行政院國家科學委員會專題研究計畫　成果報告

回程取貨車輛路線問題之巨集啟發式解法研究

計畫類別：個別型計畫
計畫編號：NSC93-2211-E-216-009-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：中華大學運輸科技與物流管理學系

計畫主持人：卓裕仁
共同主持人：韓復華
計畫參與人員：王生德　朱佑旌

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 94 年 10 月 17 日

行政院國家科學委員會補助專題研究計畫成果報告

回程取貨車輛路線問題之巨集啟發式解法研究

計畫類別：☑ 個別型計畫　　□ 整合型計畫
計畫編號：NSC93－2211－E－216－009－
執行期間：　93 年　8 月　1 日至　94 年　7 月 31 日

計畫主持人：卓裕仁
共同主持人：韓復華
計畫參與人員：王生德、朱佑旌

成果報告類型(依經費核定清單規定繳交)：☑精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：中華大學運輸科技與物流管理系

中　華　民　國　九十四 年　十 月　十五 日

摘要

回程取貨車輛路線問題(Vehicle Routing Problem with Backhauls, VRPB)是車輛路線問題(Vehicle Routing Problem, VRP)的延伸，在物流配送實務上有很高的應用價值。由於 VRPB 屬於解題複雜度很高的 NP-hard 問題，現有文獻所探討的求解方法尚不多見。

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 VRPB 之巨集啟發式解法，並藉由自行產生例題進行測試，以分析此 TA_VRPB 巨集啟發式解法之解題績效。TA_VRPB 方法之執行架構包括：起始解構建模組、鄰域搜尋模組和門檻接受模組。在起始解構建模組方面，本研究提出了一種平行式的鄰近點法；鄰域搜尋模組包括：路線內 2_opt 節線交換、路線間 1_0、1_1 節點交換及 S_S 節線交換等四種方法；門檻接受模組則設計了三種 TA 執行架構（TA1、TA2、TA3）。

本研究經由修改 Solomon 的 VRPTW 國際標竿例題產生了 81 個 VRPB 測試例題，並以 C++語言撰寫執行 TA_VRPB 的電腦程式，然後透過三階段的實驗設計進行例題測試與數值結果分析。結果發現：(1) 本研究提出之平行式鄰近點法比傳統的循序式鄰近點法具有更佳的解題能力；(2) TA2 架構的解題績效優於 TA1 與 TA3 架構；(3) 在目標值方面，平均車輛數從 9.88 降至 6.74，平均路線距離從 923.82 降至 727.44。由上可知，整個門檻接受法確實能提高解題精確度且執行時間短，顯示 TA 在處理回程取貨車輛路線問題上具有不錯的應用潛力。

關鍵字：回程取貨車輛路線問題、巨集啟發式解法、門檻接受法、平行式鄰近點法。

Abstract

The Vehicle Routing Problem with Backhauls (VRPB), which simultaneously considers the operations of delivery and pickup, is a variant of the classical Vehicle Routing Problem (VRP). Successful application of the VRPB to the real-world distribution will improve the performance of logistics.

Due that the computational complexity of VRPB is NP-hard, this paper develops a meta-heuristic approach, which integrates the Threshold Accepting (TA) meta-strategy with the implementation techniques of modified Nearest Neighbor and Exchange procedures, to solve the VRPB. The main framework of the proposed TA_VRPB meta-heuristic consists of initial solution constructor (ISC) module, neighborhood search (NS) module and Threshold Accepting (TA) module.

There are 81 instances, modified from a bank of Solomon's VRPTW benchmark instances, generated for testing and analyzing the performance of the proposed meta-heuristics. Three phases of experiments are designed and related parameters are set. Numerical results imply the following conclusions: (1) proposed Parallel Nearest Neighbor procedure is superior to the traditional Sequential Nearest Neighbor procedure; (2) among three TA frameworks, TA2 experiences better performance than others; and (3) the average fleet size is reduced from 9.88 to 6.74 and the average routing distance is diminished from 923.82 to 727.44. In sum, the proposed TA meta-heuristic actually provides an efficient and robust tool for VRPB applications.

Keywords: Vehicle Routing Problem with Backhauls, Meta-heuristics, Threshold Accepting, Parallel Nearest Neighbor.

一、前言

物流配送實務講求作業績效與資源有效利用。以國內某汽車路線貨運公司新竹營業所之每日營運作業流程為例，在配送與集貨兩項作業中，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（王春分等人, 2000）。這種利用回程時取貨的物流配送作業型態最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。美國自 1982 年以來，食品雜貨業由於對回程空車的利用，增加車輛使用率，估計每年在運輸成本上節省了一千六百億美元（Daniel 等人, 1988）。上述回程取貨型態的物流配送作業即可應用「回程取貨車輛路線問題 (Vehicle Routing Problem with Backhauls, VRPB)」的數學模式來求解，以提升實務業者的作業效率、降低營運成本。

典型的車輛路線問題(Vehicle Routing Problem, VRP)乃是由同一車種、固定容量的車隊，從單一場站出發，服務完一群需求量已知的顧客後返回中心場站；目標在使車輛路線的總運輸成本最小化。而所謂的 VRPB，係屬於 VRP 的一種衍生問題型態。VRPB 將顧客需求點分為兩部分，一為中心場站到顧客的送貨點(Linehauls)，此一部分是將貨物從中心場站運送給顧客；另一為顧客到場站的取貨點(Backhauls)，由客戶需求點收取貨物運回中心場站。VRPB 問題假設車輛必須先服務完所有送貨點顧客，然後才能開始服務取貨點顧客，如此可避免重新整理車廂貨物的排列和浪費時間。

二、研究目的

雖然 VRPB 在實務上有很廣泛的應用，然而現有文獻探討 VRPB 的求解方法尚不多見。由於 VRPB 屬於解題複雜度很高的 NP-hard 問題，其求解所需時間會隨節點數的遞增而呈指數性的成長，因此實務應用時多採用啟發式(heuristics)或巨集啟發式(meta-heuristics)方法進行求解。基於以上認知，本研究之主要目的乃在透過對目前文獻發表的各種 VRPB 求解方法進行回顧與分析，了解 VRPB 的解題特性與關鍵因素，並發展一套績效良好且適合求解 VRPB 的巨集啟發式演算法。

本研究主要內容包括：(1) 蒐集並回顧有關 VRPB 問題與求解方法之文獻，以了解 VRPB 的問題求解特性與關鍵；(2) 以 Solomon（1983）所提出之 VRPTW 國際標竿例題為基礎進行修改，以建立 VRPB 測試題庫；(3) 結合改良式的鄰近點(Nearest Neighbor)法、鄰域搜尋(Neighborhood Search)法、門檻接受(Threshold Accepting, TA)法，設計求解 VRPB 之巨集啟發式解法架構；(4) 以 C++語言撰寫上述巨集啟發式解法之電腦程式，並進行例題測試，以瞭解該巨集啟發式方法應用於 VRPB 的解題特性與執行績效。

三、文獻探討

國內外有關 VRPB 各種求解方法之文獻並不多，茲將其重點彙整於表 1。由表 1 可知，VRPB 求解方法主要可分成數學規劃法與啟發式方法兩部分，但以啟發式方法為主要趨勢，

近年來並朝向使用巨集啟發式方法之應用。此外，為測試各種求解方法之績效，學者多以 VRPTW 國際標竿題庫為基礎，設計出不同的 VRPB 測試例題。而從各文獻方法的測試結果可發現：VRPB 的求解品質受送貨點顧客與取貨點顧客間的相對位置關係影響甚鉅。

表 1、VRPB 求解方法彙整

年代	作者	使用方法	限制條件	測試例題
1984	Deif & Bodin	。節省法	。車容量限制 。路線長度限制	。取貨比例： 10%、20%、50%
1989	Goetschalckx & Jacobs-Blecha	。空間填滿曲線法 。K-中位法 。貪心法	。車容量限制	。取貨比例： 25%、50%
1997	Toth & Vigo	。整數線性規劃模式 。拉氏下界	。車容量限制	。取貨比例： 20%、34%、50%
1999	Toth & Vigo	。先分群再排路線 。拉式鬆弛法	。車容量限制	。取貨比例： 20%、34%、50%
2000	Osman & Wassan	。節省插入法 。節省指派法 。禁制搜尋法	。車容量限制	。取貨比例： 25%、50%； 20%、34%、50%
2002	Wade & Salhi	。貪心插入法	。車容量限制	。取貨比例： 25%、50%； 20%、34%、50%

四、研究方法

(一) 巨集啟發式解法架構

本研究採用門檻接受法(TA)結合改良式的鄰近點法與傳統鄰域搜尋法，構建一套巨集啟發式解法，並將之命名為 TA_VRPB。整個 TA_VRPB 的設計理念在於藉由 TA 可接受劣於現解之鄰解的機制，以便使搜尋過程能夠脫離局部最佳解的束縛，且 TA 採用確定性的接受法則，執行方式較為簡單。

整套 TA_VRPB 解題架構係依循 GIDS 法的概念，分為三個程序：第一個程序為起始解構建模組(ISC module)，第二個程序為強調深度搜尋的鄰域搜尋模組(NS module)，第三個程序為強調包容與廣度搜尋的門檻接受模組(TA module)。如圖 1 所示，起始解構建程序根據傳統的循序式鄰近點法(Sequential Nearest Neighbor, SNN)，提出改良的平行式鄰近點法(Parallel NN, PNN)以產生起始可行解；接著進行鄰域搜尋改善程序，分別以路線內交換改善模組和路線間交換改善模組，以搜尋局部最佳解。其中，1_0_b、1_1_b、S_S_b 分別代表各交換法的交換策略為最佳改善(best-improvement)策略，1_0_f、1_1_f、S_S_f 代表各交換法的交換策略為首先改善(first-improvement)策略；最後，門檻接受程序則針對鄰域搜尋模組構建的結果，使用 TA 演算法進行搜尋以跳脫局部最佳解的束縛，本研究共提出 TA1、TA2、TA3 三種執行方式。

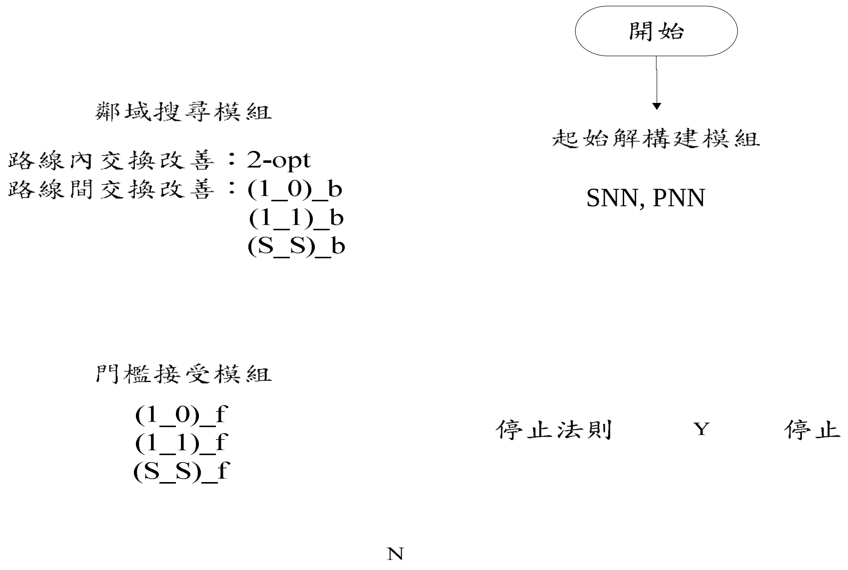

圖 1、TA_VRPB 解題架構

(1) 起始解構建模組

以往求解 VRP 相關問題的文獻多使用節省法來構建起始解，並以路線成本節省為主要考量。本研究認為傳統的鄰近點法(Nearest Neighbor, NN)僅以距離為考慮因素，較節省法容易執行且解題速度快；然而傳統鄰近點法採取循序的方式來構建路線，求得解的精確度較差。有鑑於此，本研究提出了改良式的鄰近點法來做為起始解的構建方法，將節省法中採用的平行式構建路線機制納入 NN 法中，稱之為平行式鄰近點法(PNN)。所謂循序的概念意指：一條路線構建完畢之後再構建另一條新的路線，直到所有需求點皆已納入所有路線為止；而平行的概念意指：同時構建多條路線，直到所有需求點皆已納入所有路線內為止。

(2) 鄰域搜尋模組

起始解模組將起始解構建完成後，接著進入鄰域搜尋模組，除了採用路線內 2_opt 節線交換、路線間 1_0 節點交換、1_1 節點交換等方法外，更依據先送貨後取貨的特質提出 S_S(segment-segment)整段交換法。其中，1_0 節點交換模組同時具備減少車輛數和路線成本的功能；其餘交換模組主要功能皆為降低路線成本。

本研究路線內改善模組採用 2_opt 交換法，交換的策略採用首先改善(first-improve)策略。在車容量、時間窗及先送貨後取貨的限制下，經由路線內節線交換機制，以達成縮短總路線成本的目標；在路線間交換的階段，採用 1_0 節點交換法、1_1 節點交換法；和本研究提出的 S_S 整段交換法。交換的策略則採用最佳改善(best-improve)策略；在車輛容量及先送貨後貨的限制下，經由不同路線間需求點交換的機制，以達成縮短總路線成本的目標。

