

國立交通大學

運輸科技與管理學系

碩士論文

應用可回溯式門檻接受法結合GENIUS
求解VRP問題之研究

A Metaheuristic Using BATA and GENIUS
for Vehicle Routing Problems



研究生：廖昱傑

指導教授：韓復華 教授

中華民國九十六年六月

應用可回溯式門檻接受法結合 GENIUS 求解 VRP 問題之研究

A Metaheuristic Using BATA and GENIUS
for Vehicle Routing Problems

研究生：廖昱傑


Student : Yu-Jie Liao

指導教授：韓復華

Advisors : Anthony Fu-Wha Han

國立交通大學
運輸科技與管理學系
碩士論文

A Thesis
Submitted to Department of Transportation Technology and Management
College of Management
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Transportation Technology and Management
June 2007
Hsinchu, Taiwan, Republic of China

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside the circle, there is a stylized representation of a building or a ship, with the year '1896' at the bottom. The text 'A Thesis' is positioned above the logo, and 'Submitted to Department of Transportation Technology and Management' is positioned below it.

中華民國 九十六 年 六 月

應用可回溯式門檻接受法結合 GENIUS 求解 VRP 問題之研究

學生：廖昱傑

指導教授：韓復華 教授

國立交通大學運輸科技與管理學系碩士班

摘要

可回溯式門檻接受法(Backtracking Adaptive Threshold Accepting, BATA)是由 Tarantilis et al. [38]於2001年提出，它是從門檻接受法演變而來的新型巨集啟發式解法。其主要差異在於門檻值變化並非單調遞減；當找不到可接受的交換時，可允許放鬆門檻值再重新搜尋。本研究以此方法作為基本架構，並與一般化插入/解繫法(GENIUS)和其他傳統鄰域搜尋法相結合，來求解VRP問題，以探討其求解績效。本研究之求解架構包含起始解模組、鄰域搜尋模組和可回溯式門檻接受模組三個部份。並以Christofides et al. [8]的14題國際標準題庫作為績效評估的測試例題。

本研究首先以循序式的一般化插入法(GENI)產生起始解，再以鄰域搜尋模組改進，其方法共包括解繫法(US)、1-1、2-Opt和Or-Opt等數種。並且以巨網結構的方式建構路線，使其能同時考慮「路線內」與「路線間」的交換。同時，本研究設計了一個能夠擴大解繫法鄰域搜尋範圍機制(Expanded US)，以提高解繫法的搜尋績效。而本研究以C#語言撰寫程式，並在Pentium(R) 4，CPU為3.00GHz的個人電腦測試可回溯式門檻接受模組。結果顯示，傳統上BATA的設定，即門檻回溯比值 $b < 1$ 時，在起始門檻比率0.01、門檻下降比率0.9，以及門檻數列長度180的情況下，14題標準題目的平均誤差可為1.2%。另外，本研究亦發現，若突破傳統BATA限制，而設定門檻回溯比值 $b > 1$ 時，在起始門檻比率0.02附近的範圍，門檻下降比率0.9以上附近的範圍，以及門檻數列長度180的情況下，可找到更佳的结果，其中最低的14題標準題目平均誤差更降低至0.87%。表示此種更為放鬆的設計可能可以找到較佳的解。

最後，再與近年來國際期刊之文獻進行比較。本研究14題中共有7題可找到已知最佳解，而各題最佳結果之平均誤差僅為0.26%，且相比之下求解時間亦頗快速，顯示可回溯式門檻接受法可成為相當不錯的求解架構。

關鍵詞：車輛路線問題、一般化插入法、解繫法、可回溯式門檻接受法

A Metaheuristic Using BATA and GENIUS for Vehicle Routing Problems

Student: Yu-Jie Liao

Advisor: Anthony Fu-Wha Han

Department of Transportation Technology and Management
National Chiao Tung University

Abstract

Backtracking Adaptive Threshold Accepting (BATA) was first introduced by Tarantilis et al. in 2001 for the distribution of perishable goods. This algorithm is similar to Threshold Accepting (TA) but the values of threshold are lowered or raised, depending on if an acceptable solution can be found in a fixed number of iterations. This research used a BATA structure embedded with GENIUS and other traditional exchange methods for solving the Vehicle Routing Problem (VRP). And the 14 classic instances described by Christofides et al. were selected for the evaluation of our method.

The first phase of our proposed metaheuristic is the following. A feasible initial solution was generated by a sequential GENI, and then improved by neighborhood search methods such as US, 1-1, 2-Opt, Or-Opt. Since we presented the solution as a giant tour, both the inter-route and intra-route improvements were considered simultaneously. We also proposed a mechanism named "Expanded US" to enhance the performance of US.

In the second phase, we applied BATA to further improve the giant-tour solution. We coded our metaheuristic method in C# and implemented all experiments on a Pentium 4, 3.00GHz personal computer. Results showed that the average deviation of 14 benchmark instances can be 1.2% using traditional BATA parameter $b < 1$. We also tested the case of the threshold backtracking factor $b > 1$ and found that this change could lead to even better results. The average of deviation of the 14 benchmark instances can be reduced to 0.87%. Overall, compared with the recent literatures, among the 14 instances, we found 7 best-known solutions and the average deviation is merely 0.26%. The average computer time is about 50 seconds which demonstrated the efficiency and potential applicability of our proposed method.

Keywords: Vehicle Routing Problem, GENIUS, BATA

誌 謝

論文的完成首要感謝恩師 韓復華 教授兩年來悉心的指導與教誨。老師的淵博學識與處事態度，在佛法的浸潤下更顯輝揚，也讓學生在探索學問上能有更深度的體會，在此致上由衷的謝忱！老師的恩情與給予的鼓勵，將永銘於心。

在論文研究期間，非常感謝中華大學 卓裕仁 教授。卓老師無私的教導與不辭辛勞的前來，引領學生平穩地跨入 VRP 的領域；老師隨和的態度與幽默的話語，總是讓學生對於克服困境有著光明的期待。在此祝福老師身體健康、學術生涯愉快。

論文口試期間，承蒙中華大學 卓裕仁 教授、元智大學 丁慶榮 教授不吝指教與斧正，提供許多寶貴的意見與有趣的觀點，使本論文能得到更多的補足，在此特別感謝。論文審查期間，感謝系上老師 王晉元 教授、黃寬丞 教授，細心的閱讀並給予意見，在此由衷表示謝忱。

論文完成的時刻，也代表了學生生涯在此告一段落，能在待了六年的交大結束，特別是最後兩年的網路實驗室大家庭，無疑是相當幸運的，要對一路上協助過我的人說聲感謝。感謝系上師長、學長姐與學弟妹的照顧。感謝系上助理：幸榮姐與秀蔭姐等人於行政事務上的幫忙。感謝十樓的樓友們給予的熱鬧與歡笑。感謝實驗室的學長姐，中距離超準的威哥，在籃球場上真是個好隊友。擁有百分之百無失誤正妹雷達感應裝置的俊德，感謝你總是盡心盡力的提供幫忙。經常語出驚人的裕智與俐論，是實驗室的奇景，再搭配昆論爽朗有力的笑聲、育廷阿牛式的憨厚微笑，以及淑詩無奈受窘的表情，總是替生活添加了更多精采。感謝同窗好友們，袁大帥，好吧，最後就這麼叫你了，說真的，六年了你還是我心目中國語說最好的僑生，不如就真的留下來了吧，哈哈。美食達人阿春，當初要不是你的鐵口直斷，也就不會進入網路實驗室了，很高興又多當了兩年同窗。耍色大美女小小賴，要不是你的調格式功力，這篇論文看起來就不像篇文章了吧。另外，謝謝學弟妹佑甯、仲豪和沂茹帶來的歡笑與樂趣，希望你們的論文也能順利。最後，sumire、阿誠、webber 和永朋要感謝你們的衝勁，能一起在半個台灣中都留下我們的足跡。

還有系足的夥伴們，在大一時被強拉進這個在台灣冷門的運動之後，為了四年一度的世界盃而瘋狂，在足球場上揮灑青春的汗水直到跑不動，最後兩年解決苦悶的最好方式，幸好有你們讓我捨不得離開這個隊伍。

祝福你們的未來都很好。

最後，要感謝我的家人，爸爸、媽媽、哥哥、弟弟們，感謝你們的支持與鼓勵，讓我於求學生涯沒有後顧之憂。感謝舅舅，為我們拍攝的畢業照，是此刻人生中最美麗的註腳感謝。感謝女友，多年相處幸虧有你的容忍與體諒。

要感謝的人太多了，願將此小小的成就與喜悅之情與你們共同分享，在未來人生彼此再繼續努力，再共同感受生命的脈動。

廖昱傑 謹誌

于 交大網路實驗室 2007.07

目 錄

中文摘要	i
英文摘要	ii
誌 謝	iii
目 錄	iv
表 目 錄	vi
圖 目 錄	vii
第一章 緒論	1
1.1 研究動機與目的	1
1.2 研究內容與範圍	1
1.3 研究方法與流程	2
第二章 文獻回顧	4
2.1 車輛路線問題	4
2.2 啟發式解法回顧	5
2.2.1 傳統的啟發式解法	6
2.2.2 一般化插入法與解繫法(GENIUS)	9
2.2.3 巨集啟發式解法	13
2.3 可回溯式門檻接受法	16
2.4 小結	17
第三章 起始解與鄰域搜尋法構建	18
3.1 起始解產生：一般化插入法(GENI)	18
3.2 鄰域搜尋模組之設計	18
3.2.1 於巨網結構下(Giant Tour)下執行鄰域搜尋	18
3.2.2 擴大解繫法(US)鄰域搜尋範圍之設計	20
第四章 BATA 應用於 VRP 之設計與測試	21
4.1 BATA 與 GENIUS 結合之設計	21
4.1.1 門檻值回溯幅度之探討	21
4.1.2 執行架構說明	22
4.2 測試例題說明	24
4.3 測試結果之整理與分析	25
4.3.1 實驗設計	25

4.3.2	起始解測試.....	25
4.3.3	鄰域搜尋模組測試.....	26
4.3.4	可回溯式門檻接受模組測試.....	29
4.4	演算法績效比較與分析.....	35
第五章	結論與建議.....	37
5.1	結論.....	37
5.2	建議.....	37
參考文獻	38
附 錄	最佳個案結果與路線圖.....	42



表目錄

表 2.1	多種巨集啟發式解法之解題策略比較.....	14
表 2.2	SA、TA、GDA 與 RRT 等方法之比較.....	15
表 4.1	VRP 測試例題.....	25
表 4.2	起始解之誤差(單位：%).....	26
表 4.3	以包含 US 之鄰域搜尋模組改善後之誤差(單位：%).....	27
表 4.4	以包含 Expanded US 之鄰域搜尋模組改善後之誤差(單位：%).....	27
表 4.5	單一組參數之最佳結果與測試過程中之最佳結果.....	34
表 4.6	演算法績效比較表.....	36



圖目錄

圖 1.1	研究流程圖	3
圖 2.1	VRP 的路線型態	4
圖 2.2	節省法示意圖	6
圖 2.3	插入法示意圖	6
圖 2.4	2-Opt 的概念示意圖	7
圖 2.5	Or-Opt 的概念示意圖	8
圖 2.6	1-0 交換法的概念示意圖	8
圖 2.7	1-1 交換法的概念示意圖	8
圖 2.8	1-2 交換法的概念示意圖	9
圖 2.9	Type I 插入法之示意圖	10
圖 2.10	Type II 插入法之示意圖	10
圖 2.11	Type I Unstring 之示意圖	11
圖 2.12	Type II Unstring 之示意圖	12
圖 2.13	TA、GDA 與 RRT 接受法則示意圖	15
圖 2.14	TA 門檻數列遞減型態	16
圖 2.15	BATA 門檻數列收斂示意圖	16
圖 3.1	巨網結構的路線示意圖	19
圖 3.2	路線間的解繫法示意圖	19
圖 4.1	回溯幅度大小對於搜尋影響示意圖	21
圖 4.2	本研究巨集啟發式解法求解架構流程圖	23
圖 4.3	BATA 核心搜尋細部流程	24
圖 4.4	不同鄰域搜尋模組之平均誤差(%)	28
圖 4.5	不同鄰域搜尋模組之平均耗費時間(秒)	28
圖 4.6	不同 K 值與 b 值對於求解結果的影響($T_0 = 0.01, r = 0.9$)	29
圖 4.7	門檻值與暫時最佳解變化圖—以例題 2 為例	30
圖 4.8	不同 K 值與 b 值所需之平均耗費時間(秒)($T_0 = 0.01, r = 0.9$)	31
圖 4.9	不同 r 值對於求解結果的影響($T_0 = 0.01$)	32
圖 4.10	不同 T_0 值對於求解結果之影響	34

第一章 緒論

1.1 研究動機與目的

目前全球商業的競爭態勢已然改變，從企業對企業的競爭逐漸轉為供應鏈(Supply Chain)對供應鏈的競爭。在供應鏈內上、下游不同的企業組織均需配合整個供應鏈的流程與活動，以降低成本，提升效率，並共享其利潤。而物流作業為供應鏈相當重要的一個環節，並佔有不少的成本；為滿足企業的需要，乃有整合型物流服務業者的興起，為企業提供了物流、金流和資訊流各方面之解決方案。而配送物料實體的車輛路線安排與規劃仍然是其諸多考量中不可忽視的一個課題。

車輛路線問題(Vehicle Routing Problem, VRP)與旅行銷售員問題(Traveling Salesman Problem, TSP)是最基本的兩種問題型態。隨著實務狀況的不同，更衍生出許多更複雜的應用類型。這類問題的複雜程度屬於NP-Hard；在問題規模很大時，通常以啟發式解法來求解，冀求能在有效率的時間內求得精確度很高的近似解。目前，巨集啟發式解法已成為求解這類問題的主要方法。常見的類型如：禁忌搜尋法(Tabu Search, TS)、演化法(Evolutionary algorithm, EA)、門檻接受法(Threshold Accepting, TA)和螞蟻演算法(Ant Colony Optimization, ACO)等等。這些方法均有避免陷入局部最佳解的機制，在搜尋的過程中能幫助獲得更好的解。其中，門檻接受法具有解題精確度高，運算時間短等特點，亦已應用在各式的網路組合最佳化問題。

Tarantilis et al. [38]提出可回溯式門檻接受法(Backtracking Adaptive Threshold Accepting, BATA)，其與傳統的門檻接受法同樣採取較為寬鬆的接受法則(為一門檻值)，使其接受劣於現解的鄰解，有助於在搜尋時能跳脫局部最佳解束縛。但其收斂型態則非單調的逐漸遞減，而允許有門檻值回溯的情形發生；即找不到一個可接受的暫時解時，可將門檻值予以放鬆再重新搜尋。這種設計對於VRP的解題績效相信會有更佳的结果，惟其程度為何，頗值得探討。

同時，門檻型演算法所採用的局部搜尋方法(Local Search)，或是鄰域搜尋法(Neighborhood Search)，對於解的品質優劣有著關鍵的影響。假若不能快速地找出局部最佳解，最終也無法獲得優良的解。另一方面，由於每種鄰域搜尋方法對於鄰域(Neighborhood)的概念皆有所不同，搜尋方向與範圍也跟著不同。若適當地結合數種不同的鄰域搜尋方法，相信將可以對於其鄰近點進行綿密而有效的搜尋，而迅速獲得局部最佳解。而本研究選擇GENIUS此著名的TSP啟發式解法加以調整，並搭配其他不同類型的交換法，以達成上述目的。

綜合以上所述，本研究之目的即在以此種新型的門檻接受法為基本求解架構，探討其應用在VRP上的解題精確度與速度。並由前述之鄰域搜尋概念，以GENIUS與其他類型交換法互相搭配輔助，建構出一套有效率的求解方法，期能再推廣至其他車輛路線問題。

1.2 研究內容與範圍

本研究是以可回溯式門檻接受法作為VRP主要的解題架構，並結合GENIUS與其他鄰域搜尋法工具，發展成一套有效率的巨集啟發式解法。故本研究之內容與範圍為：

- (1) 以 C#語言為 GENIUS 與其他交換法撰寫程式；並於國際標準題庫找尋 VRP 測試例題，以測試其解題的效果與應用限制，作為日後 BATA 啟發式解法架構選擇參考。
- (2) 建立以 BATA 為架構的巨集啟發式解法，並以 C#語言撰寫程式，選擇 VRP 測試例題求解，以對其整體解題績效作進一步比較分析，並對應用時所設定的參數與求解架構提出建議。
- (3) 對發展 BATA 求解架構所遭遇到的限制與發現做一整理，以供進一步修改或後續研究之參考。

1.3 研究方法與流程

本研究的重點在於以可回溯式門檻接受法作為 VRP 主要的解題架構，結合 GENIUS、2-Opt、Or-Opt、1-0、1-1 等著名的鄰域搜尋法解題工具，發展成一套有效率的巨集啟發式解法。因此，本研究之研究流程與執行步驟，將如圖 1.1 所示，並分述如下：

(1) 相關文獻蒐集與回顧

回顧 VRP 相關解法之發展，以及目前國內外關於 GENIUS 與 BATA 等啟發式解法之應用情況。

(2) 測試題庫蒐集

於網路上蒐集標準國際題庫，如 [VRP Web](#)[43]，可提供相關例題與結果。整理例題後供測試與評估績效之用。

(3) 啟發式解法方法論彙整

彙整目前所發展的 GENIUS 與其他各式啟發式解法之方法論，並加以分析，提供未來巨集啟發式解法選擇組合的參考。

(4) 起始解構建

採用一般化插入法(Generalized Insertion, GENI)建立起始解構建模組，並針對 VRP 的問題特性加以調整。

(5) 改善模組

以解繫法(Unstring-and-String, US)與其他鄰域搜尋法建立改善模組。此部份為針對前一步驟所得之起始解進行改善，獲得一個更佳的解。

(6) 巨集啟發式解法

以可回溯式門檻接受法為架構，並結合解繫法與其他鄰域搜尋法，作為本研究 VRP 巨集啟發式解法的求解架構。

(7) 實驗設計與參數設定

以不同的參數設定與執行架構以試驗其對於問題變異的敏感程度。

(8) 測試結果比較與分析

以標準的國際題庫進行例題測試，藉以獲得較佳的參數設計，與了解其解題特性與

績效，最後從中探討此求解模組的發展可行性與應用潛力。

(9) 結論與建議

根據前述各步驟的結果，整理出具體的結論，並就未來可能的重點項目與後續研究提出建議。

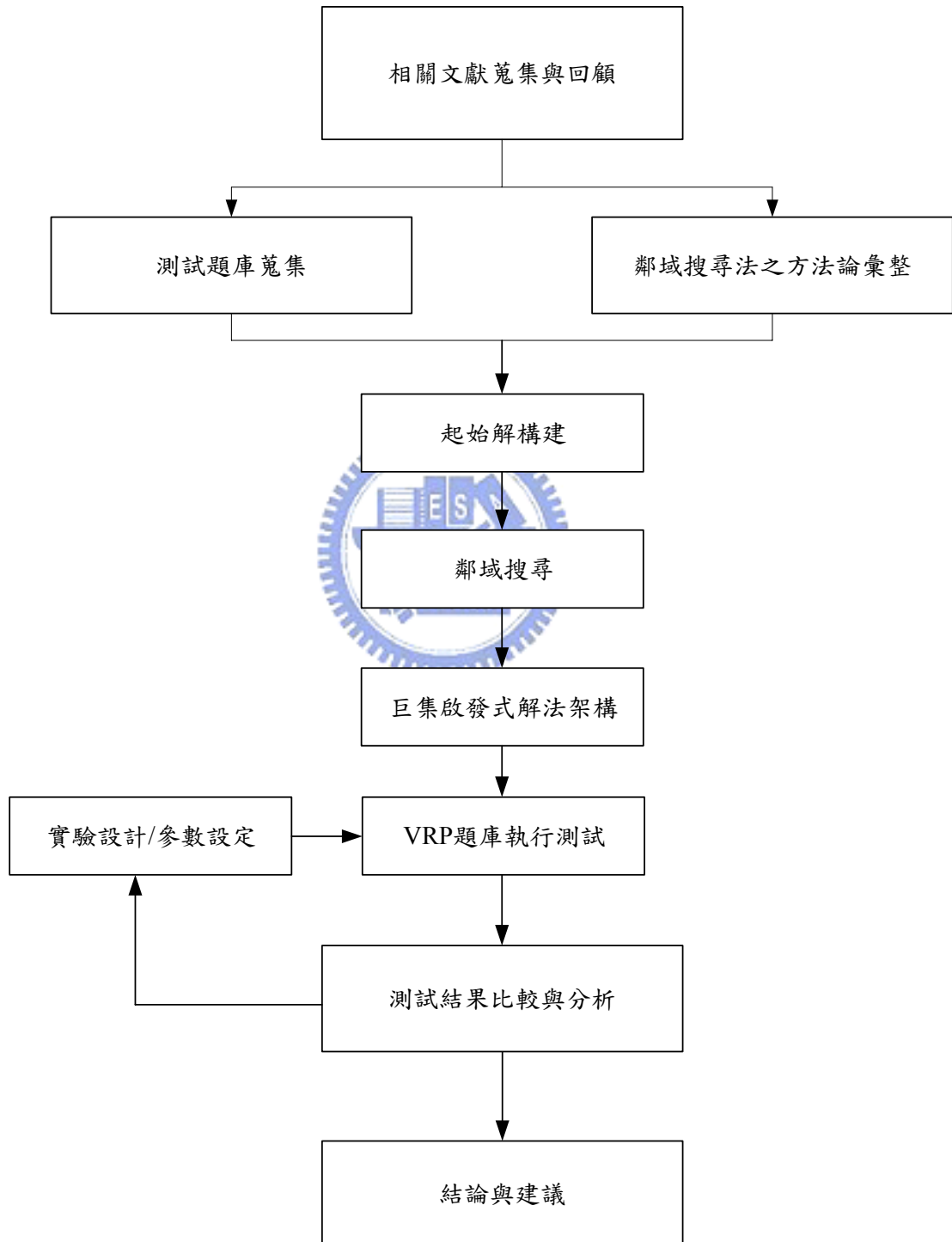


圖 1.1 研究流程圖

第二章 文獻回顧

2.1 車輛路線問題

車輛路線問題(Vehicle Routing Problem, VRP)，在運輸領域上是相當實用且重要的一個問題。於 1959 年由 Dantzig and Ramser[14]所提出後，一直是網路最佳化問題(Network Combinatorial Optimization Problems)最基本的問題型態之一。

VRP 的基本定義為：「給定一個路網 $G = \{N, L\}$ ，其中 N 為節點(Node)的集合， L 為節線(Link)集合，節點 0 代表場站， d_i 表各節點的需求量，每輛車的容量為 K_v 。定義車輛路線問題即是找出成本最小的車輛路線並滿足以下限制式：

- (1) 每一個節點都只由一部車服務一次。
- (2) 每輛車都從場站出發又回到原場站。
- (3) 每輛車所服務的節點需求量總和不超過車輛的容量。
- (4) 每一個節點的需求量都需被滿足。」

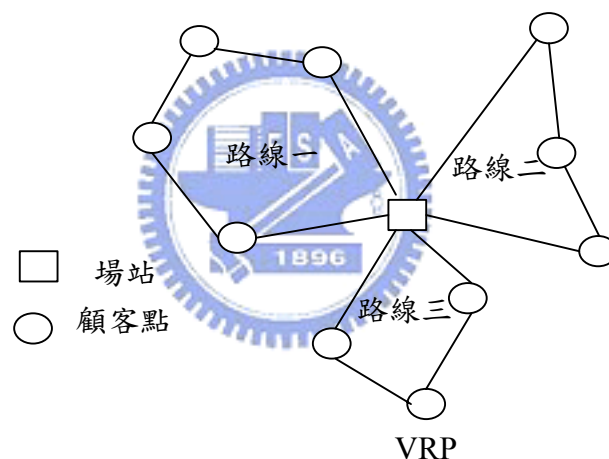


圖 2.1 VRP 的路線型態

資料來源：卓裕仁[2]

圖 2.1 為 VRP 的路線型態，表示 VRP 乃是單一場站、單一車種，多輛車且有容量限制的服務型態。VRP 常見的限制條件還有旅行時間限制，即每條路線所運行的時間(包含在路段的時間與服務時間)不超過所規定的時間。而若只有車容量限制時，常常被稱為 CVRP(Capacitated Vehicle Routing Problem)(Cordeau et al. [13])。CVRP 的數學規劃模式可如下表示(Bodin et al. [6])：

$$\text{Min} \quad \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^{MV} c_{ij} x_{ij}^v \quad (1)$$

s.t.

