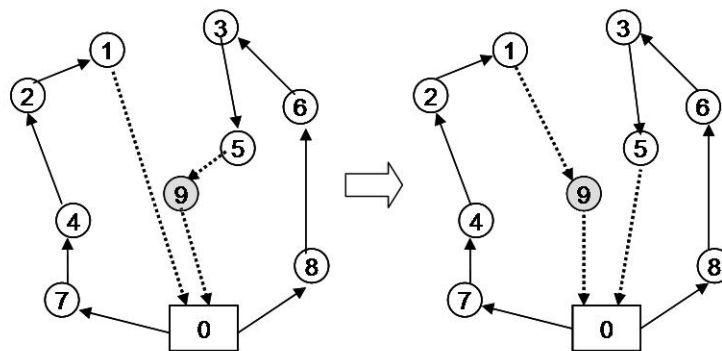


圖 8 (1-0) 節點交換法之觀念示意圖

(四)(1-1) 節點交換法

係將某兩條路線的各一節點自該路線移除後，再重新插入到對方路線的原本位置上。

如圖 9 顯示，將左路線之節點 5 與右路線之節點 1 自原路線中移除後，再插入至對方路線的原本位置以形成新路線。此方法可減少路線行駛距離或時間。

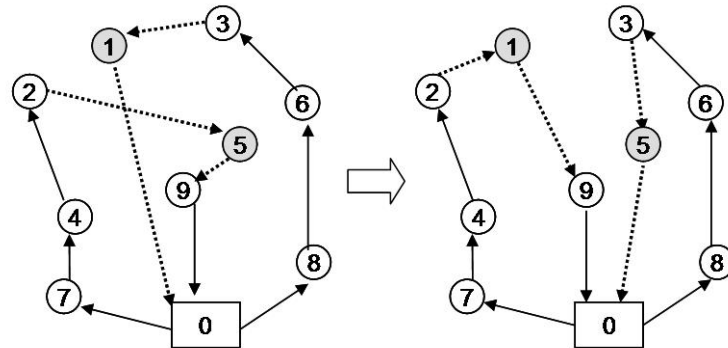


圖 9 (1-1) 節點交換法之觀念示意圖

(五)(1-1-0) 節點交換法

乃是將甲路線中的某一節點插入到乙路線中，並將乙路線中的另一節點插入至丙路線中；屬於一種簡化的 node-ejection chains 法^[21]。如圖 10 顯示，將左路線的節點 2 插入至中路線的節線 (1, 0) 中間，再將中路線的節點 9 插入至右路線的節線 (5, 0) 中間，以形成三條新路線。此方法可大幅度調整路線間的節點服務順序，以減少路線行駛距離或時間，並減少車輛數。

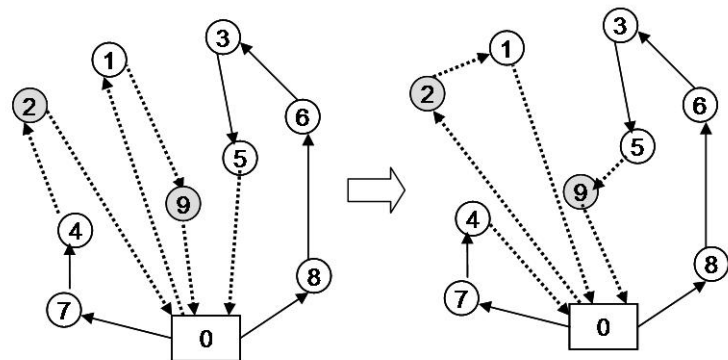


圖 10 (1-1-0) 節點交換法之觀念示意圖

執行上述五種鄰域搜尋時，皆需檢查交換後是否符合容量限制、時窗限制及先送貨後取貨限制，以確保解的可行性。

表 3 BI 策略與 SI 策略之差異：以 Or-opt ($p = 1$) 節線交換法為例

—420—

3.3 回溯型門檻接受模組

本研究根據圖 3 之兩階段 BATA 架構，分別提出兩種回溯型門檻接受模組：BTA1 與 BTA2。採用兩階段架構之目的在於，第一階段希望用較少的執行時間進行路線間的節點調整，並減少車輛數；第二階段則對現解進行較大規模的節線與節點交換，希望能同時改善車輛數與行駛時間。

