

國立交通大學

運輸科技與管理學系碩士班

碩士論文

二階段基因演算法於

具時間窗限制車輛路線問題之應用

A two-phased genetic algorithm for

the vehicle routing problem with time window

研究生：陳綠茵

指導教授：王晉元

中華民國九十七年七月

二階段基因演算法於具時間窗限制車輛路線問題之應用

A two-phased genetic algorithm for  
the vehicle routing problem with time window

研究生：陳綠茵

Student : Lui-Yin Chen

指導教授：王晉元

Advisor : Jin-Yuan Wang

國立交通大學  
運輸科技與管理學系  
碩士論文



Submitted to Institute of Transportation Technology and Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Engineering

in

Transportation Technology and Management

July 2008

Hsinchu, Taiwan, Republic of China

中華民國 九十七 年 七 月

## 二階段基因演算法於具時間窗限制車輛路線問題之應用

學生姓名：陳綠茵

指導教授：王晉元

國立交通大學運輸科技與管理學系碩士班

### 摘要

本研究目的為對於具時間窗限制車輛路線問題設計有效的演算法。本研究採用二階段基因演算法求解具時間窗限制車輛路線問題。第一階段以 Alvarenga (2007) 所提出的基因演算法為基礎並且在建構初始族群、交配以及突變上做些修改，以及啟發式解法來減少旅行成本。第二階段針對第一階段得到的路徑解做改善，並且解決違反時間問題。我們使用 Solomon(1983) 提出的 56 題國際標竿例題來進行測試與敏感度分析。測試結果顯示，本研究提出的演算法是適當的並且符合預期。

關鍵詞：具時間窗車輛路線問題、基因演算法



# **A two-phased genetic algorithm for the vehicle routing problem with time window**

Student : Lui-Yin Chen

Advisor : Jin-Yuan Wang

Department of Transportation Technology & Management  
National Chiao Tung University

## **ABSTRACT**

The purpose of this research is designing an efficient and effective algorithm for vehicle routing problem with time window. This research proposes a two-phase genetic algorithm approach for solving this problem. The first phase is based on the genetic algorithm introduced by Alvarenga (2007) with some modifications of generation of initial population, crossover and mutation techniques, and an heuristics to reduce travel cost. The second phase aims at improving routes obtained in the first phase and addresses the violated time windows. We use Solomon's 56 benchmark VRPTW instances for evaluation and sensitivity analysis. Our testing results show that the proposed algorithm is adequate and satisfactory.

Key Words: Vehicle Routing Problem with Time Window (VRPTW),  
Genetic Algorithm

## 誌謝

在撰寫論文的過程中歷經許多波折，也從中得到許多人的協助，這篇論文能夠順利完成，最要感謝的是我的恩師王晉元老師，謝謝老師這二年來的用心指導，讓我這個非本科系的學生能夠從無到有，不僅是知識上的學習，在生活上與待人處世上亦扮演重要角色。感謝中華大學卓裕仁老師以及國立高雄第一科技大學盧宗成老師於論文口試期間，撥冗閱讀我的論文並且給予我許多寶貴的問題與建議，讓我的論文能更加完整。

再來我要感謝實驗室的學長姊弟妹們，感謝思文學長在這段期間給予我許多論文以及程式上的建議與教導，讓我對於問題有不同的啟發；謝謝愛耍冷的帥猴學長，不論在計畫合作上或是生活上皆是獲益匪淺；有好東西總是與我分享的詩敏學姐，在苦悶的研究生生活中有妳真好；有個帥兒子的小松學長，給予我許多研究上的指導；程式達人英潔，謝謝你寶貴的程式建議；計畫的好伙伴，葉珮、佳儒、Daisy 與丹丹，很高興能與你們一起做計畫；還有韻佳學姐、彥佑學長、how 學長與阿誠學長，謝謝你們在研究上幫助。同實驗室的 yu，你真的很搞笑卻是個程式高手；sony lab 的邱小美、胖澤以及亞瑟，謝謝你們陪我一起吃飯、運動、搞笑以及分享生活點滴；謝謝倒數口試的成員，陳仲豪與 pony，我想只有你們理解倒數的心情；還有維真、維婷、法筑、沂茹、Daniel、大菽、總務、小花、wei、ling、同是力宏迷的之音、怡珊等同窗們，謝謝你們這二年的陪伴。我的超級好室友依雲，謝謝你一路上的支持與鼓勵，總在我最無助時適時地出現；親愛的 333 室友們，小黑以及漢娜，謝謝你們陪我度過最難熬的一年。

最後我要感謝我親愛的家人們，爸爸、媽媽以及奶奶，謝謝你們給予的一切，讓我能毫無顧慮地去追求自己的夢想。感謝你們一路上的支持，讓我能積極樂觀去迎接任何艱困的挑戰，你們永遠是我最佳的避風港。

謝謝交大所有的師長們，讓我在這研究所二年獲得豐富的求學經驗。感謝一直陪伴我成長，分享喜怒哀樂的所有朋友們，你們是最強大的後盾，很高興能與大家分享我的研究成果。

陳綠茵 誌謝

2008 年七月 於新竹交大

# 目 錄

目 錄 .....	iv
圖 目 錄 .....	v
表 目 錄 .....	vi
第一章 緒論 .....	1
1.1 研究動機 .....	1
1.2 研究目的 .....	3
1.3 研究範圍 .....	3
1.4 研究流程 .....	3
第二章 文獻回顧 .....	5
2.1 具時間窗車輛路線問題 .....	5
2.1.1 具時間窗車輛問題定義 .....	5
2.1.2 時間可行性之檢驗 .....	6
2.1.3 具時間窗車輛路線問題之文獻 .....	7
2.2 具時間窗車輛路線問解法 .....	8
2.2.1 初始解法 .....	8
2.2.2 路線改善法 .....	10
2.3 小結 .....	17
第三章 研究方法 .....	18
3.1 第一階段演算法設計 .....	18
3.2 第二階段演算法設計 .....	25
第四章 測試範例 .....	28
4.1 測試過程 .....	28
4.2 測試結果 .....	28
4.3 改善交錯的效果 .....	32
4.4 敏感度分析 .....	34
第五章 結論與建議 .....	40
5.1 結論 .....	40
5.2 建議 .....	40
參考文獻 .....	41
簡歷 .....	43

# 圖 目 錄

圖 1.1	實務上汽車路線貨運作業流程圖[31]	2
圖 1.2	汽車路線貨運業的路線形式[31]	2
圖 1.3	研究流程圖	4
圖 2.3	基因演算法流程圖	14
圖 2.4	十五個城市、四位銷售員之一條染色體範例(1)	17
圖 2.5	十五個城市、四位銷售員之二條染色體範例	17
圖 2.6	十五個城市、四位銷售員之一條染色體範例(2)	17
圖 3.1	第一階段基因演算法流程圖	19
圖 3.2	染色體示意圖	20
圖 3.3	路線交錯情況	21
圖 3.4	單一交錯情形	22
圖 3.5	單一交錯路線改善	22
圖 3.6	多重交錯情形	22
圖 3.7	多重交錯路線改善	23
圖 3.8	改善交錯流程圖	23
圖 3.9	選擇示意圖	24
圖 3.10	突變示意圖	25
圖 3.11	改善違反時間窗顧客點流程圖	26
圖 3.12	車輛途程成本改善流程圖	27
圖 4.1	突變率變動趨勢圖	35
圖 4.2	懲罰乘數變動趨勢圖	36
圖 4.3	初始族群變動趨勢圖	37
圖 4.4	初始族群與 CPU 時間趨勢圖	38
圖 4.5	顧客點間節線距離超過總旅行成本的百分比敏感度趨勢圖	39

# 表 目 錄

表 4.1	範例測試參數設定表 .....	28
表 4.2	Solomon 56 題測試題庫文獻已知最佳解.....	28
表 4.3	本研究與文獻最佳解之比較(R1 型).....	30
表 4.4	本研究與最佳解之比較(RC1 型).....	31
表 4.5	本研究與文獻最佳解之比較(R1 型).....	31
表 4.6	本研究與文獻最佳解之比較(RC1 型).....	31
表 4.7	改善交錯前後測試比較結果(群聚分佈例題) .....	32
表 4.8	改善交錯前後測試比較結果(隨機分佈例題) .....	33
表 4.9	改善交錯前後測試比較結果(隨機與群聚分佈例題) .....	33
表 4.10	敏感度分析參數設定表 .....	34
表 4.11	突變率變動敏感度分析表 .....	35
表 4.12	懲罰乘數變動敏感度分析表 .....	36
表 4.13	初始族群變動敏感度分析表 .....	37
表 4.14	顧客點間節線距離超過總旅行成本的百分比敏感度分析表 .....	38





# 第一章 緒論

## 1.1 研究動機

目前台灣物流業快速成長，物流業者的內部工作除了理貨、揀貨、訂單作業及貨物倉儲管理等之外，最重要的是配送作業的服務品質，滿足顧客的需求。不適當的路線規劃會使物流業者產生較高的營運成本且降低其競爭力。因此如何在最小運送距離等成本之下，規劃出有效率的運送路徑是物流業者所要面臨的重要課題。車輛路線問題(Vehicle Routing Problem, VRP)乃是由同一車種、固定的容量限制所組成的車隊，從單一場站出發，服務完一群需求已知的顧客後再回到中心場站，目標是追求車輛路線的總運輸成本最小化或是車輛數量最少。但目前國內物流中心的作業方式還在發展中的階段，在路線規劃方面，大多以人工的作業方式運作，其存在著許多缺點。依賴較資深的營業所人員或是司機的主觀經驗作路線規劃，並非最好的方式，因為他們所做的規劃可能和最佳化的路線規劃有些差距，若有人事上的變動，將會影響其公司營運效率和品質等，易造成其路徑規劃上的無效率以及資源的浪費，因此應尋求更好的方式來規劃路線。