$$\sum_{i=0}^n \sum_{v=1}^{MV} x_{ij}^v = 1 \quad (j = 0, \dots, n) \quad (2)$$

$$\sum_{j=0}^n \sum_{v=1}^{MV} x_{ij}^v = 1 \quad (i = 0, \dots, n) \quad (3)$$

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad (v = 1, \dots, MV; p = 1 \dots n) \quad (4)$$

$$\sum_{i=1}^n d_i \left(\sum_{j=0}^n x_{ij}^v \right) \leq K_v \quad (v = 1, \dots, MV) \quad (5)$$

$$\sum_{j=0}^n x_{ij}^v \leq 1 \quad (v = 1, \dots, MV) \quad (6)$$

$$\sum_{i=0}^n x_{ij}^v \leq 1 \quad (v = 1, \dots, MV) \quad (7)$$

$$X \in S \quad (8)$$

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

其中，(1)式為總路線成本最小的目標式；(2)式與(3)式限制每個顧客由一輛車服務為、(4)式為流量守恆限制式，表示服務顧客點 p 的車輛皆是同一輛；(5)式為容量限制式；(6)與(7)限制每輛車僅能使用一次。(8)式為子迴路限制式，其中 S 可表為：

$$S = \{(x_{ij}) : y_i - y_j + nx_{ij} \leq n - 1 \text{ for } 2 \leq i \neq j \leq n \text{ for some real numbers } y_{ij}\}$$

(9)式表示決策變數 x_{ij}^v 為二元變數，當節線 (i, j) 由車輛 v 所服務即為 1，否則為 0。

上述之數學規劃模式並非 VRP 唯一的表示方式，其它類型的 VRP 數學規劃模式可參考 Chrisdofides et al. [8]、Golden and Assad[24]、Lawler et al. [27]及 Cordeau et al. [13] 等文獻。

由於 VRP 屬於 NP-Hard 的問題，雖然目前已發展出許多精確解的解法，如分支定限法(Branch and Bound)、拉式鬆弛法(Lagrangian Relaxation)、切割平面法(Cutting Plane)、分支切面法(Branch and Cut)、變數產生法(Column Generation)等等，但在節點數多的時候，幾乎無法在有效率的時間(Polynomial time)求出解來。為應付實際需要，許多研究便致力於啟發式解法的發展，如下節所述。

2.2 啟發式解法回顧

發展啟發式解法的目的為在一合理的時間內求算出精確度很高的解。又可分成傳統的啟發式解法與 1990 年代後興起的巨集啟發式解法。將於以下小節分述。

2.2.1 傳統的啟發式解法

傳統求解 VRP 的啟發式解法可約略分成兩階段來討論，其一為構建法(Construction Methods)，另一則為改善法(Improvement Methods)。

(一) 構建法：

構建法為最初所發展的啟發式解法，目的在於根據一套特定的規則，從無到有逐步地將所有的顧客點插入路線中，完成一個可行解。因此，構建法包含了三個要件：起始準則(Initialization Criterion)、選擇準則(Selection Criterion)與插入準則(Insertion Criterion)。選擇準則為選擇欲插入顧客點的依據，插入準則決定了插入的位置(Cordeau et al. [13])。構建法又可進一步細分為循序(Sequential)與平行兩種(Parallel)。循序法將一條路線構建完成之後才跳至另一條路線，平行法在插入顧客點時同時考慮了一條以上的路線。在各種構建法中，最常被使用的方法為節省法(Savings Methods)與插入法(Insertion Methods)。

節省法由 Clarke and Wright[9]所提出，其概念如圖 2.2 所示，先計算出所有的顧客點對 (i, k) 之節省值 (S_{ik}) ；節省值公式為 $(i, 0)$ 、 $(0, k)$ 兩條節線之成本減去 (i, k) 之成本，其意涵為合併 i 與 k 兩節點所能減少的路線成本。接著依據節省值大小進行排序，得到節省值列(Saving List)。選擇節省值最大者，如圖 2.2 方式優先進行合併，其過程仍需在 不違反車輛容量限制下進行。由此可知，節省法本質上為平行法，同時考慮了多條路線(Cordeau et al. [13])。

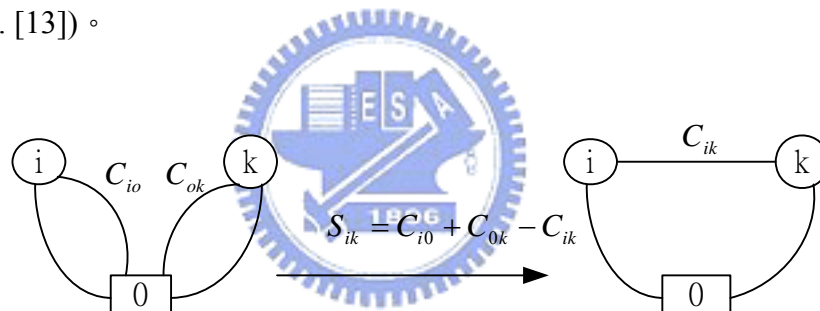


圖 2.2 節省法示意圖

插入法係由 Rosenkrantz et al. [35]所提出，其概念可如圖 2.3 所示：從一條簡單的路線開始，考慮到若插入 k 點，路線會增加了成本 I_{i0}^k ；此時便依路線所增加的成本愈小的準則決定其插入的位置。在不違反車輛容量限制的情況下，逐步加入新顧客點，直至所有顧客點皆已被服務。

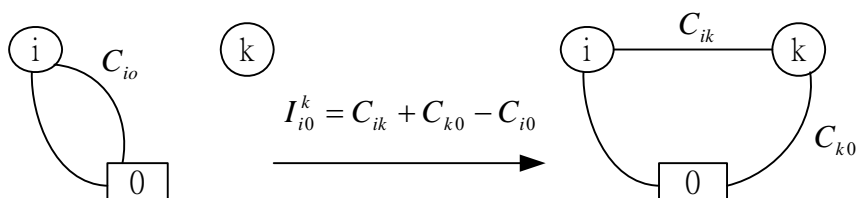


圖 2.3 插入法示意圖

由於節省法與插入法的觀念都非常簡單與容易執行，後續有多位學者採用並針對問題特性改變其計算公式或執行架構等；例如插入法便有最近插入 (Nearest Insertion)、最遠插入(Farthest Insertion)、最省插入(Cheapest Insertion)、最快插入(Quick Insertion)、任意插入 (Random Insertion) 等多種修正方法。詳細內容可參考 Bodin et al. [6] 或 Cordeau et al. [13]之整理。

(二) 改善法：

改善法乃是利用節點與節線的重新安置與交換，而使解獲得改善。因此，在第一步應以一個起始解產生機制產生起始解，並將此起始解設定為現有解。透過一個特定程序針對現有解的鄰域尋找新的解，當有一可行解的目標函數值優於現有解時，將此新解更新為現有解，稱為一次移動(Move)。如此重複此動作，直到找不到無任何可改善的可行解即停止，最後的現有解為所找到的區域最佳解。

VRP 路線改善的方法通常可約略分成節線交換(Arc Exchange)與節點交換(Node Exchange)兩種類型。節線交換法中，K-Opt 節線交換法為常用的交換法，由 Lin[28]於 1965 年提出，K 表示為每次交換的節線數，且節線間互不相鄰。K 值愈大，可行的交換型態愈多愈複雜，因此 K 值通常取 2 或 3。圖 2.4(a)顯示 2-Opt 的解題概念：原本的 $(i, i+1)$ 與 $(j, j+1)$ 兩條節線被移除，而以 (i, j) 與 $(i+1, j+1)$ 兩條節線來替代。

經由交換節線後，若找到一個可改善目標值的鄰解，即以此鄰解更新為現有解，在重覆此鄰域搜尋的動作直至找不到可改善的解為止。由於 K-Opt 原本設計用於 TSP 問題，於 VRP 問題上可被用來作為各路線內的節線交換。然而，2-Opt 亦可用於路線間的節線交換，如 Tarantilis et al. [38]所採用的方式。如圖 2.4(b)所示，可看出路線間的 2-Opt 交換，是以不同的兩路線各抽出一條節線，再分別插入一條節線，完成兩條新的路線。

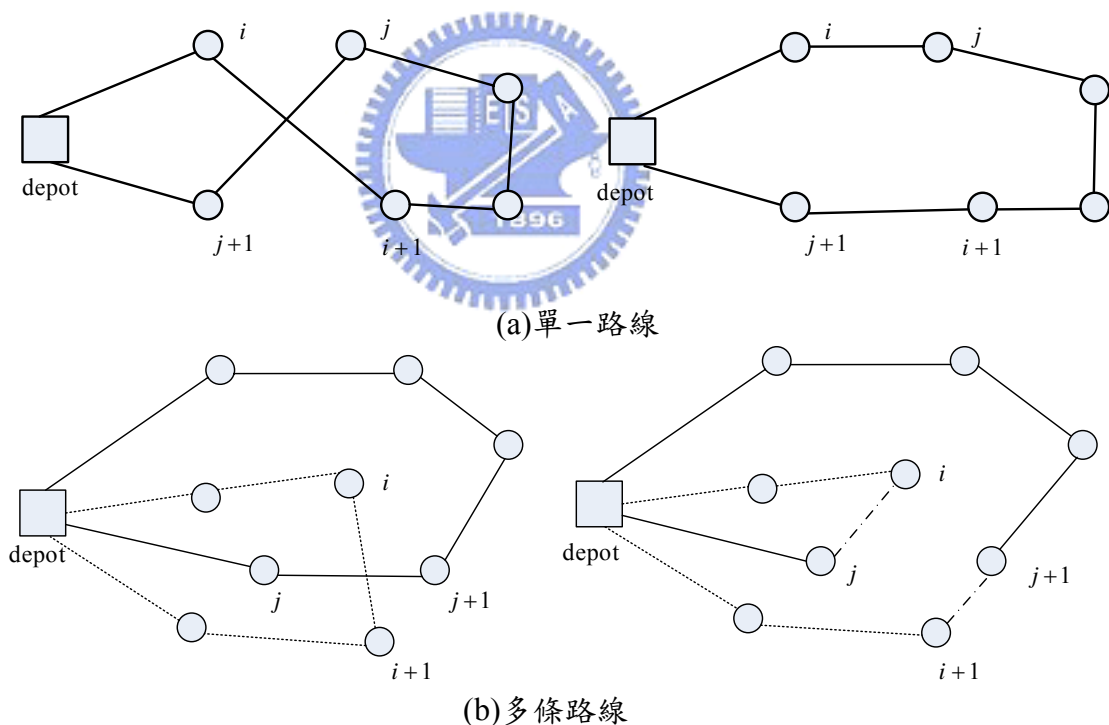


圖 2.4 2-Opt 的概念示意圖

資料來源：Tarantilis et al. [38]

由於 2-Opt 與 3-Opt 的交換方式都會造成某些節線產生需要反轉的情況，造成計算上的負擔，Or[32]便提出了 Or-Opt 節線交換法，被視為是一種 3-Opt 的變形，它可將某一節點、某一段節線、或某兩段節線自路線抽出，再插入至其他節線之間。如圖 2.5 所示。 $(i, i+1)$ 、 $(j, j+1)$ 與 $(k, k+1)$ 三條節線被斷開，而以 $(i, j+1)$ 、 (k, j) 與 $(i+1, k+1)$ 三條節線替代，重新連成一條迴路。可看到 $(i+1, j)$ 這段節線從路線中被抽出，接著再插

入至 k 與 $k+1$ 之間。

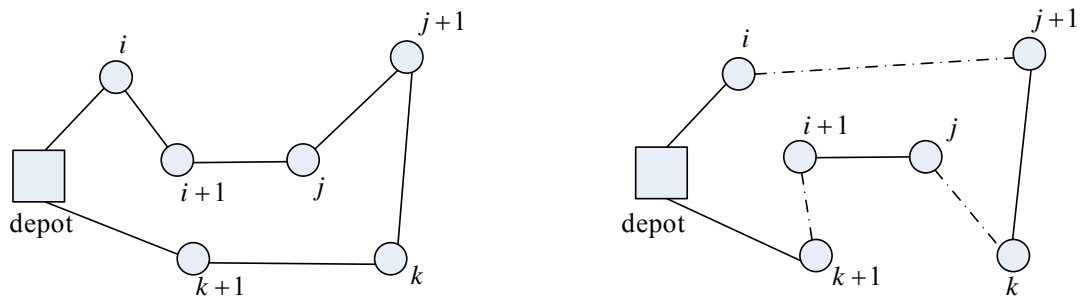


圖 2.5 Or-Opt 的概念示意圖

資料來源：卓裕仁[2]

節點交換方法選取了任兩條路線，並相互交換其部分節點，檢查交換後的結果是否能維持可行並獲得改善。常見的方式則有 1-1、1-0、1-2 等數種。由於是路線之間的交換，牽涉到車輛容量的限制，可能使得交換不容易發生。圖 2.6 為最簡單的 1-0 交換的解題觀念。

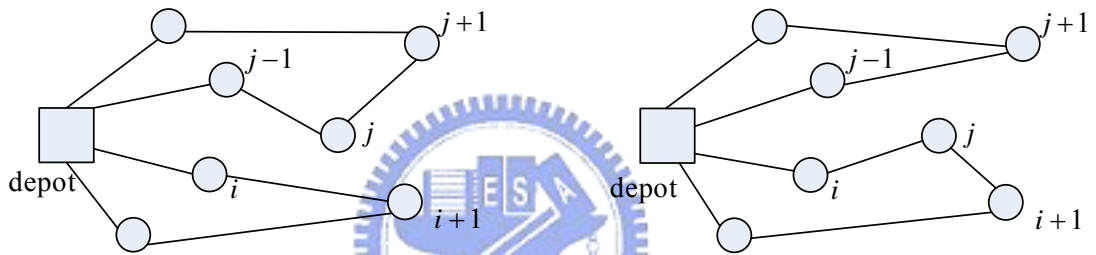


圖 2.6 1-0 交換法的概念示意圖

資料來源：Tarantilis et al. [38]

Osman[33]將 1-0 交換法稱為一個轉移過程(Shift Process)，而將 1-1、1-2 稱之為 λ -interchange， λ 表示了牽涉到的節點數。圖 2.7 為 1-1 交換法的解題概念：原本屬於路線 A 的 i 點與路線 B 的 j 點，先從原有路線斷開後，再分別重新插入彼此之前的位置。1-0、1-1 節點交換亦可允許在同一條路線內發生，而不限定為不同路線，如 Tarantilis et al. [38]所採用的方式。

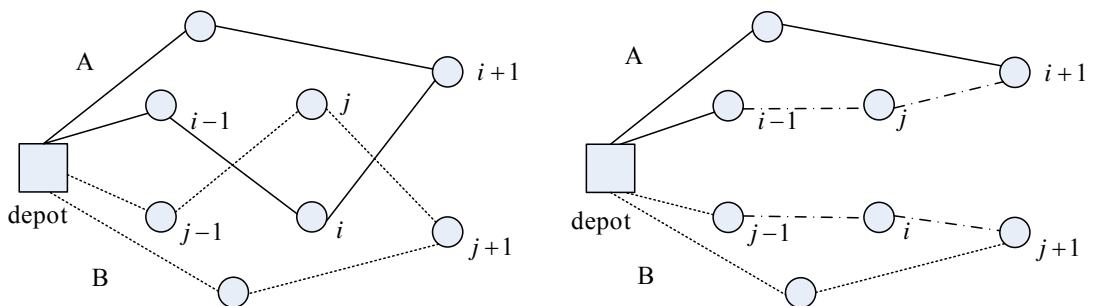


圖 2.7 1-1 交換法的概念示意圖

資料來源：Tarantilis et al. [38]

1-2 交換法的解題概念如圖 2.8 所示，原本屬於路線 B 的 s 點插入了路線 A；路線 A 的 j 點與路線 B 的 i 點則互相交換，分別插入了彼此之前的位置。

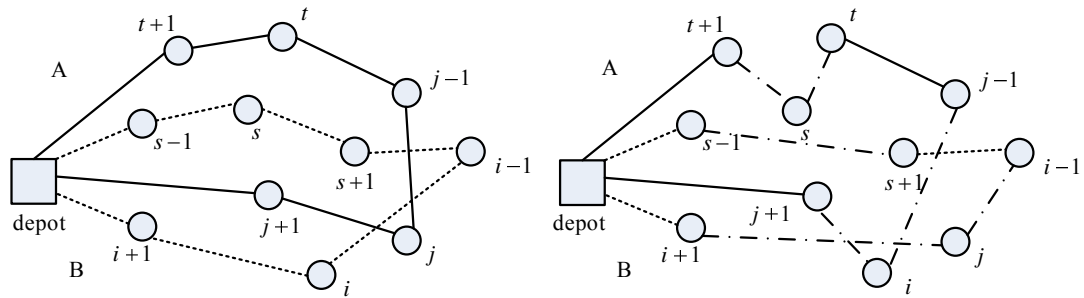


圖 2.8 1-2 交換法的概念示意圖

資料來源：陳國清[3]

另外，由 Gendreau et al. [21]於 1992 年提出一般化插入/解繫法(Generalized Insertion/Unstring and String, GENIUS)求解 TSP 問題，它包含了 GENI 與 US 兩部分，GENI 可用作一起始解構建的方法，US 則可作為路線改善的方法，其內容如下小節詳述。

2.2.2 一般化插入法與解繫法(GENIUS)

傳統的插入法要將節點 v 插入兩點中間時，此兩點必須為相鄰的，即此兩點在路線中是以節線直接相連。而 Gendreau et al. [21]所提出一般化插入法則放寬了此項限制，可以選擇不相鄰的兩點來插入，並設計出 Type I 和 Type II 兩種形式，解說如下：

首先，在一條給定方向的路線(順時針或逆時針)，欲將節點 v 插入 v_i 與 v_j 之間，規定 v_k 為 v_j 至 v_i 路徑中的一點， v_l 為 v_i 至 v_j 路徑中的一點。而對任意點 v_h ， v_{h-1} 視為其上游點， v_{h+1} 視為其下游點。

Type I 的插入法可以圖 2.9 表示，其中實線表示為連接的路徑，虛線則表示為節線。在 v_i 、 v_j 分別和其下游點 v_{i+1} 、 v_{j+1} 所連結的節線斷開，轉而與 v 相連。此時必須找到一點 v_k ，使其與下游點 v_{k+1} 所連接的節線也斷開來，再分別與 v_{i+1} 、 v_{j+1} 相連結。就形成一條包含 v 點的新路線。值得注意的是， v_{i+1} 至 v_j 與 v_{j+1} 至 v_k 這兩條路徑的方向發生了翻轉的情形。

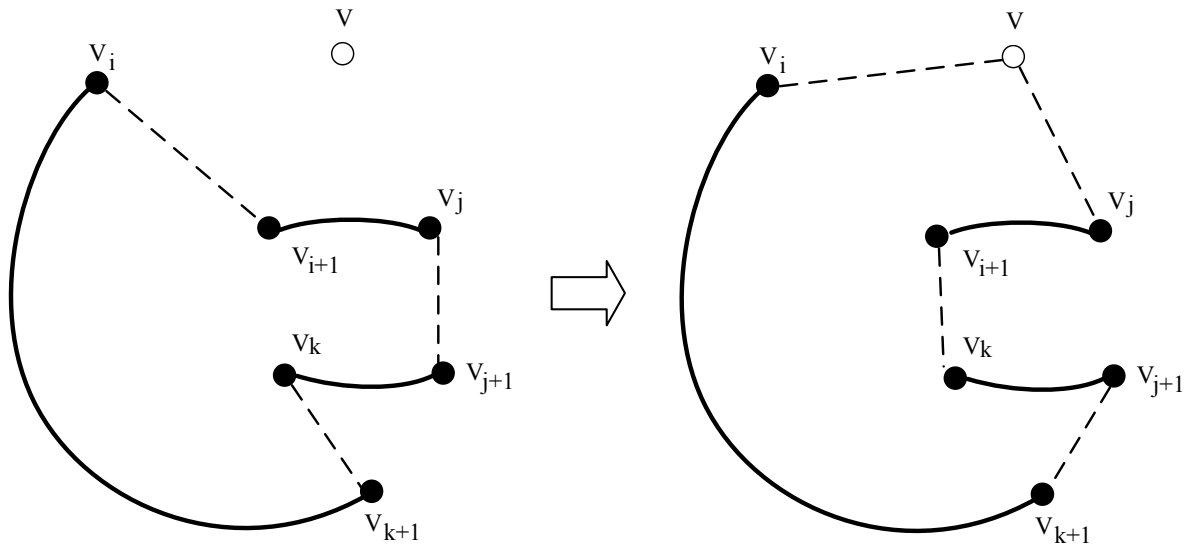


圖 2.9 Type I 插入法之示意圖

資料來源：Gendreau et al. [21]

Type II 的插入法則如圖 2.10 所示，同樣地， v_i 、 v_j 分別和其下游點 v_{i+1} 、 v_{j+1} 所連結的節線斷開，與 v 點相連。此時需選擇 v_k 、 v_l ，使其與上游點 v_{k-1} 、 v_{l-1} 的節線斷開。 v_{i+1} 、 v_{j+1} 再分別與 v_k 、 v_l 相連。最後將 v_{k-1} 、 v_{l-1} 連接，即形成一條包含 v 點的新路線。而 v_j 至 v_l 與 v_{i+1} 至 v_{l-1} 這兩條路徑的方向也被翻轉過來了。

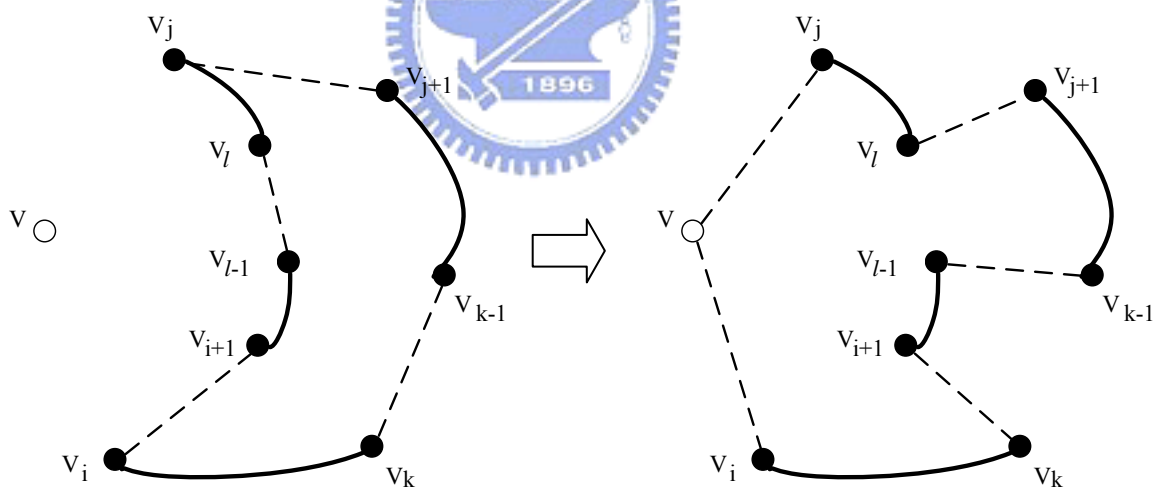


圖 2.10 Type II 插入法之示意圖

資料來源：Gendreau et al. [21]

Gendreau et al. [21] 以 $N_p(v)$ 表示為規定對任一點 v 的鄰近點集合(P-neighborhood)，其定義為在現有路線中的節點與 v 點最為靠近的 P 個點。當任一點 v 欲插入現有路線時，其所選擇的 v_i 、 v_j 屬於 $N_p(v)$ ，而 v_k 與 v_l 分別屬於 $N_p(v_{j+1})$ 和 $N_p(v_{i+1})$ ，以此方式可以其節省選擇的時間。 P 值愈大時，代表可以選擇的範圍也愈大，但是計算時間也變長，可以視需要來調整。

GENI 的演算法如下所述：

- 步驟 1. 任意選擇 3 點構建起始路線，初始化所有節點的 P-neighborhood。
- 步驟 2. 選擇尚未在路線內的任一點，需考慮路線的方向，與兩種形式的插入法。選出其中具有最小成本的方式插入。最後更新 P-neighborhood。
- 步驟 3. 所有的點都在路線中，則停止。否則重複步驟 2。

Gendreau et al. [21]亦利用此模組發展出一套路線改善的方法。即所謂的解繫法(US)。與一般交換法不同的是，解繫法的概念共有兩個階段：其一為將一個節點移出現有的路線，稱為 Unstring；再以前述之 GENI 演算法插入此路線，稱為 String。Unstring 可視為 GENI 方法的反轉(reverse-GENI)，因此亦有 Type I 和 Type II 兩種形式。如下所述：

Type I Unstring 如圖 2.11 所示，在 v_i 被移出路線後，其上游點 v_{i-1} 和下游點 v_{i+1} 分別找到了 v_k 與 v_j 相連， v_k 與 v_j 的下游點 v_{k+1} 、 v_{j+1} 再連起來，重新形成一條路線。從圖中可看出 Type I Unstring 是將 Type I Insertion 的插入步驟顛倒過來。同樣地，Type II Unstring 也是顛倒 Type II Insertion 的步驟，如圖 2.12 所示。