圖 11 顯示回溯型門檻接受模組的執行流程。如圖 11 所示，在此循序執行過程中，門檻值是在該模組內的各種鄰域搜尋方法皆依序執行一次 TA 準則之鄰域搜尋，並執行鄰域搜尋模組後，再依 2.3 節所述的 BATA 門檻遞減型態，更換成下一個門檻值。各主要步驟說明如下：

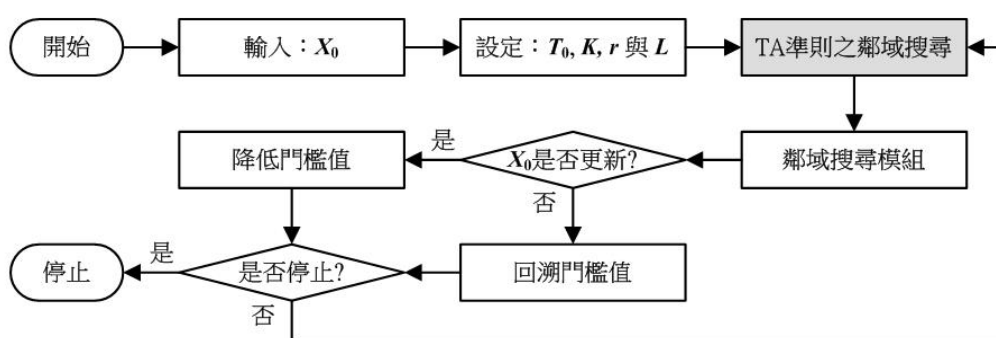


圖 11 回溯型門檻接受模組執行流程

- (一) 輸入起始解：係承續前一階段模組的最終結果，以作為本模組的起始解 (X_0)，然後對該起始解進行改善。
- (二) 設定 TA 參數：本研究採用「直線遞減型」的門檻值數列，其門檻值計算公式如式 (4) 所示。其中， $C(X_0)$ 為起始解之目標函數值， N 為例題之顧客節點數， R 為起始解之路線數， T_0 為起始門檻值比率， K 為門檻值遞減參數。除設定上述的 T_0 與 K 等參數外，尚需設定門檻值回溯比率 r 與最大連續回溯次數 L 之值。

$$\begin{cases} T_1 = \frac{C(X_0)}{N+R} \times T_0 \\ T_k = T_{k-1} - \frac{T_1}{k-1}, \text{ for } k > 1 \end{cases} \quad (4)$$

- (三) TA 準則之鄰域搜尋：此步驟乃根據式 (1) 之門檻接受法則進行鄰域搜尋，以便搜尋過程能夠跳脫目前局部最佳解的束縛。本研究共設計了 BTA1 與 BTA2 兩種模組，其執行架構與流程完全相同，差異僅在於使用鄰域搜尋方法的數量不同。上述兩種模組

所使用的各種鄰域搜尋方法，亦採取循序執行的方式，以及 SI 策略。

- (1) BTA1 模組僅採用(1-0)節點交換與(1-1)節點交換兩種方法。
- (2) BTA2 模組則採用 Or-opt ($p = 1$) 節線交換、2-opt* 節線交換、(1-0) 節點交換、(1-1) 節點交換、(1-1-0) 節點交換等方法，即鄰域搜尋模組的五種交換法。
- (四) 鄰域搜尋模組：在 TA 準則鄰域搜尋之後，再執行一次鄰域搜尋模組（嚴格的接受法則：只接受有改善的鄰解，並採 SI 策略），以便找到新的局部最佳解。
- (五) 更新門檻值：首先檢查現解是否因執行前兩步驟而有所改變。若是，則依據式 (4) 降低門檻值 ($k: = k + 1$)；否則，根據式 (2) 回溯門檻值。
- (六) 停止法則：判斷是否達到停止條件。本模組共有兩種停止條件，當符合其中任一條件時，即停止本模組：(1) 門檻值收斂至 0；(2) 連續回溯次數超過 L 次。