以汽車路線貨運業為例，係指在核定路線內，以載貨汽車運送貨物為營業者。其承載之貨物大多為零擔運輸，所以一般又稱為「零擔貨運業」。又可分為內部營運網路與外部營運網路。內部營運網路係指各營業所至轉運中心，以及轉運中心之間，利用大型車輛進行長途的運輸作業；外部營運網路則是營業所至客戶之間，利用小型車輛進行區域的集配作業。營業所視其營業規模設置於全省中樞地區，服務範圍涵蓋全省及離島地區，主要配送模式為企業至企業(B2B)和企業至個人(B2C)。

每日固定時段從全國各地的營業所使用運務車載貨以及使用集配車輛收集一般顧客的貨物，場站人員將貨物分區，依區域將貨物堆放於車輛中。每位司機都有各自所屬的責任區域，司機依據自己經驗循事前規劃好的路徑進行運送服務。在配送的途程中，一般客戶會聯絡營業所或是直接聯絡司機告知到府收貨的需求，營業所會先搜尋顧客的區域，再以電話或車機訊息通知所屬區域的司機，司機再依據經驗來重新安排路線，但無法確定到達顧客處的明確時間。此種分區的好處可以節省車輛指派的難度，缺點是常造成司機工作量與服務時間的不平均。客戶會要求業者在特定時間內到府收貨[31]。

車輛種類不同，主要分為運務車和集配車。依到著地區之不同，長途以運務車從事定期、定時、固定班次的轉運作業；短程則使用集配車到顧客處收送貨。本研究僅討論單一車種問題。圖 1.1 為實務上汽車路線貨運作業流程圖，為 A 區營業所和 B 區營業所的運送作業，A 區營業所將屬於 B 區的貨物裝在運務車中，由司機運送到 B 區營業所，由 B 區的集配車司機理貨，並將所屬區域的貨物裝至集配車中，再將貨物送到顧客處。

圖 1.2 為汽車路線貨運業的兩種運送路線形式，其中，運務車係往返於各營業所之間的大量運送，其路線為固定，且會依據顧客要求的時間內到府收貨，此屬於本研究之具時間窗限制之車輛路線問題；集配車則服務各地營業所之客戶，其路線為非固定。

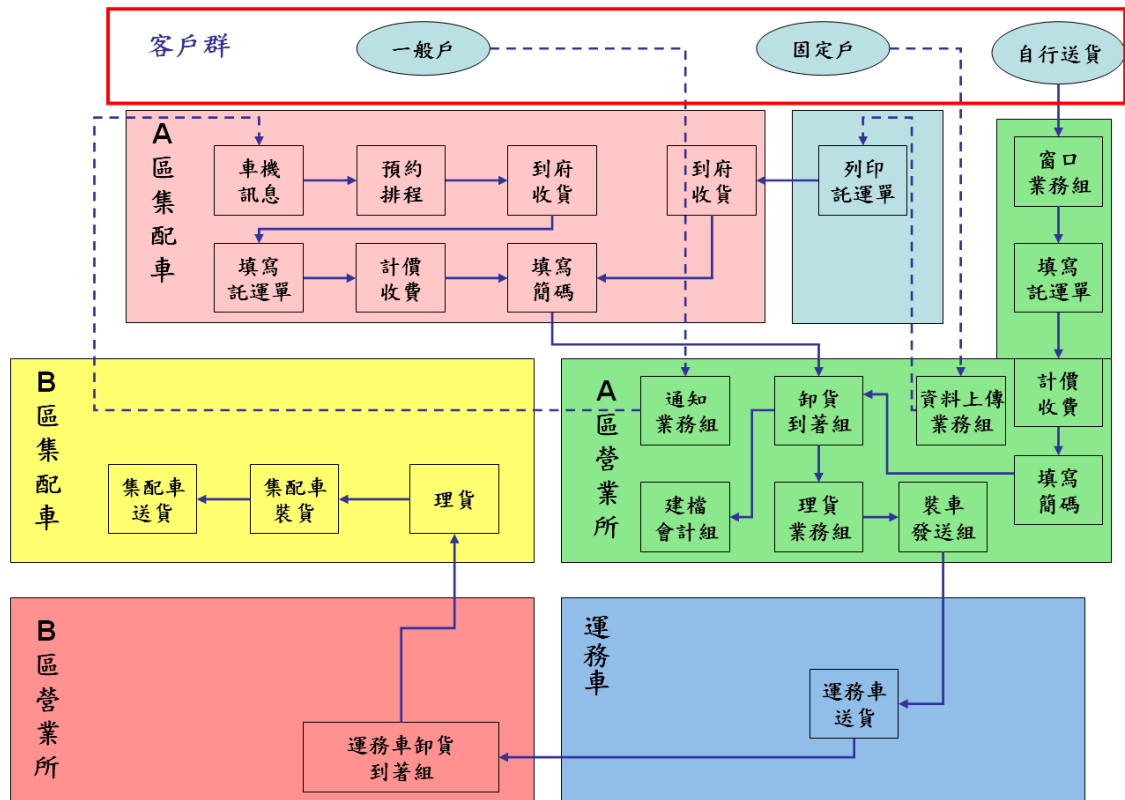


圖 1.1 實務上汽車路線貨運作業流程圖[31]

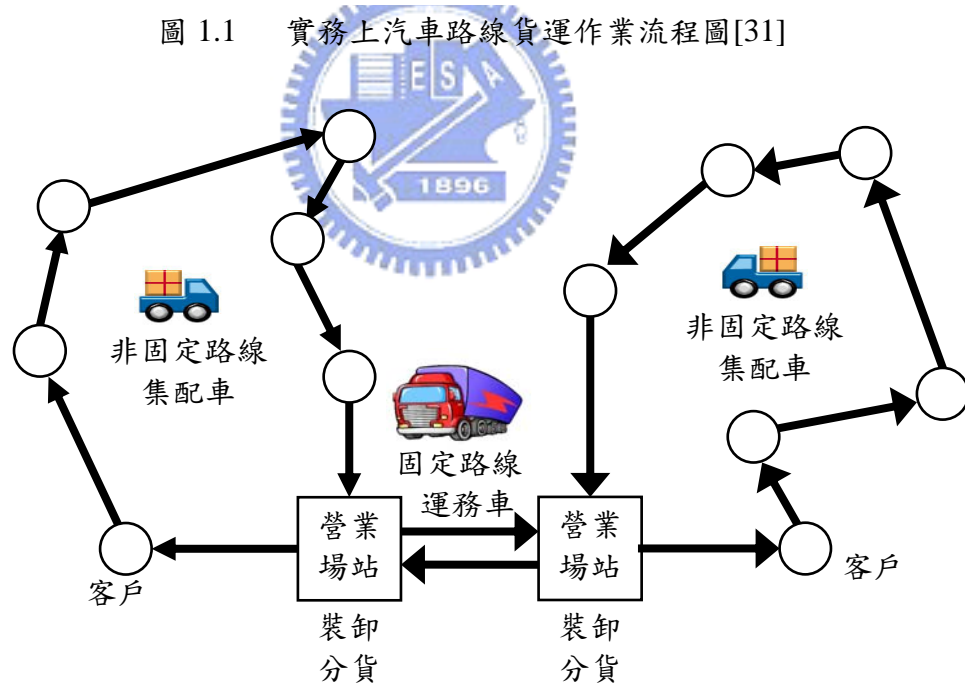


圖 1.2 汽車路線貨運業的路線形式[31]

現今物流業成長快速，若能快速回應顧客的需求，將貨物正確送到顧客手中，必能有效提升競爭力。因此，物流業者必須考慮顧客的特定需求，主要是以貨物送達顧客處的時段，為使車輛路線問題更具實用性，應加入顧客的時間窗限制。此為具時窗限制的車輛路線問題(Vehicle Routing Problem with Time Windows, VRPTW)，為解決在給定服

務時間窗限制下，使用車隊去服務顧客集合，目的是以最小化車輛數使得服務成本和總旅行成本最少。每一輛車服務一條路徑且有容量限制，每一條路徑顧客的總需求量不可以超過車輛容量限制，每一位顧客有他們自己的需求，服務時間和時間窗上下限。

基於上述之因素，若能提供求解問題的模式，且能快速提供路線規劃決策方法，是很重要的。本研究將針對 VRPTW 問題建構數學模式，設計一套演算法求解，產生有效的路徑集合，期能更節省物流成本以提升物流業的整體績效。

## 1.2 研究目的

本研究為對於物流業具時間窗限制之車輛路線問題(Vehicle Routing Problem with Time Windows, VRPTW)，由於 VRPTW 是屬於 NP-hard 問題，其求解時間會隨著問題的大小(顧客數)遞增，受限於問題的大小，此類問題不易求得精確解，因此很多研究上使用啟發式解法來解決此問題，可以兼顧求解的速度和品質。本研究針對 VRPTW，發展一套演算法進行求解，期望能達到以下目的：

1. 在滿足所有顧客需求、時間窗限制等因素下，應用二階段啟發式演算法求解，能兼顧求解速度和品質，期能產生一組最佳品質的車輛路徑解。
2. 提供較佳的車輛路徑，避免因司機依據經驗規劃路線造成的資源浪費，有效提升業者本身的競爭力。

## 1.3 研究範圍

本研究的研究範圍如下所列：

1. 司機在出發前已知確定顧客的需求點與需求量。
2. 在時間窗限制下，每位已知的顧客希望在某段時間內被服務，車輛早到需要等待，晚到給予一懲罰值，屬於軟時間窗限制。
3. 單一場站，車輛以場站為起點與終點。
4. 單一車種，各車的噸數相同。
5. 節點與節線空間關係為歐氏網路，節線成本滿足三角不等式。
6. 模式目標為最小化總車輛總旅行成本。
7. 各顧客點的需求量必須全部由某輛車一次服務完，不可分割成多次或由多輛車來服務，每位顧客的需求量須小於車輛的容量限制。
8. 每條路徑的總服務時間不得大於駕駛者的工作時間，且所有服務顧客的總需求量不得大於車輛的最大容量。