(3) 門檻接受模組

本研究在門檻接受模組，以前述之路線間交換法為核心工具，但為節省運算時間，皆採

用首先改善之執行方式且不考慮路線內 2_opt 交換法。在控制參數方面，包括起始門檻值(T_0)、門檻數列長度(K)及門檻數列收斂型態。門檻數列在門檻數列長度(K)期間，自起始門檻值(T_0)逐漸收斂到 0，本研究門檻數列收斂型態採用線性遞減門檻數列。為了探討 TA 的改善效率，本研究設計了三種執行模組，TA1、TA2 的執行過程為串聯的模式，TA3 的執行過程類似並聯的概念。

(二) 測試例題產生與實驗設計

由文獻回顧可知，大多數的文獻皆根據 Solomon (1983)所建立的 VRPTW 測試題庫為基礎，先不考慮時間窗之參數限制，然後將需求點分成送貨點顧客及取貨點顧客兩部分，並設定取貨點顧客的百分比（例如：10%、30%、50%），以產生 VRPB 測試例題。設定取貨點顧客百分比之目的乃在於分析取貨點顧客比例對求解績效之影響；至於取貨點顧客的選取，則採隨機選取方式。

Solomon 的 VRPTW 測試例題共有 56 題，依各需求點的散佈型態分成：均勻隨機(R)、聚落(C)、混合(RC)三類；各類之下又分成兩種型態：車輛容量小、路線最大長度短，車輛容量大、路線最大長度長。因此，共分成 R1、R2、C1、C2、RC1 及 RC2 等六小類，各小類下有若干例題，其需求點座標不變，僅各需求點的時間窗有所不同。

本研究亦以 Solomon 的題庫為基礎，挑選 R101、C101、RC101 三個例題，先不考慮時間窗之參數限制，並分別選擇前 25、50 及 100 個顧客；然後設定取貨點顧客的百分比為 10%、30%、50%，並採隨機方式選取取貨點顧客，將需求點分成送貨點顧客及取貨點顧客兩部分；每種情境各產生三題，總共產生了 81 個測試例題。

本研究針對方法與模組之組合方式、控制參數之設定，共設計了三個階段的測試實驗。實驗一首先針對兩種起始解法(SNN 與 PNN)進行測試，以了解改良式方法之適用性與解題效果；實驗二根據三種路線間交換法（1_0、1_1、S_S）之特性，設計了兩種鄰域搜尋組合並配合 PNN 起始解方法，來測試三種 TA 執行方式(TA1、TA2、TA3)的解題績效；實驗三則針對不同的 TA 控制參數數值範圍進行其敏感度分析。

在 TA 的控制參數方面，有兩個控制參數：起始門檻值(T_0)與執行次數(K)，並據以構成一個門檻數列長度{T_k}。三種 TA 執行方式與參數設定範圍分別為：T_0 = 0.01、0.05、0.1、0.2、0.3 與 0.5；TA1 和 TA3 之 K 設為 10，TA2 設為 5。本研究採用的是直線遞減型的門檻數列，其門檻值比率自 T_0 逐漸下降至 0，並根據 VRPB 之問題特性，提出下列的門檻數列計算公式：

$$\{T_k \mid T_k = \frac{C(X_0)}{(N+R)} \times T_0 \times (\frac{K-k}{K-1}),\ \ for\ \ k = 1 \sim K\} \qquad (1)$$

解題績效之分析包括兩項指標：(1) 解題精度：第一目標為車輛數，第二目標為路線成本；一般常使用與「真確最佳解」或「已知最佳解」之成本誤差(%)為比較基準；但本研究所產生的 VRPB 例題，並無真確最佳解或已知最佳解，故僅能比較不同階段的改善情形。因此，本研究將從所有求得的解當中挑選出暫優解當作比較基準；(2)解題速度：一般使用電腦運算所需的 CPU 時間。由於各電腦的系統、配備、並不相同，所使用的程式語言亦不一致，因此 CPU 時間僅提供為評估時之參考指標。

五、結果與討論

(一) 測試結果分析

(1) 起始解建構模組之測試結果

表 2 為 SNN 與 PNN 兩種方法在所有 81 個測試例題的平均結果比較。由表 2 可知，SNN 之平均車輛數為 7.86 輛，平均路線成本為 1075.65；PNN 所求得的平均車輛數為 9.88 輛，平均路線成本為 923.82。此測試結果與以往文獻報導關於循序式節省法及平行式節省法之比較相似，即循序式方法在車輛數目標上表現優於平行式方法，而在路線成本目標方面則以平行式方法表現較佳。

表 2、起始解建構模組測試結果彙整

指標 / 方法	平均車輛數	平均路線成本
SNN	7.86	1075.65
PNN	9.88	923.82
總平均	8.87	999.73

(2) 門檻接受模組之測試結果

由於 1_0、1_1、S_S 三種交換法中，僅 1_0 具有可以減少車輛數的功能，因此本研究設計了兩種鄰域搜尋的執行順序：N1(1_1→S_S→1_0)與 N2(1_0→1_1→S_S)。N1 表示先執行路線成本改善再執行車輛數改善，而 N2 則表示先執行車輛數改善再執行路線成本改善。門檻接受模組之測試結果彙整如表 3 所示，其中，改善幅度(%)係以表 2 的 PNN 測試結果平均值（車輛數 9.88、路線成本 923.82）為基礎進行計算；三種 TA 執行架構的起始門檻值 T_0 皆設定為 0.1，TA1 和 TA3 之 K 設為 10，TA2 設為 5。

表 3、門檻接受模組測試結果彙整

門檻接受模組		PNN_N1		PNN_N2		總平均	
		目標值	改善(%)	目標值	改善(%)	目標值	改善(%)
TA1	車輛數	6.74	35.86	8.86	10.34	7.99	19.12
	路線成本	727.44	21.26	878.23	4.93	802.83	13.10
TA2	車輛數	8.53	13.70	8.83	10.61	8.68	12.15
	路線成本	866.84	6.17	876.77	5.09	871.80	5.63
TA3	車輛數	8.78	11.17	8.86	10.34	8.82	10.76
	路線成本	877.65	5.00	878.11	4.95	877.88	4.97
總平均	車輛數	7.88	20.25	8.85	10.43	8.36	15.34
	路線成本	834.95	9.62	877.70	4.99	856.33	7.31

從表 3 之結果可發現：(1) 三種 TA 執行架構中，以 TA1 的解題表現最佳，平均車輛數為 7.99、平均路線成本為 802.83，皆優於其他兩種架構；(2) 兩種鄰域搜尋組合中，以 N1 組合的解題表現較佳，平均車輛數為 7.88、平均路線成本為 834.95，皆優於 N2 組合；(3) 所有模組組合中，以 TA1 搭配 PNN_N1 的解題表現最佳，平均車輛數為 6.74，改善幅度為 35.86%，平均路線成本為 727.44，改善幅度為 21.26%，績效特別顯著；(4) 就整體績效而言，總平均車輛數自 9.88 降至 8.36，改善幅度為 15.34%，總平均路線成本自 923.82 降至 856.33，改善幅度為 7.31%。

(3) TA 控制參數之測試結果

TA 的參數設定包括起始門檻值(T_0)與門檻數列長度(K)，表 4 針對 T_0 設定值之測試結果進行彙整。從表 4 可知：當 T_0 值為 0.01 至 0.50 時，其解題績效並無任何改變，顯示 TA_VRPB 的方法架構對控制參數並不敏感，這也顯示 TA_VRPB 是一個穩定的方法。

表 4、起始門檻(T_0)參數測試結果彙整

T_0	0.01	0.05	0.10	0.20	0.30	0.50
車輛數	6.74	6.74	6.74	6.74	6.74	6.74
路線成本	727.44	727.44	727.44	727.44	727.44	727.44

(二) 結論與建議

本研究嘗試將門檻接受法應用於求解 VRPB，整套啟發式方法包括起始解構建、鄰域搜尋與門檻接受三大模組。起始解以傳統循序式鄰近點法(SNN)為基礎，據以提出改良的平行式鄰近點法(PNN)；鄰域搜尋包括，2-opt 節線交換法、1_0 節線交換法、1_1 節線交換法及 S_S 整段交換法，其中 S_S 為本研究所設計之新方法；最後再搭配門檻接受法設計三種不同的執行模式加強搜尋改善效果。為檢視本研究提出 TA_VRPB 之解題績效，透過修改 Solomon VRPTW 測試題庫產生了 81 個 VRPB 例題並進行求解測試，所得結果彙整如下：

1. 本研究提出之平行式鄰近點法(PNN)在路線成本目標方面表現優於傳統的循序式鄰近點法(SNN)，但是在車輛數目標方面則不及 SNN；此結果顯示這兩種方法可能具有很高的互補性。
2. 在鄰域搜尋改善方面，1_0 節點交換法同時具備減少車輛數和降低路線成本的功能；鄰域搜尋方法的組合，則以先執行 2-opt、1_1、S_S 等僅能改善路線成本的方法後，最後再執行 1_0 以改善車輛數的順序(N1)為佳。
3. 三種 TA 執行架構中，以 TA1 的改善效果最佳。所有模組組合則以 TA1 搭配 PNN_N1 的表現最佳，平均車輛數 6.74(改善幅度 35.86%)，平均路線成本 727.44(改善幅度 21.26%)。
4. TA_VRPB 的解題績效不受起始門檻值(T_0)的數值範圍影響，由此可知，TA 是一個有效、穩健(Robust)的巨集啟發式方法。

後續的研究方向可考慮：

1. 由於 SNN 與 PNN 具有互補的效果，可考慮利用權重的方式將兩種方法整合成一個起始解構建法，以擷取其優點。

2. 在車輛數目標的改善方面，本研究只應用了 1_0 節點交換法，略嫌不足，後續研究可參考多車種車輛路線問題的求解方法，設計其他能改善車輛數的方法，以增進整體求解績效。

3. 本研究僅應用門檻接受法(TA)求解 VRPB，未來可嘗試結合其他的巨集啟發式方法，例如：禁制搜尋法(Tabu Search, TS)或螞蟻演算法(Ant Colony System, ACS)。

4. 可嘗試將本研究發展的平行式鄰近點法與 TA_VRPB 架構應用在其他複雜的車輛路線問題，例如：時窗限制回程取貨車輛路線問題(VRPBTW)、同時收送貨車輛路線問題(PDVRP)等。

參考文獻

王春分、吳柏嫺、林伶芝(2000)，**市區內零擔貨物運送路線問題之研究 - 以新竹貨運在新竹市區之配送為例**，畢業專題報告，中華大學交通與物流管理學系畢業專題報告。

申生元(1999)，**時窗限制車輛途程問題**，交通大學工業工程與管理研究所博士論文。

卓裕仁(2001)，**以巨集啟發式方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所博士論文。

Christofides, N. & S. Eilon (1969), "An Algorithms for Vehicle Dispatching Problem," *Operations Research Quarterly*, Vol.20, pp.568-589.

Daniel, O., B. Golden & A. Edward (1988), "Vehicle Routing with Backhauls: Models, Algorithms, and Case Studies," in B. Golden and A. Assad (eds.), *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam, pp.127-147.

Deif, I. & L. Bodin (1984), "Extension of the Clarke-Wright algorithms for solving the Vehicle Routing with Backhauling," in A. Kidder (ed.), *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, MA, pp.75-96.

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.

Dumamel, C., J. Potvin, J.Y. & J. Rousseau (1997), "A tabu search heuristic for the vehicle routing problem with backhauls and time windows," *Transportation Science*, Vol. 31, pp.49-59.

Gélinas, S., M. Desrochers, J. Desrosiers & M. Solomon (1995), "A new branching strategy for time constrained routing problems with application to backhauling," *Annals of Operations Research*, Vol. 61, pp.91-109.

Goetschalckx, M. & C. Jocabs-Blencha (1989), "The Vehicle Routing Problem with Backhauls," *European Journal of Operational Research*, Vol.42, pp.39-51.

Kontoravdis, G. & J. Bard (1995), "A GRASP for the vehicle routing problem with time windows," *ORSA Journal of Computing*, Vol. 7, pp.10-23.

Lin, S. (1965), "Computer Solution of the Traveling Salesman Problem," *Bell System Technology Journal*, Vol.11, pp.2245-2269.

Osman, I. (1993), "Metastrategy Simulation Annealing and Tabu Search Algorithms for the Vehicle Routing and Scheduling Problem with Time Windows," *European Journal of Operational Research*, Vol.41, pp.421-451.

Potvin J., C. Duhamel & F. Guertin (1996), "A Genetic Algorithms for Vehicle Routing Problem with Backhauling," *Applied Intelligence*, Vol.6, pp.345-355.

Solomon, M. (1983), *Vehicle routing and Scheduling with Time Window Constraints: Models and Algorithms*, Ph.D. Dissertation, University of PeSNNylvania.

Thangiah, R., J. Potvin & T. Sun (1996), "Heuristic Approach to Vehicle Routing with Backhauls and Time Windows," *Computers and Operations Research*, Vol. 23, No. 11, pp.1043-1057.

Toth, P. & D. Vigo (1997), "An Exact Algorithms for the Vehicle Routing Problem with Backhauls," *Transportation Science*, Vol.31, No.4, pp.372-385.

Toth, P. & D. Vigo (1999), "A Heuristic Algorithm for the Symmetric and Asymmetric Vehicle Routing Problem with Backhauls," *European Journal of Operational Research*, Vol.113, pp.528-543.