四、例題測試結果分析

4.1 測試例題與參數設定

從 2.2 節的文獻回顧得知，大多數的文獻皆採用 Gélinas 等人^[6]的 VRPBTW 例題來進行測試，故本研究也採用 Gélinas 等人^[6]的國際標竿例題。如表 4 所示，Gélinas 等人^[6]的 VRPBTW 例題共有 15 題，係以 Solomon^[20]的 VRPTW 標竿題庫中，r1 系列的前 5 個例題（即 r101, r102, r103, r104, r105）為基礎進行修改。原本的 5 個 VRPTW 例題皆為 100 個顧客點的規模，顧客點呈隨機分佈；Gélinas 等人^[6]以隨機方式分別在 100 個顧客點中選擇 10%（即 10 個顧客點）、30% 及 50% 的顧客作為取貨點顧客（原本的配送量即變成取貨量），產生了這 15 個 VRPBTW 例題。在顧客時窗寬度方面，BHR101 與 BHR105 這兩組例題為固定，分別為 10 單位與 30 單位，其他三組例題則為不固定。

此外，本研究亦蒐集並整理出目前文獻已知最佳解之結果及其來源，一併列於表 4。其中，在文獻已知最佳解方面，VRPBTW 有兩項目標：總車輛數極小化、總行駛時間極小化。在比較解題績效時，係以第一目標之總車輛數極小化為主要準則；當總車輛數相同時，再比較第二目標之總行駛時間。

TBTA_VRPBTW 的執行模組與參數設定如下：起始解模組方面，分別採用了 CIC、NTW、MWR 三種插入準則之起始解法。鄰域搜尋模組使用了路線內 Or-opt ($p = 1$)，路線間 2-opt*、(1-0)、(1-1)、(1-1-0) 等五種交換法。BTA 模組方面，採用先 BTA1 再 BTA2 的執行架構；停止條件為「門檻值收斂至 0」或「連續回溯次數超過 L 次」。

4.2 測試結果之整理與分析

本研究使用 Visual C# 語言撰寫 TBTA_VRPBTW 演算法之電腦執行程式，並在 2.2

表 4 VRPBTW 標竿例題之資料彙整

例題編號	顧客數	取貨比例	顧客時窗寬度	文獻已知最佳解		來源 [†]
				車輛數	行駛時間	
BHR101A	100	10%	固定 (10)	22	1818.86	RP
BHR101B	100	30%	固定 (10)	23	1959.56	RP
BHR101C	100	50%	固定 (10)	24	1909.84	HKK
BHR102A	100	10%	不固定	19	1653.19	RP
BHR102B	100	30%	不固定	21	1764.30	TPS
BHR102C	100	50%	不固定	21	1745.70	TPS
BHR103A	100	10%	不固定	15	1371.60	TPS
BHR103B	100	30%	不固定	15	1390.33	RP
BHR103C	100	50%	不固定	16	1486.56	ZC
BHR104A	100	10%	不固定	11	1084.17	RP
BHR104B	100	30%	不固定	11	1154.84	RP
BHR104C	100	50%	不固定	11	1191.38	RP
BHR105A	100	10%	固定 (30)	15	1561.28	RP
BHR105B	100	30%	固定 (30)	16	1583.30	RP
BHR105C	100	50%	固定 (30)	16	1710.19	RP
平均值				17.1	1559.007	

註：[†] RP = Ropke 與 Pisinger^[17]，HKK = Hasama 等人^[23]，TPS = Thangiah 等人^[13]，ZC = Zhong 與 Cole^[15]。

GHz P4-CPU、1GB RAM 的個人電腦上執行例題測試。解題績效之分析係以「解的精確度」為指標，用以衡量 TBTA_VRPBTW 演算法所得之最佳解與「文獻已知最佳解」之間的誤差(%)；共包括兩項目標值：第一目標為車輛數，第二目標為行駛時間。此外，電腦的 CPU 運算時間也會一併列出，以作為演算法執行效率之評估參考。

為了解 TBTA_VRPBTW 各階段模組，以及 BTA 模組組合方式之績效，本研究首先將 BTA 模組之參數值設定為：門檻值比率 $T_0 = 1.0$ 、門檻值遞減參數 $K = 500$ 、門檻值回溯比率 $r = 0.9$ 、最大連續回溯次數 $L = 500$ ，然後進行解題測試。表 5 彙整 TBTA_VRPBTW 之模組組合測試結果，包括三種插入準則下之起始解經鄰域搜尋模組改善後之平均解題績效(15 個例題之平均值)，以及上述結果經各種 BTA 模組及其組合改善後之平均解題績效。