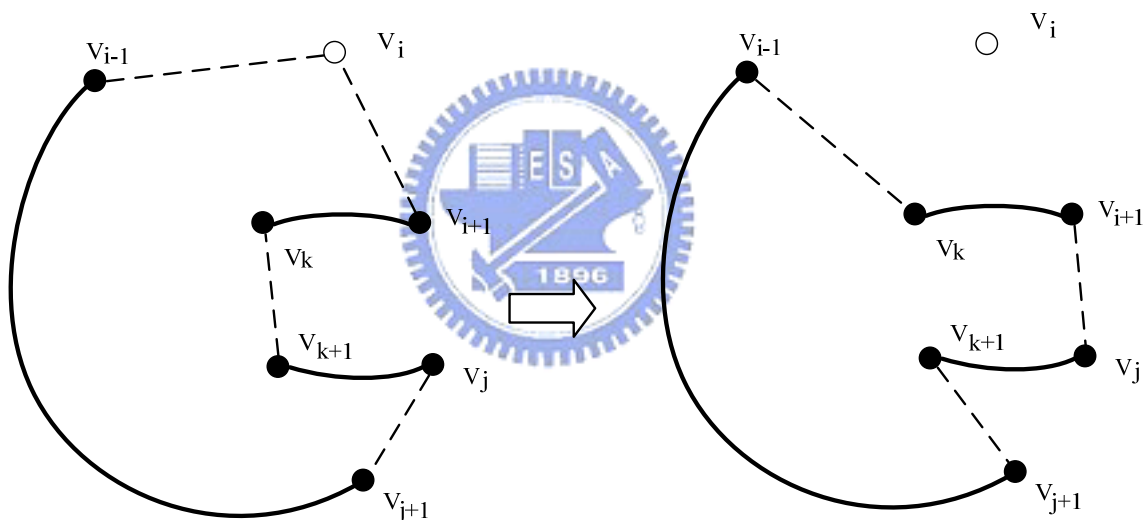


圖 2.11 Type I Unstring 之示意圖

資料來源：Gendreau et al. [21]

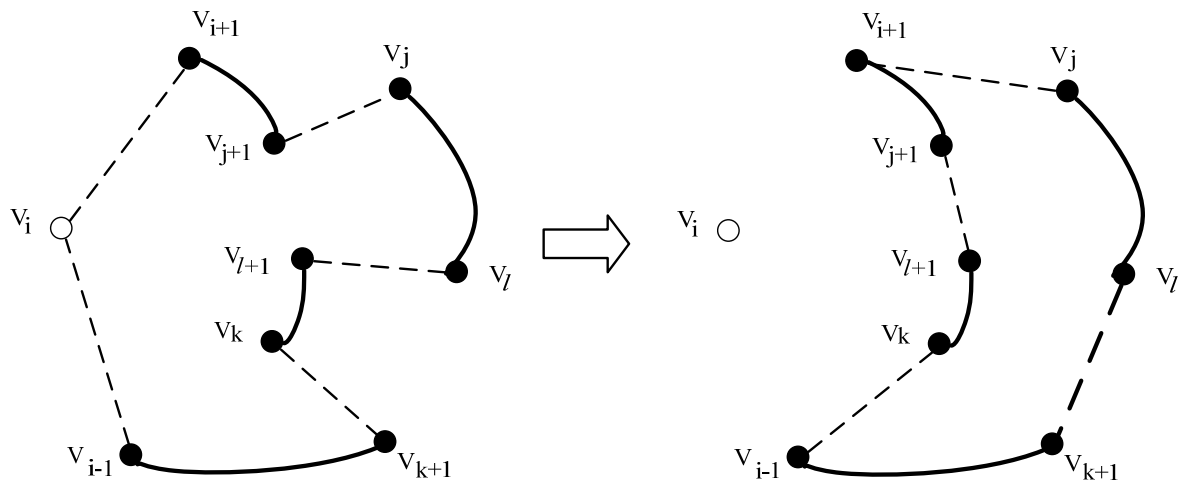


圖 2.12 Type II Unstring 之示意圖

資料來源：Gendreau et al. [21]

而解繫法詳細的執行步驟則可如下所述：

步驟 1. 現有路線 τ 的成本為 z ，設定 $\tau^* := \tau$ ， $z^* := z$ ，以及計數器 $t := 1$

步驟 2. 對 v_i 而言，進行 Unstring 和 String，同樣地需考慮路線的方向，與兩種形式的操作。選擇其中具有最小成本 z' 的路線 τ' ，並使 $\tau := \tau'$ ， $z := z'$ 。

若 $z < z^*$ ，則 $\tau^* := \tau'$ ， $z^* := z'$ ，以及 $t := 1$ ；重複步驟 2。

若 $z \geq z^*$ ，設定 $t := t + 1$ 。

若 $t = n + 1$ ，停止，並回報 τ^* 和成本 z^* ，否則重複步驟 2。

2.2.3 巨集啟發式解法

由於傳統的啟發式解法，必須找到優於所紀錄最佳解的解，才允許更新，因此在執行上容易掉入局部的最佳解。為了克服此缺點，後續的趨勢逐漸演變成以高階的策略作為指引，結合傳統的啟發式解法，來跳脫出局部最佳解的束縛，形成巨集啟發式解法的架構。

巨集啟發式方法發展至今，已演變出許多解題概念不盡相同的方法。較為著名的方法諸如：禁制搜尋法(Tabu Search, TS)、基因演算法 (Genetic Algorithm, GA)、類神經網路 (Neural Network)、螞蟻演算法(Ant Colony Optimization, ACO)、模擬鍛鍊法(Simulated Annealing, SA)等不一而足。

卓裕仁[2]曾整理多種巨集啟發式解法包含的解題策略與觀念，而歸納出數種基本類型。如表 2.1 所示。這些巨集啟發式解法除了上述之巨集啟發式解法，還包括：門檻接受法 (Threshold Accepting, TA)、大洪水法 (Great Deluge Algorithm, GDA)、紀錄更新法 (Record-to-Record Travel, RRT)、變動鄰域搜尋法 (Variable Neighborhood Search, VNS)、噪音擾動法 (Noising Method, NM)、兩極跳躍法 (Flip-Flop, FF)、搜尋空間平滑法 (Search Space Smoothing, SSS)、貪心隨機適應搜尋法(Greedy Random Adaptive Search Procedure, GRASP)，以及分散搜尋法 (Scatter Search, SS) 等等。其策略解釋如下：

- (1) 接受劣解：可接受劣於現有解之鄰解，以便搜尋過程能夠跳出局部最佳解。
- (2) 變換鄰域：於搜尋過程交替更換不同的鄰域搜尋法以產生相異的搜尋空間。
- (3) 擾動成本：擾動成本函數以改變解空間，並在擾動後的解空間繼續搜尋。
- (4) 多重起點：分別自不同的起點（起始解）進行鄰域搜尋。
- (5) 保留資訊：記錄求解過程中之重要資訊，並利用該資訊導引搜尋的方向。
- (6) 自我學習：應用人工智慧的方式，讓方法能夠自搜尋過程中累積求解經驗。

表 2.1 多種巨集啟發式解法之解題策略比較

策略解法	接受劣解	變換鄰域	擾動成本	多重起點	保留資訊	自我學習
禁制搜尋法	✓	✓		✓	✓	
模擬鍛鍊法	✓					
門檻接受法	✓					
大洪水法	✓					
紀錄更新法	✓					
變動鄰域搜尋法		✓				
噪音擾動法			✓			
兩極跳躍法		✓	✓			
搜尋空間平滑法			✓			
跳躍搜尋法				✓		
貪心隨機適應搜尋法	✓			✓		
基因演算法				✓	✓	
分散搜尋法				✓	✓	
類神經網路法					✓	✓
螞蟻演算法					✓	✓

資料來源：卓裕仁[2]

其中，模擬鍛鍊法(SA)、門檻接受法(TA)、大洪水法(GDA)與紀錄更新法(RRT)同屬於門檻型演算法，有著類似的解題策略。此類方法之基本觀念乃是在鄰域搜尋陷入局部最佳解時，鬆綁其接受法則接受劣於現解之鄰解，以便跳脫局部最佳解而能搜尋到更佳的解。因此，SA、TA、GDA 與 RRT 等方法的執行架構與傳統鄰域搜尋法之架構相似，差別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；門檻型演算法則可接受暫劣之鄰解。

SA 可說是 TA、GDA、RRT 的概念來源，其基本觀念最早由 Metropolis 等人於 1953 年所提出，然後由 Kirkpatrick et al. [26]加以應用到組合最佳化問題之求解上。SA 法的接受法則為機率性接受暫劣解：利用一個隨機產生的數值與門檻值做比較，此門檻值是鄰解與現有解之目標值差額及溫度的函數；所謂的「溫度」是做為控制門檻值高低的參數；降溫則是為了使 SA 能夠逐漸收斂(卓裕仁[2])。

TA 法的觀念源自於 SA 法，由 Dueck and Scheuer[16]於 1990 年發表，1993 年 Dueck[15]又根據 TA 的觀念衍生出 GDA 與 RRT。此三種方法皆採用確定性的接受法則，茲以圖 2.13 說明其接受法則的異同：傳統之路線改善法採取嚴格的「接受法則」，即當交換後的可行解之目標值低於原有解時($C(X') < C(X_c)$ ， $C(X_c)$ 為現有解 X_c 之目標值， $C(X')$ 為鄰解 X' 的目標值)，才接受此解。但 TA 法所採用的接受法則為 $C(X') < C(X_c) + T_k$ ， T_k

則是事先產生的一組固定門檻值數列，並依次取用之門檻值；而 GDA 法乃先設定一個起始水位，只要有改善就降低水位（固定的下降速度），其接受法則為 $C(X') < L$ ；至於 RRT 法則是將目前的暫優解設為記錄值，取記錄值之固定百分比率做為門檻值，其接受法則為 $C(X') < C(X_c) + R \times p$ 。

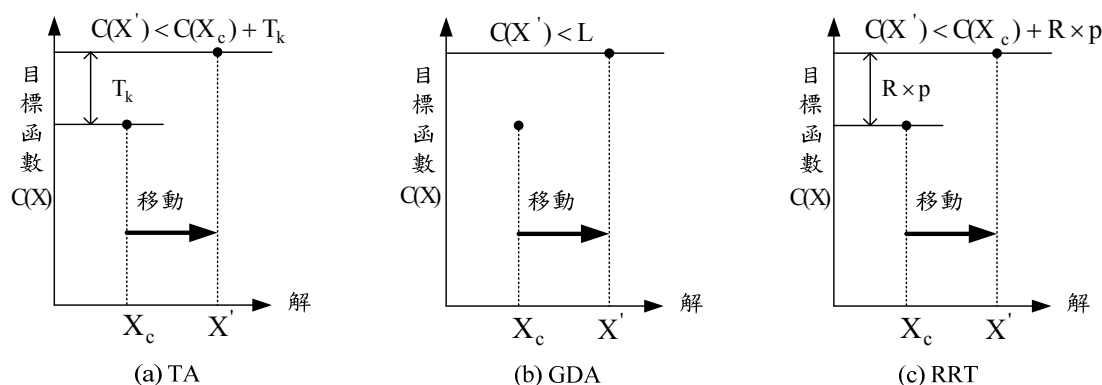


圖 2.13 TA、GDA 與 RRT 接受法則示意圖

資料來源：卓裕仁[2]

表 2.2 說明了 SA、TA、GDA 與 RRT 等方法重要步驟之相似與相異之處。其中「控制參數」指用以控制演算法執行與停止之參數；「接受法則」為判斷是否從現解 X_c 移動至某鄰解 X' 之準則；「收斂法則」是為確定搜尋過程會收斂，在現解移動後對其控制參數進行調整之方式；「停止法則」則規範演算法停止搜尋之標準。

表 2.2 SA、TA、GDA 與 RRT 等方法之比較

方 法	SA	TA	GDA	RRT
控制參數	<ul style="list-style-type: none"> ◦ 溫度(T) ◦ 機率值($0 \leq r \leq 1$) ◦ 次數(K) 	<ul style="list-style-type: none"> ◦ 門檻(T) ◦ 次數(K) 	<ul style="list-style-type: none"> ◦ 水位(L) ◦ 速度(D) 	<ul style="list-style-type: none"> ◦ 誤差率($p < 1$) ◦ 記錄值(R) ◦ 次數(K)
接受法則	機率性接受： $r < \exp\left[\frac{C(X_c) - C(X')}{T}\right]$	確定性接受： $C(X') < C(X_c) + T$	確定性接受： $C(X') < L$	確定性接受： $C(X') < C(X_c) + R \times p$
收斂法則	T 遞減	T 遞減	$L = L - D$	更新R值
停止法則	完成K次迴圈	完成K次迴圈	所有 $C(X') \geq L$	完成K次迴圈

資料來源：卓裕仁[2]

楊智凱[4]曾對 TA 架構應用於 VRP 求解做了深入的研究。楊智凱[4]指出，TA 在應用上必須建立一個門檻數列，根據這個門檻數列 TA 可以來決定是否接受新的解，以及何時停止。因此，TA 的概念在於其接受法則與傳統啟發式解法不同，而其影響求解品質的關鍵則在於其門檻數列。

門檻數列有三項組成要素，包括起始門檻、門檻數列長度與門檻數列遞減型態。此三項組成要素皆會影響 TA 整個求解架構的計算時間與解題精確度。起始門檻太高會使得搜尋方向太過發散，太低則不易跳脫局部最佳解。門檻數列的長度愈長，計算時間也愈冗長。門檻數列遞減型態一般則有直線遞減、梯形遞減和等比遞減三種，如圖 2.14 所示。而門檻值若下降的太快，一旦在過程中找不到可接受的解，可能導致之後的執行

也無法搜尋可接受的解，使得 TA 快速的收斂，無法達到預期的效果。由此可知門檻數列的設定關係著 TA 的求解品質。

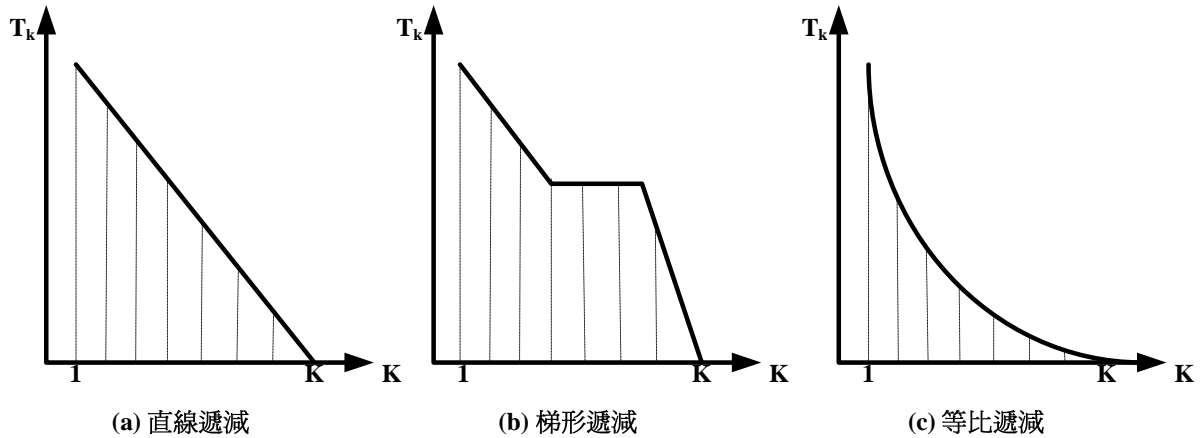


圖 2.14 TA 門檻數列遞減型態

資料來源：楊智凱[4]

2.3 可回溯式門檻接受法

可回溯式門檻接受法(Backtracking Adaptive Threshold Accepting, BATA)是由 Tarantilis et al. [38]所提出的一種改良式門檻接受法。它與傳統的門檻接受法最大的不同在於門檻數列的型態：以往門檻接受法的門檻數列皆呈現出逐漸遞減的型態。而可回溯式門檻接受法在完成一次搜尋迴圈之後，若是有找到至少一次的可接受解，即可降低門檻；反之，則必須放鬆門檻值再進行交換。如此門檻數列的型態，便呈現出有增有減而逐漸收斂的情形。如圖 2.15 所示。

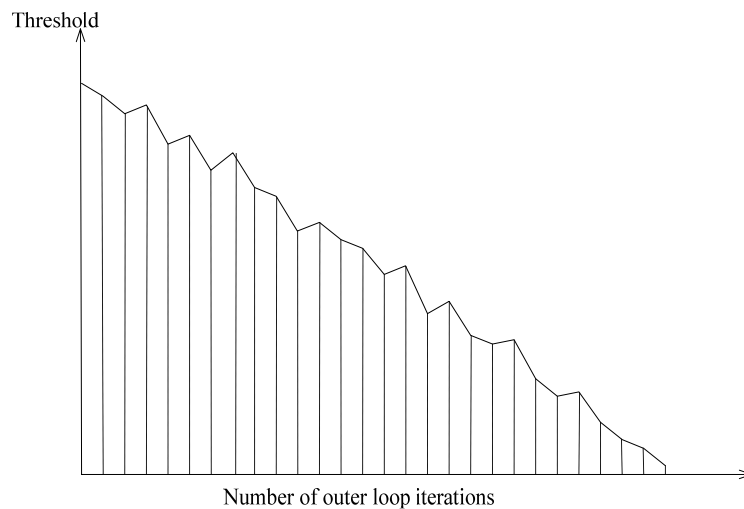


圖 2.15 BATA 門檻數列收斂示意圖

資料來源：朱佑旌[1]

而此方法回溯門檻值的機制，是假設在每次進行鄰域搜尋之後，若在期間有至少一次移動(找到可接受的解)，則減低門檻值；否則即提高門檻值，但不可超過上一次被接受的門檻值。而其回溯公式為： $T_n = T_c + (T_p - T_c) \times b$ ，其中 T_n 為得到的新門檻值， T_c

為目前的門檻值， T_p 為前一次被接受的門檻值； b 則為門檻回溯的比例值，由此公式可知 b 值不得超過 1，以避免違反上述規定。

可回溯式門檻接受法另一個與傳統門檻接受法的差異，在於停止法則除了達到原先預定的門檻數列長度之外，當已經連續回溯一個預設次數，卻沒有找到任何一個可接受的解，即直接停止。

可回溯式門檻接受法應用於組合最佳化問題在文獻上還不多見。Tarantilis et al. [38] 提出此方法時，被設計應用於 HFFVRP (Heterogeneous Fixed Fleet Vehicle Routing Problem)。其後，朱佑旌[1]則以 BATA 方法設計一套兩階段可回溯式門檻接受法 (TBTA)，求解 HVRPBTW (Heterogeneous Fleet Vehicle Routing Problem with Backhauls and Time Windows)。

2.4 小結

目前，巨集啟發式解法所發展的概念可大致歸納為深度搜尋(Intensification Search)與廣度搜尋(Diversification Search)兩種高階策略。乃是想像在一龐大的解空間當中，不僅要策略地決定搜尋的方向，也要有優秀的搜尋工具快速地搜尋到其區域最佳解。可回溯式門檻接受法調控門檻值的概念即符合此項作法，當門檻值縮小，意味著偏重深度搜尋；門檻值加大，則偏向了廣度搜尋(Tarantilis et al. [40])。本研究即欲利用此概念靈活切換深度搜尋與廣度搜尋，並以 GENIUS 與其它數種鄰域搜尋法構成有效的搜尋工具，構建出一套有效求解 VRP 的巨集啟發式解法。



第三章 起始解與鄰域搜尋法構建

本章內容將詳細說明本研究對於 GENIUS 如何應用於 VRP，以及本研究所採用鄰域搜尋法和所做的改良方法。

3.1 起始解產生：一般化插入法(GENI)

本研究採用一般化插入法作為起始解的構建方法。一般化插入法不同於傳統插入法，可選擇將一點插入至不相鄰的兩點之間，其插入的方法可分成型一與型二兩種。而在考慮插入的位置時，先考量在順時針或逆時針的迴路中，型一與型二方法分別得到的成本增加值，再選擇擁有最小值的方法來插入。

另外，一般化插入法原先是以 TSP 為求解的問題(Gendreau et al. [21])，故應用在多條路線的 VRP 時，可採用常見的循序或是平行兩種構建法則。本研究選擇較為簡單的循序構建，其步驟流程詳細如下：

- 步驟 0. 將所有顧客點設定為尚未插入路線的集合 U 。
- 步驟 1. 找出 U 中距離場站最遠的一點 a ，與距離 a 最近的一點 b ，與場站連成一條起始路線，將 a 點與 b 點自 U 移除。同時，對所有節點均設定其鄰近清單 (Neighborhood list); 某節點之鄰近清單為同路線中 p 個距離此點最近之節點的集合。
- 步驟 2. 檢查 U 中每一點其是否插入後可以滿足車容量限制，若無任何點滿足則跳至步驟 1，以新增路線。反之，對所有滿足的點計算出其兩個方向與兩種型的成本增加值。選擇擁有最小成本增加值的點 v ，並以其所對應的插入法執行插入；將 v 點自 U 移除，更新所有顧客點的鄰近清單。
- 步驟 3. 若 U 為空集合即停止。否則，重複步驟 2。

3.2 鄰域搜尋模組之設計

本研究採用的鄰域搜尋方法包含了解繫法(Unstring-and-String, US)與其他常見的鄰域搜尋法。另外，在程式的資料結構上，本研究採用巨網(Giant Tour) [20]的概念，將多路線的 VRP 問題視為一條整體的路線。並設計一個擴大解繫法搜尋範圍的機制使其找到更多可能的交換，期以這樣的方法，來擴大其搜尋的鄰域，從而找到最佳的解。

3.2.1 於巨網結構下(Giant Tour)下執行鄰域搜尋

本研究在鄰域搜尋模組的程式撰寫中，利用了巨網結構(Giant Tour)[20]的概念。如圖 3.1 所示，圖 3.1 (a)是實際上 VRP 問題的路線結構，而在巨網結構的構想中則是用多個虛擬的場站點將不同的路線隔開，使得每條路線均成為一條連結不同場站的路徑(Path)，而整個網路則可連結成單一迴路(Cycle)。如圖 3.1 (b)所示，路線一最後在一虛擬的場站結束；而路線二則是從此場站出發，並到達另一個虛擬的場站，最後經由路線三返回最初的場站。

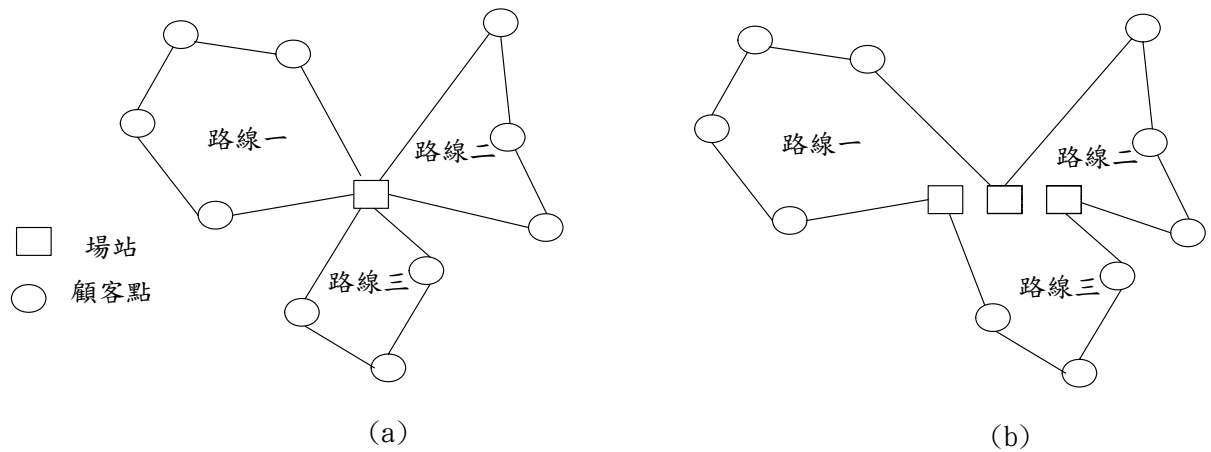


圖 3.1 巨網結構的路線示意圖

由於解繫法原本應用於 TSP 問題，而此種設計將可以使其應用於 VRP 問題時，除了必須考慮每條路線的容量限制問題之外，其他仍然與應用於 TSP 的情況大同小異。也因此其所設定鄰近清單是由路網上最為鄰近的點所組成，而不考慮到其實際所在之路線；所以執行搜尋時，其所選的 V_i 、 V_j 、 V_k 和 V_l 四點均在同一條路線時，為「路線內」的交換，而如果至少其中一點與其他點不為同一條路線，則可視為「路線間」的交換。圖 3.2 表示了一個路線間交換的範例。其中， V_j 和 V_k 屬於同一條路線， V_i 則為另一條路線，當被拔出路線外的第 8 點重新插入路線 1 後，第 7 點反而被移至路線 1，第 1 點則移至路線 2。由此可想見，當 V_i 、 V_j 、 V_k 和 V_l 此四點所分屬的路線愈多，其執行的交換也愈複雜，特別是「路線內」與「路線間」兩種不同的交換在此結構下均可能發生，使得在此所執行解繫法之變異性也大大地增加了。