Wade, A. & S. Salhi (2002), "An investigation into a new class of vehicle routing problem with backhauls," *Omega*, Vol.30, pp.479-487.

Yano, T., L. Chan & K. Richter (1987), "Vehicle Routing at Quality Stores," *Interfaces*, Vol.17, No.2, pp.52-63.

計畫成果自評

一、本計畫之研究內容及項目，皆與原計畫書之內容一致，相符程度達 100%，並能達成預期目標。

	原計畫書內容	成果報告內容
研究目的	發展一套績效良好且適合求解 VRPB 的巨集啟發式演算法，並透過對目前文獻發表的各種 VRPB 求解方法進行回顧與分析，了解 VRPB 的解題特性與關鍵因素	透過對目前文獻發表的各種 VRPB 求解方法進行回顧與分析，了解 VRPB 的解題特性與關鍵因素，並發展一套績效良好且適合求解 VRPB 的巨集啟發式演算法
研究內容	1. 蒐集國內外文獻中已發表的 VRPB 例題或修改其他 VRP 例題，以建立 VRPB 測試題庫 2. 結合多起始解建構、深度化包容搜尋、廣度化擾動搜尋等策略群組，設計求解 VRPB 之巨集啟發式解法架構 3. 以 C 語言在電腦上撰寫演算法程式，並進行例題測試，藉以瞭解巨集啟發式方法應用於 VRPB 測試例題的解題特性與執行績效	1. 蒐集並回顧有關 VRPB 問題與求解方法之文獻，以了解 VRPB 的問題求解特性與關鍵 2. 以 Solomon（1983）所提出之 VRPTW 國際標竿例題為基礎進行修改，以建立 VRPB 測試題庫 3. 結合改良式的鄰近點(Nearest Neighbor)法、鄰域搜尋(Neighborhood Search)法、門檻接受(Threshold Accepting, TA)法，設計求解 VRPB 之巨集啟發式解法架構 4. 以 C++語言撰寫上述巨集啟發式解法之電腦程式，並進行例題測試，以瞭解該巨集啟發式方法應用於 VRPB 的解題特性與執行績效
研究成果	1. 完成相關文獻之蒐集與回顧，整理出各種巨集啟發式方法所運用到的巨集策略，並了解各巨集策略的解題成效與適用情形，以強化 GIDS 方法的解題能力 2. 完成 VRPB 問題之特性與解題關鍵因素分析，並建立對應的巨集策略與求解機制 3. 完成 VRPB 測試例題建立與實驗設計，並進行系統化 GIDS 應用測試，分析其解題績效 4. 完成期末報告之撰寫，並將研究成果發表於國際或國內學術期刊上	1. 完成相關文獻之蒐集與回顧（詳見本報告第三節），並發表一篇相關論文（詳見計畫成果自評第二項） 2. 完成並建立對應 VRPB 問題特性的巨集策略與 TA_VRPB 求解機制（詳見本報告第四節），並據以強化 GIDS 方法的解題能力 3. 完成 VRPB 測試例題建立與實驗設計，並進行系統化 GIDS 應用測試，分析其解題績效（詳見本報告第五節） 4. 完成期末報告撰寫，並已發表研究成果於國際學術期刊上（詳見計畫成果自評第二項）

二、研究成果之學術或應用價值

本計畫的研究成果將有助於國內從事巨集啟發式方法或車輛路線相關問題求解之研究人員參考，進而提升相關領域在理論上的解題績效。在實務的應用價值方面，國內的公路路線貨運業者（大榮貨運、新竹貨運），快遞業者（宅急便、統一速捷），甚至郵局的包裹與快遞業務，皆可應用本研究發展的回程取貨車輛路線問題模式與求解方法，來改善物流配送作業之效率、降低物流成本。

對於參與本研究之人員與研究生，可建立其在相關領域的專業知識與技能，並藉由研究成果的發表與交流，達到知識分享之目的。如下所列，本計畫相關之成果已發表了三篇學術論文，並有一篇正在投稿中；本計畫的經費與研究內容也協助了一位研究生完成碩士論文，以及一位正在進行碩士論文的研究生。

(1) 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 Transportation Studies*, Vol.6, pp.3022-3037, 2005.（全文詳見附件）

(2) 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」，第一屆台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，國立台北科技大學(台北)，作業研究與管理科學類，188 至 194 頁(光碟)，民國 93 年 11 月。（全文詳見附件）

(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十八屆學術研討會，國立交通大學(新竹)，運輸網路類，編號 1413(光碟)，民國 92 年 12 月。

(4) 卓裕仁、王生德、朱佑旌，「以巨集啟發式方法求解回程取貨車輛相關路線問題之研究」，運輸計畫季刊（投稿中）。

(5) 王生德，以巨集啟發式方法求解時窗限制回程取貨車輛路線問題(VRPBTW)之研究，中華大學科技管理研究所運輸科技與物流管理組碩士論文，民國 93 年 6 月。

(6) 朱佑旌，巨集啟發式解法應用於時窗限制多車種回程取貨車輛路線問題之研究，中華大學運輸科技與物流管理學系暨研究所碩士論文（進行中）。

三、本計畫的研究成果相當適合在學術期刊上發表，但是因尚未開發完整的電腦輔助決策系統與使用者介面，故不適合申請專利。演算法方面已於學術期刊上發表，屬於個人智慧財產權，歡迎引用。

四、本研究主要發現車輛路線相關問題的種類相當多，而且隨著限制條件的不同，必須設計符合其問題特性的求解方法。以本研究的對象 VRPB 而言，由於有「先送貨後取貨」的順序限制，因此傳統的路線內 2-opt 節線交換即無法發揮效用。此外，若再加上「時間窗」限制的話，則其鄰域搜尋改善更加困難。這些問題皆需要再投入人力與時間進行更深入與廣泛的研究，也是本人後續的重要研究課題。

可供推廣之研發成果資料表

□ 可申請專利　☑ 可技術移轉　　日期：94 年 10 月 15 日

國科會補助計畫	計畫名稱：回程取貨車輛路線問題之巨集啟發式解法研究 計畫主持人：卓裕仁 計畫編號：NSC-93-2211-E-216-009　學門領域：土木工程(交通)
技術/創作名稱	TA_VRPB 巨集啟發式方法
發明人/創作人	卓裕仁
技術說明	中文： TA_VRPB 為一可用於求解 VRPB 問題之巨集啟發式方法，共包含三個求解程序：起始解構建(ISC)模組、鄰域搜尋(NS)模組、門檻接受(TA)模組。
	英文：
可利用之產業及可開發之產品	1. 可利用之產業：公路路線貨運業、快遞業者、郵局及其他以配送為主的物流業者。 2. 可開發之產品：物流配送車輛路線派遣之決策支援系統
技術特點	TA_VRPB 採用巨集啟發式方法技術，可以在很快的時間內求得精確度不錯的近似解。
推廣及運用的價值	可提升業者車輛路線派遣之績效，並有效減少運輸成本。

附件：已發表論文二篇

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

Yuh-Jen CHO
Assistant Professor
Department of Transportation Technology and Logistics Management
Chung Hua University
707, Sec. 2, Wu Fu Rd., Hsin Chu,
300, Taiwan, R.O.C.
Fax: +886-3-5186545
E-mail: yjcho@chu.edu.tw

Sheng-Der WANG
Master (In Army)
Institute of Technology Management
Chung Hua University
707, Sec. 2, Wu Fu Rd., Hsin Chu,
300, Taiwan, R.O.C.
Fax: +886-3-5186545
E-mail: m9103066@chu.edu.tw

Abstract: This paper presents a meta-heuristic, which is based on the Threshold Accepting combined with modified Nearest Neighbor and Exchange procedures, to solve the Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW). The VRPBTW assumes that trucks initially start from the depot, deliver goods to the linehaul customers, successively pickup goods from the backhaul consumers, and finally return to depot. Eighty-one instances are generated to identify the performance of the proposed meta-heuristic. Four experiments are designed and related parameters are set. Numerical results imply the following conclusions: (1) modified Nearest Neighbor procedures are superior to the traditional Nearest Neighbor; (2) among three TA frameworks, TA2 experiences better performance than others; and (3) the average fleet size is reduced from 18.54 to 15.18 and the average routing time is diminished from 1770.83 to 1267.09. In sum, the proposed TA meta-heuristic actually provides an efficient and robust tool for VRPBTW applications.

Key Words: meta-heuristic, threshold accepting, vehicle routing problem, backhauls, time windows

1. INTRODUCTION

The Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW), which simultaneously considers the operational sequence of delivery and pickup as well as the service time interval of customers, is an extensive variant of the classical Vehicle Routing Problem (VRP). The VRPBTW involves two different subsets of customers known as linehauls and backhauls. A linehaul customer requires a given quantity of goods from a central depot, whereas a given quantity of goods is collected from a backhaul customer and brought back to the depot. Moreover, the backhauls must be visited after the linehauls in each route. Additionally, each customer must be serviced within a specified time interval (or time window). The lower and upper bounds of the time window define the earliest and latest time for the beginning of service at the customer. Hence, a vehicle is not allowed to begin service at a customer location after its time window's upper bound. Furthermore, a waiting time is incurred if a vehicle reaches a customer before the lower bound. Each customer also has a specified maneuvering time which is the time spent by the vehicle to load or unload the goods. The total route time of a vehicle is the sum of travel time (which is proportional to the distance traveled), waiting time and maneuvering time. The first objective of the VRPBTW is to minimize the fleet size (i.e., number of vehicles). Then, for the same fleet size, minimize the total routing time to service all customers, without violating the capacity of each vehicle and the time windows of each customer.

The backhauling operation of trucks comes from practical considerations. Since most vehicles are rear-loaded, it is often inconvenient to rearrange onboard delivery loads to accommodate new pick-up loads. This precedence constraint is frequently adopted in order to reduce empty travel and to save cost for the LTL (Less than Truck-Load) transport carriers and home delivery companies. Examples of vehicle routing problems with backhauls are mostly found in land-based distribution systems (e.g., grocery industry, where linehaul customers are supermarkets and backhaul customers are grocery suppliers). Successful application of the VRPBTW to the real-world cases will improve the performance of distribution. For example, Yano *et al*. (1987) describe an interesting application for Quality Stores, a Michigan-based retail chain of approximately 40 stores with one centralized distribution center located near Toledo, OH. Quality Stores owns or leases 11 trucks, which are nearly identical with respect to weight and volume of shipments. This fleet is used to deliver goods to stores and pickup goods from nearby suppliers. Quality Stores has found that the savings realized from using its fleet rather than common carriers to backhaul goods from suppliers located near its stores helped it to save US$450,000 in 1986.

Although many different heuristic and exact algorithms may be found in the literature for vehicle routing problems with backhauls (VRPB) or for vehicle routing problems with time windows (VRPTW), only a few recent papers are devoted to the VRPBTW. Due that the computational complexity of VRPBTW is NP-hard (Thangiah *et al.*, 1996), efficient heuristic approaches are eagerly necessary for solving the VRPBTW. In this article, we present a meta-heuristic approach, which integrates the Threshold Accepting (TA) meta-strategy with the implementation techniques of traditional local search algorithms, to solve the VRPBTW. This paper is organized as follows. Section 2 presents a mathematical definition and formulation of the VRPBTW, then, surveys the existing solution methods of the VRPBTW and the concepts of the TA approach. Section 3 describes the details of TA meta-heuristic for the VRPBTW. Computational results are reported in Section 4 on 81 newly created VRPBTW instances with 100 customers. Finally, Section 5 concludes the paper.

2. PROBLE DEFINITION AND LITERATURE REVIEW

2.1 Mathematical Formulation of VRPBTW

Let $G = (N, A)$ be a complete undirected graph with node set N and arc set $A = \{(i, j)| i, j \in \mathrm{N}\}$. The node set is partitioned into $N = \{\{0\}, L, B\}$, where 0 is the depot, L is the set of linehaul customers, $L = \{1, \ldots, n\}$, and B is the set of backhaul customers, $B = \{n+1, \ldots, n+m\}$. Each node $i \in N$ is associated with: (1) a non negative quantity a_i or b_i of goods to be delivered or picked up; (2) a time window $[e_i, l_i]$, where e_i and l_i are the lower and upper bound of the time window, respectively (and where e_0 is the earliest start time of each vehicle from the depot and l_0 the latest return time); and (3) a maneuvering time s_j for loading or unloading the goods (with $s_0 = 0$). Finally, a symmetric traveling time matrix $T = [t_{ij}]$ satisfying the triangle inequality is defined on E.

Given a fleet of identical vehicles, $V = \{1, \ldots, v\}$, each with capacity d, the VRPBTW then consists of finding a set of vehicle routes with minimal costs (fleet size and routing time), originating from and terminating at the depot, such that: (1) each vehicle services one route; (2) each customer node i is visited exactly once; (3) the quantity of goods on-board never exceeds the vehicle capacity; (4) the linehaul customers precede the backhauls on each vehicle route; (5) the start time on each vehicle route is greater than or equal to the time window's lower bound at depot; (6) the return time on each vehicle route is less than or equal to the time window's upper bound at depot; (7) the time of beginning of service at each node *i* is less than or equal to the time window's upper bound l_i; and (8) the time of beginning of service at each vertex *i* is greater than or equal to the time window's lower bound e_i. The mathematical formulation of VRPBTW, which is modified from Goetschalckx and Jacobs-Belcha (1989), can be described as follows, where x_{ijk} = 1 if arc (i, j) is traversed by vehicle *k*, 0 otherwise; u_{ik} = 1 if linehaul customer *i* is served by vehicle *k*, 0 otherwise; v_{ik} = 1 if backhaul customer *i* is served by vehicle *k*, 0 otherwise; t_i = real variable to denote the time starting to serve customer *i*; and *T* is a extremely large value.