由表 5 可知：(1) 起始解經鄰域搜尋模組改善後，以 CIC (最省插入成本) 準則之解題績效最佳，其平均車輛數為 21.1、平均行駛時間為 1730.008。(2) 就 BTA 模組之改善幅度而言，以 BTA1 模組在 CIC 準則下之解題績效為例，其平均車輛數改善幅度為 12.8%、平均行駛時間改善幅度為 4.4%，改善績效明顯。(3) 所有 BTA 模組組合測試中，以 NTW (最

窄時間窗) 準則搭配兩階段 (BTA1 + BTA2) 之結果最佳, 其平均車輛數為 17.7、平均行駛時間為 1617.300。(4) 就整體績效而言, CIC 準則與 NTW 準則略優於 MWR 準則, 兩階段 BTA 模組優於單一 BTA 模組; 而兩階段 BTA 模組中則以 BTA1 + BTA2 組合稍優於 BTA2 + BTA1 組合。

表 5 TBTA_VRPBTW 之模組組合測試結果彙整

BTA 模組	CIC 準則		NTW 準則		MWR 準則	
	車輛數	行駛時間	車輛數	行駛時間	車輛數	行駛時間
起始解 + 鄰域搜尋改善	21.1	1730.008	21.9	1807.312	21.7	1778.046
BTA1	18.4	1653.835	18.6	1649.483	18.7	1650.807
BTA2	18.0	1626.459	18.1	1626.963	18.1	1633.315
BTA1 + BTA2	18.0	1629.050	17.7	1617.300	18.1	1622.980
BTA2 + BTA1	18.0	1619.857	18.1	1618.619	18.1	1625.580

為進一步了解參數值設定對 TBTA_VRPBTW 之解題績效的影響, 本研究再以上述測試表現最佳的 BTA1 + BTA2 模組組合為基礎, 進行參數敏感度分析。在參數值的範圍設定方面, 門檻值比率 T_0 為 0.01、0.1、0.5、1.0, 門檻值遞減參數 K 固定為 500, 門檻值回溯比率 r 為 0.9、0.95、0.99, 最大連續回溯次數 L 為 500 與 1000; 再搭配起始解構建模組的三種插入準則, 總共有 72 種參數設定組合。表 6 彙整了上述各種參數值設定求解 15 個 VRPBTW 標竿例題的測試結果。

觀察表 6 之結果可歸納出以下五點: (1) 在門檻值比率 T_0 方面, 解題績效隨 T_0 值增加而改善, 但是第一目標車輛數在 $T_0 \geq 0.1$ 之後並無差異; (2) 在門檻值回溯比率 r 方面, 解題績效隨 r 值增加而稍有改善, 但是並不明顯; (3) 在最大連續回溯次數 L 方面, $L = 1000$ 之結果明顯優於 $L = 500$ 之結果; (4) 三種插入準則方面, 平均績效表現以 NTW 準則最優且穩定, 其次為 CIC 準則; (5) 最佳績效表現則發生在 NTW 準則下的幾組參數組合, 平均車輛數為 17.9、平均行駛時間為 1619.459。整體而言, TBTA_VRPBTW 在上述設定的參數範圍內表現尚稱穩定。

表 7 則彙整 TBTA 與其他文獻方法求解 VRPBTW 之最佳結果比較。因限於頁面篇幅, 本研究僅列出目前國際文獻上表現最佳的三個方法進行比較, 這三個方法依序來自 Ropke 與 Pisinger^[17] (簡稱 RP)、Reimann 與 Ulrich^[19] (簡稱 RU)、Thangiah 等人^[13] (簡稱 TPS)。由表 7 可知, 目前單一方法求解 Gélinas 等人^[6] VRPBTW 國際標竿例題表現最好的是 RP 的方法, 平均車輛數為 17.3 輛 (誤差 1.17%)、平均行駛時間為 1561.114 (誤差 0.14%); 其次為 RU 的方法, 平均車輛數為 17.4 輛 (誤差 1.95%)、平均行駛時間為 1596.162 (誤差 2.38%); 第三為 TPS 的方法, 平均車輛數為 18.3 輛 (誤差 7.03%)、平均行駛時間為 1603.53 (誤差 2.86%)。至於 CPU 運算時間方面, RP 法之平均值為 113.5 秒、RU 法之平均值為 75