## 1.4 研究流程

圖 1.3 為本研究之研究流程圖。流程圖各階段內容的簡述如下：

### 1. 界定問題與研究範圍

透過研究動機與研究目的界定本研究的問題和範圍。

### 2. 蒐集及回顧文獻

蒐集並整理國內外相關具時間窗車輛路線問題之文獻，回顧解決此問題的相關啟發式解法與演算法之觀念及其架構等。從中挑出符合本研究問題的文獻所提出的演算法，並做優缺點之比較以及在演算法上做適當的改良。

### 3. 模式求解方法

本研究以基因演算法為主要架構，發展演算法，求解具時間窗限制之車輛路線問題。

### 4. 撰寫程式

本研究採用 C++ 進程式撰寫，在 Microsoft Visual Studio 2005 中編譯。

### 5. 測試範例求解與分析

撰寫程式求解測試範例，並對其結果做分析，評估演算法的品質績效。

### 6. 結論與建議

根據本研究所得成果，提出適當的結論與建議，期能提供學術研究與實務之參考。

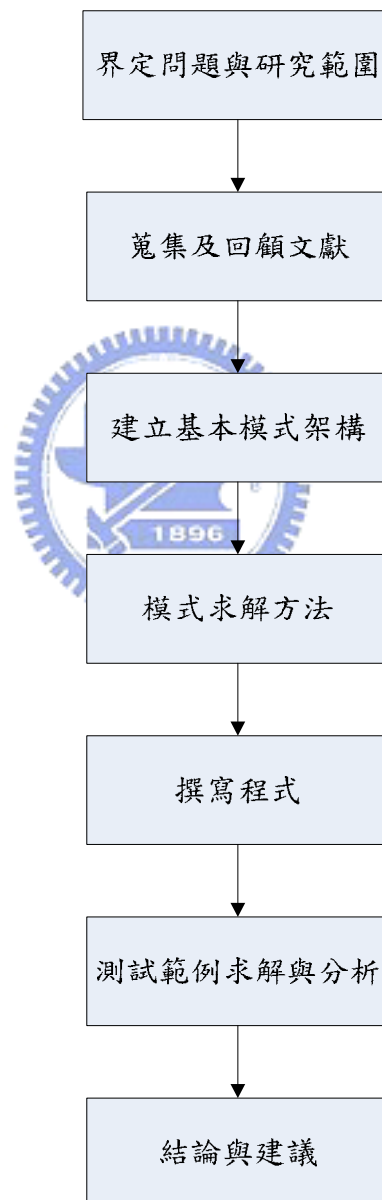


圖 1.3 研究流程圖

## 第二章 文獻回顧

本研究為探討具時窗限制車輛路線問題，此類問題屬於解題複雜度很高的 NP-Hard 問題，亦即當問題規模很大時，幾乎無法在有效的時間內求得精確最佳解，精確解大多以數學規劃方法來求解，像分枝定限法(Branch-and-Bound)、變數產生法(Column Generation)、拉氏鬆弛法(Lagrangian Relaxation)等解法。雖然這些解法能求到精確解，但無法在短時間內求得最佳解，缺乏效率，在問題的規模上也有執行的困難。由於多數的實務應用都屬於大規模的問題，因此得採用能夠求得較佳近似解的啟發式解法來進行求解，啟發式解法求解速度快，且不受問題規模的限制，能求得品質較高的解。

本章 2.1 節首先介紹具時間窗限制車輛路線問題之定義、時間可行性之檢驗與具時間窗限制車輛路線問題的文獻回顧；2.2 節針對具時間窗限制車輛問題解法做回顧；2.3 節為本章之小結。

### 2.1 具時間窗車輛路線問題

#### 2.1.1 具時間窗車輛問題定義

時窗限制分為兩種，分別為軟性時窗(Soft Time Window)和硬性時窗(Hard Time Window)。軟性時窗能夠容忍在非時窗範圍的時間到達，但針對太早或太晚服務顧客點必須依其違反的程度給予一懲罰成本。硬性時窗限定車輛必須於顧客所要求的時窗範圍內到達。以下是根據 Solomon(1983)之硬性時窗車輛路線問題之數學模式。其數學模式如下：

Minimize  $K$

Minimize  $\sum_i \sum_j \sum_k c_{ijk} x_{ijk}$

Subject to

$$\sum_v y_{ik} = \begin{cases} K & i = 0 \\ i & i \in \{1, \dots, n\} \end{cases} \quad (1)$$

$$\sum_i q_i y_{ik} \leq V_k \quad k \in \{1, \dots, k\} \quad (2)$$

$$\sum_i x_{ijk} = y_{jk}, \quad j \in \{0, \dots, n\} \quad k \in \{1, \dots, K\} \quad (3)$$

$$\sum_j x_{ijk} = y_{ik}, \quad i \in \{1, \dots, n\} \quad k \in \{1, \dots, K\} \quad (4)$$

$$b_j \geq b_i + s_i + t_{ij} - (1 - x_{ijk})T, \quad i, j \in \{1, \dots, n\} \quad k \in \{1, \dots, K\} \quad (5)$$

$$b_j \leq l_i \quad i \in \{1, \dots, n\} \quad (6)$$

$$b_j = \text{Max}\{e_i, b_i + s_i + t_{ij}\}, \quad i, j \in \{1, \dots, n\} \quad (7)$$

$$e_i \geq t_{0i}, \quad i \in \{1, \dots, n\} \quad (8)$$

$$e_0 \leq t_0^k \leq l_0 \quad k \in \{1, \dots, K\} \quad e_0 \geq 0 \quad (9)$$

變數定義：

$K$ ：車輛數。

$c_{ijk}$ ：顧客點  $i$  到顧客點  $j$  之旅行成本。其中  $i \neq j \quad i, j \in \{0, 1, \dots, N\}$ 。

$x_{ijk} = \begin{cases} 1, & \text{當車輛從顧客點 } i \text{ 至顧客點 } j \text{ 時。} \\ 0, & \text{其他。} \end{cases} \quad i \neq j \quad i, j \in \{0, 1, \dots, N\};$

$x_{ijk} = :$  代表車輛  $k, k \in \{1, \dots, K\}$ 。  $K$  為總車輛數。

$y_{ik}$ ：為 0 或 1；若車輛  $k$  服務顧客  $i$  時為值為 1，否則值為 0。

$q_i$ ：顧客  $i$  之需求量。

$V_k$ ：車輛  $k$  之容量限制。

$N$ ：總顧客數。

$t_{ij}$ ：從顧客點  $i$  到顧客點  $j$  之旅行時間，其中  $i \neq j \quad i, j \in \{0, 1, \dots, N\}$ 。

$t_{0i}$ ：場站(depot)至顧客點  $i$  的旅行時間。

$b_i$ ：開始服務顧客  $i$  的時間。

$e_i$ ：顧客點  $i$  之時窗開始時間。

$s_i$ ：顧客點  $i$  的服務時間。

$T$ ：為一很大正值常數，通常為最大可服務時間。

$l_i$ ：在顧客點  $i$  之時窗結束時間。

$e_0$ ：最大可服務之時窗下限。

$l_0$ ：最大可服務之時窗上限。



此問題之第一目標式總車輛數最小，第二目標式為極小化總旅行成本。式(1) 為確保每一條路徑皆由場站(depot)出發，且最後回到中心場站，每位顧客只有拜訪一次；式(2)為車容量限制；式(3)與式(4)為限制每位顧客點恰巧被一條路線進入與離開一次；式(5)為限制服務的時間順序並且防止子路徑的形成；式(6)與式(7)為硬性時窗限制，若車輛早到必需等待至顧客時間窗下限才能服務；式(8)與式(9)定義二元變數。

本研究的 VRPTW 問題為軟性時窗時窗限制，目標式只考慮極小化總旅行成本，因此不考慮式(6)與式(7)的硬性時窗限制，若早到或晚到必須給予一懲罰值。

### 2.1.2 時間可行性之檢驗

在 VRPTW 問題中，每個顧客點皆有其時間窗限制，因此在插入新顧客時必須先檢查顧客的時間可行性，判斷此顧客以及後續受到影響的顧客在時間窗上界是否有被違反。Solomon[19]所提出檢查時間可行性的法則做一說明：

假設目前將準備測試時間可行性的顧客  $u$  插入路線  $z$  中的  $p$  與  $p-1$  點之間 ( $0 \leq p \leq m, 0 = m = depot$ )，則顧客  $z_p$  的「後推時間項(Push Forward, PF)」如(10)式所示，定義為  $z$  路線在插入顧客  $u$  後， $p$  點新的開始服務時間減去插入顧客  $u$  前  $p$  點的原始開始服務時間：

符號定義：

$u$ ：目前插入且測試可行性的顧客點。

$m$ ：場站。

$z_p$ ：路線  $z$  中的顧客點  $p$  ( $0 \leq p \leq m, 0 = m = depot$ )

$b_{u,z_p}^{new}$ ：在  $z_p$  點前插入新顧客點  $u$  後的新開始服務時間。

$b_{z_r}$ ：開始服務  $z_p$  的時間。

$l_u$ ：顧客點  $u$  的時間窗上界。

$$PF_{z_p} = b_{u,z_p}^{new} - b_{z_p} \geq 0 \dots\dots\dots(10)$$

$r$  點是指路線  $z$  中  $p$  點後面的顧客點，它們的 PF 值如(11)所示：

$$PF_{z_{r+1}} = \max\{0, PF_{z_r} - w_{z_{r+1}}\} \quad , \quad p \leq r \leq m-1 \dots\dots\dots(11)$$

而時間可行性之充分且必要條件如(12)式所示，顧客  $u$  的開始服務時間必須不大於自己本身的時間窗上界，且顧客  $u$  在插入後，它後續的所有顧客點均不能違反各自的時間窗上界：

$$b_u \leq l_u \text{ and } b_{z_r} + PF_{z_r} \leq l_{z_r} \quad , \quad p \leq r \leq m \dots\dots\dots(12)$$

在時間可行性檢查的過程中，提前確定時間可行(即  $PF_r = 0$ )或提前確定時間不可行(即  $b_{u,z_p}^{new} \geq l_{z_r}$ )可提早停止檢查外，否則，需檢查至此路線迄點為止。