Minimize: $$\sum_{j=1}^{n+m} \sum_{k=1}^{v} x_{0jk} \tag{1}$$

Minimize: $$\sum_{i=0}^{n+m} \sum_{j=0}^{n+m} \sum_{k=1}^{v} t_{ij} \cdot x_{ijk} \tag{2}$$

Subject to $$\sum_{k=1}^{v} u_{ik} = 1 \qquad i = 1 \sim n \tag{3}$$

$$\sum_{k=1}^{v} v_{ik} = 1 \qquad i = n+1 \sim n+m \tag{4}$$

$$\sum_{i=1}^{n} a_i \cdot u_{ik} \le d \qquad k = 1 \sim v \tag{5}$$

$$\sum_{i=n+1}^{n+m} b_i \cdot v_{ik} \le d \qquad k = 1 \sim v \tag{6}$$

$$\sum_{i=0}^{n+m} x_{ijk} = \begin{cases} u_{jk} & if\ j = 1 \sim n \\ v_{jk} & if\ j = 0,\ n+1 \sim n+m \end{cases} ;\ k = 1 \sim v \tag{7}$$

$$\sum_{j=0}^{n+m} x_{ijk} = \begin{cases} u_{ik} & if\ i = 0 \sim n \\ v_{ik} & if\ i = n+1 \sim n+m \end{cases} \qquad ;\ k = 1 \sim v \tag{8}$$

$$\sum_{i=0}^{n} \sum_{j=0,n+1}^{n+m} x_{ijk} = 1 \qquad k = 1 \sim v \tag{9}$$

$$t_j \ge t_i + s_i + t_{ij} - (1 - x_{ijk})T \qquad i, j = 1 \sim n+m;\ k = 1 \sim v \tag{10}$$

$$e_i \le t_i \le l_i \qquad i = 1 \sim n+m \tag{11}$$

$$u_{0k} = 1; u_{ik} = 0\ or\ 1 \qquad i = 0 \sim n;\ k = 1 \sim v \tag{12}$$

$$v_{0k} = 1; v_{ik} = 0\ or\ 1 \qquad i = 0,\ n+1 \sim n+m;\ k = 1 \sim v \tag{13}$$

$$x_{ijk} = 0\ or\ 1;\quad t_i \ge 0 \qquad i, j = 0 \sim n+m;\ k = 1 \sim v \tag{14}$$

Equations (1) and (2) are objective functions which minimize the fleet size and the routing time respectively. Constraint sets (3) and (4) separately state that each linehaul customer and each backhaul customer must be served by exactly one vehicle. Constraint sets (5) and (6) indict that the aggregated demand for delivery and pick-up in vehicle k could not exceed the capacity of vehicle. Constraint sets (7) and (8) commonly describe the relationships of flow conservation. Constraint set (9) states that precedence of linehaul and backhaul in each vehicle k, where backhaul customers must be served after linehaul customers. Constraint set (10), which calculates the earliest starting time to serve customer j, guarantees that sub-tour could be breaking. Equation set (11) is the time window constraint of serving customer i. Constraint sets (12), (13) and (14) define the domain of decision variables respectively.

2.2 Existing Methods for VRPBTW

In this sub-section, we review the exact algorithm, heuristics, and meta-heuristics for solving the VRPBTW. Gélinas *et al.* (1995) proposed an exact algorithm, based on a column generation technique for solving a set partitioning formulation of the VRPBTW. This algorithm found optimal solutions to different problems, with up to 100 customers, derived from those found in Solomon’s (1983) VRPTW test set.

A parallel insertion heuristic, Push-Forward Insertion Heuristic (PFIH), for the VRPBTW is proposed in Kontoravdis and Bard (1995). This heuristic uses an efficient method for inserting customers into the routes, and was applied to problems with up to 100 customers and 3 vehicles. Thangiah *et al.* (1996) propose a heuristic based on the insertion procedure of Kontoravdis and Bard (1995) to obtain initial solutions to the VRPBTW. Then, the initial solutions are improved through the application of λ-interchanges and 2-opt* exchanges, which previously developed for the VRPTW only. This two-phase heuristic was used to solve problems of size 25, 50 and 100 (Gélinas *et al.*, 1995), for which the optimal solution is known in most cases. In addition, the heuristic was tested on 24 newly created problems of size 250 and 500.

Potvin *et al.* (1996) present a Genetic Algorithm (GA) which is combined with a greedy route construction heuristic. The greedy heuristic inserts the customers one by one into the routes, using a fixed ordering of customers. The genetic algorithm is used to find good orderings for the insertion heuristic. In Duhamel and Potvin (1997), a Tabu Search (TS) meta-heuristic is proposed to solve the VRPBTW. The TS includes a greedy insertion heuristic from Kontoravdis and Bard (1995) for constructing an initial feasible solution, and a tabu search procedure based on three different neighborhood search heuristics: extended 2-opt, Or-opt and Swap. Shen (1999) proposed a two-stage meta-heuristic, RNETS, based on a route neighborhood structure to solve the VRPBTW. In the first stage, a nested parallel method is used to construct initial feasible routes from the lower bound direction. Then, those routes are improved by local search in the stage II.

Table 1 summarizes the previous works. Most researchers adopt the insertion-based heuristics to construct feasible routes firstly. Several methods execute the local search heuristics to improve the initial solution. Meta-heuristics, such as Tabu Search (TS) and Genetic Algorithm (GA), present good performance on solving the VRPBTW. On the other hand, the testing instances almost are modified from the Solomon’s VRPTW benchmark instances (Solomon, 1987). The size of the instances with known exact solutions is up to 100 customers.

2.3 TA Meta-heuristic

The Threshold Accepting (TA), a variant of the well-known Simulated Annealing (SA), was introduced by Dueck and Scheurer (1990). Both TA and SA belong to the generic search methods which would allow the overall search move through some bad solutions when bypass the trap of a local optimum. The essential difference between TA and SA consists of the different acceptance rules. TA adopts a deterministic acceptance criterion to choose a neighbor solution and does not require the generation of random numbers and exponential functions for which SA does.

Table 1. The Existing Solution Methods for VRPBTW

Year	Authors	Methods	Size of Instances
1995	Kontoravdis and Bard	Parallel Insertion (PFIH)	25, 50, 100
1995	Gélinas *et al.*	Column Generation	25, 50, 100
1996	Thangiah *et al.*	Parallel Insertion Local Search	25, 50, 100, 250, 500
1996	Potvin *et al.*	Greedy Insertion Genetic Algorithm	25, 50, 100
1997	Duhamel and Potvin	Greedy Insertion Local Search Tabu Search	25, 50, 100
1999	Shen	Route Neighborhood Local Search	100

We explain the concept and procedure of TA more precise. TA is an iterative and convergent meta-heuristic, which guides the subordinate heuristic to search. For a cost-minimized problem, without loss of generalization, a feasible solution is as the current configuration, named $X^{current}$, with value of cost function $C(X^{current})$.TA firstly generates a non-increasing series of threshold values $\{T_k\}$. Then, TA guides the neighborhood search process moving from the current configuration to a new configuration, said as X^{new}, which is not much worse than the incumbent. As shown in Figure 1, TA accepts the new configuration only if $C(X^{new})$ is less than $C(X^{current}) + T_k$, at iteration k. While the k^{th} iteration of neighborhood search completed, TA starts the next iteration with threshold T_{k+1}. TA repeats the previous procedure until all of the thresholds in the series are adopted.

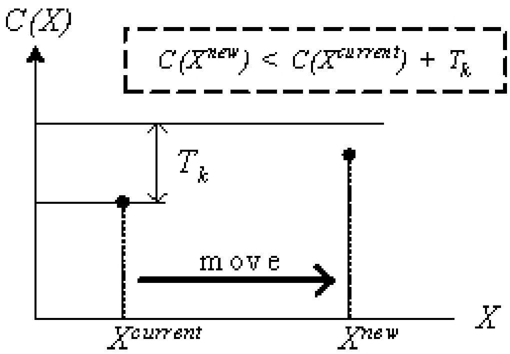

Figure 1. Concept of Threshold Accepting

Even simpler structured than SA, TA has been referenced less often. In Dueck and Scheurer (1990), TA was utilized to solve the Traveling Salesman Problem (TSP) of 442 cities and yielded very-near-to-optimum routes. Han and Cho (2000) presented several modifications on parameters setting and implementation of TA, and applied them to solve TSP and VRP (Vehicle Routing Problem). Moreover, Han and Cho (1999) combined TA with GDA (Great Deluge Algorithm) and FFS (Flip-Flop Search) to form a GIDS (Generic Intensification and Diversification Search) framework. The GIDS was applied to solve complicated VRPs, the

Fleet Size and Mix Vehicle Routing Problem and the Periodic Vehicle Routing Problem (Han and Cho, 2002).

3. TA META-HEURISTIC FOR VRPBTW

In this article, we propose a TA approach for solving the VRPBTW, named as TA_VRPBTW. The main framework of the proposed TA_VRPBTW meta-heuristic, as shown in Figure 2, consists of Initial Solution Constructor (ISC) module, Local Search (LS) module and Threshold Accepting (TA) module. In the ISC module, we modified the traditional Nearest Neighbor algorithm by considering the influences of time windows and the parallel implementation. In the LS module, we adopt 2_opt intra-route arc exchange, (1_0) and (1_1) inter-route node interchange, and (S_S) inter-route arc exchange procedures. Additionally, three TA implemental structures are designed in the TA module.

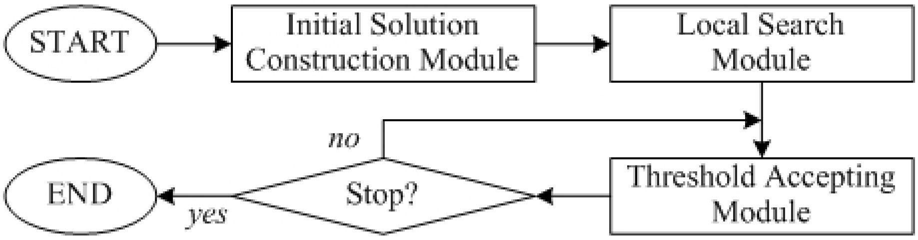

Figure 2. Main Framework of TA_VRPBTW

3.1 Initial Solution Construction Module

The ISC module, which is modified from the Nearest Neighbor algorithm originally applied to the traditional VRP, aims to construct an initial solution of the VRPBTW. The original definition of nearest neighbor is really to indict the candidate node with the minimal distance or travel time (Christofides and Eilon, 1969). For the sake of introducing the factor of time window, we extend the concept of "nearest neighbor" to the following two new aspects.

- The first new aspect considers that from a specific node to some candidate node, which has the earliest lower bound of time window, is the nearest neighbor of this specific node.
- The second new aspect thinks that the nearest neighbor of a specific node is some candidate node whose waiting time is the minimal. Here, the waiting time means that the different between the lower bound of time window and the arrival time from the specific node to this candidate.

To distinguish between these aspects, we name the algorithm under original definition of nearest neighbor as NN1_S, the algorithm under the first new aspect as NN2_S, and the algorithm under the second aspect as NN3_S, where "S" means the algorithm sequentially construct routes. On the other hand, NN1_P, NN2_P and NN3_P separately present the algorithms concurrently construct several routes under above corresponding aspects. Therefore, we totally design six Nearest Neighbor algorithms, i.e., NN1_S, NN2_S, NN3_S, NN1_P, NN2_P and NN3_P for the ISC module.

We explain these aspects more precise by an example shown in Figure 3. Suppose that node A is the last node of the constructing route, and the finished time (*ft*) to serve node A is 20. Additionally, nodes B, C and D are candidates, which satisfy the capacity constraint if be included to the constructing route. According to the original definition of nearest neighbor, we have to select node D as the next node of the constructing route because the travel time (t = 5) leaving from node A is the minimal among all candidates. Under the first new aspect, node C is the nearest neighbor of node A because the lower bound of time window of node C (i.e., 31) is the earliest among candidates. Furthermore, node B, whose waiting time (wt = 0) is the minimal among candidates, is the nearest neighbor of node A under the definition of second new aspect.