秒、TPS 法之平均值為 17 秒；但是 RU 法與 TPS 法並沒有個別列出各題的運算時間。

再者，表 7 亦列出本研究提出之 TBTA_VRPBTW 方法在測試的各種模組與參數組合下，所得到的各題最佳結果，其平均車輛數為 17.7 輛（誤差 3.52%）、平均行駛時間為 1602.741（誤差 2.81%），平均 CPU 運算時間為 33.69 秒。與表 7 之其他文獻方法相比，解

表 6 TBTA_VRPBTW 之參數設定測試結果彙整

參數範圍設定			CIC 準則		NTW 準則		MWR 準則	
T_0	r	L	車輛數	行駛時間	車輛數	行駛時間	車輛數	行駛時間
0.01	0.9	500	18.6	1637.504	18.3	1618.608	18.5	1627.185
		1000	18.5	1623.318	18.3	1618.100	18.4	1627.389
	0.95	500	18.5	1638.727	18.3	1617.201	18.4	1623.186
		1000	18.5	1635.876	18.3	1616.929	18.3	1622.663
	0.99	500	18.0	1625.079	18.3	1616.604	18.4	1623.140
		1000	18.5	1633.423	18.3	1616.664	18.3	1622.683
0.1	0.9	500	18.0	1625.079	18.0	1622.491	18.1	1626.130
		1000	18.0	1619.653	17.9	1619.485	18.0	1617.890
	0.95	500	18.0	1625.079	18.0	1622.491	18.1	1626.130
		1000	18.0	1619.653	17.9	1619.485	18.0	1617.890
	0.99	500	18.0	1625.079	18.0	1622.491	18.1	1626.130
		1000	18.0	1623.393	17.9	1619.485	18.0	1617.890
0.5	0.9	500	18.0	1624.375	18.0	1622.465	18.2	1626.460
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
	0.95	500	18.0	1626.611	18.0	1622.465	18.2	1626.460
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
	0.99	500	18.0	1626.611	18.0	1622.465	18.2	1626.460
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
1.0	0.9	500	18.0	1626.611	18.0	1622.465	18.2	1626.460
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
	0.95	500	18.0	1626.611	18.0	1622.465	18.2	1626.460
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
	0.99	500	18.0	1626.611	18.0	1622.465	18.1	1627.385
		1000	18.0	1624.924	17.9	1619.459	18.1	1618.221
平均值			18.1	1626.618	18.1	1620.066	18.2	1622.805
標準差			0.22	4.8776	0.16	2.1161	0.14	3.9562

表 7 TBTA 與其他文獻方法求解 VRPBTW 之結果比較

例題編號	文獻已知最佳解		Ropke 與 Pisinger ^[17]		Reimann 與 Ulrich ^{[19]*}		Thangiah 等人 ^{[13]**}		本研究 (TBTA)	
	車輛數	行駛時間	車輛數	行駛時間	車輛數	行駛時間	車輛數	行駛時間	車輛數	行駛時間
BHR101A	22	1818.86	22	1818.86	22	1853.45	24	1842.3	22	1819.39
BHR101B	23	1959.56	23	1959.56	23	1985.23	24	1928.6	23	1959.56
BHR101C	24	1909.84	24	1939.10	24	1964.04	25	1937.6	24	1939.10
BHR102A	19	1653.19	19	1653.19	19	1663.16	20	1654.1	19	1668.63
BHR102B	21	1764.30	22	1750.70	22	1759.02	21	1764.3	22	1764.36
BHR102C	21	1745.70	22	1775.76	22	1782.91	21	1745.7	22	1788.15
BHR103A	15	1371.60	15	1387.57	15	1454.25	15	1371.6	16	1399.04
BHR103B	15	1390.33	15	1390.33	15	1407.29	16	1477.6	15	1465.56
BHR103C	16	1486.56	17	1456.48	17	1478.48	17	1543.2	17	1518.44
BHR104A	11	1084.17	11	1084.17	11	1153.06	13	1220.3	12	1351.16
BHR104B	11	1154.84	11	1154.84	11	1228.61	12	1303.5	12	1307.89
BHR104C	11	1191.38	11	1191.38	11	1306.97	13	1346.6	12	1302.68
BHR105A	15	1561.28	15	1561.28	16	1570.11	17	1553.4	16	1538.27
BHR105B	16	1583.30	16	1583.30	16	1646.11	18	1706.7	16	1595.67
BHR105C	16	1710.19	16	1710.19	17	1689.74	18	1657.4	17	1623.22
平均值	17.1	1559.007	17.3	1561.114	17.4	1596.162	18.3	1603.53	17.7	1602.741
誤差 (%)			1.17	0.14	1.95	2.38	7.03	2.86	3.52	2.81