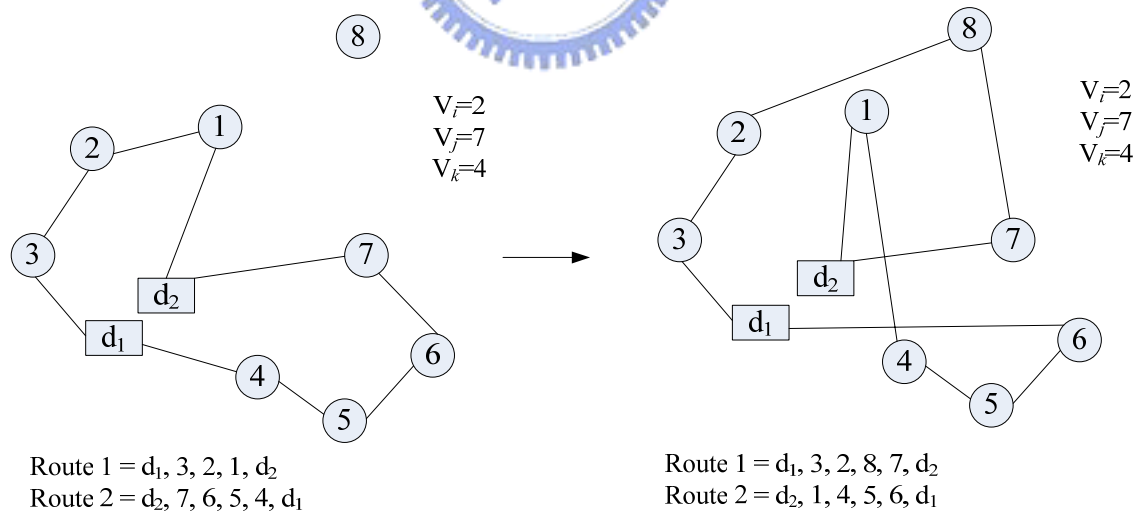


圖 3.2 路線間的解繫法示意圖

同樣地，原本適用於 TSP 問題的交換法，2-Opt 與 Or-Opt 亦可以在此結構下同時考慮「路線內」與「路線間」的交換，因此本研究將其納入為鄰域搜尋模組的交換方法。而原本屬於路線間交換法的 1-0、1-1，進行路線間或路線內的交換，亦均可能獲得改善，但因 Or-Opt 方法內將一個節點自路線內抽出，再將其插入其他節線中的作法，已與 1-0 交換法相當類似，故不將 1-0 交換獨立出來，僅將 1-1 列為鄰域搜尋的交換方法。

因此，本研究之鄰域搜尋模組共包含了 US、2-Opt、Or-Opt 和 1-1 四種方法。而在執行順序上共有 24 種組合，但實際上無法有所依據斷言哪一種組合在最後會產生較佳的結果。然而本研究希望在一開始執行較複雜的解繫法，藉此將互相鄰近的點先逐漸整理至同一條路線，以獲得一個不錯的解。接著以 1-1 交換進行微調。而由於 Or-Opt 可做較為仔細的交換，因此先以 2-Opt 進行快速地搜尋，最後則以 Or-Opt 把關，確保較少的交換會被遺漏。故鄰域搜尋模組執行的順序為：(1) US、(2) 1-1、(3) 2-Opt、(4) Or-Opt。

3.2.2 擴大解繫法(US)鄰域搜尋範圍之設計

因為車容量的限制，使得在鄰域搜尋中大部分的交換並不可行；而解繫法以建立鄰近清單來縮小其搜尋範圍，但也使得其可接受的交換變得更少。特別是解繫法是透過路線中某個節點的拔出與重新插入來尋求改善，計算上較為繁複且較不容易找到可接受的交換。

本研究認為，在 Gendreau et al. [21]所提之解繫法的架構下，任一個節點無論是 Unstring 或是 String 均有 Type I 和 Type II 兩種不同型態的方法去比較，再選出具有最佳改善的方式。這樣目的應是著眼於不同的鄰域搜尋法其鄰域均不太相同，若同時考慮到數種鄰域搜尋法可能的交換，將使得其鄰域擴大而更有機會找到較佳的解。以此觀之，在執行解繫法的時候均需依順序針對每一個節點，進行點的重新拔出與插入來尋求改善，而若此同時能比較此點與另一點的節點互換(1-1)之改善效果，或此點與另一點之節線交換(2-Opt)，或是將此點、或此點之後的一段節線或兩段節線抽出再插入其他節線位置(Or-Opt)，再選擇其中具有最佳改善的方法，相信將可進一步擴大其鄰域搜尋範圍，進而達成上述之目的。本研究將此在 US 的架構下，能綜合考慮各種交換法的機制，稱之為 Expanded US，並將於第四章中以實際測試說明其效果。

另外，在核心交換法的選擇策略上通常有(1) 最佳改善(Best Improvement)、(2) 首先改善(First Improvement)和(3) 半最佳改善(Semi-best Improvement)三種：最佳改善意指檢查所有可能的交換後選擇接受具有最多改善的交換；首先改善在當找到可行的交換後即接受。半最佳改善則針對特定一點所能找到的鄰解中，選擇具有最多改善的交換。由於此三種策略在改善效果方面尚未證實有明顯差異，但在執行時間上首先改善應為最優。因此，本研究在鄰域搜尋模組所使用的策略均為首先改善。

第四章 BATA 應用於 VRP 之設計與測試

4.1 BATA 與 GENIUS 結合之設計

本章內容探討 BATA 方法中門檻值回溯幅度的大小對於搜尋的影響，以及提出本研究的求解架構，說明如何將 BATA 與 GENIUS 相結合，最後以國際標竿題庫進行求解的績效測試。

4.1.1 門檻值回溯幅度之探討

傳統的門檻接受法所採用的門檻數列為遞減型態，隨著門檻值的降低，接受法則也漸趨嚴格，找不到可接受解的機率也就升高，這是 Tarantilis et al. [38] 提出可回溯式門檻接受法主要著眼改善的地方。而由於其採用的遞減型態為等比下降，若發生門檻值回溯的情形，通常已經經過若干次下降，這也意味著現有門檻值與上一個門檻值的差距相當的小。所以將門檻值回溯之後，可以搜尋解的區域亦有限，很可能依然無法跳脫局部最佳解。因此，本研究認為若將 b 值的範圍擴大，不受其小於 1 的限制，使下一個門檻值能夠高於上一個被接受的門檻值，將可以使其跳脫現有局部最佳解的束縛。如下圖 4.1 所示，由 A 點搜尋至 B 點後，現有解與最佳解均獲得更新，門檻值則進一步下降。當此時找不到可接受的解，在回溯比率 $b < 1$ 的情況下，可接受的範圍上升至 C 點如實線箭頭，不僅難以再進一步放鬆搜尋範圍，且可能於下一次搜尋時又掉至局部最佳解；相反地，若 $b > 1$ 則可接受的範圍將如虛線箭頭所示上升至 D 點，且在往後搜尋中將可能到達全域最佳解 E 點。

然而如此一來，若連續二次以上回溯門檻值，由於 T_c 會高於 T_p ，將使得 $T_p - T_c$ 的值有可能為負，門檻值反而會降低。為了避免此種情形發生，本研究以絕對值確保其差值為正，修正後的公式為 $T_n = T_c + |T_p - T_c| \times b$ 。在後續小節中，將分別對回溯比率 b 小於 1 或大於 1 的情形進行測試，以驗證此想法。

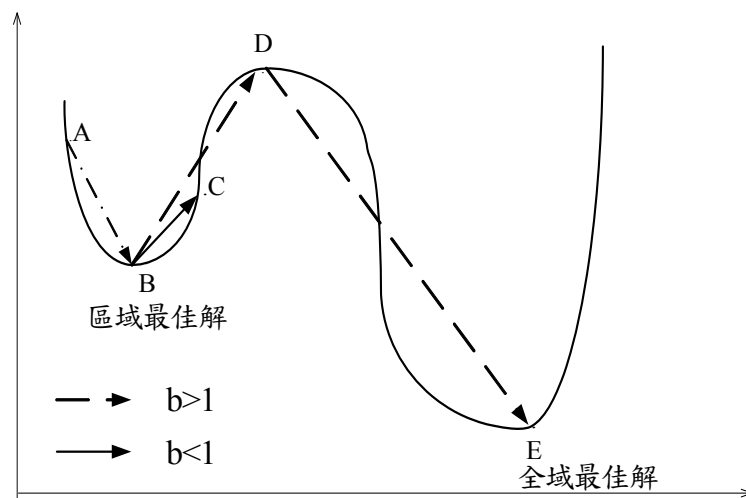


圖 4.1 回溯幅度大小對於搜尋影響示意圖

4.1.2 執行架構說明

本研究的巨集啟發式解法是以 Tarantilis et al. [38]所使用的 BATA 架構為基礎，其核心交換模組為以第三章所述之鄰域搜尋模組，而整體的求解架構流程如圖 4.2 所示。由圖中可知，本研究首先以一般化插入法構建一個起始解，並由前述之鄰域搜尋模組進行改善，最後將此改善的解投入 BATA 的架構中以找尋更佳的解。由此可知，本研究之執行架構共可包含起始解模組、鄰域搜尋模組和可回溯式門檻接受模組三個模組。

在 BATA 的架構中，核心搜尋以前述之鄰域搜尋模組為主，包含(1) US 或 Expanded US (2) 1-1 (3) 2-Opt (4) Or-Opt 四種方法順序，其基本流程如圖 4.3 表示：首先搜尋一個暫存解 X' ，若其成本 $C(X')$ 與現有解 $C(X_c)$ 之差值小於現有門檻值 T_c ，則接受交換；若現有解 $C(X_c)$ 小於暫時最佳解 $C(X_b)$ ，亦進行更新。

此模組整個執行完一次之後，若在此迴圈之中有發生過接受交換的情形($H>0$)，門檻值即遞減。而門檻值遞減型態與 Tarantilis et al. [38]相同，採用等比數列，並有門檻下降比率 r 來控制下降速度；另外，回溯幅度受到門檻回溯比值 b 的控制，其公式以 $T_n = T_c + |T_p - T_c| \times b$ 代替。其中 T_p 是前一個曾經接受交換的門檻值， T_c 為現有門檻值，下一個門檻值為 T_n 。因此在控制參數方面有起始門檻 T_0 、門檻下降比率 r 、門檻回溯比率 b 以及解繫法中鄰近清單大小 p ，停止法則為完成 K 次的鄰域搜尋，與連續放鬆 C 次的門檻皆無法找到可接受的解。而最後所得結果即記錄中之暫時最佳解。



T: 起始門檻;
 I: 門檻數列, $I = 1, 2, \dots, K$
 K: 門檻數列長度
 J: 連續放寬門檻值次數,
 $J = 0, 1, 2, \dots, C$
 C: 連續放寬門檻最大次數
 H: 接受交換次數, $H = 0, 1, 2, \dots$
 r: 降低比率值
 b: 放寬比率值
 T_p : 前一個被接受的門檻值
 T_c : 現有門檻值
 T_n : 下一個門檻值

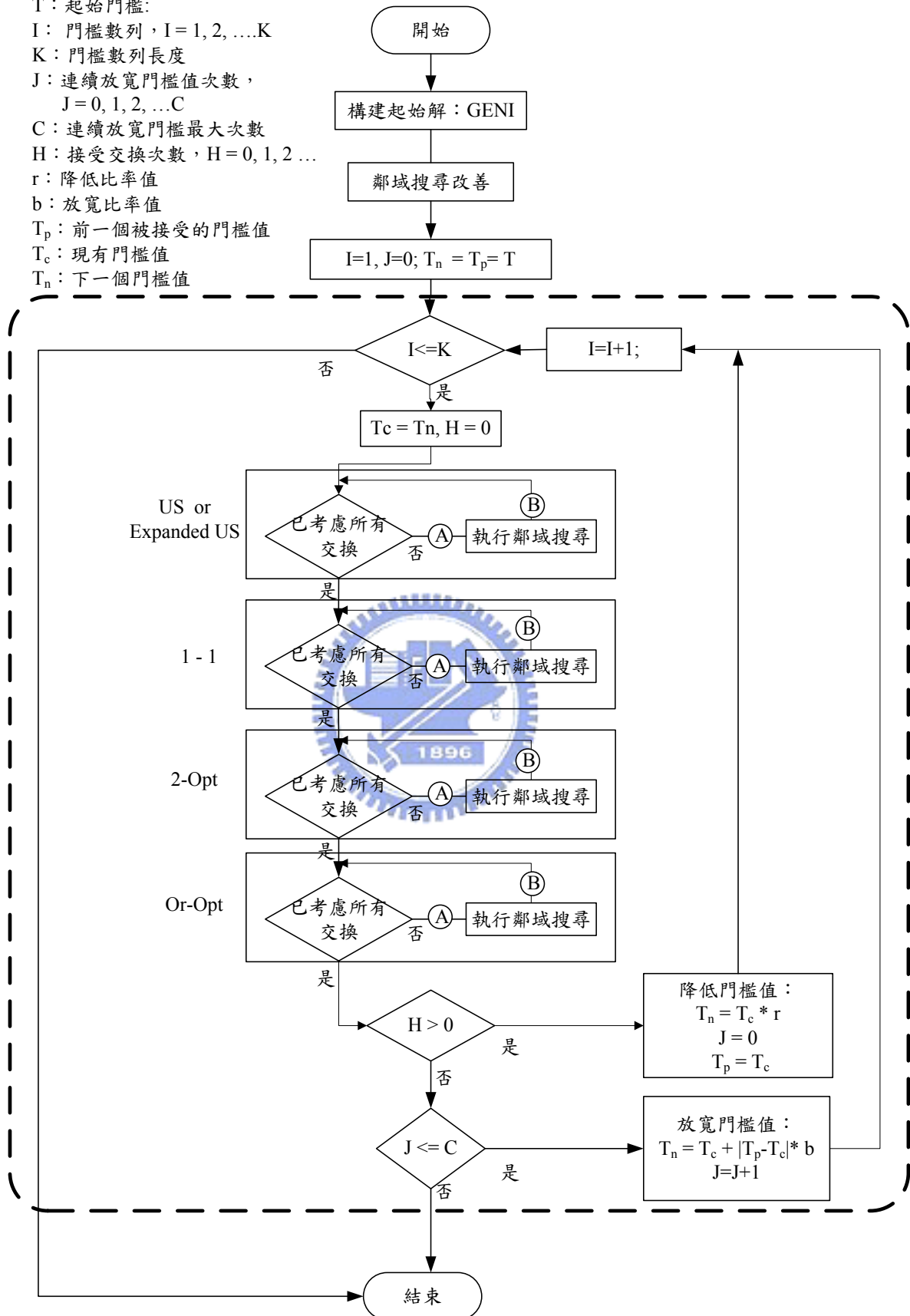


圖 4.2 本研究巨集啟發式解法求解架構流程圖

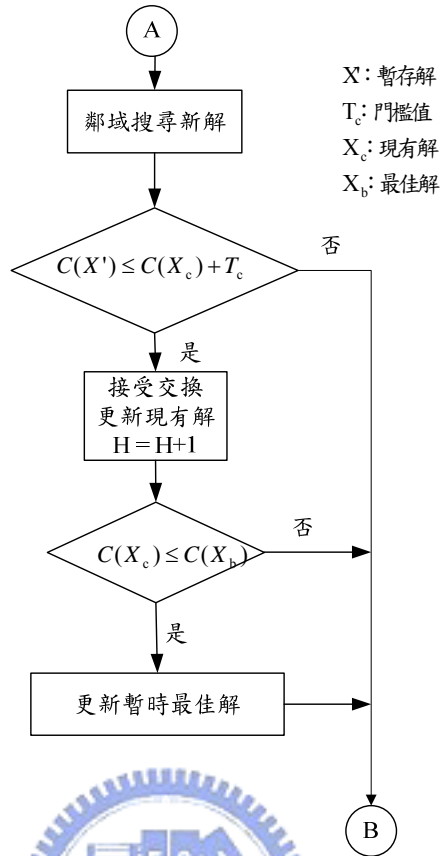


圖 4.3 BATA 核心搜尋細部流程

4.2 測試例題說明

本研究以Cristofides et al. [8]的14題標竿例題作為測試例題，這些例題均可從網站 [VRP Web](#)[43]中下載得來。而已知最佳解則由發表於國際期刊的文獻與部份網站[29、34、36、43]整理而得。這些例題的服務顧客數從50到199不等，且均符合以下三個前提假設：1. 路網為對稱(Symmetric)且無方向性(Undirected)。2. 各節點間皆有節線相連之完全性路網(Complete Graph)。3. 路網為二維平面空間分佈，各點間距離符合三角不等式。其中，1至10題的節點分布為隨機分布，11至14題為群聚分布(Cluster)；且6至10題與13、14題同時具有容量限制與最大路線時間限制，其餘則僅有容量限制。

表4.1為例題相關資料的整理，其中所列之文獻最佳解，均有詳細之路線資料可供參考，除了第5題之外。此題於Mester and Bräysy[29]此篇文獻已獲得最佳的解，但是此文獻中並無提供詳細的路線資料。

表4.1 VRP測試例題

例題	顧客數	車容量	最大 旅行時間	服務 時間	已知最佳解	
					總成本	來源
1	50	160	∞	0	524.61	Tallard[36]
2	75	140	∞	0	835.26	Tallard[36]
3	100	200	∞	0	826.14	Tallard[36]
4	150	200	∞	0	1028.42	Tallard[36]
5	199	200	∞	0	1291.29	Mester and Bräysy [29]
6	50	160	200	10	555.43	Tallard[36]
7	75	140	160	10	909.68	Tallard[36]
8	100	200	230	10	865.94	Tallard[36]
9	150	200	200	10	1162.55	Tallard[36]
10	199	200	200	10	1395.85	Rochat and Taillard[34]
11	120	200	∞	0	1042.11	Tallard[36]
12	100	200	∞	0	819.56	Tallard[36]
13	120	200	720	50	1541.14	Tallard[36]
14	100	200	1040	90	866.37	Tallard[36]

4.3 測試結果之整理與分析

4.3.1 實驗設計

本研究的實驗測試共包含三部份：(1)起始解測試：比較不同的鄰近清單大小(p)所獲得起始解的結果，p 值測試範圍為 3~7，共 5 種結果。(2)鄰域模組測試：以鄰域搜尋模組針對前述所獲得的起始解結果進行改善。並以兩種模組的測試結果來比較，一種包含 US、1-1、2-Opt、Or-Opt 等鄰域搜尋法，另一種則將 US 替換成前章所述之 Expand US，其餘相同。另外，起始解與鄰域搜尋改善所採用的鄰域清單大小均相同，因此每一種模組亦可得到 5 種測試結果。(3)可回溯式門檻接受模組：以此模組所需設定的參數進行測試，如起始門檻比率(T_0)、門檻下降比率(r)、門檻回溯比率(b)、門檻數列長度(K)等。

本研究以 C# 進程式撰寫，並在 Microsoft Visual Studio 2005 中編譯，測試環境為使用 Windows XP SP2 的作業系統、Pentium(R) 4，CPU 為 3.00GHz 的個人電腦。

4.3.2 起始解測試

首先針對起始解進行測試，結果如表 4.2，表中所列為各題與已知最佳解之誤差(%)。可發現以一般化插入法，並採用循序構建方式所建立之起始解，其平均誤差隨著 p 值的增加並無明顯的變化，誤差值約在 32%~39%之間，求解效果並不佳。然而較好的起始解並不等於最後巨集啟發式解法所獲得之解亦較好，因此，效果不佳的原因本研究

先不予討論。

表 4.2 起始解之誤差(單位：%)

例題 \ p值	3	4	5	6	7	平均誤差
1	30.31	24.93	16.26	16.26	16.26	20.8
2	35.78	34.13	33.83	36.87	36.77	35.48
3	45.41	42.31	31.84	32.5	26.22	35.66
4	52.05	41.53	44.39	43.62	40.01	44.32
5	44.53	47.47	43.25	39.57	40.82	43.13
6	28.67	11.2	15.19	15.19	15.19	17.09
7	25.56	26.38	30.8	21.48	21.48	25.14
8	31.34	25.57	20	30.51	24.09	26.3
9	29.96	23.4	26.23	19.71	22.89	24.44
10	35.93	28.58	26.68	25.51	27.95	28.93
11	52.6	64.18	50.56	49.62	53.18	54.03
12	47.17	44.93	36.67	43.23	43.23	43.05
13	47.41	57.23	48.54	49.6	49.6	50.48
14	39.05	41.39	36.06	31.72	31.62	35.97
平均誤差	38.98	36.66	32.88	32.53	32.09	

4.3.3 鄰域搜尋模組測試

接著，進行鄰域搜尋模組測試。本研究以兩種模組進行測試。一種包含 US、1-1、2-Opt 和 Or-Opt 四種方法，並以此順序執行之；另一種則將 US 替換成前述之 Expanded US 方法，其餘則相同。此部分的結果是針對前述之起始解進行改善，而起始解與鄰域搜尋改善所採用的鄰域清單大小相同，因此亦可得到如表 4.3、表 4.4 的結果，同樣列出各題與已知最佳解之誤差(%)。由表中可看出，在此階段各題的結果均可以再獲得進一步改善，14 題平均誤差可在 5~10%之間。

圖 4.4 是由 14 題平均誤差與 p 值繪製而成的圖，由圖中可以發現，包含 US 的鄰域搜尋模組僅在 p = 3 的時候略優於包含 Expanded US 的鄰域搜尋模組，或在 p = 6 的時候為平手，但是其餘的情況下，包含 Expanded US 之鄰域搜尋模組均較包含 US 之鄰域搜尋模組為佳，整體來說可見 Expanded US 確實能更為有效地搜尋到更好的解，因此在以下之可回溯式門檻接受模組的測試均採用包含 Expanded US 之鄰域搜尋模組做為核心交換模組。

由圖 4.5 的結果可觀察出平均時間與 p 值的關係。圖中顯示，隨著 p 值的增加，平均所耗費時間約為指數成長，但平均誤差卻只有緩步的下降。在誤差與時間的綜合考量下，在 p = 4 的情況下的績效已經相當不錯，故以下小節中對可回溯式門檻接受模組之測試所採用的 p 值均為 4。

表 4.3 以包含 US 之鄰域搜尋模組改善後之誤差(單位：%)

例題 \ p值	3	4	5	6	7	平均誤差
1	15.5	5.38	3.06	4.27	3.19	6.28
2	12.6	11.38	8.41	3.39	5.36	8.23
3	3.45	5.24	2.37	4.75	2.48	3.66
4	6.26	10.95	6.03	6.04	9.47	7.75
5	14.26	15.59	7.14	6.68	7.01	10.14
6	4.32	2.56	2.3	1.35	1.35	2.38
7	8.98	10.44	4.77	3.91	3.91	6.4
8	7.31	7.46	4.8	5.67	2.19	5.49
9	7.38	6.95	4.69	7.61	4.98	6.32
10	7.35	8.58	5.88	6.75	6.35	6.98
11	22.42	31.57	26.81	16.39	19.86	23.41
12	5.91	7.1	14.91	6.36	4.56	7.77
13	7.03	11.44	5.14	6.18	7.65	7.49
14	8.05	9.03	17.82	4.82	7.49	9.44
平均誤差	9.34	10.26	8.15	6.01	6.13	

表 4.4 以包含 Expanded US 之鄰域搜尋模組改善後之誤差(單位：%)

例題 \ p值	3	4	5	6	7	平均誤差
1	9.29	7.54	2.09	3.6	2.46	5
2	11.03	12.05	9.33	4.44	4.57	8.28
3	1.97	1.42	1.94	3.89	2.49	2.34
4	7.02	9.61	3.87	6.85	2.53	5.98
5	11.93	13.56	12.57	8.73	7.69	10.9
6	2.39	2.22	1.35	1.35	1.66	1.79
7	5.73	3.92	6.28	2.28	3.74	4.39
8	8.1	6.38	3.48	3.31	1.49	4.55
9	8.15	5.87	4.77	9.42	5.08	6.66
10	8.15	7.53	8.94	5.99	8.37	7.8
11	26.77	18.21	25.29	14.07	17.2	20.31
12	12.04	5.48	11.98	12.33	5.78	9.52
13	11.81	14.61	9.42	3.18	7.24	9.25
14	8.26	7.24	6.6	4.65	4.28	6.21
平均誤差	9.47	8.26	7.71	6.01	5.33	

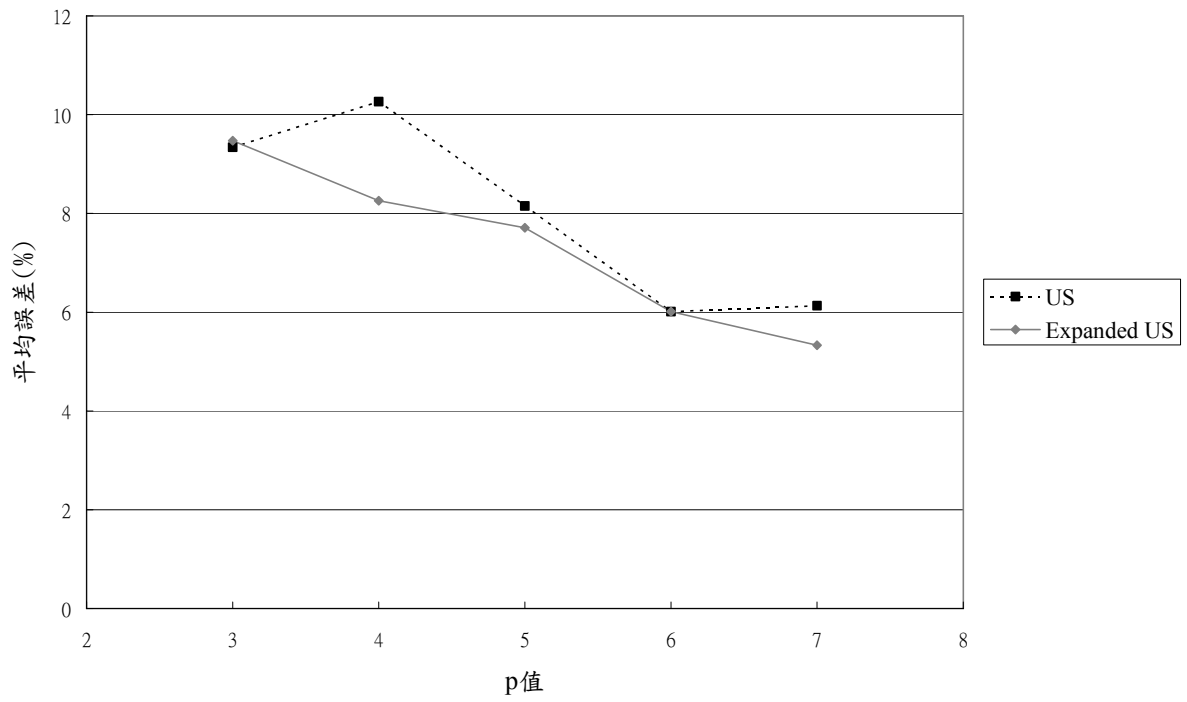


圖 4.4 不同鄰域搜尋模組之平均誤差(%)