### 2.1.3 具時間窗車輛路線問題之文獻

Tan et al.[10]比較四種演算法，人工智慧(Artificial Intelligent, AI)演算法、模擬退火法、Tabu Search(TS)與基因演算法。在基因演算法部份使用 PFIH 產生起始解。測試結果顯示基因演算法在 VRPTW 上的績效最好。

Mak and Guo[11]考慮不確定顧客的 VRPTW 問題，使用軟性時間窗，違反時間窗限制時，早到達服務點仍須等候，若遲到則需負擔懲罰金。作者使用 Age based Genetic Algorithm(Age-GA)，將母代染色體給予歲數，母代可以做數次的演化交配，直到 age=4 時，母代會死亡。以服務顧客的順序做為編碼方式，交配方式為 two-point，當發生不可行解時，將第一個交配點之前和第二個交配點之後被重覆服務的點以未被服務的點來取代。突變的方式隨機選擇一點做為突變點，此點之後的顧客需求點做擾亂，產生新的個體。Age-GA 由於母代存活時間較長，因此有較多的機會產生好的子代，可以得到品質較好的解。

Alvarenga et al.[12]使用二階段分區的基因演算法來求解 VRPTW 問題。解法的架構動機為對於 VRPTW 的區域最佳解(local minimum)的涵蓋的路徑有顯著的可能，也能找到全域最佳解(global optimum)。第一階段求解出區域最佳解的路徑集合，每一獨立執行 GA 執行視為一世代的進化，直到世代數達到指定個數時則停止。第二階段將第一階段產生的路徑集合應用於 SPP Model 中求解，從中隨機挑選  $N$  條路徑，將這些路徑中所服務的顧客再執行一次 GA，使得顧客的子集合減少 30%的問題規模，即為產生一組新的 Reduced Problem。此能脫離區域最佳解，顧客重新組合做調整，期能得到更好品質的解，最後求得改善品質的全域最佳解路徑集合。然而，第二階段排除了先前路徑集合中，因演化過程導致顧客服務順序錯誤或是路線交換錯誤。使用二階段的方式所產生的路徑集合可以有效達到最小的旅行成本。

Potvin 和 Laporte [13]利用貪婪法插入顧客點來構建 VRPTW 初始路徑，再利用基因演算法求解，並與多種演算法做比較，發現在總旅行成本方面與禁制搜尋法不相上下，在 C 型路網上表現較好。

Koskosidis et al. [14]以一般指派法為基礎做延伸，求解軟時間窗車輛路線問題，首先是提出硬時窗限制的模型，接著將每個顧客點的配送時間窗放鬆到目標式為一軟時間窗問題，由於放鬆後的目標式較為複雜難解，故使用一近似替代模型取代之，並將問題拆解為一容量限制分群問題與含時窗限制之旅行銷售員問題進行演算法求解，透過隨機找取數據與標竿問題對演算法進行測試。

Chiang and Russell [15]結合模擬退火法與禁制搜尋法之禁制列表特性求解 VRPTW 問題，並以 Solomon 例題評估求解績效，結果優於原本使用單一模擬退火法之架構，也證明結合兩種演法特性所衍生之新型態演算法值得深入探討的。

Chiang and Russell [16]以  $\lambda$  exchange 為核心的反應式(Reactive)禁制搜尋法求解硬時間窗車輛路線問題(Vehicle Routing Problem With Hard Time Window, VRPTHW)，並發展其他三種禁制搜尋法架構做比較，以 Solomon 例題來衡量績效表現。

Jim and Hsu [17]提出一混合的 GA 演算法。先使用 DP 求解，若求解時間超過設定的門檻，則改採用 GA，並且比較四種複製與三種突變的方法。

## 2.2 具時間窗車輛路線問題解法

本節針對求解 VRPTW 的初始解法、路線改善法、巨集啟發式解法與基因演算法做回顧。

### 2.2.1 初始解法

文獻中常見的初始解法包括插入法(Insertion)、掃瞄法(Sweep Method)、節省法(Saving Method)、最近鄰點法(Nearest Neighbor Algorithm)等。以下針對這些解法做簡單的介紹。

#### 1. 插入法(Insertion Algorithm)

由 Rosenkrantz et al.所提出，Solomon(1983)將其應用於 VRPTW 問題上。首先，從一條簡單的路線開始，逐步考慮尚未包含在路徑內的節點，選擇插入一個節點到路徑內以產生可行解。插入節點有幾項準則：在路徑內兩個連續節點的最大角、增加最小距離、離路線最遠等。其最終目的在於將顧客點插入後，能夠使增加的路線成本愈小愈好，且須滿足車輛容量限制。

#### 2. 掃瞄法(Sweep Method)

由 Gillett and Miller(1971)[1]所提出，Solomon(1983)將其應用於 VRPTW 問題上。以座標來表示顧客點的位置。以下為求解步驟：

- (1) 以場站(depot)的位置作為原點 0，依序為每個配送點以 1,2,3,.....編號，並計算各配送點極座標之角度值。
- (2) 以 depot 做為起點，編號 1 號的點作為第一個配送點，依逆時針方向選取比第一個配送點的角度值大一個順位的點，作為第二個配送點，以此類推，直到滿足一輛貨車之最大載貨限制條件，即回到物流配送中心，完成第一條路線的建構。
- (3) 再選取物流配送中心做為起點，選取比上一條路線最後一個配送點之角度值大



- 一個順位的點作為第一個配送點，重複步驟(2)的程序。
- (4) 重複步驟(3)的程序，直到所有配送點皆納入規劃的路線中，即完成一組起始可行解。
  - (5) 將座標軸旋轉一個角度，使點 1 變為最後一點，點 2 變成第一個點，重複步驟(1)到(4)找到另一組可行解，直到所有點都當過求解的起始解。
  - (6) 重複步驟(1)到(5)，但為反方向搜尋。

### 3. 節省法(Saving Method)

Larke and Wright(1964)提出，應用於 VRP 問題，Solomon(1983)將其應用於 VRPTW 問題上。假設每個顧客點皆有一條路線由場站至顧客點直接服務，藉由逐步合併路線的方式來構建可行解；路線合併的準則為，在不違反車輛容量限制下，合併後所能產生之路線成本節省值，節省值愈大者將優先考慮合併。依據節省法運作之不同可分成循序節省法和平行節省法。以下為求解步驟：

- (1) 已知需求點之間距離  $d(i,j)$ ，各車裝載容量。計算各點間之節省值  $S(i,j)$ 。

$$S(i, j) = 2 \times d(0, i) + 2 \times d(0, j) - \{d(i, 0) + d(0, j) + d(i, j)\} \\ = d(i, 0) + d(0, j) - d(i, j)$$

- (2) 依節省值大小次序列表，從表中最大之節省值開始執行

#### (3a) 平行法

在不違背裝載及其他限制下，考慮該兩點之連結與路線合併。直到節省值用完為止。

#### (3b) 循序法

在節省值次序列表中尋找第一條可連結延伸現有路線的節省值(不違背營運條件)，若找不到可行之節省值，找最大值重新定義一條路線。重覆 a、b 直到節省值用完為止。

### 4. 最近鄰點法(Nearest Neighbor Algorithm)

由 Rosenkrantz, Sterns and Lewis(1977)提出。先任意選一個起始點，找距離起始點最近的節點將其兩者相連結，繼而再找與剛加入之節點距離最近的節點相連結，一直反覆進行上述動作，若不違反車輛容量限制，則進行連接直到所有節點皆加入為止，最後再將起使點與最後的節點相連形成一個封閉的網路，再依序進行改善。

### 5. 後推時間項插入法(Push Forward Insertion Heuristic, PFIH)

Solomon(1987)所提出的 VRPTW 初始解解法。此為計算插入一新顧客  $C_i$  到目前路徑中之有效方法。假設一路徑  $R_p = \{C_1, \dots, C_m\}$ ， $C_1$  為第一個顧客， $C_m$  為最後一位顧客。插入一可行的顧客到  $R_p$  中，必須先檢查其顧客在目前路徑中所有節線成本，並將顧客  $C_i$  插入到成本最低的節線中。顧客  $C_i$  欲插入到顧客  $C_0$  和  $C_1$  中，必須先檢查其後推時間項，後推時間項的改變可能影響  $C_i$  所有後繼點的到達時間。因此對於  $C_i$  插入的可能性，需要去檢查  $C_i$  後續顧客的後推時間。

$$C_i = \alpha d_{0i} + \beta b_i + \gamma((p_i / 360)d_{0i}) \dots \dots \dots (13)$$

以下為 PFIH 之步驟：

- (1) 開始時為空的路徑，從場站開始出發，令  $r=1$ 。
- (2) 如果所有的顧客都被服務，則到步驟 8。對所有尚未被服務的顧客  $j$ ：根據式(13)

計算顧客成本，並將成本做排序。

- (3) 從尚未被服務的顧客中，挑選成本最低的顧客為第一個顧客點  $j^*$ 。
- (4) 加入  $j^*$  到目前路徑  $r$  中，並且更新路徑的容量。
- (5) 對於所有尚未被服務的顧客  $j$ ：對於當前路徑中所有節線  $\{k, l\}$ ，計算在  $k$  與  $l$  間每一尚未服務之顧客的成本。
- (6) 挑選一在節線  $\{k^*, l^*\}$  中未被服務之顧客  $j^*$ 。如果插入顧客  $j^*$  在  $k^*$  和  $l^*$  間，符合時間可行性與車容量限制，則插入顧客  $j^*$  在  $k^*$  和  $l^*$  間，更新當前路徑  $r$  之容量，並且到步驟 5。若不符合則到步驟 7。
- (7) 從場站開始一新路徑，令  $r=r+1$ 。到步驟 2。
- (8) 所有顧客都被服務完，結束 PFIH。