註：□ 執行環境：1.5 GHz P4 個人電腦，CPU 時間單位為秒。

* 執行環境：1.5 GHz P4 個人電腦，CPU 時間單位為秒；但各題的運算時間無法取得 (not available)。

** 執行環境：33 MHz 68040 個人電腦，CPU 時間單位為秒；但各題的運算時間無法取得 (not available)。

□ 執行環境：2.2 GHz P4 個人電腦，CPU 時間單位為秒。

的精確度雖不及最佳的兩個方法，但已超越排名第三的方法；而在運算時間方面，儘管表 7 中各文獻所使用的電腦設備效能不同，本研究方法所耗費的電腦運算時間應屬合理且可接受之範圍。整體而言，兩項目標的誤差分別低於 4% 與 3%，表現尚稱良好。

五、結論與建議

本研究嘗試將回溯型門檻接受法應用於求解 VRPBTW，整套啟發式方法包括起始解構建、鄰域搜尋與回溯型門檻接受三大模組。起始解方法以循序插入法為基礎，設計出三種插入準則：最省插入成本 (CIC)、最窄時間窗 (NTW) 與最小等待時間比例 (MWR)。鄰域搜尋方法包括：路線內 Or-opt ($p = 1$) 節線交換法，路線間 2-opt* 節線交換法、(1-0) 節點交換法、(1-1) 節點線交換法與 (1-1-0) 節點交換法。最後，再搭配兩階段回溯型門檻接受模組 (BTA1 與 BTA2) 來加強搜尋改善效果。

為檢視本研究提出 TBTA_VRPBTW 之解題績效，採用 Gélinas 等人^[6]之 VRPBTW 國際標準題庫進行測試，結果發現，與文獻已知最佳解相比，平均車輛數為 17.7 輛 (誤差 3.52%)，平均行駛時間為 1602.741 (誤差 2.81%)，平均 CPU 運算時間為 33.69 秒，績效表現尚稱良好。在參數敏感度測試方面，TBTA_VRPBTW 在本研究所設定之參數範圍內解題績效表現穩定。上述結果證實了 TBTA 的解題架構對於高複雜度的 VRPBTW 問題，確實具有解題效果與實務應用的潛力。

此外，針對未來可能的研究方向，本文亦提出下列幾點建議：

1. 可針對各種 BATA 參數 (T_0 、 K 、 L 、 r) 或門檻數列遞減型態進行更廣泛的設定與測試，以獲得較適的參數範圍。
2. 在門檻值回溯比例方面，亦可嘗試大於 1 之回溯比例，或其他的門檻設定方式，例如：清單型門檻 (list based threshold accepting, LBTA)^[24]。
3. 可使用其他更有效率的鄰域搜尋方法，例如：node-ejection chains 法^[21]，或是不同的鄰域搜尋法執行順序，並探討其求解之績效。
4. 可嘗試結合其他的巨集啟發式方法於 TBTA 的架構中，例如：螞蟻演算法 (AS) 或導引式區域搜尋法 (GLS)。
5. 可考慮將本研究提出之 TBTA 架構應用於其他高複雜度的問題，例如：時窗限制之同時收送車輛路線問題 (pickup and delivery problem with time windows, PDPTW)。

參考文獻

1. Yano, C., Chan, T., Richter, L., Cutler, T., Murty, K., and McGettigan, D., "Vehicle Routing at Quality Stores", *Interfaces*, Vol. 17, No. 2, 1987, pp. 52-63.