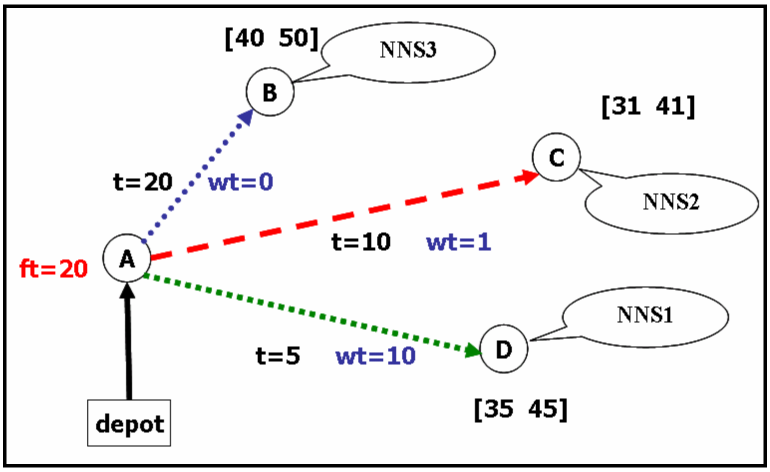

Figure 3. An Example to Explain the Aspects of Nearest Neighbor

3.2 Local Search Module

The LS module, which is based on the exchange procedures, aims to improve the initial solution generated by the ISC module. Classical exchange procedures (Lin, 1965), such as 2_opt, (1_0), can also be easily adapted to the VRPBTW. Hence, we adopt four exchange procedures: 2_opt intra-route arc exchange, (1_0) inter-route node exchange, (1_1) inter-route node exchange, and (S_S) inter-route arc exchange. Where, the (S_S) exchange is an extension of the 2_opt* exchange procedure (Thangiah *et al.*, 1996; Duhamel and Potvin, 1997). In order o distinguish 2_opt* from traditional 2_opt, the name of (S_S) is used instead of the extended 2_opt*, where (S-S) means the “segment-to-segment”.

- The 2_opt exchange procedure deletes two nonadjacent arcs on the linehaul-segment (or backhaul-segment) in a route, and re-links the route feasibly by adding two new arcs.
- The (1_0) exchange procedure extracts a linehaul (or backhaul) from one route, and inserts this linehaul (or backhaul) between two linehauls (or backhauls) or the last linehaul and the first backhaul in another route.
- The (1_1) exchange procedure interchanges two nodes (linehaul or backhaul) in two different routes. The new locations for inserting these two nodes must satisfy the precedence constraint of linehaul and backhaul.
- The (S_S) exchange procedure initially breaks two routes, says A and B, into four segments by deleting one arc in each route. Then, the first node on the rear-segment of route A is linked to the last node on the head-segment of route B, and the first node on the rear-segment of route B is linked to the last node on the head-segment of route A. Figure 4 presents the above concept of the (S_S) exchange procedure.

In Thangiah *et al.* (1996), the 2_opt* exchange only considers that the breakpoint in a route is chosen between the last linehaul and the first backhaul nodes. In Duhamel and Potvin (1997), the 2_opt* exchange considers that the breakpoint must not only occur between the last linehaul and the first backhaul on each route, but also occur between two linehauls or two backhauls on both routes. In this paper, the (S_S) exchange considers more feasible breakpoints as follows. As shown in Figure 4, the route A can be broken between two linehauls (*l-l*), two backhauls (*b-b*), or the last linehaul and the first backhaul (*l-b*). In the first case, the feasible breakpoint in the other route B could occur between (*l-l*) and (*l-b*). In the second case, the feasible breakpoint in route B could be chosen between (*b-b*) and (*l-b*). Finally, in the (*l-b*) case, the breakpoint could occur at any location, i.e., (*l-l*), (*b-b*) and (*l-b*), in route B.

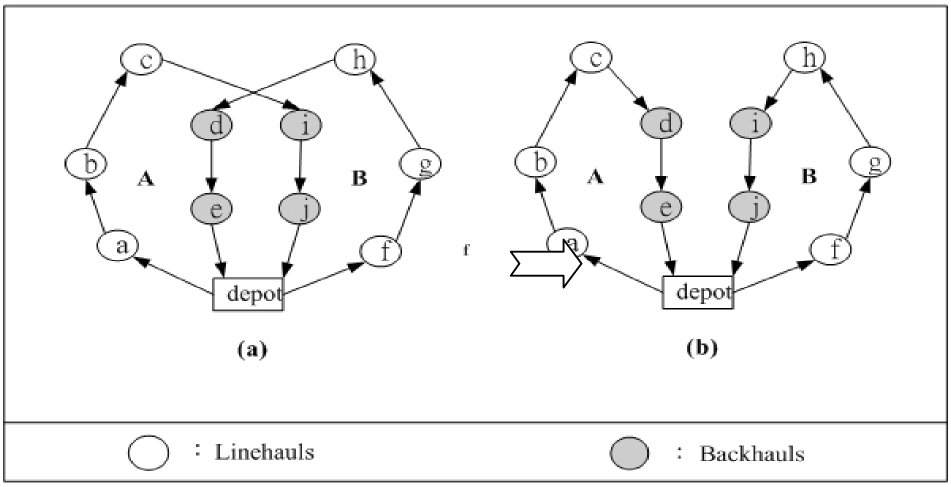

Figure 4. Concept of (S_S) Exchange

Once the generation of the neighborhood of the current solution is established, a strategy must be decided for the acceptance of a new solution in the neighborhood. There are two strategies have been adopted: (1) The first-improvement strategy selects the first solution met in the neighborhood that improves the objective value of the current solution; and (2) The best-improvement strategy checks all neighbor solutions and selects the solution that obtains the largest improvement in the objective value of the current solution. The best-improvement strategy is utilized to rule the selection of neighbor solution in the LS module. Moreover, all of the four exchange procedures need to satisfy the capacity and time window constraints of routes, too.

3.3 Threshold Accepting Module

The TA module, which guides the subordinate exchange procedures described in previous subsection, aims to intensify the process of search and escape from the captivity of the local optimum. As described in Subsection 2.2, the TA has to decide the times of iterations (K), and create a decreasing series of threshold values, $\{T_k\}$, whose length is equal to K. At each run of iteration, TA guides the local search procedure to accept new neighbors according to a specific threshold value T_k. In this study, we design three processes to implement the TA module, named as TA1, TA2 and TA3 (see Figure 5).

- The TA1 module, as shown in Figure 5(a), sequentially executes a set of LS procedures K times. At each execution of iteration, this set of LS procedures accepts new neighbors according to the same threshold T_k.

- The TA2 module, as shown in Figure 5(b), sequentially executes a set of LS procedures K

times. At each execution of iteration, this set of LS procedures accepts new neighbors according to the same threshold T_k, and finally executes the LS module once.

- The TA3 module, as shown in Figure 5(c), separately executes a set of LS procedures in a specific order. At each execution of LS procedure, which consists of K runs of iteration, the series of threshold values $\{T_k\}$ is independently adopted to rule the acceptance of new neighbors.

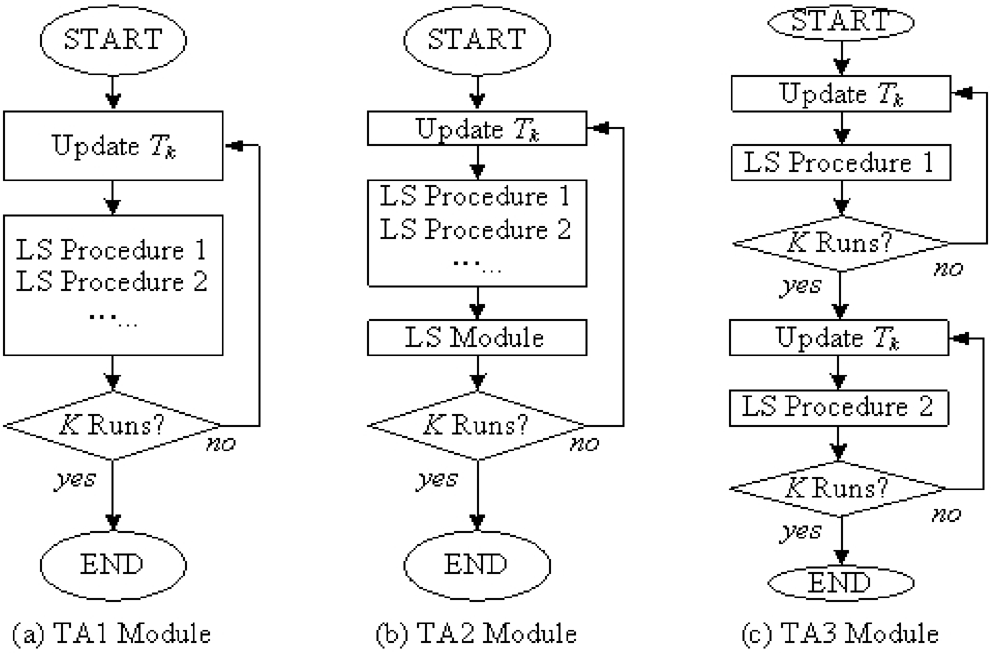

Figure 5. Executive Framework of TA Modules

As mentioned previously, there are two strategies for selecting neighbors, i.e., the first-improvement strategy and the best-improvement strategy. In these TA modules, we choose the first-improvement strategy utilized to the LS procedures. Note that the LS module executed in the TA2 module still adopts the best-improvement strategy.

Furthermore, the decreasing pattern of the threshold series guides the generation of threshold values. As shown in Equation (15), a linear decreasing pattern which refers to the previous study of Han and Cho (2000), is proposed. Where, $C(X_0)$ is the routing time of initial solution, R is the fleet size of initial solution, N is the number of customers, T_0 is a pre-defined initial threshold rate, and K is the times of iteration. The ratio of $C(X_0)$ to $(N + R)$ means the average routing time of a pair of customers, which is considered as high relative to the exchange procedure.

$$\{T_k | T_k = \frac{C(X_0)}{(N+R)} \times T_0 \times (\frac{K-k}{K-1}), \quad for \quad k = 1 \sim K\} \tag{15}$$

4. EXPERIMENTAL ANALYSIS

There are various different versions and parameters setting which could be designed to implement a meta-heuristic procedure. An important issue is which one is superior and powerful in solving problems. It is generally believed that empirical analysis of the average case is appropriate for identifying the performance of a specific procedure. Therefore, we adopt a series of numerical experiments to analyze the average-case performance of the proposed TA_VRPBTW meta-heuristic procedure.

4.1 Testing Instances Bank

There are 81 instances, modified from a set of Solomon's VRPTW benchmark instances, generated for testing and analyzing the performance of TA_VRPBTW. Solomon (1983) originally generated a set of 56 VRPTW instances, which are classified into three patterns of customers' geographical spread: randomized (R), clustered (C), and mixed of randomized and clustered (RC). All of the VRPTW instances comprise 100 customer nodes with corresponding (x, y) coordinates, volume of demand, lower bound and upper bound of required time window, and a fixed amount of 10 time units for maneuvering (e.g. unload or load the goods at each customer location). In these instances, the traveling times are equal to the corresponding Euclidean distances. Real, double-precision distances were used (i.e., neither rounding nor truncation took place during the computations). The overall routing time is the sum of traveling time, waiting time, and maneuvering time.

Initially, we chose three VRPTW instances, R101, C101 and RC101, from the Solomon's benchmark instances. Then, the first 25, 50 and 100 customers were extracted from the original VRPTW instances to individually construct the base of VRPBTW instances in three different sizes. Sequentially, the VRPBTW instances were created by randomly assigning 30, 50 and 70% of backhaul customers in each base. Therefore, 27 kinds of scenarios based on three patterns of geographical spread (R101, C101 and RC101), three sizes of instance (25, 50 and 100 customers), and three ratios of backhaul customers (30, 50 and 70%) are available. We randomly generated three instances in each scenario, and finally built a bank of 81 VRPBTW instances. Note that the capacity of vehicle for every instance is equal to 400 units.

4.2 Experimental Designs

As described in Section 3, there are three kinds of modules, i.e., ISC, LS and TA combined in the TA_VRPBTW. Where, ISC module first constructs an initial feasible solution, and then, this initial solution is improved by LS and TA modules. The implementation of these three modules is sequential. Hence, four phases of experiments are designed to identify the average-case performance of various modules and overall TA_VRPBTW. The quality of solution is represented in minimizing two objectives of the fleet size and the routing time respectively.

- Experiment I. Six nearest neighbor procedures for ISC module are separately tested on the 81 VRPBTW instances. The purpose of Experiment I is to identify the effects of new proposed aspects of nearest neighbor and parallel construction strategy.

- Experiment II. Based on six initial solutions constructed by ISC module, four exchange-base procedures for LS module are separately tested. In Experiment II, we want to evaluate the performance of individual exchange procedure.

- Experiment III. According to the results of Experiment II, we combine effective exchange procedures to a sequential execution. Experiment III aims to find out the best combination of these exchange procedures.

- Experiment IV. Based on the results of Experiments III, three executive framework of TA module are tested. In Experiment IV, several parameters for implementing TA module are set, and the performance of the overall TA_VRPBTW is finally identified. As shown in

Equation (15), the value of parameters T_0 and K must be set.

Table 2 summarizes the procedures and parameters setting in four phases of experiments. Even though the performance of 2_opt is inferior to the other exchange procedures in Experiment II, we design six combinations of (1_0), (1_1) and (S_S), in which 2_opt intra-route exchange procedure is still executed before these three inter-route exchange procedures. Because that TA2 module executes more times of exchange procedures than other TA modules in each run, we set the times of iteration, K = 5 for TA2 and K = 10 for TA1 and TA3 to balance their CPU execution time.

Table 2. Symbols of Procedures and Parameters for Experiments

ISC Module	LS Module		TA Module	Parameters	
	Separation	Combination		T_0	K^*
NN1_S	2_opt	2_opt + (1_0) + (1_1) + (S_S)	TA1	0.01	5
NN2_S	(1_0)	2_opt + (1_0) + (S_S) + (1_1)	TA2	0.05	10
NN3_S	(1_1)	2_opt + (1_1) + (1_0) + (S_S)	TA3	0.10	
NN1_P	(S_S)	2_opt + (1_1) + (S_S) + (1_0)		0.15	
NN2_P		2_opt + (S_S) + (1_0) + (1_1)		0.20	
NN3_P		2_opt + (S_S) + (1_1) + (1_0)			

* K = 5 is specific to the TA2 module, and K = 10 is set for TA1 and TA3 modules.