## 6. 路徑掃描法(Path-scanning algorithm)

Golden et al.(1983)提出。從場站(depot)出發，考慮容量限制，考慮未服務的節點，已服務過的節線不可以再選。若在某一節點，所有連續 2 節線均已包含在其他路徑中，則找最近的節線且其具有尚未被服務過的節線。當容量已達上限或已沒有未服務的線可選時，則以最短路徑回到場站(選最短路徑時，不須受到已服務節線的限制)。

## 7. 一般化指派法( Generalized Assignment Heuristic)

Fisher et al.(1981)提出。假設有  $n$  車輛數，車輛有容量限制。找最遠的  $n$  個點當種子(seeds)。透過(14)式算出某一點加入各路線時增加的距離，選出增加的距離最小的那個點，繼續尋找下個點直到路網中各點皆被搜尋過則結束。Bolkan (1986)提出一般化分配法可以求解小於 25 個顧客的小問題，但對於規模較大的實際應用上不易求解。

$$d_{ik} = \min[c_{li} + c_{iik} + c_{ikl}, c_{lik} + c_{iki} + c_{il}] - [c_{iik} + c_{ikl}] \dots \dots \dots (14)$$

( $d_{ik}$  : 加入節點  $i$  增加的成本)

### 2.2.2 路線改善法

一般傳統 VRP 問題路線改善法進行時需符合車容量限制，而 VRPTW 問題之路線改善法是以 VRP 問題之路線改善法作為基礎，在進行節點或節線交換時需另外考慮時間窗限制，符合其限制條件方為可行解。

因此，VRPTW 問題的路線改善法可採用傳統的 VRP 路線改善法，並加入檢查時間可行性。以下介紹幾種常見的路線改善法。

#### 1. K-opt

Lin(1965)提出。k 表示每次交換的節線數目。藉由交換同一路線中 k 條不相鄰節線之連接方式，檢查交換後的結果是否能夠維持可行解並且獲得改善，進而決定接受某種交換形式以獲得較佳的路線。一般而言，k 值不宜太大，通常為 2 或 3。

#### 2. Or-opt

Or(1978)提出。步驟一：任選連續三點( $S=3$ )， $i$ 、 $i+1$ 、 $i+2$ 。步驟二：其餘節點中依序選出連續兩點  $j$ 、 $j+1$ 。步驟三：將  $i$ 、 $i+1$ 、 $i+2$  插入  $j$ 、 $j+1$ ，與原來路網比較，如果成本較節省則插入，否則保留。步驟四、重複步驟一到三。步驟五：令  $S=2$ 、 $S=1$ ，重覆步驟一到四，直到在所有節省值中找出最佳解為止。

### 3. 2-opt\*

Potvin et al.[2]提出。是從 2-Opt 變化而來的，用來改善 MTSP，也就是多路線的 TSP，主要特色在於兩條 TSP 的節線作交換，另外對於有 TW 限制的問題能更有效的改善。

### 4. 插入解繫法(Generalized Insertion/Unstringing and Stringing,GENIUS)[3]

Gendreau et al.[1992]提出。可視為是節點插入法與節線交換法的結合。其中，GENI 可作為起始解構建的方法，US 則可作為路線改善的方法。GENI 與 US 的基本型態是相同的，共有 Type I 與 Type II 兩種；Type I 如圖 2.1 所示，將節點  $V$  插入  $V_i$  和  $V_j$  中間，刪掉三條節線  $(V_i, V_{i+1})$ 、 $(V_j, V_{j+1})$  與  $(V_k, V_{k+1})$ ，加入四條節線  $(V_i, V)$ 、 $(V, V_j)$ 、 $(V_{i+1}, V_k)$  與  $(V_{j+1}, V_{k+1})$ 。Type II 如圖 2.2 所示，將  $V$  插入  $V_i$  和  $V_j$  中間，刪掉四條節線  $(V_j, V_{j+1})$ 、 $(V_l, V_{l+1})$ 、 $(V_i, V_{i+1})$  與  $(V_{k-1}, V_k)$ ，加入五條節線  $(V, V_j)$ 、 $(V_l, V_{j+1})$ 、 $(V, V_i)$ 、 $(V_{l+1}, V_{k-1})$  與  $(V_k, V_{i+1})$ 。節點的插入位置不一定限於相鄰的兩點之間，但是插入後該兩點將與插入點相鄰。

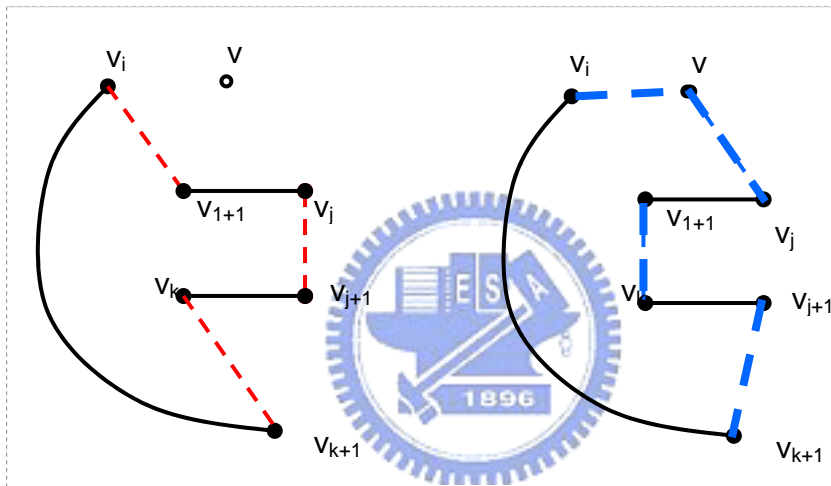


圖 2.1 GENI Type I

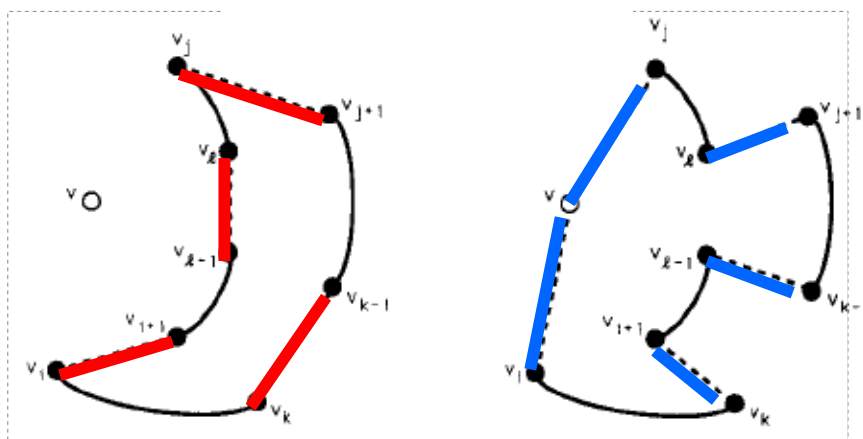


圖 2.2 GENI Type II

### 5. 路線間節點交換法

對於任兩條路線相互交換其部分節點，在不違反兩條路線各自的車輛容量限制下，檢查交換後的結果是否可行並獲得改善。節點交換的型式很多，例如：1 點對 0 點(1-0)、1 點對 1 點(1-1)、1 點對 2 點(1-2)等。

### 2.2.3 巨集啟發式解法

傳統的鄰域搜尋或交換型啟發式方法只在目標值有改善時才進行交換，加上受起始解組合型態的影響，因此在求解過程中易陷入局部最佳解 (Local Optimum) 而無法繼續尋找較佳的解。一般通用型的搜尋法即是克服傳統交換型啟發式方法之缺點，此類方法發展至今已逐漸成為求解組合最佳化問題之主流方法，並被命名為「巨集啟發式方法 (Meta-Heuristics)」。巨集啟發式方法係以傳統的啟發式方法為基礎，並根據一些高階的搜尋策略 (Meta-Strategies) 啟發式方法，以跳脫局部最佳解的束縛並增廣其搜尋空間範圍。以下介紹幾種較為著名的巨集啟發式方法，包括：禁忌搜尋法 (Tabu Search, TS)、基因演算法 (Genetic Algorithm, GA)、螞蟻演算法 (Ant Colony Optimization, ACO)、門檻型演算法 (Threshold Algorithms, TAs) 等演算法。

- (1) 禁制搜尋法：由Glover及Hansen (1986)所提出，是鄰域搜尋的一種變型方法，用來解決組合最佳化的問題。TS在目前解的鄰域中進行搜尋，透過彈性記憶之運用，將已經搜尋過的解記錄在禁忌清單 (Tabu List) 中，以避免重覆搜尋情況，直到整個鄰域都搜尋完畢後，再選擇一個最佳的目標進行移動，以逐漸逼近最佳解。TS法的關鍵即在於記憶機制的設計，像是記憶機制中應該記錄何種資料或是記憶體長度的設定等。而TS的記憶機制又可分為：短期記憶及長期記憶兩種型態，短期重點在於加速達到區域最佳化，在長期則使用深度搜尋與多樣性以求得更佳的解。
- (2) 基因演算法：最早是由Holland (1975)所發展而來，其原始構想是利用生物染色體基因架構的觀念，來表示一個複雜系統的結構。舉例說明，在一個0-1整數規劃問題中，每個決策變數皆可視為是一個基因 (Genes)，而每個可行解則可視為是染色體 (Chromosome)；若任意產生一些可行解做為「父代 (Parents)」，再經由基因間的複製、交配 (Crossover)、突變 (Mutation) 產生新的可行解「子代 (Offspring)」，接著保留最佳的幾個解再產生下一組子代。重複執行此過程多次後，將可找到最佳解或甚佳的近似解。在2.2.4節中針對基因演算法有詳細的介紹。
- (3) 螞蟻演算法：由Dorigo et al. (1999)提出。其概念來自於模擬螞蟻群尋找食物的行為，然後將其轉換成一套有系統的人工智慧搜尋方法。研究學者觀察到螞蟻在尋找食物的過程中，會沿途留下一種稱為費洛蒙 (Pheromone) 激素的物質，來標示其蹤跡；透過遺留在路途中的費洛蒙濃度，可以讓其他螞蟻判斷如何依循前進以找到食物並得以返回巢穴。費洛蒙濃度越高，該路線被螞蟻選擇的機率也越大。許多學者也藉此建立最佳化問題的解決方法。  
自從ACO的基本概念被提出後，衍生出多種不同的螞蟻演算法架構，如：螞蟻系統 (Ant System, AS)、螞蟻群落系統 (Ant Colony System, ACS)、評等螞蟻系統 (Rank-based Ant System, ASrank)。目前ACO已成功地被應用在許多複雜的問題上，例如：旅行推銷員問題 (TSP)、排程 (Scheduling)、電信網路 (Telecommunication Network) 等方面。
- (4) 門檻型演算法：基本觀念乃是在區域搜尋陷入局部最佳解時，採取較放鬆的接受法則，此法則通常為一門檻值 (門檻值大，容易接受較差的解；門檻愈小，愈不容易接受較差的解) 能接受較差的暫劣解，以脫離局部最佳解的束縛而能繼續搜尋。屬於這類的的方法有：門檻接受法 (Threshold Accepting, TA)、大洪水法 (Great Deluge)