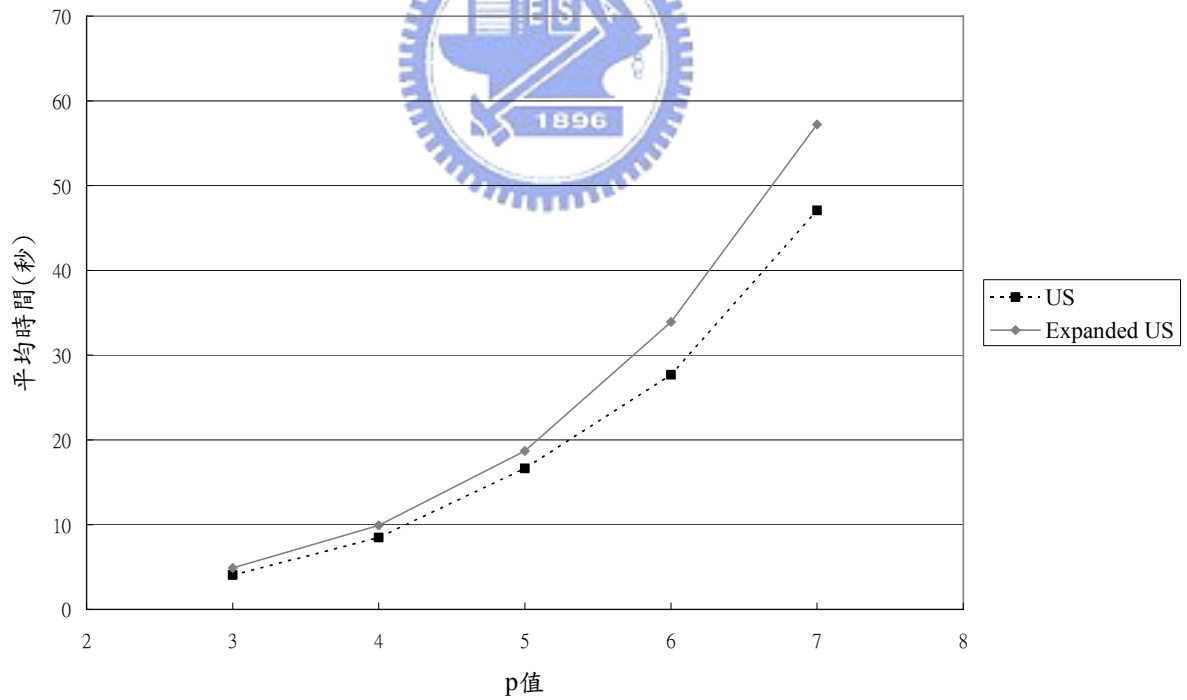


圖 4.5 不同鄰域搜尋模組之平均耗費時間(秒)

4.3.4 可回溯式門檻接受模組測試

在此小節中，欲測試的可回溯式門檻接受模組之參數包含了起始門檻比率 T_0 、門檻下降比率 r 、門檻回溯比率 b 、門檻數列長度 K 等。另外，核心交換法中的 Expanded US 之鄰近清單大小 p 值如前述設為 4，最大連續回溯門檻次數則設為 20。

首先是 b 值與 K 值的測試。為了驗證 4.1.1 小節中關於門檻值回溯幅度的討論，分別選取 $b < 1$ 與 $b > 1$ 的若干個數值進行比較。其中， $b < 1$ 測試的數為 0.5, 0.55, 0.6, ... 0.95，共 10 個數； $b > 1$ 測試的數則為 1~40 的整數值。在 K 值方面，雖然楊智凱[4]的 TA 門檻數列長度建議值為 30 或 60，但考量到因數列長度太短，可能導致來不及回溯或回溯之後尚未搜尋到更好的解即停止，將使得回溯效果不明顯，所以 K 值的選擇為除了 60、90 之外，還包括 180。測試的結果如圖 4.6 所示，橫軸為不同的 b 值，縱軸則為 14 題之平均誤差(%)，而其餘參數設定則為 $T_0 = 0.01$, $r = 0.9$ 。結果可以發現， $b < 1$ 時所得的平均誤差於 $K = 60$ 的情況下，均落在 2.4% 左右，相差不大；且當 K 值增大後，平均誤差亦無明顯的變化。反之，同樣 $K = 60$ 的情形下， $b > 1$ 時所得的平均誤差變化較大，且均優於 $b < 1$ 的結果；當 K 值增大至 180 時，可見其平均誤差均可再進一步獲得改善。且當 $b > 1$ 之後平均誤差即緩步下降，雖然 $b = 5$ 之後平均誤差起伏較不穩定，但是均可以低於 2%。可見其確實有脫離局部最佳解的效果。

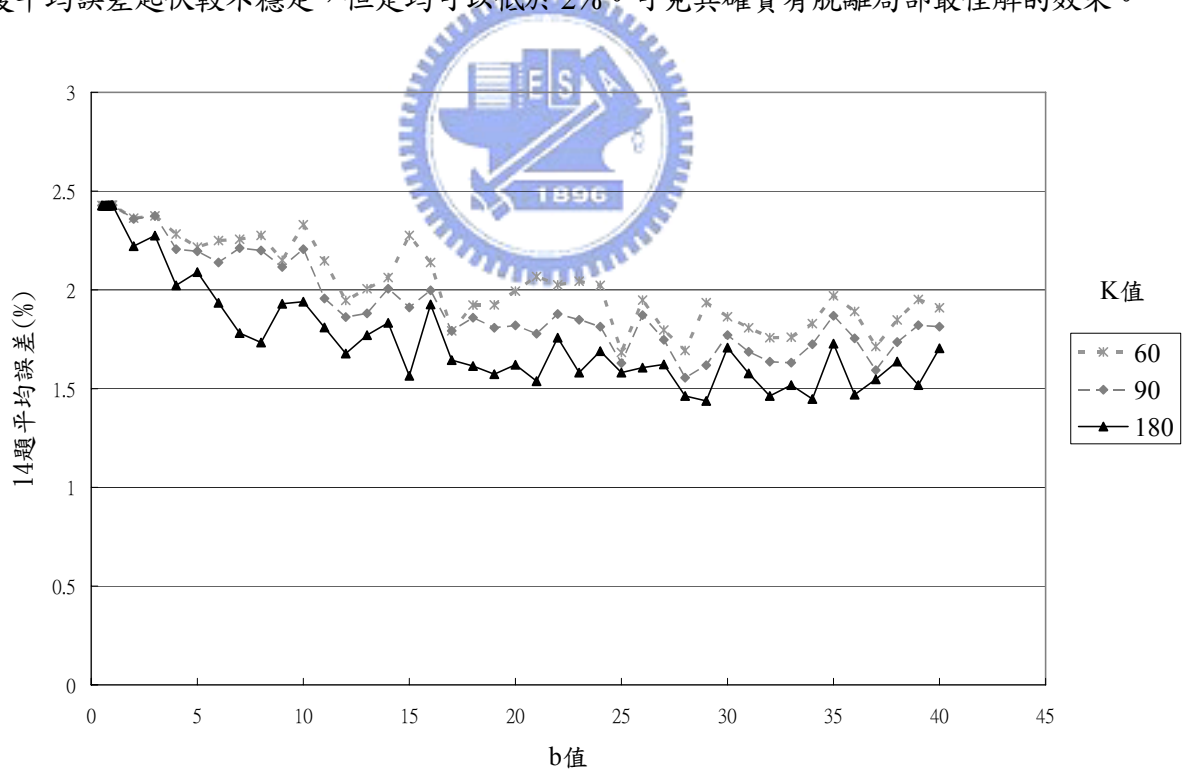


圖 4.6 不同 K 值與 b 值對於求解結果的影響($T_0 = 0.01$, $r = 0.9$)

接著，我們以第 2 題為例，在 $T_0 = 0.01$, $r = 0.9$, $K = 180$ 的設定下，觀察其在 b 值分別為 0.95 與 30 的時候，門檻值與暫時最佳解的變化情形，結果如圖 4.7 所示，圖 4.7 (a) 表示了門檻值的變化，圖 4.7 (b) 則為暫時最佳解的變化。可看到在經過 23 次門檻值下降之後，已獲得一個不錯的

暫時最佳解，而此時已找不到任何可接受的交換，於是有門檻值回溯的情形。而 $b = 0.95$ 為小幅的門檻回溯，在附近的區域進行了綿密的搜尋，使暫時最佳解進一步更新，但由於其門檻回溯幅度愈來愈低，再經過幾次門檻回溯之後，因達到連續門檻回溯次數的停止法則，使得還未到60次搜尋即停止。反之， $b = 30$ 的在第一次回溯即大幅超過前一次門檻，暫時最佳解亦在後續的搜尋獲得更新，經過幾次門檻回溯，暫時最佳解數獲得更新，而得到更佳的结果。

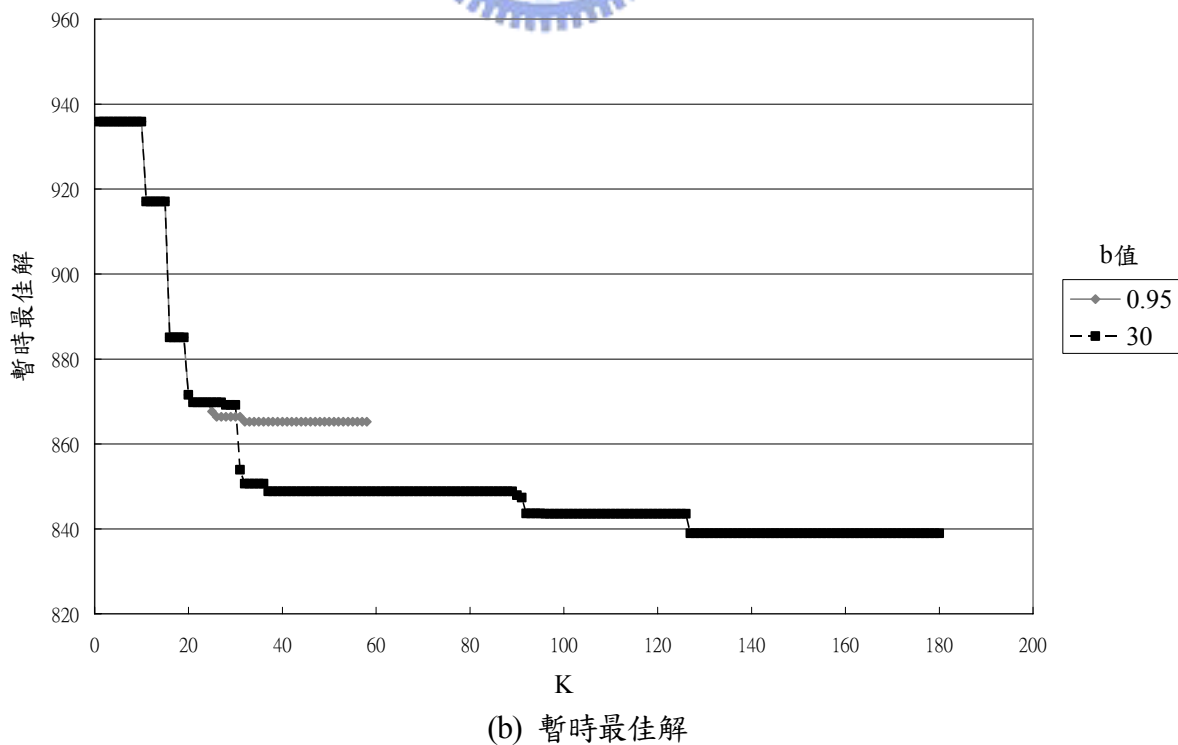
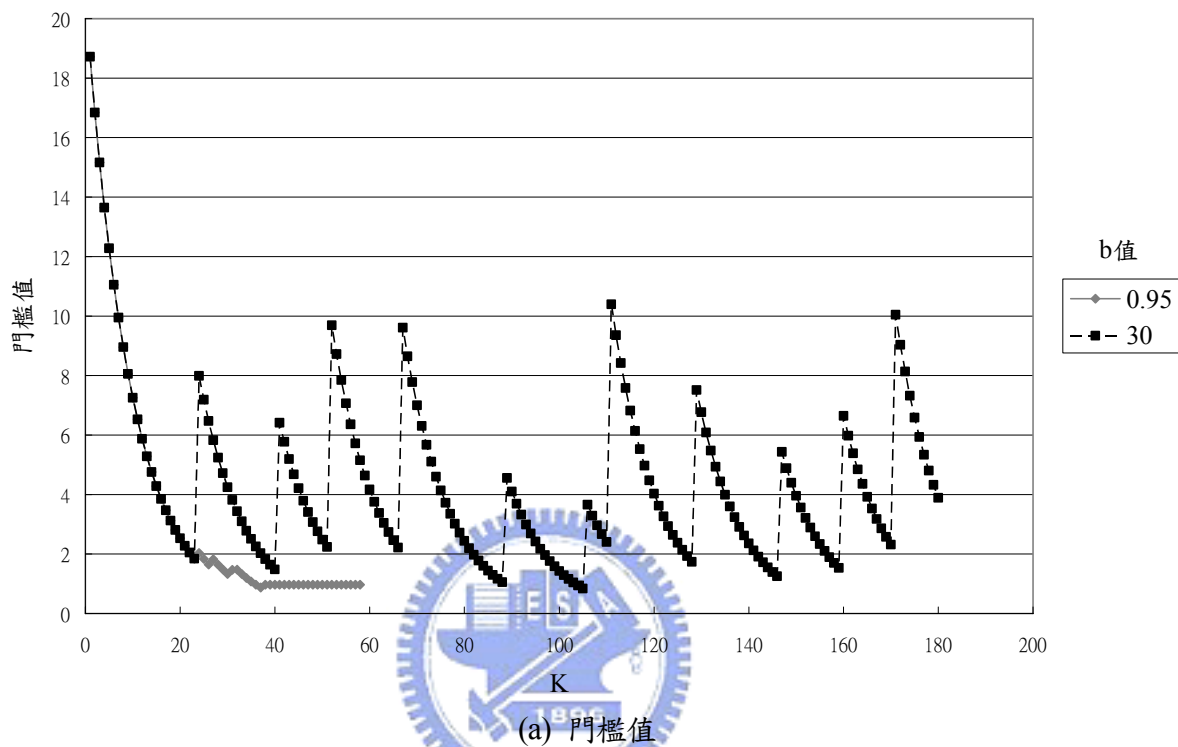


圖 4.7 門檻值與暫時最佳解變化圖—以例題 2 為例

另一方面，如圖 4.8 所示，隨著 K 值的增加，平均耗費時間亦增加，而在 K 值 180 的時候，因 $b < 1$ 的回溯幅度小，導致某些例題其連續回溯次數達到最大限制，而提前結束搜尋，所以時間較短。而 $b > 1$ 的設定由於其回溯幅度大，較不容易達到最大連續回溯次數的限制，在較為末端的門檻也還有可能找到最佳的解，因此使用較大的門檻數列長度其求解效果會比較佳比較穩定，而且在 K 值 180 的時候平均所耗費的時間也僅僅約 50 秒左右而已，因此在以下的 T_0 、 r 值的測試，門檻值都設為 180，以保證有較為一致的結果。

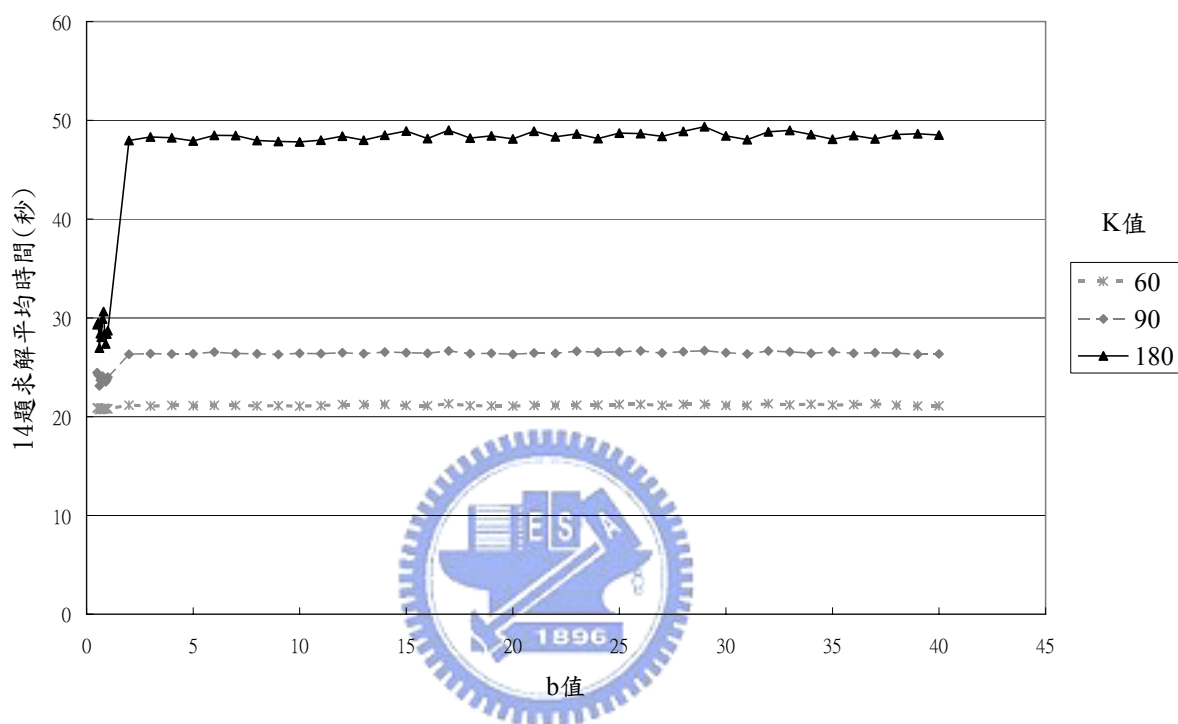


圖 4.8 不同 K 值與 b 值所需之平均耗費時間(秒)($T_0 = 0.01, r = 0.9$)

接下來，本研究以 r 值為 0.7、0.8、0.9、0.95、0.99 進行測試， b 值範圍如前所述， T_0 值則為 0.01，K 值為 180。如圖 4.9 所示，可以觀察到在 $b < 1$ 的情況下，隨著 r 值的增加，整體的平均誤差是往下降的；而當 $b > 1$ 時，可看到顯現出曲線相互交錯的情形，並不一定符合此趨勢，但仍可觀察出較大的 r 值擁有較佳且較為穩定的績效；較小的 r 值，如 0.7、0.8，其振盪的幅度較大，只有在少數的 b 值下，其平均誤差有很優秀的表現。

另外，由於較大的 r 值表示其門檻數列下降的速率較為緩慢，在相同的門檻數列長度下，其回溯次數可能也會變得較少，使得 b 值的影響力減低，所以可看到圖中 r 值愈大， $b < 1$ 與 $b > 1$ 的差距也愈不明顯。可見在應用上，若求解時間較少，可選擇較大的 r 值以確保穩定求解績效；反之，則可選擇較小的 r 值，讓 b 值的影響增加，可能可以找到更好的結果。

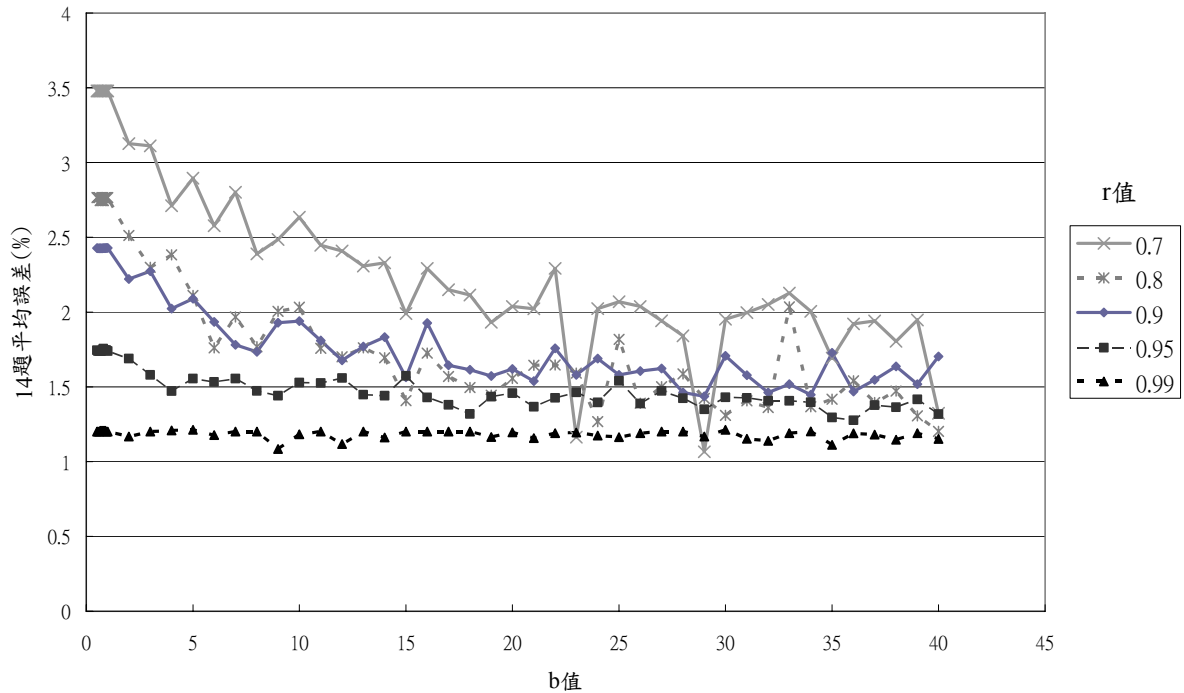
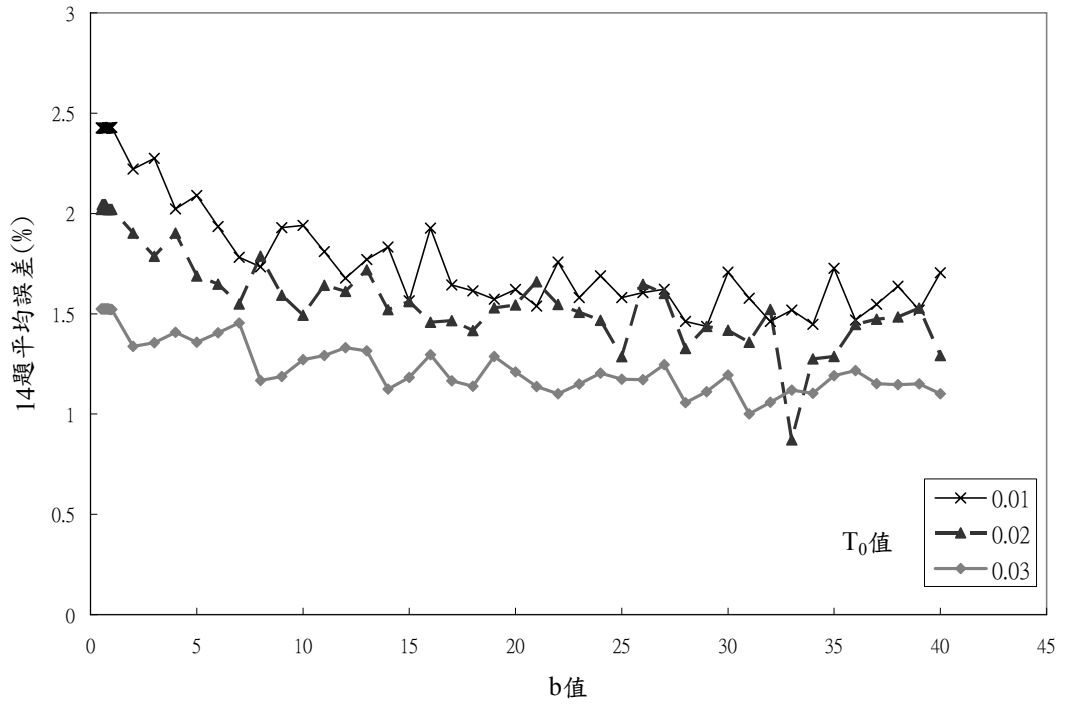