4.3 Computational Results

The proposed TA_VRPBTW meta-heuristic was coded by the Borland C++ programming language and tested on the bank of instances mentioned in Subsection 4.1. The numerical experiments were executed on a personal computer with Pentium III 932MHz CPU, 384MB RAM and Windows XP.

Table 3 presents the numerical results of Experiment I which tests the six nearest neighbor procedures, i.e., three aspects of nearest neighbor implemented by sequential or parallel construction strategies, on 81 VRPBTW instances. Among six nearest neighbor procedures, NN2_P (the earliest lower bound of time window and parallel construction) performs superior than other procedures in both of average fleet size (13.98) and average routing time (1427.68). Dramatically, NN2_S (the earliest lower bound of time window and sequential construction) obtains the worst results in both of objectives; average fleet size is 24.56 and average routing time is 2104.53. Consider that the average performance of three aspects of nearest neighbor, NN3 (the minimal waiting time) generates the least number of vehicles (fleet size is 15.27), and NN1 (the minimal traveling time) expends the least routing time, 1609.10. On the other hand, the parallel construction strategy experiences better results (average fleet size is 17.17, and average routing time is 1655.55) than the sequential construction strategy does (19.80 for fleet size, and 1909.79 for routing time). Summarily, the overall performances of ISC module (six nearest neighbor procedures) are 18.49 vehicles for average fleet size and 1782.67 units for average routing time. It is apparent that the proposed new aspects of nearest neighbor (earliest lower bound of time window and minimal waiting time) and parallel construction strategy are effective for creating good initial solutions.

Table 4 shows the computational results of Experiment II on individually testing four exchange procedures, i.e., 2_opt intra-route exchange, and (1_0), (1_1) and (S_S) inter-route exchanges. All of the exchange procedures are executed to improve the 6 initial solutions constructed by above six nearest neighbor procedures respectively. Judging from Table 4, (1_0) exchange is the best procedure in performance of fleet size (17.03) and routing time

(1549.73). Note that (1_0) exchange is the only one procedure which is capable of reducing the vehicles of fleet (percentage of improvement is 7.90%). On the other hand, all of four exchange procedures generate smaller routing time, in the sequence of (1_0), (1_1), (S_S) and 2_opt, than initial solutions. The percentages of improvement in routing time are significantly different between exchange procedures (from 13.07%, 10.55% and 2.70% to 0.84%). Notice that the inferior effect on diminishing routing time of 2_opt is possibly due to the strict limitations of precedence and time window between the customers in the same route. Though that 2_opt performs poorly, we still adopt it in the following two experiments by executed after the ISC module done.

Table 3. Results of Experiment I for Six Nearest Neighbor Procedures

Nearest Neighbor	Average Fleet Size			Average Routing Time		
	Sequential	Parallel	Average	Sequential	Parallel	Average
NN1	19.55	22.29	20.92	1631.52	1586.68	1609.10
NN2	24.56	13.98	19.27	2104.53	1427.68	1766.11
NN3	15.28	15.25	15.27	1993.33	1952.28	1972.80
Average	19.80	17.17	18.49	1909.79	1655.55	1782.67
St-Dev.†			2.91			182.42

† St-Dev. denotes the standard deviation of the objective values.

Table 4. Results of Experiment II for Individual Exchange Procedure

Procedures	Average Fleet Size	Improvement (%)	Average Routing Time	Improvement (%)
ISC Module	18.49	---	1782.67	---
2_opt	18.49	0.00%	1767.74	0.84%
(1_0)	17.03	7.90%	1549.73	13.07%
(1_1)	18.49	0.00%	1594.67	10.55%
(S_S)	18.49	0.00%	1734.58	2.70%

In Experiment III, six combinations of (1_0), (1_1) and (S_S) inter-route exchange procedures are tested. Similar to Experiment II, all of the combinations are executed to improve six initial solutions respectively. As shown in Table 5, Combinations (1_1)+(S_S)+(1_0) and (1_1)+(1_0)+(S_S) gain the least average fleet size, 16.58 and 16.60 separately. Additionally, Combination (1_0)+(S_S)+(1_1) obtains the least routing time (1421.45) followed by Combinations (1_1)+(S_S)+(1_0) and (1_1)+(1_0)+(S_S). The total averages of fleet size and routing time are 16.82 and 1446.57, as well as the corresponding average percentages of improvement are 9.06% and 18.85%. Such a result indicts that these exchange procedures of the LS module perform well in routing time than in fleet size. Moreover, even though the differences of performances between various combinations are insignificant, Combinations (1_1)+(S_S)+(1_0) and (1_1)+(1_0)+(S_S) seem to perform superior to others in both of objectives.

In order to identify the effects of LS module on improving different initial solutions, Table 6 summarizes the performances of ISC module and LS module according to different initial solutions. Each initial solution gains improvements on average fleet size and average routing time in varied degree. Interestingly, inferior initial solutions seem to obtain more improvements than superior initial solutions. This situation implies that LS module diminishes the difference between various initial solutions and put it tend toward good direction. NN2_P still generates the least fleet size (13.93) and routing time (1307.67), but the worst result of fleet size (19.61) is produced by NN1_P, where NN2_S still gets the longest average routing time (1575.96).

Table 5. Results of Experiment III for Combination of Exchange Procedures

Combinations	Average Fleet Size	Improvement (%)	Average Routing Time	Improvement (%)
ISC Module	18.49	---	1782.67	---
2_opt+(1_0)+(1_1)+(S_S)	17.03	7.90%	1456.44	18.30%
2_opt+(1_0)+(S_S)+(1_1)	17.03	7.90%	1421.45	20.26%
2_opt+(1_1)+(1_0)+(S_S)	16.60	10.22%	1431.11	19.72%
2_opt+(1_1)+(S_S)+(1_0)	16.58	10.33%	1442.38	19.09%
2_opt+(S_S)+(1_0)+(1_1)	16.92	8.49%	1478.34	17.07%
2_opt+(S_S)+(1_1)+(1_0)	16.73	9.52%	1449.71	18.68%
Total Average	16.82	9.06%	1446.57	18.85%
Standard Deviation	0.21		20.02	

Table 6. Improvements of Performance from ISC Module to LS Module

Initial Solutions	Average Fleet size			Average Routing Time		
	ISC	LS	Improvement	ISC	LS	Improvement
NN1_S	19.55	18.04	7.72%	1631.52	1411.81	13.47%
NN2_S	24.56	19.11	22.19%	2104.53	1575.96	25.12%
NN3_S	15.28	15.09	1.24%	1993.33	1505.80	24.46%
NN1_P	22.29	19.61	12.02%	1586.68	1386.99	12.59%
NN2_P	13.98	13.93	0.36%	1427.68	1307.67	8.41%
NN3_P	15.25	15.06	1.25%	1952.28	1491.14	23.62%
Average	18.49	16.82	9.06%	1782.67	1446.57	18.85%

Due to the variety of combining ISC and LS modules (total of 36 combinations), we select four combinations of executing ISC and LS modules to test following Experiment IV. We combine NN1_P and NN2_P with 2_opt+(1_0)+(1_1)+(S_S) and 2_opt+(1_1)+(S_S)+(1_0) respectively to form the four executive combinations. As shown in Table 7, TA2 module obtains the best results on average fleet size (15.06) and average routing time (1256.28), followed by TA1 and TA3. The superiority of TA2 to TA1 and TA3 implies that the immediate compensation strategy of executing LS module for intensive search after the diverse TA search is an effective and efficient mechanism.

Moreover, the total average percentages of improvement are 7.92% for fleet size and 6.25% for routing time. Although the improving ratio of TA module is smaller than that of LS module, the difference between various executive frameworks is continuously reduced (judging from the standard deviations reported in Tables 3, 5 and 7). Such a reduction implies that TA meta-heuristic is possibly robust and effective.

Figure 6 depicts the decreasing trends of objective values in fleet size and routing time during the process of implementing TA_VRPBTW. For the objective of minimizing fleet size, the averages are 18.49 of ISC module, 16.82 of LS module, and 15.18 of TA module. It is almost a linear reduction in various phases of modules. Similarly, the average routing times, which are 1782.67 of ISC module, 1446.57 of LS module, and 1267.09 of TA module, present a trend of slightly slow reduction.

Table 7. Results of Experiment IV for Three TA Executive Frameworks

Executive Frameworks	Average Fleet Size	Improvement (%)	Average Routing Time	Improvement (%)
ISC + LS	16.49	---	1351.60	---
TA1	15.25	7.52%	1270.84	5.98%
TA2	15.06	8.67%	1256.28	7.05%
TA3	15.24	7.58%	1274.14	5.73%
Total Average	15.18	7.92%	1267.09	6.25%
Standard Deviation	0.11		9.50	

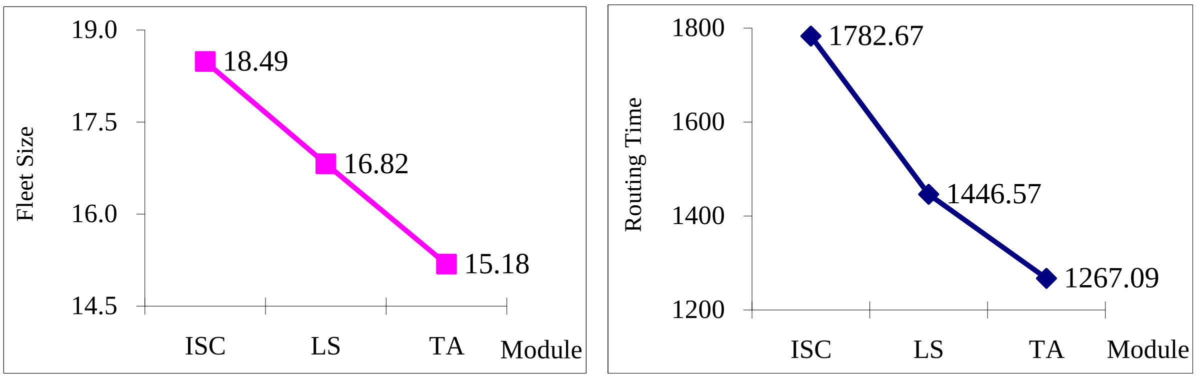

Figure 6. Trends of Objective Values for VRPBTW Testings

Table 8 lists the qualities of solutions generated by TA modules under various parameter values of initial threshold rate, T_0. Observing from Table 8, T_0 = 0.05 performs superior to other values of T_0 in both objectives of fleet size and routing time. Overall, the difference of objective values between these T_0 is very slightly. It seems that the quality of solutions is insensitive to the parameters T_0.

Table 8. Computational Results of Parameters T_0

	Average Fleet size				Average Routing Time			
T_0	TA1	TA2	TA3	Average	TA1	TA2	TA3	Average
0.01	14.76	14.71	14.76	14.74	1272.25	1256.54	1269.71	1266.17
0.05	14.76	14.46	14.76	14.66	1271.13	1253.75	1269.71	1264.86
0.10	14.76	14.71	14.76	14.74	1285.94	1262.13	1269.34	1272.47
0.15	14.76	14.71	14.76	14.74	1286.65	1265.99	1272.93	1275.19
0.20	14.76	14.71	14.76	14.74	1286.65	1265.99	1272.93	1275.19

Finally, Table 9 gives the CPU running time of executing the TA_VRPBTW according to different sizes of instances. As shown in Table 9, TA2 framework expends less average CPU times (11.06 seconds) than TA1 and TA3 frameworks. Although the CPU times of three TA modules are almost same in the size of 25 and 50 customers, TA2 experiences about half times of TA1 or TA3 in the size of 100 customers. In average, the CPU running time of TA_VRPBTW is less than 20 seconds.

Table 9. CPU Running Time of TA_VRPBTW (Unit: Second)

Size of Instances	TA1 Module	TA2 Module	TA3 Module	Average
25	0.76	0.83	0.46	0.67
50	1.25	1.28	0.93	1.15
100	64.81	31.07	76.74	57.54
Average	22.27	11.06	26.04	19.79

5. CONCLUSIONS AND SUGGESTIONS

This study proposed a TA (Threshold Accepting) meta-heuristic for solving VRPBTW, which is important in practical logistics and distributions, such as home delivery. A bank of 81 new VRPBTW instances is adopted to test the performance of various modules of TA_VRPBTW under four phases of experiments and related parameters. Findings and conclusions obtained from the numerical results are summarized as follows.

- Our proposed new aspects of nearest neighbor, the earliest lower bound of time window and the minimal waiting time, can generate better initial solutions than the original aspect (the minimal traveling time) under the situations of Experiment I. Additionally, the proposed parallel construction strategy also performs well than the traditional sequential construction strategy.

- Inter-route exchange procedures gain more improvements than intra-route exchange procedure. Combination of various exchange procedures to execute LS module is effective. According to results of Experiment III, the percentages of improvement on fleet size and routing time are 9.06% and 18.85% respectively.

- Not only in quality of solutions but also in CPU running time, TA2 framework, which implements LS module once more after sequentially executing exchange procedures under the corresponding threshold value T_k at each run of iteration, is superior to other TA frameworks.