Algorithm, GDA) 與記錄更新法 (Record-to-Record Travel, RRT), 等。SA、TA、GDA與RRT等方法的執行架構與傳統鄰域搜尋法架構類似, 主要差異之處在於使用不同的接受法則。其中模擬退火法基本觀念最早是由Metropolis et al. (1953)所提出。Li and Lim (2001)使用SA法對時窗限制之同時收送貨車輛路線問題 (PDPTW) 進行求解, 其結果在車輛數方面有不錯的表現。Dueck and Scheuer [6]提出門檻接受法 (TA), 並以一個442節點的TSP例題來驗證其可行性; TA已被應用於許多最佳化問題, 如: 工作排班問題 (Job Scheduling Problem)、旅行推銷員問題 (TSP) 等, 皆有不錯的績效。隨後, Dueck [7]又根據TA的觀念衍生出GDA與RRT兩種新方法。Tarantilis et al. 提出一個改良式的門檻接受法, 稱之為可回溯式門檻接受法 (Backtracking Adaptive Threshold Accepting, BATA)。

#### 2.2.4 基因演算法回顧

本研究採用基因演算法, 必須先對於GA運作過程深入了解。隨著問題模式的不同, 其染色體的編碼方式也不同。透過演算法中的適合度函數來決定適合保留下來的群體。接著, 經由三個演算過程, 選擇、交配、突變進行演化。圖 2.3 為基因演算法流程圖。



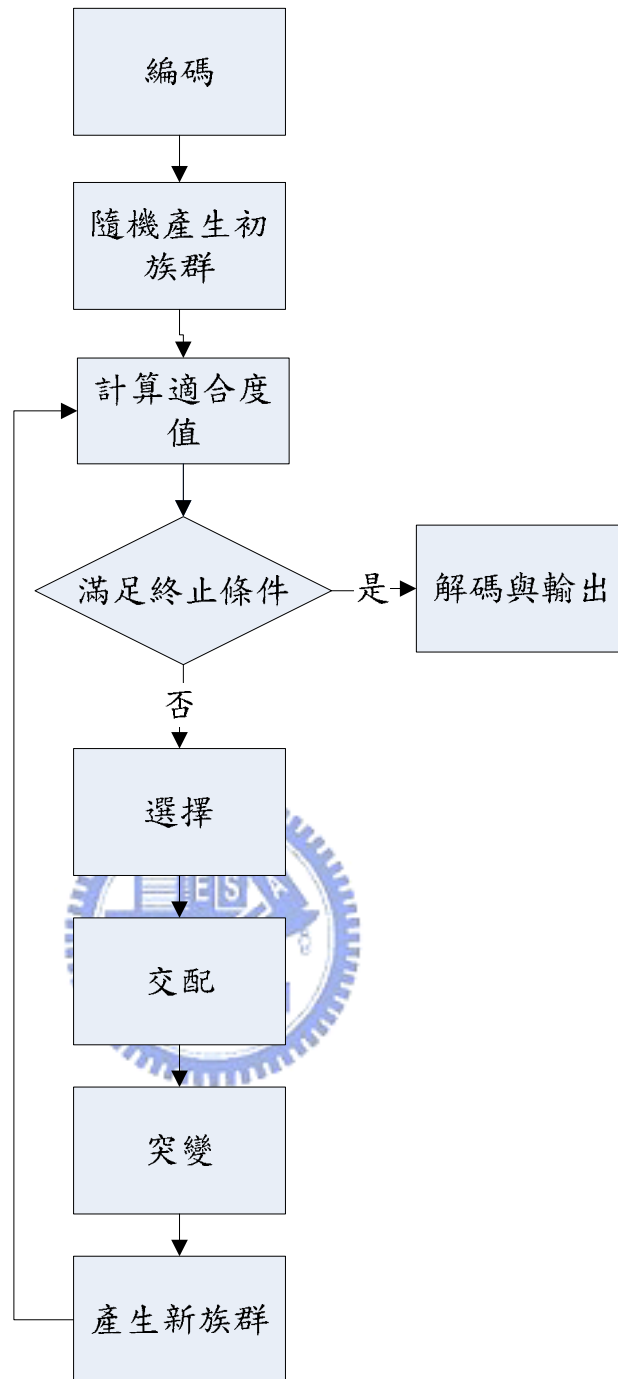


圖 2.3 基因演算法流程圖

基因演算法包含下列幾個步驟：

#### 一、編碼(Encoding)

係將問題的變數以編碼的方式表示，經過編碼程序後的解稱作染色體。一般編碼方式分為二進位編碼、實數編碼、整數編碼及資料結構等多種型態。一般最常用二進位來編碼，其二進位與十進位的對應，如果將二進位字串 00110 解碼，則可以對應十進位的值 5。其可視為一條染色體。以二進位資料型態來編碼需考量解的空間大小。

#### 二、產生初始族群(Initial Population)

初始族群即為第一代演化之親代，即為初始解所組成的集合。一般基因演算法中初始族群的產生方式，最常見為隨機產生。然而族群的規模大小會影響搜尋效率，族群大有較大的可能找到最佳解，但求解速度較慢；若族群小容易陷入區域最佳解，但求解速度較快。因此一個好的始族群能有助於搜尋的效率。

### 三、適合度函數(Fitness Function)

適合度數函是用以評估族群中染色體的優劣指標。個體的適合度函數值越高，代表個體的適應力越高；反之則越低。藉由適合度函數數值的高低來決定染色體保留到下一代的存活率。

### 四、選擇(Selection)

選擇即是在演算法過程中選擇族群中較優良的染色體，將之保存下來以演化出更優良的後代。依據每一個染色體適合度函數值的高低做選擇，擁有較高適合度數值，則有較大機率被選擇出來進行複製。一般最常用的選擇方法是 Goldberg 於 1989 年提出的「輪盤式選擇(Roulette Wheel Selection)」，其程序如下：

1. 將目前族群中每個個體的適合度函數計算出來，且將所有適合度數值加總。
2. 求每個個體占總值的比例。
3. 劃分區塊，區塊大小即為個體適合度數值所占的份量。
4. 隨機產生一界於 0 到 1 的數字，此數落於何區的範圍內即被選擇。適合度值越大所占的區塊也越大，因此被選擇的機率相對較高。
5. 重複選取直到交配槽中的個數等於族群數目為止。

另一種選擇方式是競爭法(Tournament Selection)，其程序如下：

1. 從母代族群中設定一個合理的競爭比較個數，假設為 N。
2. 從母代中隨機選取 N 個個體做比較，選擇適合度函數值最高的個體到交配槽中。
3. 重複選取直到交配槽的個體數目等於族群中個體數目為止。

### 五、交配(Crossover)

將經過隨機選擇後交配槽的個體以兩兩成對的方式作交配，彼此交換部份的基因，產生兩個新的個體。基因交配主要目的是希望能產生更優秀的個體。一般較為常用的交配方式有三種：單點交配(One-Point Crossover)、兩點交配(Two-Point Crossover)以及均勻交配(Uniform Crossover)。

#### (一) 單點交配(One-Point Crossover)：

進行單點交配時，會隨機亂數決定一個切斷點，將欲進行交配的染色體從此切斷點分成兩部分，再將切開的部份做基因互換，重新組合成一對新的染色體。

#### (二) 兩點交配(Two-Point Crossover)：

與單點交配的程序相似，不同之處在於進行兩點交配時，會隨機亂數決定兩個切斷點，將欲進行交配的染色體切成三部分，進行基因互換，重新組合成一對新的染色體。

#### (三) 均勻交配(Uniform Crossover)：

Syswerda(1989)所提出。首先會隨機產生一個長度相同二進位字串的子染色體，而子染色體中的位元決定隨機產生的指標，當位元值為 1 時，則進行親代基因位置上的互換，反之若位元值為 0 時，則不進行互換。

## 六、突變(Mutation)

染色體在其選擇和交配的演算過程中，可能會發生過早收斂為區域最佳值，而不能夠演化出整體最佳解的情況。突變可使染色體間相異性提高，來加族群的多樣性，而不侷限在少數個體上。以生物的觀點來看，突變的機率很低。但在應用時，會提高突變率以產生適應性較強的物種。

## 七、終止條件

在正常情況下，演化過程會持續，為了能結束演算法，會訂定一個終止的法則，一般常見的終止條件有以下幾種：

- (1) 演化世代數達到所指定的代數時
- (2) 適合度函數值停滯不前已無法有效改進時
- (3) 同一世代中染色體的同質性達到預設值時
- (4) 達到設定的運算時間

Backer and Ayecheew[8]提出混合基因演算法和鄰域搜尋法求解單一場站和旅行成本限制之 VRP 問題。使用混合隨機和建構法產生初始族群，編碼方式為將顧客按照極座標順序構成染色體集合。交配方式為 two-point，當子代被產生時，立刻被放到母體中，並取代母代中最差者。作者將母代分成四類：

$$\begin{aligned} S_1 &= \{p \in P : f(p) \geq f(c), u(p) \geq u(c)\}, \\ S_2 &= \{p \in P : f(p) < f(c), u(p) \geq u(c)\}, \\ S_3 &= \{p \in P : f(p) \geq f(c), u(p) < u(c)\}, \\ S_4 &= \{p \in P : f(p) < f(c), u(p) < u(c)\}. \end{aligned}$$