圖 4.9 不同 r 值對於求解結果的影響($T_0=0.01$)

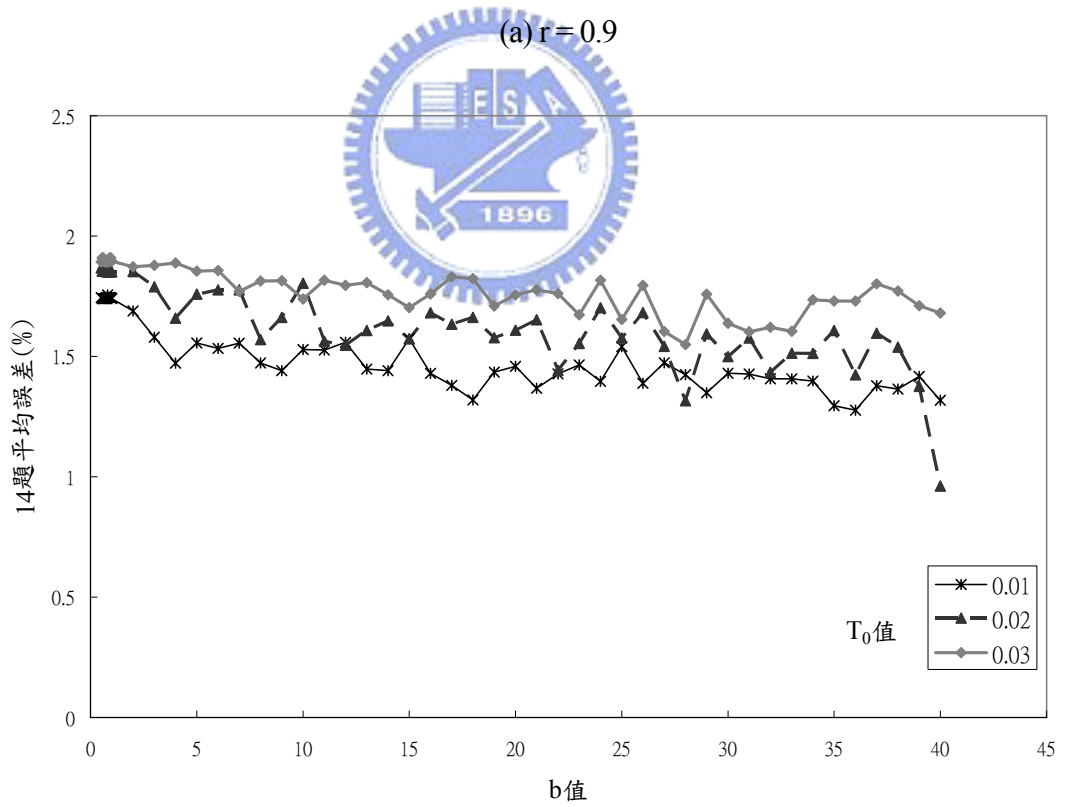
本研究繼續分別以 r 值為 0.9、0.95、0.99 的情況下，對起始門檻比率(T_0)進行測試。如圖 4.10 所示，當 $b < 1$ 時有著較一致的表現：r 值 0.9 在 T_0 值為 0.01、0.02、0.03 時均可有不錯的平均誤差，且當 T_0 愈大，表現愈佳。r 值在 0.95 時，反而當 T_0 愈大，解題績效會變得較差。r 值在 0.99 時，隨著 T_0 增加，更明顯地表現出其解題績效會愈差。可見 r 值愈大，其下降愈緩慢，必須搭配較小的 T_0 值，以避免一開始發散太遠，在後續搜尋無法找到更好的解。所以，當 r 值為 0.99，而 T_0 值為 0.01 時， $b < 1$ 的情況可在此找到整個測試中最佳的平均誤差，為 1.2%。

而在 $b > 1$ 的設定時，雖然其表現較為不一致，但是其趨勢也與 $b < 1$ 時類似：當 r 值為 0.9 時， T_0 愈大，平均誤差愈低；當 r 值為 0.95、0.99 時， T_0 愈大，平均誤差愈高。而由於 $b > 1$ 時，其回溯幅度較高，可以使門檻值放鬆更多，若在較小的 r 值、較大的 T_0 值的設定下，其發揮的影響力也更大，很可能因此找到更佳的结果，因此我們可看到其最佳的平均誤差可低至 0.87%。

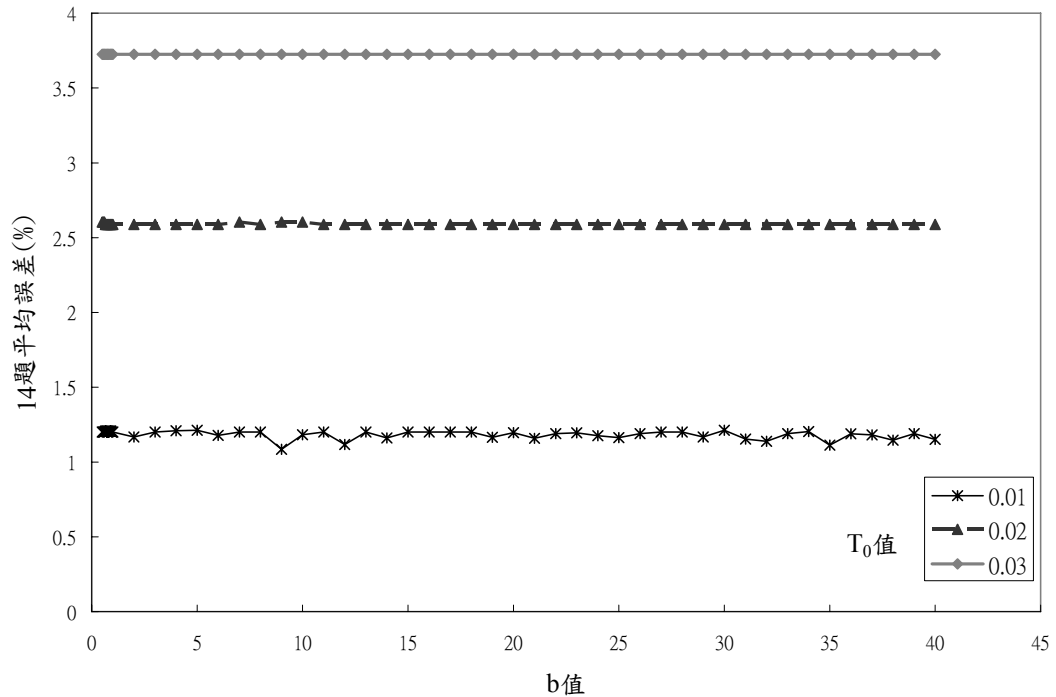
綜合以上的測試結果來看，本研究以傳統 BATA 限制 $b < 1$ 的設定下，可獲得相當不錯的结果。而若突破傳統 BATA 的限制，以 $b > 1$ 的設定進行測試，則可發現可以找到較佳的结果，這也表示了本章一開始對於回溯幅度的探討是相當正確的。但是由實驗中無法得知， $b > 1$ 時其 b 值大小對於求解結果之對應關係為何。因在搜尋中 b 值均是固定的大小，很可能在某次搜尋時，所適用的回溯幅度，反而於下次搜尋時導致其搜尋太過發散，而找不到更佳的结果。因此，未來將可繼續針對此部分進行努力與研究。



(a) $r=0.9$



(b) $r=0.95$



(c) $r = 0.99$

圖 4.10 不同 T_0 值對於求解結果之影響

最後，將本研究於測試中所獲得具體結果整理如表 4.5。表 4.5 包含了本研究測試後所得單一組特定參數下之最佳結果的整理，以及整個測試過程中所獲得之各題最佳結果。單一組參數是在 $T_0 = 0.02$ 、 $r = 0.9$ 、 $b = 33$ 、 $K = 180$ 中所得，其 14 題平均誤差 0.87%，且有 5 題找到最佳的結果。而表中亦列出其各題求解所需時間，單位為秒，由表中結果顯示，平均所耗費時間低於 50 秒，且個案最大花費時間可在 2 分鐘以內結束，可見其執行時間相當快。

而最佳結果則同樣列出其各題誤差，以及其所對應之參數值，其中有多個題目雖然有多組參數可求得最佳結果，但在此僅列出其中一組參數。由表中結果可知，其 14 題平均誤差可以低至 0.26%，且 14 題中可找到 7 題最佳結果，而個案最大的誤差則為 1.28%，由此可瞭解到本研究的解題能力亦相當不錯。

由於影響本研究之求解時間的參數主要為鄰近清單大小 p 值與門檻數列長度 K 值，由以上的測試可知，這兩項參數愈大對於解題績效愈佳，同時也會增加求解時間。若能對此兩項參數進行更進一步測試，相信將能在犧牲部分時間效率下獲得更佳的结果，以及能在求解績效與求解時間中取得平衡之最適參數數值。

表 4.5 單一組參數之最佳結果與測試過程中之最佳結果

題號	已知最佳解	最佳單組參數結果		最佳結果						
		誤差(%)	時間(秒)	誤差(%)	控制參數					
					T ₀	r	b	p	K	C
1	524.61	0	10.7	0*	0.02	0.9	33	4	180	20
2	835.26	1.38	15.39	0.05	0.03	0.95	7	4	180	20
3	826.14	0.27	33.09	0*	0.01	0.9	25	4	180	20
4	1028.42	1.46	60.52	0.32	0.02	0.95	35	4	180	20
5	1291.29	1.96	100.8	1.28	0.01	0.95	36	4	180	20
6	555.43	0	12.05	0*	0.02	0.9	33	4	180	20
7	909.68	0	19.69	0*	0.02	0.9	33	4	180	20
8	865.94	0.18	37.59	0*	0.02	0.95	40	4	180	20
9	1162.55	0.97	64.81	0.56	0.02	0.9	16	4	180	20
10	1395.85	2.06	111.83	0.8	0.02	0.9	25	4	180	20
11	1042.11	3.48	64.16	0.27*	0.01	0.99	9	4	180	20
12	819.56	0	42.73	0*	0.02	0.9	33	4	180	20
13	1541.14	0.43	66.06	0.29	0.02	0.9	40	4	180	20
14	866.37	0	53.27	0*	0.02	0.9	33	4	180	20
平均		0.87	49.48	0.26						

* 表示有多組參數求得此最佳結果，表中僅列出其中一組參數

4.4 演算法績效比較與分析

本研究以近年來發表於國際期刊之文獻做為演算法比較對象，整理如表 4.6。其中，DFB 欄(Deviation from best-known results)表示為單組特定參數下所得之目標成本與已知最佳解之誤差百分比，DFB^b 欄表示為各方法測試過程中所得最佳結果之目標成本與已知最佳解之誤差百分比。而除了 Mester and Bräysy [30]於時間欄內表示為以各題最佳結果所花費時間，其餘時間欄均表示為各方法於單組特定參數下所花費的運算時間，並皆以秒為計時單位。表中最後一列為各方法於程式測試中所使用之執行環境。

由表 4.6 中可以發現，以所獲得結果來看，各篇文獻中以 Mester and Bräysy [30]最佳，其餘文獻則緊追在後，而本研究於 14 題標竿題庫中可找到 7 題最佳結果，且最佳結果之平均誤差亦與各篇文獻相差不遠。另外，在執行時間方面，Mester and Bräysy [30]除了第 5 題之外，其餘題目所花費時間均相當短暫。而由於執行時間受到電腦執行環境、作業系統與程式語言，以及程式撰寫技巧的影響，所以執行時間的比較在此僅供參考，但可以很明顯地感受到，本研究所需要的時間也是非常的少，由此可見本研究之執行架構仍不失為一個不錯的求解方法。

表 4.6 演算法績效比較表

題號	Ergun et al. (2006) [17]			Tarantilis (2005) [37]			Mester & Bräysy (2007) [30]		Ho & Gendreau (2006) [25]		本研究		
	DFB	時間	DFB ^b	DFB	時間	DFB ^b	DFB ^b	時間	DFB	時間	DFB	時間	DFB ^b
1	0	325.2	0	0	24.6	0	0	0.2	0	46.8	0	10.7	0
2	0.91	2020.2	0.02	0	291.6	0	0	5.5	0.13	116.4	1.38	15.39	0.05
3	0.02	1680.6	0	0	514.2	0	0	1	0.26	144.6	0.27	33.09	0
4	0.55	3202.8	0.45	0.12	594	0	0	10.2	0.55	331.8	1.46	60.52	0.32
5	1.29	3761.4	0.9	1.56	787.2	1.56	0	2160	1.59	628.2	1.96	100.8	1.28
6	0	1400.4	0	0	23.4	0	0	4.2	0	45	0	12.05	0
7	0	2270.4	0.4	0	61.2	0	0	0.8	0	98.4	0	19.69	0
8	0	2550.6	0	0	225.6	0	0	0.8	0.09	148.8	0.18	37.59	0
9	1.2	2197.2	0.05	0.06	404.4	0	0	25.8	1.24	406.2	0.97	64.81	0.56
10	0	2619	1.02	0.81	1292.4	0.81	0.38	52.2	1.78	633	2.06	111.83	0.8
11	0.11	993.6	0	0	19.2	0	0	1.1	0.08	216	3.48	64.16	0.27
12	1.13	100.8	0	0	17.4	0	0	0.2	0	175.8	0	42.73	0
13	0.41	1121.4	0.14	0.19	447	0.19	0	13.5	1.79	339.6	0.43	66.06	0.29
14	0	2804.4	0	0	22.8	0	0	1.7	0	140.4	0	53.27	0
平均	0.40	1932	0.21	0.2	337.5	0.18	0.03	162.66	0.54	247.93	0.87	49.48	0.26
執行環境	Sun Sparc IIe 500MHz			Pentium II 400 MHz			Pentium IV 2800 MHz		Pentium 4 2.53 GHz		Pentium(R) 4 3.00GHz		

註：DFB表示在單組特定參數下之所得結果與已知最佳解之誤差百分比，DFB^b則是測試過程中所得最佳結果與已知最佳解之誤差百分比。

第五章 結論與建議

5.1 結論

本研究以 Tarantilis et al. [38]所提出之可回溯式門檻接受法(BATA)為基礎，並與 GENIUS 和其他傳統鄰域搜尋法相結合，來求解 VRP 問題。在鄰域搜尋模組的改良上，主要有兩項：(1) 以巨網結構的方式使鄰域搜尋能同時考慮「路線內」與「路線間」的交換。(2) 以擴大解繫法鄰域搜尋範圍的設計(Expanded US)，使其能做更有效的搜尋。在可回溯式門檻接受模組中，則以修正的門檻值回溯公式，分別測試門檻回溯比值 b 小於 1 或大於 1 的情況。本研究是以 Christofides et al. [8]的 14 題國際標竿測試例題，進行求解狀況分析，並與國際期刊之文獻所發表結果進行比較與分析。本研究所歸結之結論分述如下：

- (1) 本研究以巨網結構的方式使鄰域搜尋能同時考慮「路線內」與「路線間」的交換，而擴大解繫法鄰域搜尋範圍的設計確實有其功效，因此鄰域搜尋模組能有效地進行搜尋。
- (2) 傳統上門檻回溯比值 $b < 1$ 的設定可在起始門檻比率 0.01、門檻下降比率 0.9，以及門檻數列長度 180 的情況下，獲得 14 題平均誤差為 1.2%的結果
- (3) 另外，本研究使用修正的門檻值回溯公式，來測試 $b > 1$ 的情況。結果發現，在起始門檻比率 0.02 附近的範圍，門檻下降比率 0.9 以上附近的範圍，以及門檻數列長度 180 的設定下，其結果均相當不錯。其中最低的 14 題平均誤差更可低至 0.87%。表示此種更為放鬆的設計可能可以找到較佳的解。
- (4) 本研究所得出之最佳結果與近年來國際文獻報導的數個績效最好的方法相比。發現本研究於 14 題例題中共有 7 題找到最佳結果，而平均誤差 0.26%。且本研究的求解時間相當快，顯示此可回溯式門檻接受法可成為相當不錯的巨集啟發式方法。

5.2 建議

本研究針對未來可能的相關研究做以下兩點建議僅供參考：

- (1) 本研究測試門檻回溯比值 $b > 1$ 的情況下，可能有較佳的求解績效；未來可依此結論嘗試應用於其他問題架構，如 VRPTW。
- (2) 本研究關於門檻值回溯幅度的討論，表明了適當的回溯幅度有助於搜尋脫離區域最佳解，但無法依此確定其搜尋時建議的回溯幅度。因此，未來或可嘗試修正門檻值回溯公式，而能夠在搜尋中視情況調整回溯幅度，來獲得更佳的解。如隨著連續回溯次數的增加，適當增加回溯幅度。

參考文獻

1. 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, 碩士論文, 民國 95 年。
2. 卓裕仁,「以巨集啟發式解法求解多車種與週期性車輛路線問題之研究」, 國立交通大學, 博士論文, 民國 89 年。
3. 陳國清,「GDA 與 RRT 啟發式解法在 VRP 問題上之應用」, 國立交通大學, 碩士論文, 民國 87 年。
4. 楊智凱,「以門檻接受法改善 TSP 及 VRP 路網成本」, 國立交通大學, 碩士論文, 民國 84 年。
5. Blasum, U. and Hochstättler, W., "Application of the Branch and Cut Method to the Vehicle Routing Problem," Technical Report ZPR2000-386, Zentrum für Angewandte Informatik Köln, 2000.
6. Bodin, L., Golden, B., Assad, A. and Ball, M., "Routing and scheduling of vehicle and crew: the state of art," *Special Issue of Computers and Operations Research*, Vol.10, No.2, 1983, pp. 63-211.
7. Christofides, N., Eilon, S., "An algorithm for the vehicle-dispatching problem," *Operational Research Quarterly*, Vol. 20, No. 3, 1969, pp. 309-318.
8. Christofides, N., Mingozzi, A., Toth, P. and Sandi, C., *Combinatorial Optimization*, John Wiley & Sons, Inc., 1979.
9. Clarke, G and Wright, J., "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, Vol. 12, 1964, pp.568-589.
10. Cordeau, J-F., Gendreau, M., Laporte, G, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, Vol. 30, N0.2, 1997, pp.105-119.
11. Cordeau, J-F., Gendreau, M., Laporte, G, Potvin, J-Y. and Semet, F., "A guide to vehicle routing heuristics," *Journal of the Operational Research Society*, Vol. 53, 2002, pp.512-522.
12. Cordeau, J-F., Laporte, G and Mercier, A., "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, Vol. 52, 2001, pp.928-936.

13. Cordeau, J-F., Laporte, G, Savelsbergh, M.W.P. and Vigo, D., "Vehicle routing," in: C. Barnhart and Laporte, G (eds), *Handbooks in Operations Research and Management Science: Transportation*, Vol.14, Elsevier B.V., 2005.
14. Dantzig, G. B. and Ramser, R.H., "The truck dispatching problem," *Management Science*, Vol. 6, 1959, pp.80–91.
15. Dueck, G, "New optimization heuristics: the great deluge algorithm and the record-to-record travel," *Journal of Computational Physics*, Vol. 104, 1993, pp. 86-92.
16. 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.
17. Ergun, Ö., Orlin, J.B. and Steele-Feldman, A., "Creating very large scale neighborhoods out of smaller ones by compounding moves," *Journal of Heuristics*, Vol. 12, 2006, pp. 115-140.
18. Franceschi, R.D., Fischetti, M. and Toth, P., "A new ILP-based refinement heuristic for Vehicle Routing Problems," *Mathematical Programming*, Vol. 105, 2006, pp. 471-499.
19. Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M.P.D., Reis, M., Uchoa, E., and Werneck, R.F., "Robust branch-and-cut-and-price for the capacitated vehicle routing problem", *Mathematical Programming*, Vol. 106, No. 3, 2006, pp.491-511.
20. Funke, B., Grünert, T. and Irnich, S., "Local search for vehicle routing and scheduling problems: review and conceptual integration," *Journal of Heuristics*, Vol. 11, 2005, pp.267–306.
21. Gendreau, M., Hertz, A. and Laporte, G., "New insertion and postoptimization procedures for the Traveling Salesman Problem.," *Operations Research*, Vol. 40, 1992, pp.1086-1094.
22. Gendreau, M., Hertz, A. and Laporte, G., "A tabu search heuristic for the vehicle routing problem," *Management Science*, Vol. 40, No. 10, 1994, pp.1276-1290.
23. Gendreau, M., Hertz, A., Laporte, G. and Stan, M., "A generalized insertion heuristic for the traveling salesman problem with time windows," *Operations Research*, Vol. 43, No.3, 1998, pp.330-335.
24. Golden, B.L. and Assad, A.A., *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers B.V., 1988.
25. Ho, S.C. and Gendreau, M., "Path relinking for the vehicle routing problem," *Journal of Heuristics*, Vol.12, 2006, pp. 55-72.

26. Kirkpatrick, S., Gelatt, C., and Vecchi, M., "Optimization by Simulated Annealing," *Science*, Vol.220, 1983, pp. 671-680.
27. Lawler, E., Lenstra, J., Rinnooy Kan, A. and Shmoys, D., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons, Inc., 1985.
28. Lin, S., "Computer solutions of the traveling salesman problem," *The Bell System Technical Journal*, 1965, pp.2245-2269.
29. Mester, D. and Bräysy, O., "Active guided evolution strategies for large-scale vehicle routing problems with time windows," *Computers & Operations Research*, Vol. 32, 2005, 1593–1614.
30. Mester, D. and Bräysy, O., "Active guided evolution strategies for large-scale capacitated vehicle routing problems," *Computers & Operations Research*, Vol. 34, 2007, 2964–2975.
31. Network Lab: <http://140.113.119.114/network/>, 07/27/2007.
32. Or, I., *Traveling Salesman-type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking*, PhD. Dissertation, Northwestern University, 1996.
33. Osman, I.H., "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research*, Vol. 41, 1993, pp.421-451.
34. Rochat, Y. and Taillard, É.D., "Probabilistic diversification and intensification in local search for vehicle routing," *Journal of heuristics*, Vol. 1, No. 1, 1995, pp. 147-167.
35. Rosenkrantz, D., Sterns, R. and Lewis, P., "An analysis of several heuristics for the traveling salesman problem," *SIAM Journal of Computing*, Vol. 6, 1977, pp.563-581.
36. Taillard, É.D., "Parallel Iterative Search Methods for Vehicle Routing Problems," *Networks*, Vol.23, No.8, 1993, pp.661-673.
37. Tarantilis, C.D., "Solving the vehicle routing problem with adaptive memory programming methodology," *Computers & Operations Research*, Vol. 32, 2005, pp.2309–2327.
38. Tarantilis, C.D. and Kiranoudis, C.T.. "A meta-heuristic algorithm for the efficient distribution of perishable foods," *Journal of Food Engineering*, Vol. 50, 2001, pp. 1-9.
39. Tarantilis, C.D., Kiranoudis, C.T. and Vassiliadis, V.S., "A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem." *Journal of the Operational Research Society*, Vol. 54, 2003, pp. 65-71.
40. Tarantilis, C.D., Kiranoudis, C.T. and Vassiliadis, V.S., "A threshold accepting metaheuristic for

the heterogeneous fixed fleet vehicle routing problem.,” *European Journal of Operational Research*, Vol. 152, 2004, pp.148-158.

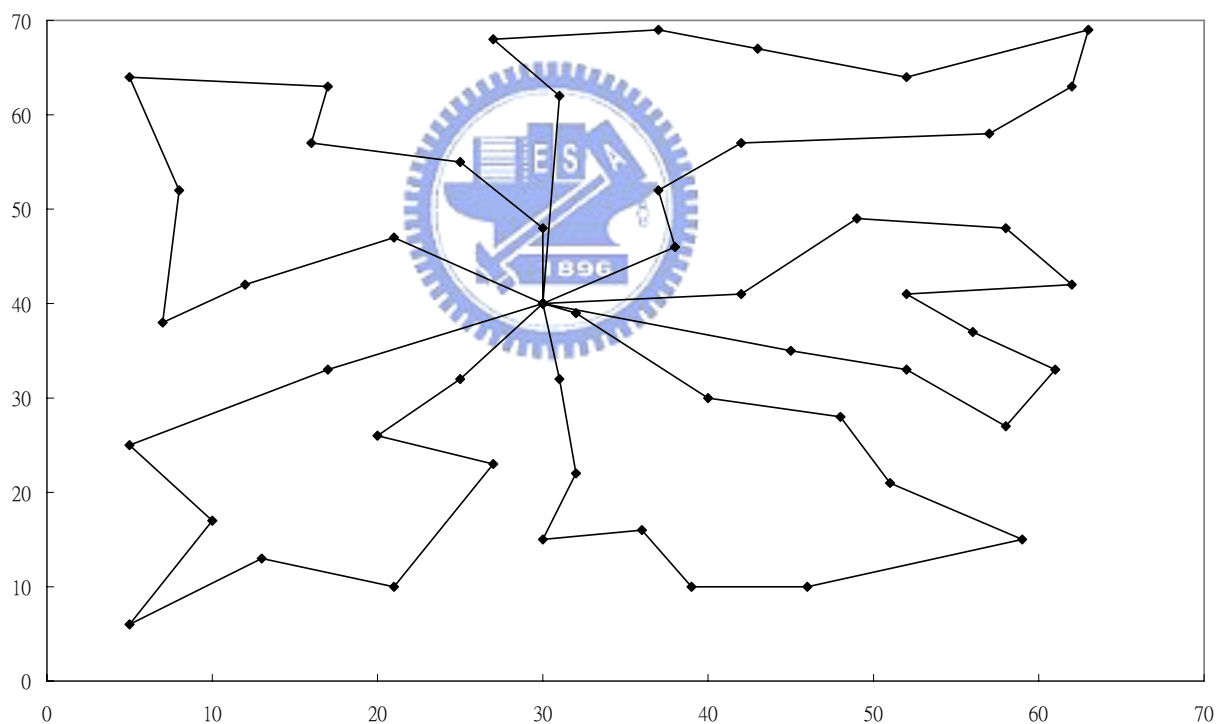
41. Università di Bologna-D.E.I.S.- Operations Research, <http://www.or.deis.unibo.it/>, 07/27/2007.
42. Xu, J. and Kelly, J.P., “A network flow-based tabu search heuristic for the vehicle routing problem,” *Transportation Science*, Vol. 30, No.4, 1996, pp. 379-393.
43. VRP Web: <http://neo.lcc.uma.es/radi-aeb/WebVRP/>, 07/27/2007.