- Experimental results of TA_VRPBTW experiences a trend of continuous improvement. The differences of qualities between various initial solutions, which are constructed by ISC module, are significantly diminished by following LS and TA modules. It implies that TA_VRPBTW is a robust and effective approach to solve the VRPBTW.

There are several issues for further research: (1) practical applications of VRPBTW on the physical distribution. For example, the home-delivery carriers can transfer the amount of deliveries and pickups for customers to the demands of linehauls and backhauls, and consider the different time intervals of service required by customers; (2) adoption of other meta-heuristics, such as Tabu Search, Ant Colony System and GIDS (Cho, 2001), to design VRPBTW solving tools; (3) extension of the modified nearest neighbor procedures and parallel construction strategy to other related VRP, for example, VRPTW; and (4) enhancement of modules to reduce the fleet size.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the National Science Council of the R.O.C. in Taiwan under grant number NSC93-2211-E-216-009.

REFERENCES

Cho, Y.J. (2001) **A New Metaheuristic Approach for HVRP and PVRP.** Ph.D. Dissertation, National Chiao Tung University, R.O.C.

Christofides, N. and Eilon, S. (1969) An algorithms for vehicle dispatching problem. **Operations Research Quarterly, Vol. 20,** 568-589.

Dueck, G. and Scheuer, T. (1990) Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. **Journal of Computational Physics, Vol. 90,** 161-175.

Dumamel, C., Potvin, J.Y. and Rousseau, J.M. (1997) A tabu search heuristic for the vehicle routing problem with backhauls and time windows. **Transportation science, Vol. 31,** 49-59.

Gélinas, S., Desrochers, M., Desrosiers, J. and Solomon, M. (1995) A new branching strategy for time constrained routing problems with application to backhauling. **Annals of Operations Research, Vol. 61,** 91-109.

Goetschalckx, M. and Jacobs-Belcha, C. (1989) The vehicle routing problem with backhauls. **European Journal of Operational Research, Vol. 42,** 39-51.

Han, A.F. and Cho, Y.J. (2000) Implementation and parameters setting of several meta-heuristics to TSP and VRP. **Proceedings of 5th Conference of Transportation Network**, Taichung, Taiwan, R.O.C., 57-69. (in Chinese)

Kontoravdis, G. and Bard, J. (1995) A GRASP for the vehicle routing problem with time windows. Working Paper, **ORSA Journal of Computing, Vol. 7,** 10-23.

Lin, S. (1965) Computer solution of the traveling salesman problem. **Bell System Technology Journal, Vol. 11,** 2245-2269.

Potvin J., Duhamel, C. and Guertin, F. (1996) A genetic algorithms for vehicle routing problem with backhauling. **Applied Intelligence, Vol. 6,** 345-355.

Solomon, M. (1983) **Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms.** Ph.D. Dissertation, University of Pennsylvania, U.S.A.

Thangiah, R., Potvin, J. and Sun, T. (1996) Heuristic approach to vehicle routing with backhauls and time windows. **Computers and Operations Research, Vol. 23, No. 11,** 1043-1057.

Yano, T., Chan, L. and Richter, K. (1987) Vehicle routing at quality stores. **Interfaces, Vol. 17, No. 2,** 52-63.

Shen, S.Y. (1999) **Time Window Constrained Vehicle Routing Problems.** Ph.D. Dissertation, National Chiao Tung University, R.O.C.

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

卓裕仁 [1]、王生德 [2]

[1] 運輸科技與物流管理學系，助理教授
中華大學
新竹市五福路二段 707 號
yjcho@chu.edu.tw

[2] 科技管理研究所運輸科技與物流管理組，碩士
中華大學
新竹市五福路二段 707 號

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，在物流配送實務上有很高的應用價值。本研究結合門檻接受法(Threshold Accepting, TA)與傳統啟發式方法，設計一套巨集啟發式解法，並藉由自行產生例題進行測試，以分析此巨集啟發式解法之解題績效。

關鍵字：時窗限制回程取貨車輛路線問題、巨集啟發式解法、門檻接受法。

壹、前言

近年來，物流產業的發展逐漸受到重視，許多新興行業(例如宅配業、快遞業)的出現，為整個物流環境注入了活力。宅配業者為提升其配送服務水準，必須增加車輛的發車班次，實務上多利用上午將前晚到達的物件遞送給客戶，然後直接前往託運顧客處收件取貨後，再返回營業所場站。上述的作業方式稱為「回程取貨(Backhauling)」，不僅可縮短車輛的行程與時間，還能避免重新整理車廂貨物的排列。

在作業研究的理論上，常將車輛配送路線規劃的問題轉換成「車輛路線問題(Vehicle Routing Problem, VRP)」來進行求解，其問題定義乃是由同一車種、固定容量的車隊，從單一場站出發，服務完一群需求量已知的顧客後返回中心場站；目標在使車輛路線的總運輸成本最小化。VRP 可視實務運作的條件與限制，衍生出許多不同的問題型態，例如：「時窗限制回程取貨車輛路線問題(Vehicle Routing Problem with Backhaul and Time Windows, VRPBTW)」即是針對上述宅配業的經營方式所提出。VRPBTW 將顧客需求點分為兩部分，送貨點(Linehauls)與取貨點(Backhauls)；假設車輛必須先服務完所有送貨點顧客，然後才能開始服務取貨點顧客，並限制車輛到達顧客點的時間必須在給定的時間窗之內；目標為總車輛數與總路線成本最小化。

VRP 與 VRPBTW 皆屬於高複雜度(NP-hard)之組合最佳化問題，在求解較大規模問題時，往往需要花費相當多的時間。因此發展有效率的啟發式解法，才能在可接受的時間內求出精確度高的近似解。有鑑於此，本研究之目的乃希望發展一套求解績效良好且時間合理的 VRPBTW 巨集啟發式演算法，以提升 VRPBTW 實務應用之求解績效，降低物流成本。

本文後續章節安排如下：第二節簡要回顧 VRPBTW 之定義與求解方法；第三節說明門檻接受法（TA）應用於 VRPBTW 之解法架構與模組設計；第四節介紹例題產生方式與實驗設計；第五節彙整例題測試的結果；最後於第六節提出本研究的結論與建議。

貳、問題定義與求解方法回顧

本節定義 VRPBTW 問題如下:「顧客需求點分為送貨點(line-hauls)及取貨點(back-hauls)兩部分,配送方式是從場站(depot)出發將貨物運送給送貨點顧客,然後再到取貨點顧客收取貨物運回場站,配送過程不得違反時間窗及車輛容量限制;目標為車輛總數及車輛路線的總運輸成本最小化。VRPBTW 問題假設車輛必須先服務完所有送貨點顧客後,才能開始服務取貨點顧客,如此可避免重新整理車廂貨物所花費的排列時間。」至於 VRPBTW 之數學規劃模式請參考王生德(2004)碩士論文。

國內外有關 VRPBTW 文獻並不多,在精確解之數學規劃方面有 Yano 等人(1987)探討,至於啟發式方法則有 Potvin 等人(1996)與 Thangiah 等人(1996)兩篇論文探討;國內有徐俊誠等人曾分別針對 VRPBTW 進行研究,但是其問題定義與國際文獻定義的 VRPBTW 稍有不同。限於篇幅,僅將國內外相關文獻及其求解方法彙整於表 1。

表 1　VRPBTW 求解方法彙整表

年代	作者	使用方法	限制條件	模式假設
1987	Yano et al.	◦集合涵蓋法	◦車容量限制 ◦時間窗限制	◦先送貨後取貨
1996	Potvin et al.	◦貪心插入法　◦基因演算法	◦車容量限制 ◦時間窗限制	◦先送貨後取貨 ◦取貨比例:10%、30%、50%
1996	Thangiah et al.	◦循序插入法　◦區域搜尋法	◦車容量限制 ◦時間窗限制	◦先送貨後取貨 ◦取貨比例:10%、30%、50%
1999	申生元	◦路線鄰域法　◦區域搜尋法 ◦擾動演算法	◦車容量限制 ◦時間窗限制	◦先送貨後取貨 ◦取貨比例:10%、30%、50%
2000	徐俊誠	◦鄰近點法　◦掃描法 ◦改良型插入法◦模擬退火法	◦車容量限制 ◦無/有時間窗限制	◦可同時收送貨 ◦取貨比例:20%、34%、50% 10%、30%、50%
2000	曾維豪	◦鄰近點法　◦循序插入法 ◦平行插入法　◦禁制搜尋法	◦車容量限制 ◦時間窗限制	◦可同時收送貨 ◦取貨比例:10%、30%、50%
2001	魏宗徵	◦鄰近點法　◦基因演算法	◦車容量限制 ◦時間窗限制	◦可同時收送貨 ◦取貨比例:10%、30%、50%
2003	莊英群	◦節省法　◦禁制搜尋法	◦車容量限制 ◦時間窗限制	◦同時收送貨 ◦取貨比例:25%、50%

參、VRPBTW 之門檻接受法架構設計

本研究採用門檻接受法(Threshold Accepting, TA)結合傳統鄰域搜尋方法,構建一套 VRPBTW 之巨集啟發式解法,並將其命名為 TA_VRPBTW。門檻接受法(Dueck & Scheuer, 1990)之基本觀念乃是在鄰域搜尋陷入局部最佳解時,採取較鬆的接受法則(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。TA 的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 則可接受暫劣之鄰解。茲以圖 1 的示意圖說明 TA 之接受法則:TA 法事先產生一組固定的門檻值數列(通常為遞減),依次使用數列中的門檻值,其接受法則為 $C(X^{new}) < C(X^{current}) + Tk$。

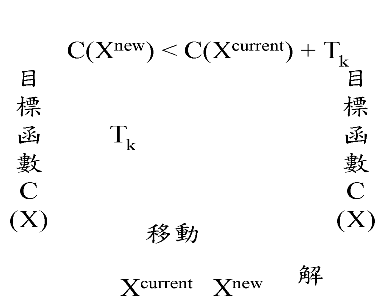

圖 1　TA 接受法則示意圖(2001)

一、整體解題架構

整套 TA_VRPBTW 解題架構如圖 2 所示，可分為三個程序：(1) 起始解構建程序，根據傳統鄰點法(Nearest Neighbor, NN)提出三種改良式的鄰點法，用以構建不同的起始解；(2) 鄰域搜尋改善程序，分別以路線內交換法和路線間交換法來搜尋局部最佳解；(3) 門檻接受程序，針對鄰域搜尋改善程序的結果，使用 TA 法進行搜尋以跳脫局部最佳解的束縛。圖 2 中，(1_0)_b、(1_1)_b、(S_S)_b 分別代表各交換法的交換策略為最佳改善(best-improvement)策略；(1_0)_f、(1_1)_f、(S_S)_f 代表各交換法的交換策略為首先改善(first-improvement)策略。各程序之模組設計細節說明於下。

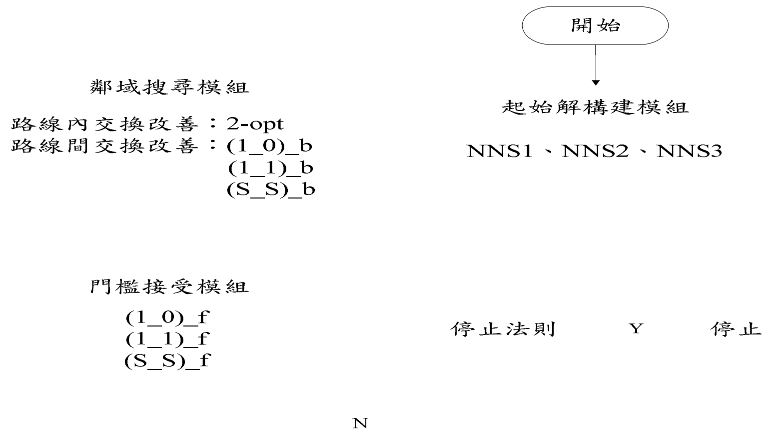

圖 2　VRPBTW 之門檻接受法解題架構

二、模組細部設計

本研究認為傳統的鄰點法(Nearest Neighbor, NN)僅以距離為考慮因素，在時間窗問題中，應可直接將時間窗因素納入考量。因此本研究提出了改良式的鄰點法做為起始解的構建方法，將時間窗因素做為選擇鄰點的準則，一為需求點的服務時窗下界最早：考慮以需求點的服務時窗下界(開始服務時間)的先後順序來決定服務順序；二為需求點的等待服務時間最少：車輛可在時間窗下界之前到達需求點，惟必須等到時間窗下界開始才可進行服務，這段時間即為等待服務時間，依等待時間長短決定服務順序。以循序的方式將需求點一一納入路線中；所謂循序的概念意指：一條路線構建完畢之後再構建另一條新的路線，直到所有需求點皆已納入所有路線為止。改良式的鄰點法包括：循序-需求點間的距離最短(NNS1)、循序-需求點的服務時窗下界最早(NNS2)、循序-需求點的等待服務時間最少(NNS3)等三種模組。

圖 3 說明三種起始解法的概念：假設有四個需求點，ft 為 A 點的服務結束時間；t 為兩點間之旅行時間(距離)；wt 為等待服務時間，[t1, t2]為時間窗上下界。點與點間的距離最短其時間窗不一定是最早，且等待時間也不一定是最少，因此若以 NNS1 為準則，t = 5 距離最短，挑選的為 D 點；以 NNS2 為準則，[31, 41]為最早，挑選的為 C 點；以 NNS3 為準則，wt = 0 為最少，挑選的為 B 點。不管是哪一種準則，從 A 點到各需求點都必須符合該點的時間窗之內。