2. Casco, D, Golden, B., and Wasil, E., "Vehicle Routing with Backhauls: Models, Algorithms, and Case Studies", in Golden, B. and Assad, A. (eds.), *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam, 1988.
3. 卓裕仁、王生德, 「回程取貨車輛路線問題 (VRPB) 之解法回顧」, 中華民國運輸學會第 18 屆學術論文研討會論文集, 第五冊, 中華民國運輸學會, 民國九十二年, 頁 1022-1031。
4. 卓裕仁、王生德, 「時窗限制回程取貨車輛路線問題之巨集啟發式解法設計與測試」, 第一屆臺灣作業研究學會學術研討會暨 2004 年科技與管理學術研討會論文集, 臺灣作業研究學會, 民國九十三年, 頁 188-194。
5. Deif, I. and Bodin, L., "Extension of the Clarke-Wright Algorithms for Solving the Vehicle Routing with Backhauling", *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson College, 1984, pp. 75-96.
6. Gélinas, S., Desrochers, M., Desrosiers, J., and Solomon, M., "A New Branching Strategy for Time Constrained Routing Problems with Application to Backhauling", *Annals of Operations Research*, Vol. 61, 1995, pp. 91-109.
7. Potvin J., Duhamel, C., and Guertin, F., "A Genetic Algorithms for Vehicle Routing Problem with Backhauling", *Applied Intelligence*, Vol. 6, 1996, pp. 345-355.
8. Duhamel, C., Potvin, J., and Rousseau, J., "A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows", *Transportation Science*, Vol. 31, No. 1, 1997, pp. 49-59.
9. Reimann, M., Doerner, K., and Hartl, R., "Insertion Based Ants for Vehicle Routing Problems with Backhauls and Time Windows", *LNCS*, Vol. 2463, 2002, pp. 135-148.
10. Dueck, G. and Scheuer, T., "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", *Journal of Computational Physics*, Vol. 90, 1990, pp. 161-175.
11. Lin, C. K. Y., Haley, K. B., and Sparks, C., "A Comparative Study of Both Standard and Adaptive Versions of Threshold Accepting and Simulated Annealing Algorithms in Three Scheduling Problems", *European Journal of Operational Research*, Vol. 83, 1995, pp. 330-346.
12. Tarantilis, C, Kiranoudis, C., and Vassiliadis, V., "A Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem", *European Journal of Operation Research*, Vol. 152, 2004, pp. 148-158.
13. Thangiah, R., Potvin, J., and Sun, T., "Heuristic Approach to Vehicle Routing with Backhauls and Time Windows", *Computers and Operations Research*, Vol. 23, No. 11, 1996, pp. 1043-1057.
14. 申生元, 「時窗限制車輛途程問題」, 國立交通大學工業工程與管理研究所博士論文, 民國八十八年。
15. Zhong, Y. and Cole, M., "A Vehicle Routing Problem with Backhauls and Time Windows: A Guided Local Search Solution", *Transportation Research Part E*, Vol. 41, 2005, pp. 131-144.
16. Cho, Y. J. and Wang, S. D., "A Threshold Accepting Meta-Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows", *Journal of the Eastern Asia Society for Transpor-*

- tation Studies*, Vol. 6, 2005, pp. 3022-3037.
17. Ropke, S. and Pisinger, D., “A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls”, *European Journal of Operational Research*, Vol. 171, 2006, pp. 750-775.
 18. Cheung, R. and Hang, D., “Multi-Attribute Label Matching Algorithms for Vehicle Routing Problems with Time Windows and Backhauls”, *IIE Transactions*, Vol. 35, 2003, pp. 191-205.
 19. Reimann, M. and Ulrich, H., “Comparing Backhauling Strategies in Vehicle Routing Using Ant Colony Optimization”, *Central European Journal of Operations Research*, Vol. 14, No. 2, 2006, pp. 105-123.
 20. Solomon, M., “Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms”, Ph.D. Dissertation, University of Pennsylvania, 1983.
 21. Rego, C., “Node-Ejection Chains for the Vehicle Routing Problem: Sequential and Parallel Algorithms”, *Parallel Computing*, Vol. 27, 2001, pp. 201-222.
 22. Osman, I., “Metastrategy Simulation Annealing and Tabu Search Algorithms for the Vehicle Routing and Scheduling Problem with Time Windows”, *European Journal of Operational Research*, Vol. 41, 1993, pp. 421-451.
 23. Hasama, T., Kokubugata, H., and Kawashima, H., “A Heuristic Approach Based on the String Model to Solve Vehicle Routing Problem with Backhauls”, Proceedings of the 5th World Congress on Intelligent Transport Systems (ITS), ITS Korea, 1998.
 24. Tarantilis, C., Kiranoudis, C., and Vassiliadis, V., “A List Based Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem”, *Journal of the Operational Society*, Vol. 54, 2003, pp. 65-71.