附錄 最佳個案結果及路線圖

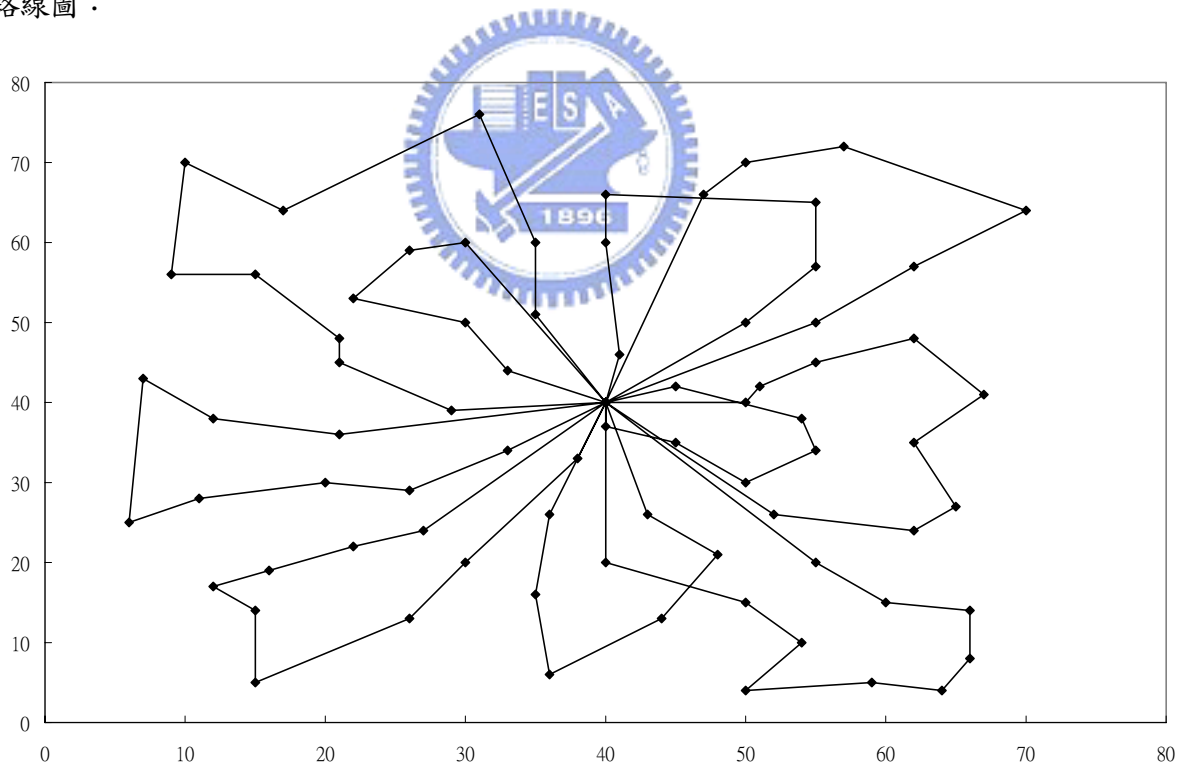
例題編號：1			已知最佳解：524.61			路線成本：524.61 (0.0%)									
顧客數：50		車輛容量：160		最大旅行時間： ∞		服務時間：0									
路線編號	經過節點順序											承載量	成本		
1	0	27	48	23	7	43	24	25	14	6	0	152	98.45		
2	0	32	1	22	20	35	36	3	28	31	26	8	0	149	118.52
3	0	11	2	29	21	16	50	34	30	9	38	0	159	99.33	
4	0	18	13	41	40	19	42	17	4	47	0	157	109.06		
5	0	12	37	44	15	45	33	39	10	49	5	46	0	160	99.25

路線圖：



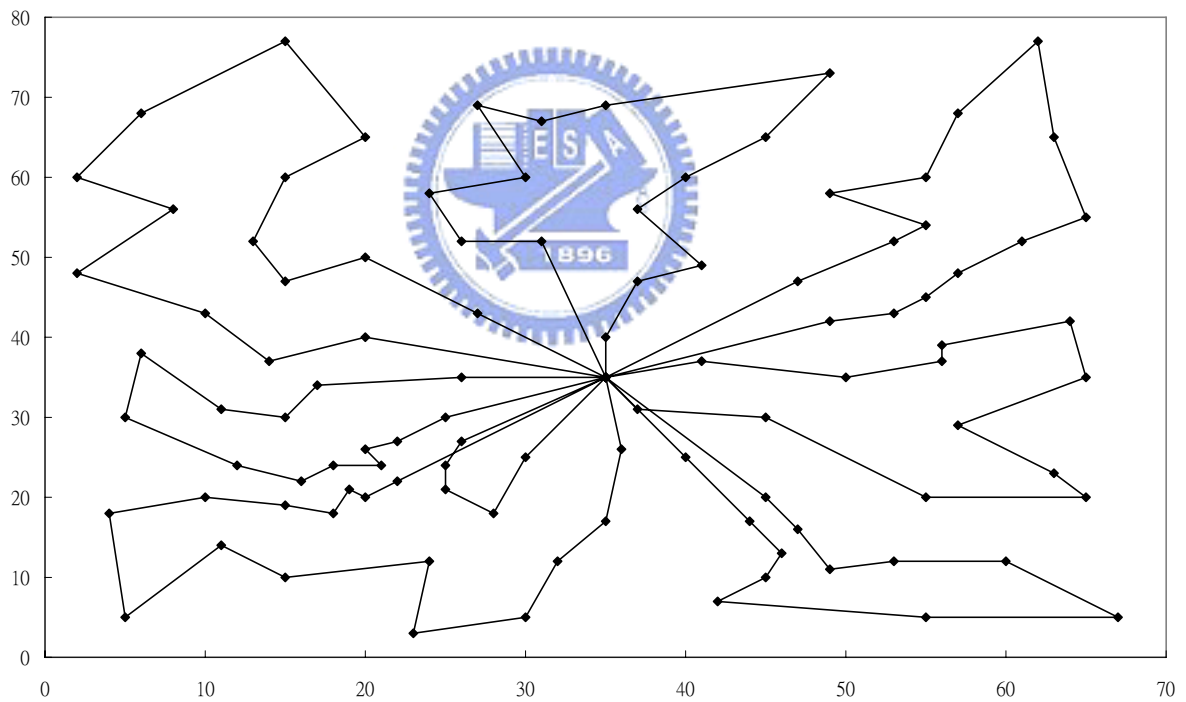
例題編號：2		已知最佳解：835.26				路線成本：835.67 (0.05 %)								
顧客數：75		車輛容量：140		最大旅行時間： ∞			服務時間：0							
路線編號	經過節點順序											承載量	成本	
1	0	73	1	43	41	42	64	22	62	68	0	136	94.63	
2	0	12	72	31	25	55	18	50	44	3	51	0	138	119.32
3	0	17	40	32	9	39	0						126	57.01
4	0	75	4	45	27	52	67	0					137	41.22
5	0	34	46	8	19	54	13	57	15	29	0	140	82.69	
6	0	7	53	11	10	58	26	0					139	71.9
7	0	30	48	21	61	28	2	0					140	75.62
8	0	16	49	24	56	23	63	33	6	0			140	92.69
9	0	5	37	20	70	60	71	69	36	47	74	0	133	107.57
10	0	38	65	66	59	14	35	0					135	93.03

路線圖：



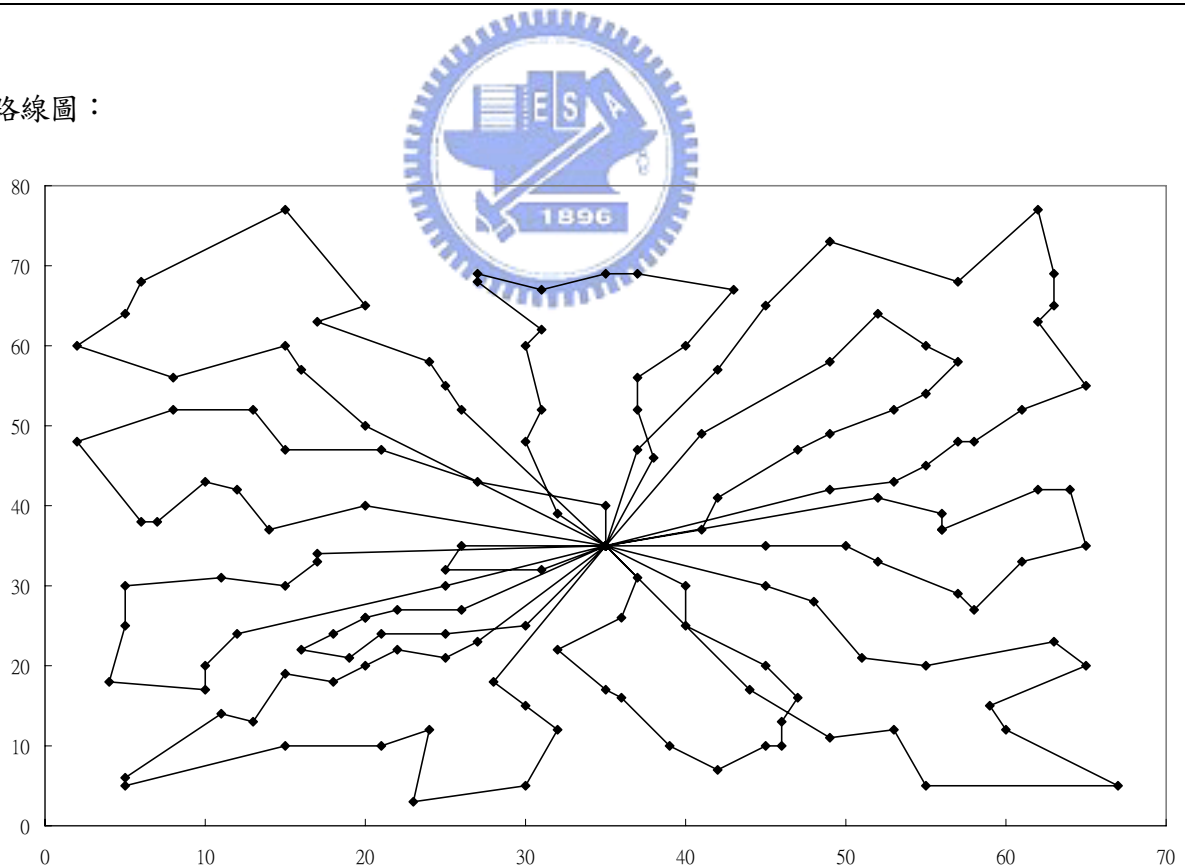
例題編號：3		已知最佳解：826.14		路線成本：826.14 (0.0%)																
顧客數：100		車輛容量：200		最大旅行時間： ∞																
				服務時間：0																
路線編號	經過節點順序														承載量	成本				
1	0	6	96	99	59	93	85	61	17	45	84	5	60	89	0	196	82.73			
2	0	31	88	62	10	63	90	32	66	20	30	70	1	69	27	0	199	113.93		
3	0	18	83	8	46	47	36	49	64	11	19	48	82	7	52	0	199	138.79		
4	0	92	37	98	100	91	16	86	38	44	14	42	43	15	57	2	58	0	198	126.66
5	0	40	73	74	22	41	23	67	39	56	75	72	21	0	194	106.06				
6	0	28	12	80	68	29	24	54	55	25	4	26	53	0	165	98.25				
7	0	50	33	81	51	9	71	65	35	34	78	79	3	77	76	0	199	118.79		
8	0	13	87	97	95	94	0	108	40.91											

路線圖：



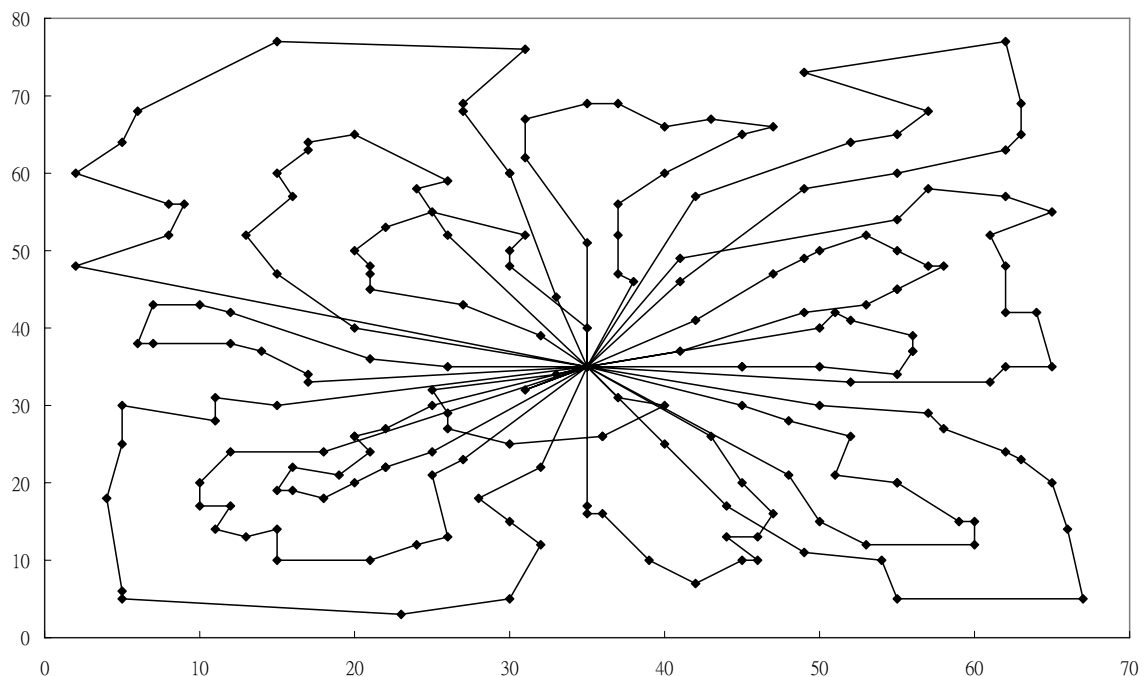
例題編號：4	已知最佳解：1028.42	路線成本：1031.71 (0.32 %)
顧客數：150	車輛容量：200	最大旅行時間： ∞ 服務時間：0
路線編號	經過節點順序	承載量 成本
1	0 1 51 103 9 120 81 33 102 50 111 28 0	195 75.2
2	0 60 118 5 84 17 113 86 141 16 61 6 0	199 83.96
3	0 117 97 92 37 100 91 119 44 140 38 14 142 42 43 15 57 144 87 0	195 118.79
4	0 89 147 112 0	69 23.16
5	0 88 148 62 107 11 64 49 143 36 47 19 123 7 0	196 120.33
6	0 94 96 104 99 93 85 98 59 95 13 0	200 50.98
7	0 146 127 31 10 108 126 63 90 32 131 128 30 70 101 132 0	199 88.22
8	0 18 83 114 8 125 45 46 124 48 82 106 52 27 0	200 92.09
9	0 73 75 56 23 67 39 139 25 55 4 110 149 26 0	199 110.88
10	0 116 68 150 80 121 29 24 134 130 54 109 12 138 0	195 79.03
11	0 76 77 3 79 129 78 34 135 35 136 65 71 66 20 122 69 0	200 120.62
12	0 53 58 137 2 115 145 41 22 133 74 72 21 40 105 0	188 68.44

路線圖：



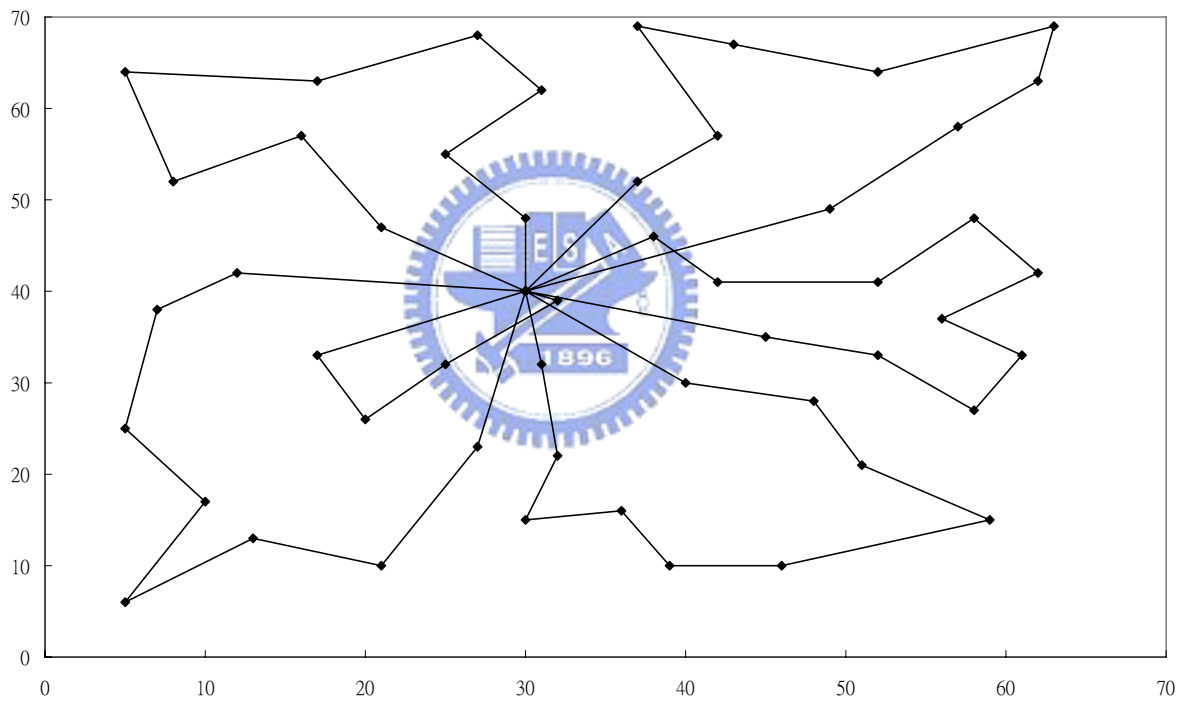
例題編號：5		已知最佳解：1091.29		路線成本：1307.87 (1.28%)														
顧客數：199		車輛容量：200		最大旅行時間： ∞														
				服務時間：0														
路線編號	經過節點順序											承載量	成本					
1	0	112	0										29	10				
2	0	132	69	101	70	30	20	188	128	160	131	32	90	108	162	0	200	86.83
3	0	195	54	130	165	55	25	170	67	23	186	75	73	40	0	198	100.41	
4	0	89	166	114	8	174	45	125	199	83	60	118	0	195	66.84			
5	0	184	196	116	68	150	80	177	12	154	138	0	194	49.2				
6	0	28	76	77	3	158	129	79	185	33	157	102	50	111	0	199	58.49	
7	0	26	149	179	110	155	4	139	187	39	56	197	198	0	197	75.56		
8	0	18	82	48	123	19	107	175	11	159	62	88	0	199	82.84			
9	0	95	151	92	37	100	193	91	85	98	59	104	99	96	6	0	195	57.19
10	0	137	87	144	57	15	43	38	140	86	113	17	173	84	5	0	195	118.06
11	0	27	127	190	31	148	182	7	194	106	153	52	146	0	198	55.55		
12	0	117	97	172	42	142	14	192	119	44	191	141	16	61	93	0	199	84.51
13	0	53	105	152	58	13	94	183	147	156	0	200	39.49					
14	0	1	81	120	164	34	78	169	121	29	24	163	134	109	0	196	98.82	
15	0	46	124	168	47	36	143	49	64	181	63	126	189	10	167	0	200	136
16	0	2	178	115	145	41	22	133	171	74	72	21	180	0	196	67.8		
17	0	122	103	161	71	66	65	136	35	135	9	51	176	0	196	120.29		

路線圖：



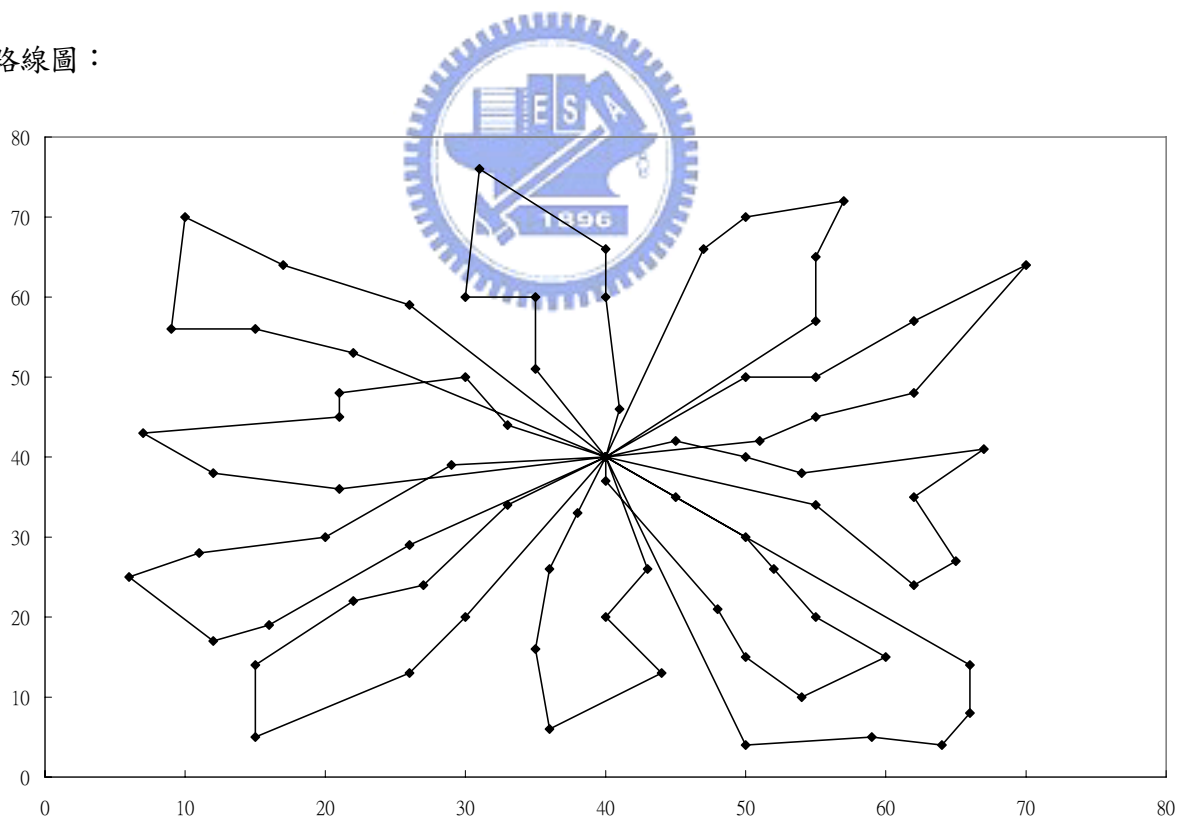
例題編號：6		已知最佳解：555.43				路線成本：555.43 (0.0%)								
顧客數：50		車輛容量：160			最大旅行時間：200			服務時間：10						
路線編號	經過節點順序											承載量	成本	
1	0	18	4	47	46	0							80	42.33
2	0	2	20	35	36	3	28	31	22	1	0	137	108.08	
3	0	27	48	8	26	7	43	24	23	6	0	133	100.64	
4	0	17	42	19	40	41	13	25	14	0	131	109.94		
5	0	38	9	30	34	50	21	29	16	11	32	0	141	95.33
6	0	12	37	44	15	45	33	39	10	49	5	0	155	99.12