其中  $f(p)$  為母代的適合度值， $f(c)$  為子代的適合度值； $u(p)$  為母代的非適合度值， $u(c)$  為子代的非適合度值。按照適合度值與非適合度值的程度做取代選擇，若  $S_1$ 、 $S_2$ 、 $S_3$  均為空集合，則不取代。突變機制為隨機選取二個基因並且交換他們的值，也就是交換其服務的車輛，排除兩顧客在同一輛車的狀況。

不可行解的處理方式分為兩個階段，第一階段有三個程序，第一，對於違反容量限制的車輛，選擇一顧客並將其重新指派到下一輛車的最佳位置，如果非適合度值的總合減少，則維持重新指派。第二、當所有違反容量限制的車輛均考慮時，重覆此第一程序，不同的是考慮前一台車輛。第三、對所有的路徑使用 2-opt 做最適化。第二階段對於敏 10,000 個世代進行鄰域搜尋法，有三個程序。第一、將所有路徑加在一起，執行 2-opt。第二、 $\lambda$ -interchange( $\lambda=1$ )，將某一顧客從一路徑中移除，並插入到另一條路徑或是交換兩個顧客的服務車輛。第三、程序一、二完成後，對每一路徑進行 2-opt。作者的方法，和傳統基因演算法、Tabu search 及模擬退火法的計算時間、品質做比較，發現求解速度和品質超過傳統基因演算法，並與後兩者不相上下。

Carter and Ragsdale[9]使用基因演算法解決多送貨員的問題。問題假設有  $n$  個服務點， $m$  位送貨員。初始族群以貪婪法(greedy heuristic)產生。三種染色體編碼方式：(1) 一條染色體：使用一條長度為  $n+m-1$  的染色體，此為隨機產生，用負號整數來表示送貨員的改變，如圖 2.4 所示。(2) 二條染色體：將城市和送貨員分為兩條染色體，城市的順序為隨機產生，另一條染色體為均勻分配  $m$  位送貨員，為每個城市所對應的銷售員，而此種方式解答空間最大，如圖 2.5 所示。(3) 二段式(two-part)染色體：將染色體分為兩部份，需求點和每位送貨員服務需求點的數目，第一部份為隨機產生，此種方式解答空間



最小，如圖 2.6 所示。測試結果顯示，第三種染色體的績效較其他兩種好，能夠獲得較小的搜尋空間，在很多方面，與之前的方法相比，也能產生更好的解。

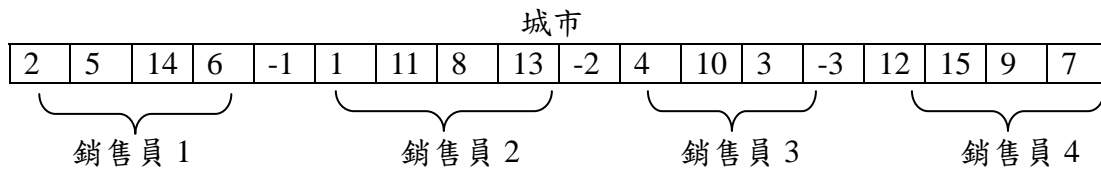


圖 2.4 十五個城市、四位銷售員之一條染色體範例(1)



圖 2.5 十五個城市、四位銷售員之二條染色體範例



圖 2.6 十五個城市、四位銷售員之一條染色體範例(2)

## 2.3 小結

VRPTW 的複雜度很高，屬於 NP-Hard 問題，當問題規模很大時，在短時間內很難求得真確最佳解。但實務上大多是屬於大規模的問題，顧客數越多問題規模較大，能在可接受的運算時間內求得一組可行路徑是重要的課題，因此大多數文獻採用啟發式解法，以在有效時間內求出近似最佳解。目前有許多 VRPTW 的相關文獻使用基因演算法，且能求出品質相當不錯的解以及績效。但近年來，有關於 VRP 或 VRPTW 等問題若使用一般傳統的基因演算法，其結果會有路線交錯的問題，因此許多研究的解法方法並非只是單純的基因演算法，通常採用二階段的解決方法，第一階段以啟發式解法求得初始解；第二階段再針對初始解予以改良品質，通常採取鄰域搜尋法(Neighborhood Search)或局部搜尋法(Local Search)等方法來改善目標函數。

Alvarenga et al.[12]利用二階段分區求解 VRPTW 問題。其第一階段使用基因演算法求得近似最佳解的路徑集合；第二階段使用第一階段所求得的路徑集合並從中隨機選出 N 條路徑構成子問題，再進行基因演算法來求解。本研究將對於使用二階段方法之來解決 VRPTW 之課題作進一步的探討，第一階段採用基因演算法；第二階段則提出一演算法來改善第一階段的路徑集合解，期能快速且有效率求得品質較佳的解。

## 第三章 研究方法

本研究求解具時窗限制車輛路徑問題( Vehicle Routing Problem with Time Window , VRPTW)，在車輛出發前，規劃已知需求點的車輛行駛路徑。本研究參考 Alvarenga[12]提出的二階段分區求解 VRPTW 問題。第一階段使用基因演算法；第二階段的演算法為解決第一階段路徑集合解所產生的問題。本章首先介紹 VRPTW 問題的數學模式，後續再針對[12]和本研究的方法做深入探討。

在 2.2.4 小節中已針對基因演算法的流程作詳細的介紹。本研究的演算法設計分為二階段求解 VRPTW 問題。3.1 節為本研究演算法的第一階段基因演算法，此階段為參考 Alvarenga [12]的基因演算法流程並且在建構初始族群、交配以及突變上做些修改；3.2 節為本研究演算法的第二階段，此階段目的為改善第一階段的解。

### 3.1 第一階段演算法設計

圖 3.1 為第一階段演算法流程，以下將針對每項流程作詳細說明。



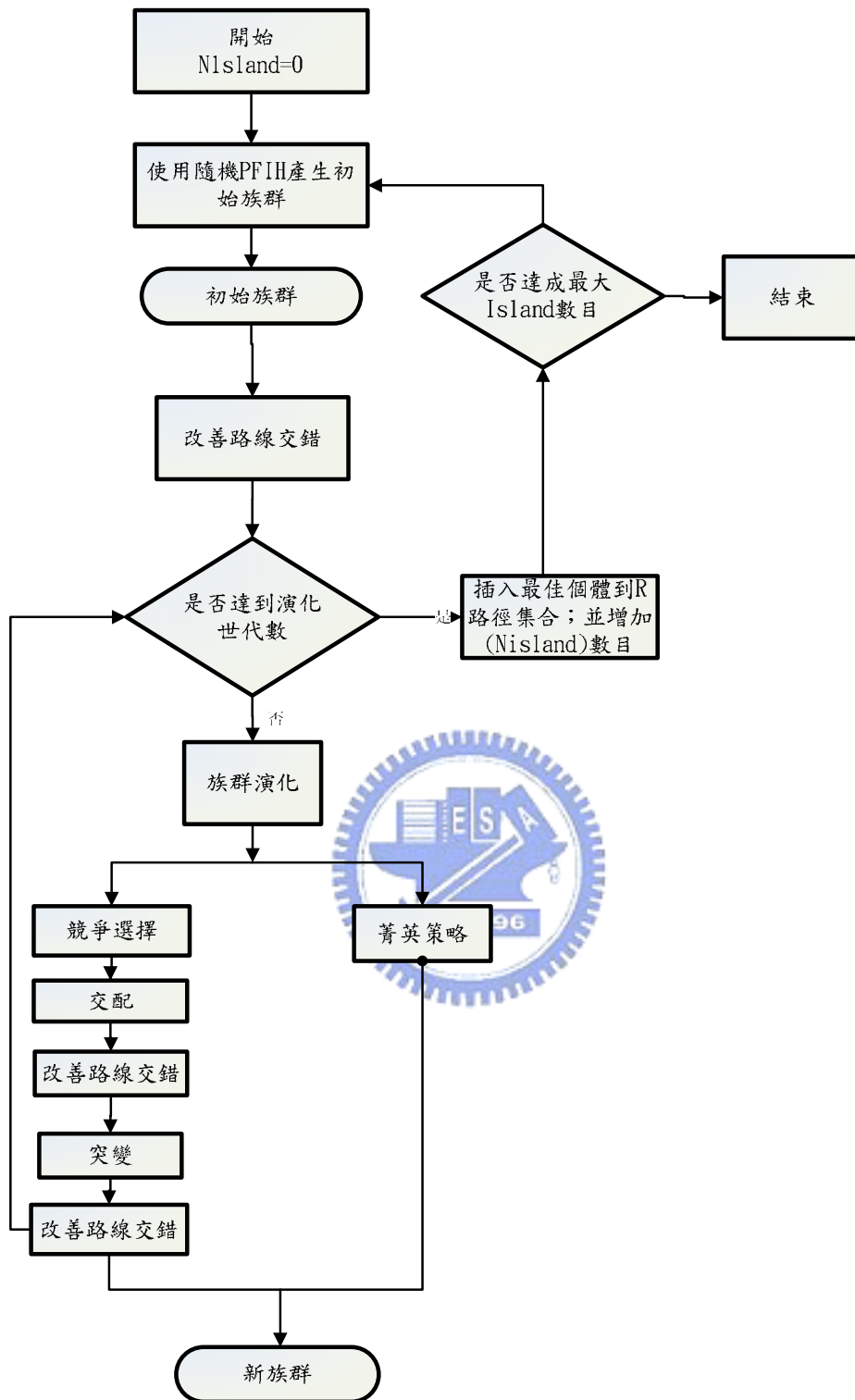


圖 3.1 第一階段基因演算法流程圖

### 一、編碼(Encoding)

本研究採用整數編碼，每一顧客有整數的定義， $i \in \{1, \dots, N\}$ ， $N$  為顧客數，一基因代表一顧客點。一條染色體代表一組解，即一條染色體表示多車之繞行路徑，路徑中不包含場站，每一染色體定為整數字串，一條路徑被一輛車服務。