路線內改善模組採用 2_opt 交換法(Lin, 1965)，交換策略為「首先改善」策略。路線間交換模組則採用 Christofides & Eilon (1969)所提出的(1_0)節點交換法、(1_1)節點交換法，以及本研究提出的(S_S)整段交換法；交換策略為「最佳改善」策略。在車輛容量、時間窗及先送貨後取貨的限制下，經由路線內節線交換與路線間節點交換等機制，以達成縮短總路線成本的目標。

在門檻接受程序方面，以前述之路線間交換模組為核心工具，但為節省運算時間，皆採用「首先改善」之執行方式。之所以沒有考慮路線內 2_opt 交換法，是因為受到時間窗及先送貨

後取貨等限制，路線內交換法很難產生顯著的改善效果，故不考慮引入。在控制參數方面，包括起始門檻值(T_0)、門檻數列長度(K)及門檻數列收斂型態。門檻數列在門檻數列長度(K)期間，自起始門檻值(T_0)逐漸收斂到 0，本研究門檻數列收斂型態採用線性遞減門檻數列。

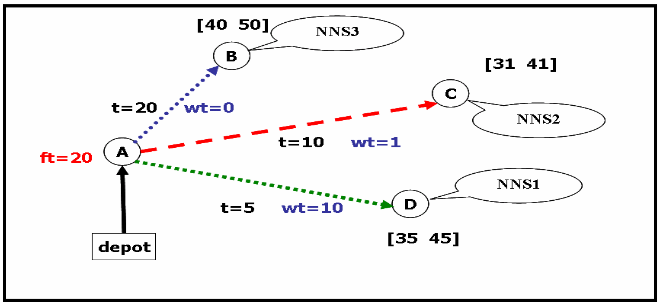

圖 3　起始解模組概念圖

TA 執行時必須先設定起始門檻值與門檻數列，首先以路線間(1_0)_f 節點交換法為鄰域搜尋找尋可行解，接著按照 TA 的接受法則決定是否更新現有解。當鄰域搜尋找到新的可行解時，必須判斷此可行解與現有解之差值是否在門檻值之下，若是則接受交換並更新暫優解。若此暫優解優於搜尋過程中目前之最佳解，則更新最佳解。當所有交換情形皆考慮後，接著執行路線間(1_1)_f 節點交換及 S_S 整段交換(執行順序可調整)，當交換模組皆執行完後，將目前之門檻值更新為下一個門檻值，繼續執行核心搜尋，重新考慮交換情況。直到所有門檻值皆執行完畢後，便可決定 TA 的停止時機。

肆、測試題庫建立與實驗設計

由於表 1 所列文獻皆以 Solomon (1983)所建立的 VRPTW 測試題庫為基礎，加以修改產生 VRPBTW 之測試例題，因此本研究亦從 Solomon 題庫中挑選 R101、C101、RC101 三個例題，分別選擇其前 25、前 50 及 100 個顧客，再以隨機方式設定 30%、50%、70%為回程取貨的顧客，各產生 3 題，總計有 81 個測試例題。

本研究針對 TA_VRPBTW 三個程序所使用的方法、模組組合方式、控制參數設定等因素進行測試之實驗設計。整個測試過程可分成以下三個實驗，整理如表 2 所示。

表 2　實驗設計

	目的	控制因子與參數
實驗一	起始解(IS)之解題績效	NNS1、NNS2、NNS3
實驗二	鄰域搜尋執行組合(NS)之解題績效	2-opt、(1_0)_b、(1_1)_b、(S_S)_b
實驗三	門檻接受法(TA)之解題績效	T_0、K；(1_0)_f、(1_1)_f、(S_S)_f

在實驗三中，TA 有兩個控制參數：起始門檻值(T_0)與執行次數(K)，並據以構成一個門檻數列長度$\{T_k\}$。本研究根據 VRPBTW 之問題特性，提出下列的修正公式：

$$\{T_k | T_k = \frac{C(X_0)}{(N+R)} \times T_0 \times (\frac{K-k}{K-1}), \ \ for \ \ k = 1 \sim K\} \tag{4.1}$$

上式中，$C(X_0)$為起始解之目標函數值(成本)；N 為顧客節點數(題目規模)；R 為起始解之車輛數。由式(4.1)可知，本研究採用的是直線遞減型的門檻數列，其門檻值比率自 T_0 逐漸下降至 0，數列長度即為執行次數(K)。$\frac{C(X_0)}{(N+R)}$ 為平均一條節線的長度，執行交換法時，即以此平均數的概

念來執行。TA 之參數設定範圍，T_0 分別設定為 0.01、0.05、0.1、0.15 及 0.2；K 設為 10 次。

伍、例題測試結果之整理與分析

表 3 為三種起始解方法在所有 81 個例題的平均測試結果比較。NNS3 表現最佳，車輛數為 15.28 輛，路線成本為 1993.33；NNS2 表現較差，車輛數為 24.56 輛，路線成本為 2104.53；整體的平均車輛數為 19.80 輛，路線成本為 1909.79。

表 3 所有測試例題平均起始解測試結果

方法＼目標	平均車輛數		平均路線成本	
NNS1	19.55	19.80	1631.52	1909.79
NNS2	24.56		2104.53	
NNS3	15.28		1993.33	

表 4 的六種鄰域搜尋模組組合(N1~N6)可分成三個型態：先降低車輛數(1_0)再改善路線成本；把降低車輛數放到中間執行；先改善路線成本最後再降低車輛數。以降低車輛數的項目來看，N1 組合降低 7.3%最多，N6 組合降低 4.7%為最少；在路線成本方面，以 N2 改善 17.08%最多，N5 改善 12.73%最少。整體而言以 N1 組合的績效較佳，因此以 N1 進行後續的 TA 測試。

表 4　整體交換組合測試結果

模組組合	平均車輛數	平均路線成本	改善幅度(%) 車輛數	改善幅度(%) 路線成本
N1: 1_1→S_S→1_0	17.18	1485.07	-7.3%	-16.14%
N2: 1_0→S_S→1_1	17.19	1468.36	-7.28%	-17.08%
N3: 1_0→1_1→S_S	17.19	1511.54	-7.28%	-14.64%
N4: S_S→1_1→1_0	17.23	1495.51	-7.07%	-15.55%
N5: 1_1→1_0→S_S	17.59	1545.43	-5.1%	-12.73%
N6: S_S→1_0→1_1	17.67	1481.24	-4.7%	-16.35%
總平均	17.41	1497.86	-6.46%	-15.42%

表 5 顯示鄰域搜尋改善模組可以大幅增進起始解之精確度，其中 NNS2 方法由於在起始解構建階段表現不佳，相對於其他兩種方法而言，具有較大的改善空間，因此車輛數和路線成本的改善幅度均是最大的，達到 22.19%、24.97%；

表 5　鄰域搜尋改善成效表

	起始解+2_opt		鄰域搜尋		改善%	
目標＼解法	平均車輛數	平均路線成本	平均車輛數	平均路線成本	平均車輛數	平均路線成本
NNS1	19.55	1630.29	18.04	1411.81	-7.72%	-13.4%
NNS2	24.56	2100.32	19.11	1575.96	-22.19%	-24.97%
NNS3	15.28	1928.88	15.09	1505.8	-1.24%	-21.93%
平均	19.80	1886.50	17.41	1497.86	-10.38%	-20.10%

實驗三考慮 TA 方法在分別經過三種起始解法與 N1 鄰域搜尋組合之後的解題績效與改善效果，在參數 T_0 = 0.01、0.05、0.1、0.15、0.2，K=10 之下，所得到的平均測試結果彙整於表 6。整體而言，經由 TA 程序可將平均車輛數自 17.17 降至 16.65，平均路線成本自 1485.07 降至 1359.79，改善幅度分別為 2.98%與 8.28%。就三種起始解法來看，仍以 NNS3 的車輛數最少，NNS1 的路線成本最少；值得注意的是，三種起始解經 TA 改善後彼此間的差距縮小。

表 6　TA 整體測試整理比較

		NNS1_N1	NNS2_N1	NNS3_N1	平均
TA 改善前	車輛數	17.86	18.7	15	17.17
	路線成本	1368.63	1585.96	1500.61	1485.07
TA 改善後	車輛數 (改善幅度)	17.33 (-2.97%)	17.72 (-5.24%)	14.89 (-0.73%)	16.65 (-2.98%)
	路線成本 (改善幅度)	1297.42 (-5.20%)	1397.61 (-11.88%)	1384.33 (-7.75%)	1359.79 (-8.28%)

從起始解構建經過鄰域搜尋改善最後再執行門檻接受程序，不管是車輛數或路線成本皆有明顯大幅的改善，茲以圖 4 與圖 5 的解題績效趨勢圖說明改善情形。圖 4 為起始解到 TA 整個執行過程，車輛數每階段的改善趨勢圖，起始解為 19.80 輛，經過鄰域搜尋後降為 17.17，最後執行 TA 更降至 16.65 輛。圖 5 為起始解到 TA 整個執行過程，路線成本每階段的改善趨勢圖，起始解為 1909.79，經過鄰域搜尋後降為 1485.07，最後執行 TA 更降至 1359.79。顯示本研究之鄰域搜尋和 TA 皆具不錯的改善效果。

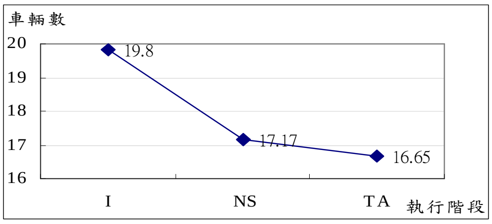

圖 4　車輛數下降趨勢圖

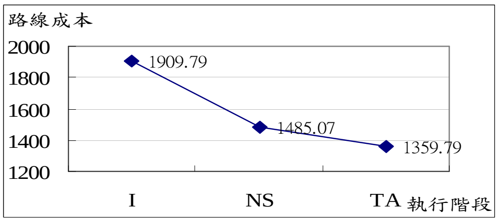

圖 5　路線成本下降趨勢圖

陸、結論與建議

茲彙整本研究之貢獻、重要結論與建議如下：

1. 本研究以傳統鄰點法的概念為核心，提出另外改良式的鄰點法做為起始解的構建方法。經由例題測試証明循序-需求點的等待服務時間最少，所求得的起始解優於傳統鄰點法。
2. 在鄰域搜尋改善方面，除了採用路線內(2_opt)節線交換、路線間(1_0)節點交換、(1_1)節點交換法外，更依據先送貨後取貨的特質提出 S_S(segment-segment)整段交換法。
3. 起始解搭配鄰域搜尋的組合以[NNS3 + 2_opt + (1_1) + (S_S) + (1_0)]表現最佳。顯示在多種鄰域搜尋改善組合中，先減少路線成本，最後再降低車輛數，所得到的平均改善效果最好。
4. 經過 TA 改善程序後，均能有效提升解的精確度，拉近所有組合間的差距，由此可知，TA 是一個有效且穩健(Robust)的巨集啟發式方法。
5. 本研究僅應用門檻接受法(TA)求解 VRPBTW，未來可嘗試結合其他傳統或巨集啟發式方法，例如：GIDS 法。
6. 可嘗試將本研究發展的改良式鄰點法應用在其他車輛路線問題，例如：時間窗車輛路線問題(VRPTW)、同時收送貨車輛路線問題(PDVRP)等。

柒、參考文獻

1. Christofides, N. & S. Eilon (1969). An Algorithms for Vehicle Dispatching Problem. Operations Research Quarterly, 20: 568-589.
2. Lin, S., (1965). Computer Solution of the Traveling Salesman Problem. Bell System Technology

Journal, 11: 2245-2269.

3. Dueck, G. and T. Scheuer (1990). Threshold Accepting: a General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing. Journal of Computational Physics, 90: 161-175.

4. Potvin J., C. Duhamel and F. Guertin (1996). A Genetic Algorithms for Vehicle Routing Problem with Backhauling. Applied Intelligence, 6: 345-355.

5. Solomon, M. (1983). Vehicle routing and Scheduling with Time Window Constraints: Models and Algorithms, Ph.D. Dissertation, University of Pennsylvania.

6. Thangiah, R., and J. Potvin, and T. Sun (1996). Heuristic Approach to Vehicle Routing with Backhauls and Time Windows. Computers and Operations Research, 23(11): 1043-1057.

7. Yano, T., L. Chan and K. Richter (1987). Vehicle Routing at Quality Stores. Interfaces, 17(2): 52-63.

8. 申生元(1999)，時窗限制車輛途程問題，交通大學工業工程與管理研究所博士論文。

9. 徐俊誠(2000)，車輛途程含取貨問題解法研究，大業大學工業工程研究所碩士論文。

10. 莊英群(2003)，應用禁忌搜尋法於混合送收貨之車輛途程問題，逢甲大學工業工程研究所碩士論文。

11. 曾維豪(2000)，軟性時窗與回程撿收之車輛途程問題研究，元智大學工業工程研究所碩士論文。

12. 魏宗徹(2001)，整合時窗限制與回程撿收之多車種車輛途程問題，元智大學工業工程研究所碩士論文。

13. 卓裕仁(2001)，以巨集啟發式方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所博士論文。

本研究蒙國科會計畫（編號: NSC93-2211-E-216-009）經費補助，特此致謝。