路線圖：



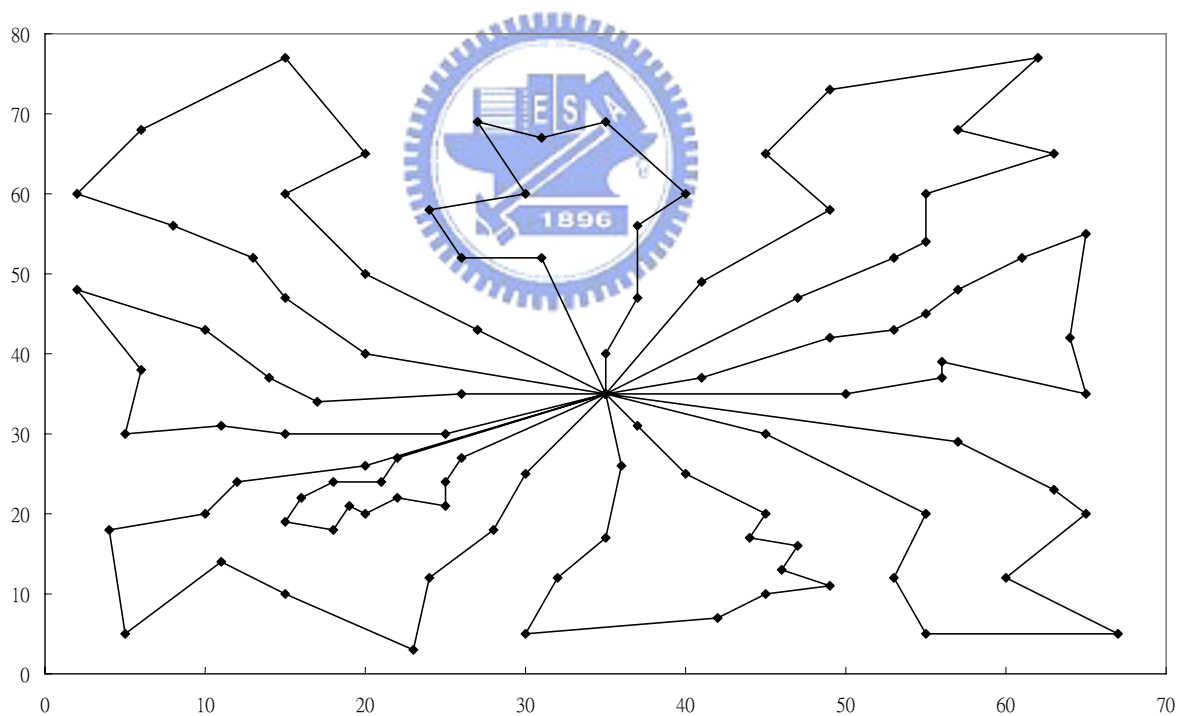
例題編號：7			已知最佳解：909.68				路線成本：909.68 (0.0%)					
顧客數：75			車輛容量：140			最大旅行時間：160			服務時間：10			
路線編號	經過節點順序										承載量	成本
1	0	4	20	70	60	71	69	0			87	98.76
2	0	6	73	1	42	64	22	62	0		112	89.92
3	0	16	49	24	3	44	40	17	0		132	76.84
4	0	12	72	39	31	10	58	26	0		123	81.69
5	0	75	48	47	36	37	5	29	45	0	140	73.82
6	0	27	15	57	13	54	52	34	67	0	135	77.54
7	0	68	2	28	61	21	74	30	0		140	74.38
8	0	53	11	66	65	38	0				129	77.16
9	0	51	63	23	56	41	43	33	0		115	85.24
10	0	9	25	55	18	50	32	0			113	92.97
11	0	7	35	14	59	19	8	46	0		138	81.36

路線圖：



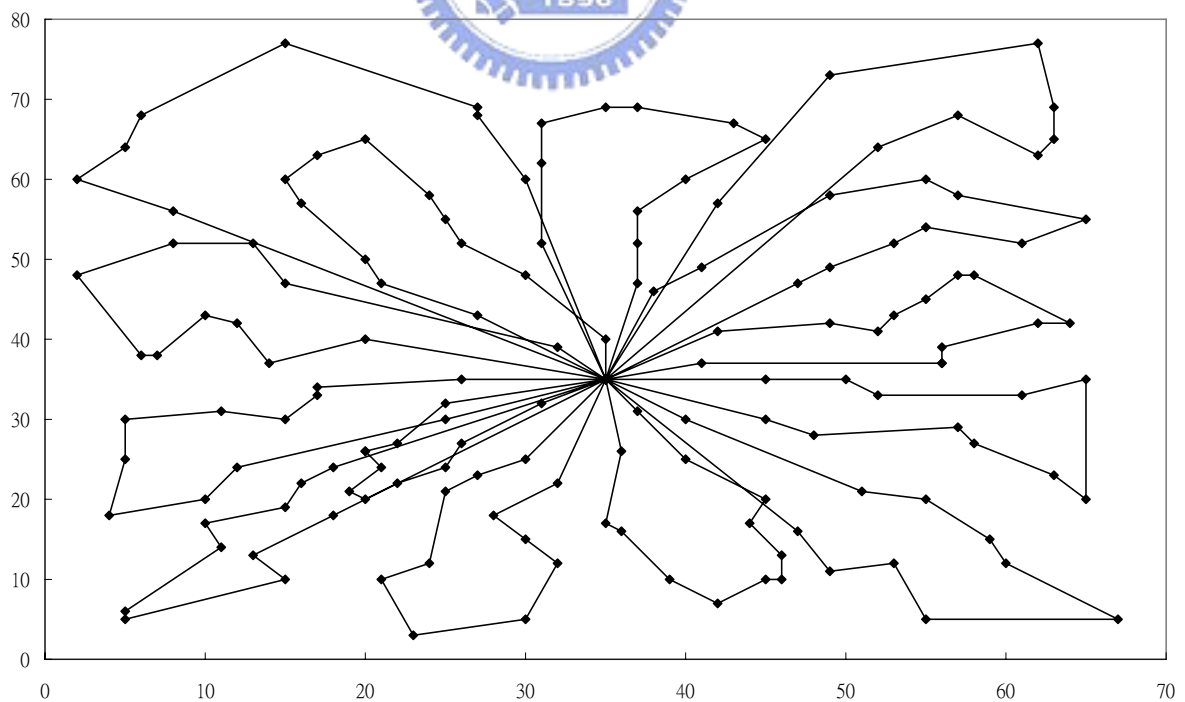
例題編號：8			已知最佳解：865.94				路線成本：865.94 (0.0%)										
顧客數：100			車輛容量：200			最大旅行時間：230			服務時間：10								
路線編號	經過節點順序														承載量	成本	
1	0	28	76	77	3	79	78	34	29	24	68	80	12	0	169	90.26	
2	0	94	95	97	92	37	98	100	91	85	93	59	96	0	199	59.35	
3	0	1	51	20	66	65	71	35	9	81	33	50	0		163	117.93	
4	0	58	2	57	15	41	22	75	74	72	73	21	40	53	0	157	83.1
5	0	18	82	48	47	36	49	64	11	19	7	52	0		178	117.55	
6	0	26	4	56	23	67	39	25	55	54	0				153	107.08	
7	0	89	60	83	8	46	45	17	84	5	6	0			93	89.16	
8	0	13	87	42	43	14	44	38	86	16	61	99	0		191	111.4	
9	0	27	69	70	30	32	90	63	10	62	88	31	0		155	90.12	

路線圖：



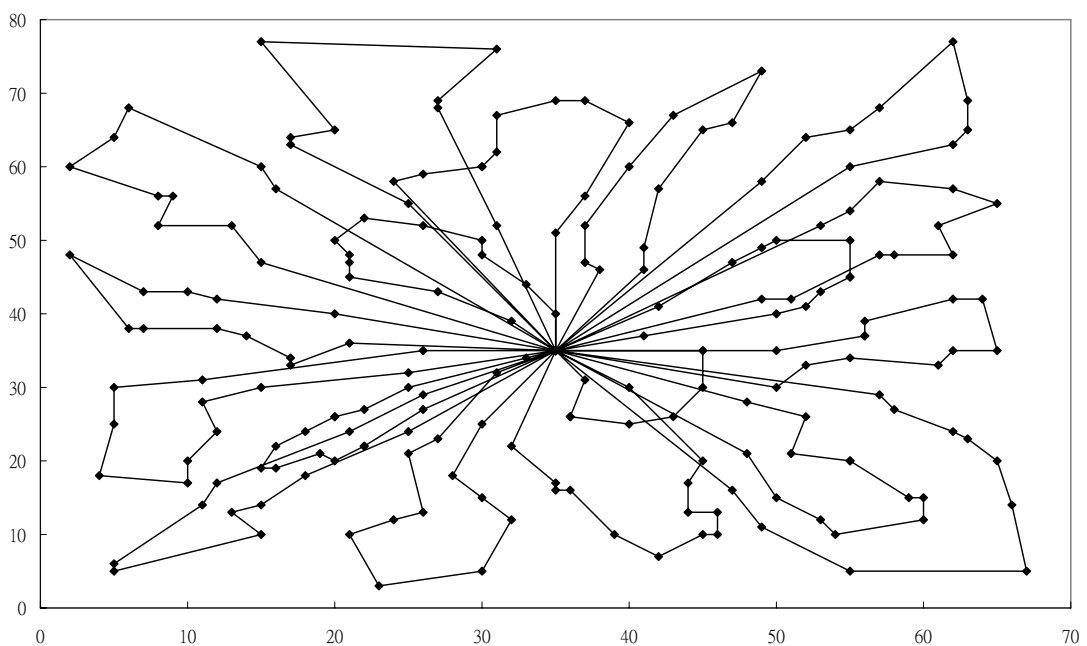
例題編號：9				已知最佳解：1162.55				路線成本：1169.09 (0.56%)									
顧客數：150				車輛容量：200				最大旅行時間：200				服務時間：10					
路線編號	經過節點順序														承載量	成本	
1	0	26	149	54	130	55	25	24	134	109	12	138	0	154	82.39		
2	0	47	36	143	49	64	63	126	10	0	115	112.73					
3	0	50	102	33	81	78	34	120	9	51	1	132	0	173	84.33		
4	0	122	66	65	136	35	135	71	103	0	115	115.54					
5	0	72	75	56	23	67	39	139	4	110	105	0	193	96.13			
6	0	111	76	116	77	3	79	129	29	121	68	80	150	28	0	188	69.64
7	0	69	101	70	30	20	128	131	32	90	108	31	0	149	81.33		
8	0	93	85	91	141	44	140	38	14	119	100	0	178	91.69			
9	0	112	94	95	92	37	98	59	104	99	96	147	0	178	47.43		
10	0	58	2	115	145	41	22	133	74	73	21	40	53	0	142	64.6	
11	0	6	61	16	86	113	17	84	5	118	60	89	0	187	81.23		
12	0	137	87	144	57	15	43	142	42	97	117	13	0	129	78.33		
13	0	52	106	7	123	19	107	11	62	148	88	127	27	0	169	74.6	
14	0	146	82	48	124	46	45	125	8	114	83	18	0	165	89.13		

路線圖：



例題編號：10		已知最佳解：1395.85		路線成本：1407.03 (0.8%)														
顧客數：199		車輛容量：200		最大旅行時間：200														
				服務時間：10														
路線編號	經過節點順序											承載量	成本					
1	0	9	135	35	136	65	71	161	103	51	0	145	104.63					
2	0	76	196	79	129	169	78	34	164	120	81	33	0	197	81.02			
3	0	72	75	23	67	170	25	55	165	130	54	0	172	98.54				
4	0	12	80	150	68	121	29	24	163	134	177	109	195	0	178	73		
5	0	198	197	56	186	39	187	139	4	155	110	179	149	0	192	77.1		
6	0	28	184	116	77	158	3	185	157	102	50	111	0	180	54.38			
7	0	138	154	26	180	40	58	152	53	0	151	36.33						
8	0	94	151	92	37	98	193	91	85	93	104	99	96	6	156	0	198	53.77
9	0	162	70	160	131	32	90	108	10	189	159	62	0	187	81.63			
10	0	137	2	178	115	145	41	22	133	74	171	73	21	105	0	188	67.78	
11	0	147	5	173	61	16	141	86	113	17	84	89	0	198	85.47			
12	0	148	107	175	11	64	181	63	126	31	0	140	109.89					
13	0	183	59	191	44	140	38	14	119	192	100	95	0	188	89.56			
14	0	146	52	153	106	194	7	182	88	190	127	167	27	0	183	51.64		
15	0	18	114	8	174	46	45	125	199	83	60	118	166	0	193	79.65		
16	0	13	87	144	57	15	43	142	42	172	97	117	112	0	161	79.09		
17	0	132	69	101	30	128	66	188	20	122	1	176	0	154	84.11			
18	0	82	48	124	168	47	36	143	49	19	123	0	181	99.44				

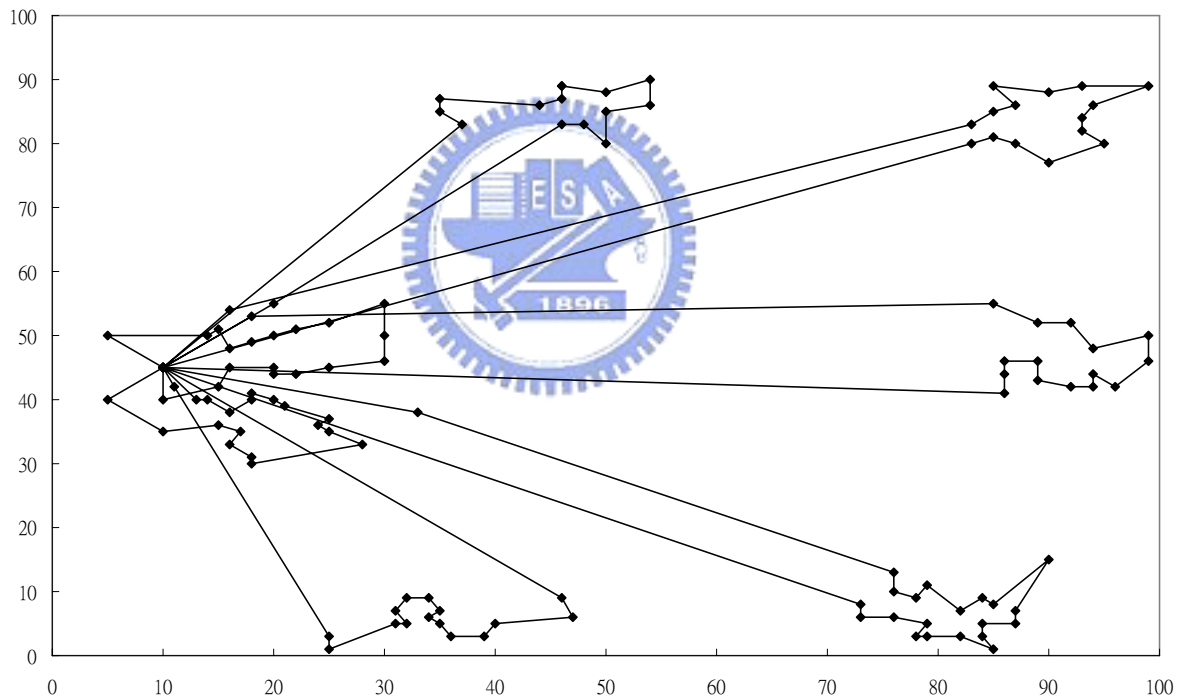
路線圖：



例題編號：11	已知最佳解：1042.11	路線成本：1044.95 (0.27%)
顧客數：120	車輛容量：200	最大旅行時間： ∞ 服務時間：0

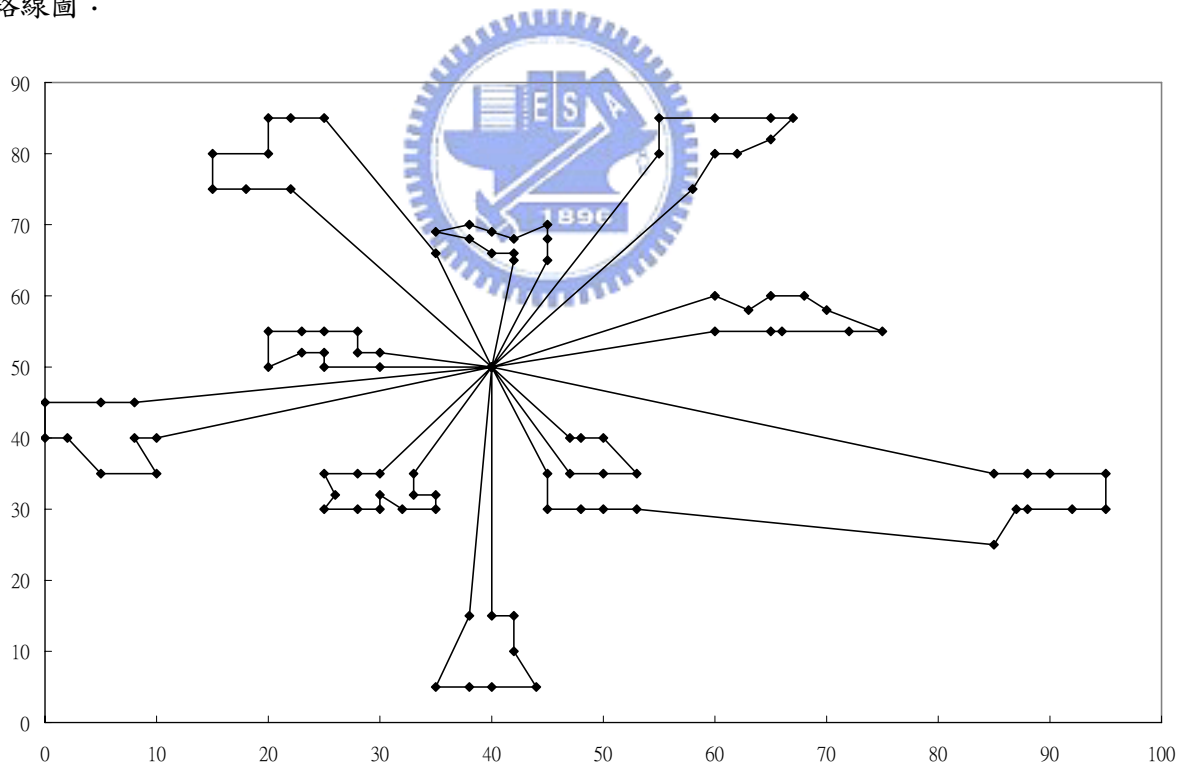
路線編號	經過節點順序	承載量	成本
1	0 82 87 95 96 93 94 97 115 110 98 116 100 99 101 102 106 105 120 0	192	74.16
2	0 88 2 1 3 4 5 6 7 9 10 11 15 14 13 12 8 0	199	134.96
3	0 52 54 57 59 65 61 62 64 66 63 60 56 58 55 53 107 0	197	215.15
4	0 119 81 112 84 117 113 83 108 118 18 114 90 91 92 89 85 86 111 0	192	66.03
5	0 109 21 20 23 26 28 32 35 29 36 34 31 30 33 27 24 22 25 19 16 17 0	197	207.94
6	0 103 73 76 68 77 79 80 78 72 75 74 71 70 69 67 0	200	144.41
7	0 37 38 39 42 41 44 46 47 49 50 51 48 45 43 40 104 0	198	202.3

路線圖：



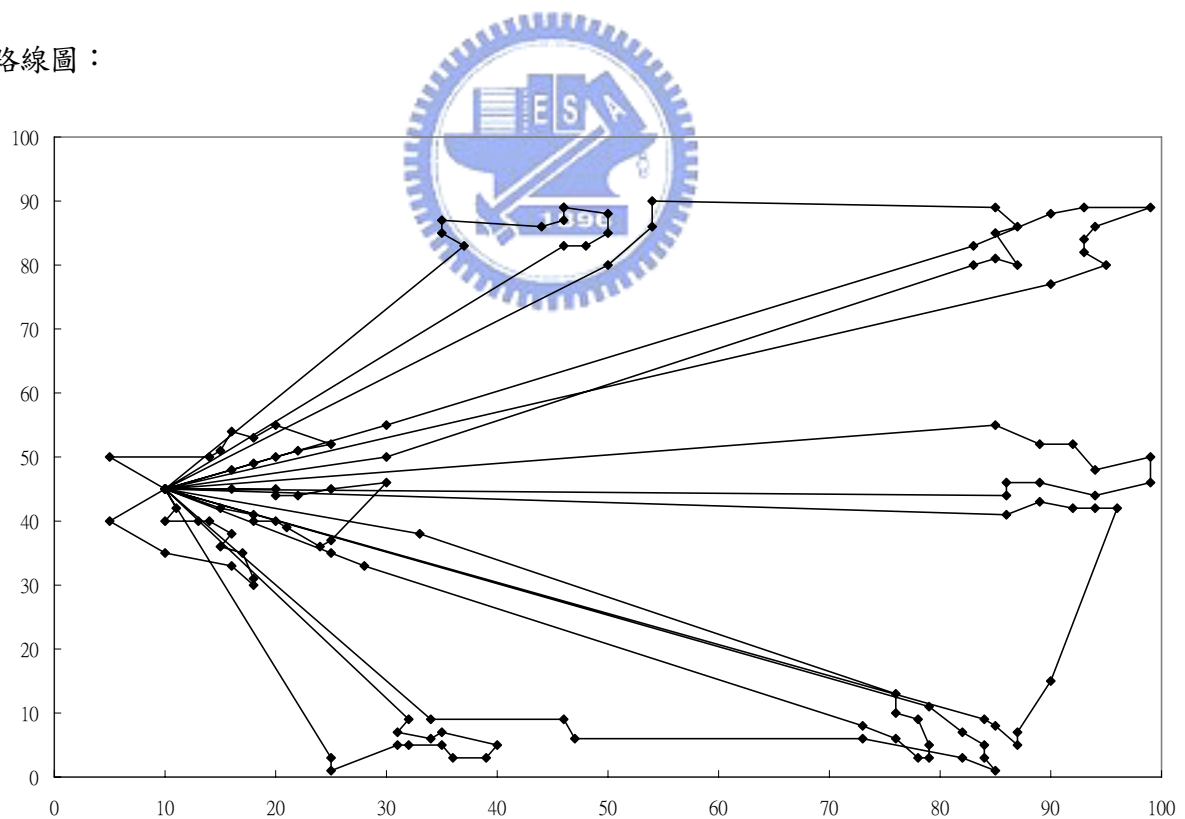
例題編號：12		已知最佳解：819.56		路線成本：819.56(0.0%)														
顧客數：100		車輛容量：200		最大旅行時間： ∞														
				服務時間：0														
路線編號	經過節點順序											承載量	成本					
1	0	47	49	52	50	51	48	45	46	44	40	41	42	43	0	160	64.81	
2	0	91	89	88	85	84	82	83	86	87	90	0	170	76.07				
3	0	66	62	74	63	65	67	0	150	43.59								
4	0	5	3	7	8	11	9	6	4	2	1	75	0	170	56.17			
5	0	34	36	39	38	37	35	31	33	32	0	200	97.23					
6	0	20	24	25	27	29	30	28	26	23	22	21	0	170	50.8			
7	0	69	68	64	61	72	80	79	77	73	70	71	76	78	81	0	200	137.02
8	0	13	17	18	19	15	16	14	12	10	0	200	96.04					
9	0	98	96	95	94	92	93	97	100	99	0	190	95.94					
10	0	57	55	54	53	56	58	60	59	0	200	101.88						

路線圖：



例題編號：13		已知最佳解：1541.14			路線成本：1545.54 (0.29%)												
顧客數：120		車輛容量：200		最大旅行時間：720			服務時間：50										
路線編號	經過節點順序													承載量	成本		
1	0	110	52	54	57	55	58	56	80	79	68	0	106	192.59			
2	0	40	43	45	48	51	50	47	42	39	38	0	127	189.3			
3	0	37	41	44	46	49	29	36	34	35	32	0	86	212.39			
4	0	73	76	77	78	75	72	74	71	70	69	67	0	162	129.03		
5	0	120	105	106	107	104	103	116	100	99	101	102	0	99	48.12		
6	0	6	5	10	9	13	14	15	11	4	3	1	2	0	148	119.67	
7	0	118	108	17	19	22	24	25	23	20	21	109	0	124	166.46		
8	0	87	92	89	91	90	18	114	115	97	94	93	96	95	0	162	50.62
9	0	26	28	31	30	33	27	16	12	8	7	0	114	186.51			
10	0	59	65	61	62	64	66	63	60	53	98	0	131	204.96			
11	0	119	81	117	83	113	84	112	85	86	111	82	88	0	116	45.89	

路線圖：



例題編號：14				已知最佳解：866.37				路線成本：866.37 (0.0%)						
顧客數：100		車輛容量：200		最大旅行時間：1040				服務時間：90						
路線編號	經過節點順序											承載量	成本	
1	0	21	22	24	25	27	29	30	28	26	23	0	160	49.41
2	0	90	87	86	83	82	84	85	88	89	91	0	170	76.07
3	0	69	66	68	64	61	72	74	62	65	67	0	150	57.79
4	0	20	49	52	50	51	48	45	46	47	0	110	61.56	
5	0	1	99	100	97	93	92	94	95	96	98	0	200	96.7
6	0	43	42	44	40	41	0	60	45.47					
7	0	63	80	79	77	73	70	71	76	78	81	0	200	128.04
8	0	13	17	18	19	15	16	14	12	10	0	200	96.04	
9	0	57	59	60	58	56	53	54	55	0	200	101.88		
10	0	32	33	31	35	37	38	39	36	34	0	200	97.23	
11	0	75	2	4	6	9	11	8	7	3	5	0	160	56.17

路線圖：