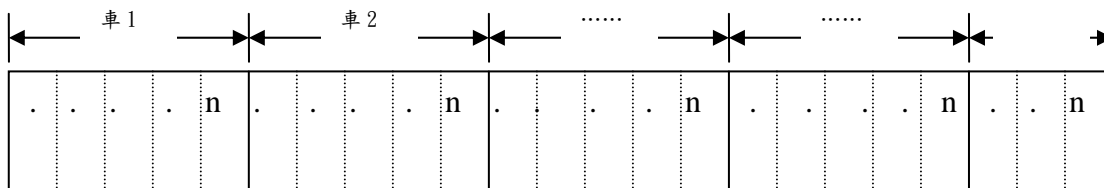


圖 3.2 染色體示意圖

## 二、 初始族群(Initial Population)

Solomon[13]所提出的 Push Forward Insertion Heuristic(PFIH)，對於 PFIH 的詳細敘述則參考[13]，此方法目前已有許多研究使用。原始的 PFIH 方法第一個顧客非隨機選擇，本研究沿用[12]針對原始 PFIH 方法做修改，每條新路徑的第一位顧客為隨機選擇，其後的顧客點則依據式(15)所計算出的成本大小，選擇最小成本的顧客點插入路徑中。

$$C_i = \alpha d_{0i} + \beta b_i + \gamma((p_i/360)d_{0i}) \dots\dots\dots(15)$$

參數  $\alpha$  設為 0.7； $\beta$  設為 0.1； $\gamma$  設為 0.2； $d_{0i}$  顧客點  $i$  到中心場站的距離； $b_i$  為允許最晚開始服務顧客  $i$  的時間； $p_i$  為從顧客點  $i$  到中心場站的極座標值。

隨機 PFIH 步驟如下：

1. 依據式(15)算出每一顧客的成本並由小到大做排序。
2. 從場站開始出發，路徑  $r=1$
3. 如果所有的顧客都服務完，到第 9 步驟。
4. 從尚未服務的顧客中挑選時窗上限最小之節點作為路徑的第一位顧客點，並更新路徑之車容量。
5. 從尚未考慮之顧客中挑選成本最小者。
6. 若符合時間可行性，到第 7 步驟；若不符合，到第 5 步驟。
7. 如果符合車容量限制，更新路徑之車容量，到第 5 步驟；若不符合則到第 8 步驟。
8. 從場站開始出發，路徑  $r=r+1$ 。到第 3 步驟。
9. 所有的顧客都服務完，則結束。

## 三、改善路線交錯

一個好的初始族群，通常能夠縮短搜尋時間，在短時間內找到不錯的解。然而，在初始族群中可能會有路線交錯的情況發生，因此在執行基因演算法前先改善路線交錯的情況，期能產生一個較好的初始族群。路線交錯的情況如圖 3.3 所示。

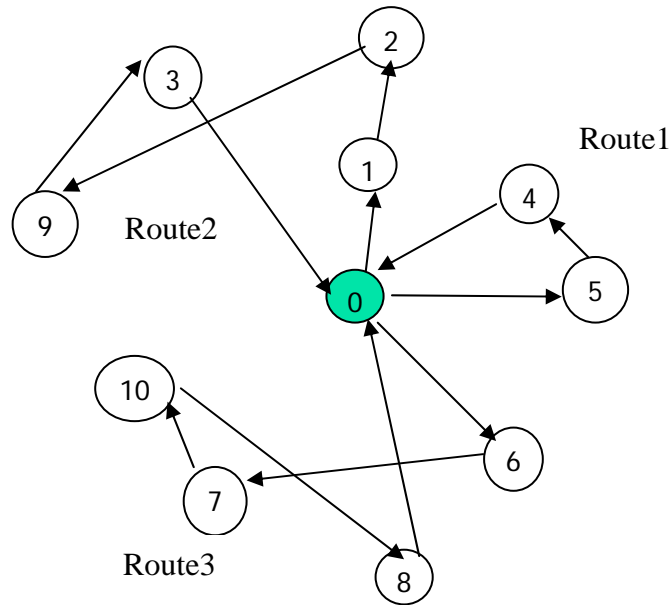


圖 3.3 路線交錯情況

首先，檢查初始族群中所有染色體的路徑間是否有交錯的情況，其判斷的準則如下 [28]：

判斷路線有無交錯：

設任意二條節線 AB 與 CD，其座標值分別為  $(a_1, a_2), (b_1, b_2), (c_1, c_2), (d_1, d_2)$ 。若座標值符合下列情形之一時，則 AB 與 CD 兩條線段必定不交錯。

$$\text{MIN}(c_1, d_1) > \text{MAX}(a_1, b_1) \dots \dots \dots (16)$$

$$\text{MIN}(a_1, b_1) > \text{MAX}(c_1, d_1) \dots \dots \dots (17)$$

$$\text{MIN}(c_2, d_2) > \text{MAX}(a_2, b_2) \dots \dots \dots (18)$$

$$\text{MIN}(a_2, b_2) > \text{MAX}(c_2, d_2) \dots \dots \dots (19)$$

其中 MIN 與 MAX 分別表示取最小值和最大值。

式(16)代表意義為：在四點中，如果 C、D 兩點中最小的 x 座標值比 A、B 兩點中最大的 x 座標值還大的話，則 AB 與 CD 兩條線段不會交錯；式(17)依此類推。式(18)代表意義為：在四點中，如果 C、D 兩點中最小的 y 座標值小比 C、D 兩點中最大的 y 座標值還大的話，則 AB 與 CD 兩條線段不會交錯；式(19)依此類推。

本研究將交錯分為兩種情形，第一種為單一交錯，第二為多重交錯。單一交錯如圖 3.4 所示。我們從圖 3.4 中可看出 0->1 與 4->5 兩條路線交錯，首先，將 0->1 與 4->5 兩條路線刪除；接著連接兩條線段的起點(顧客點 0 和顧客點 4)和迄點(顧客點 1 和 5)；最後反轉路線之連接順序。經過以上程序則可以改善交錯路線，如圖 3.5 所示。

圖 3.6 為多重交錯，圖中有兩交錯情形第一個為 0->2 與 1->6，第二個為 7->5 與 8->4。解決方式為分次處理交錯，首先，將 0->2 與 1->6 兩條路線刪除；接著連接兩條線段的起點(顧客點 0 和顧客點 1)和迄點(顧客點 2 和 6)；接著將 7->5 與 8->4 兩條路線刪除；連接兩條線段的起點(顧客點 7 和顧客點 8)和迄點(顧客點 5 和 4)；最後反轉路線之連接順序。

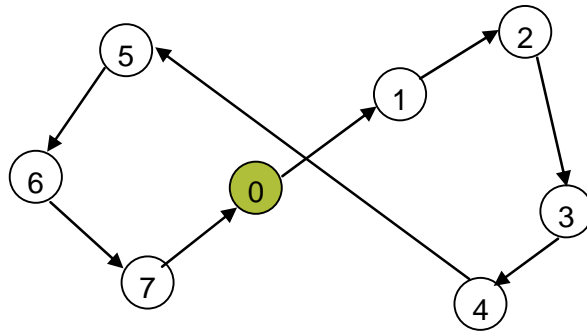


圖 3.4 單一交錯情形

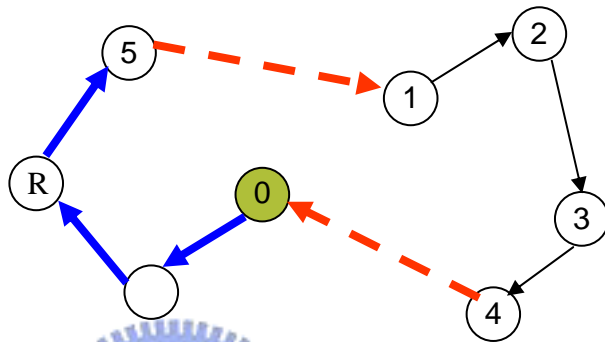


圖 3.5 單一交錯路線改善

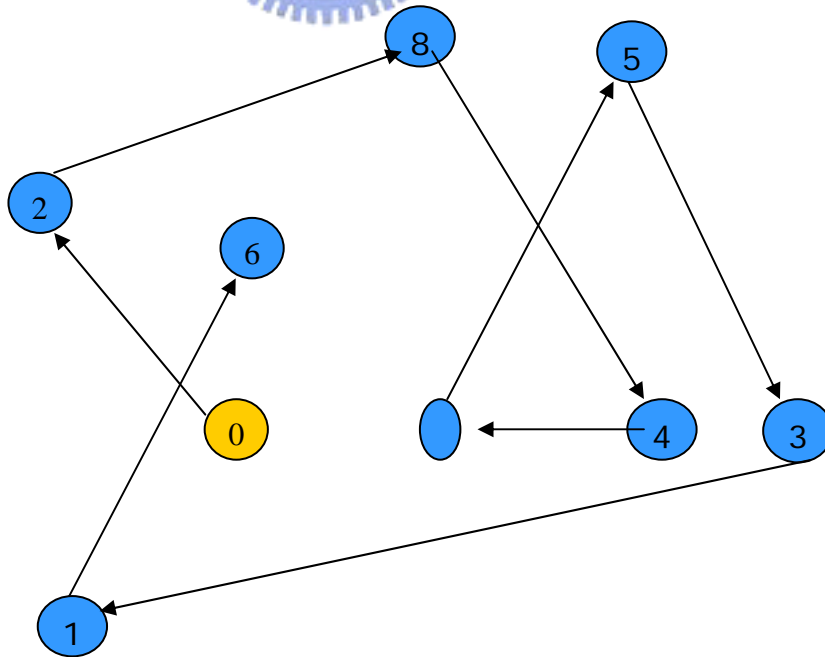


圖 3.6 多重交錯情形

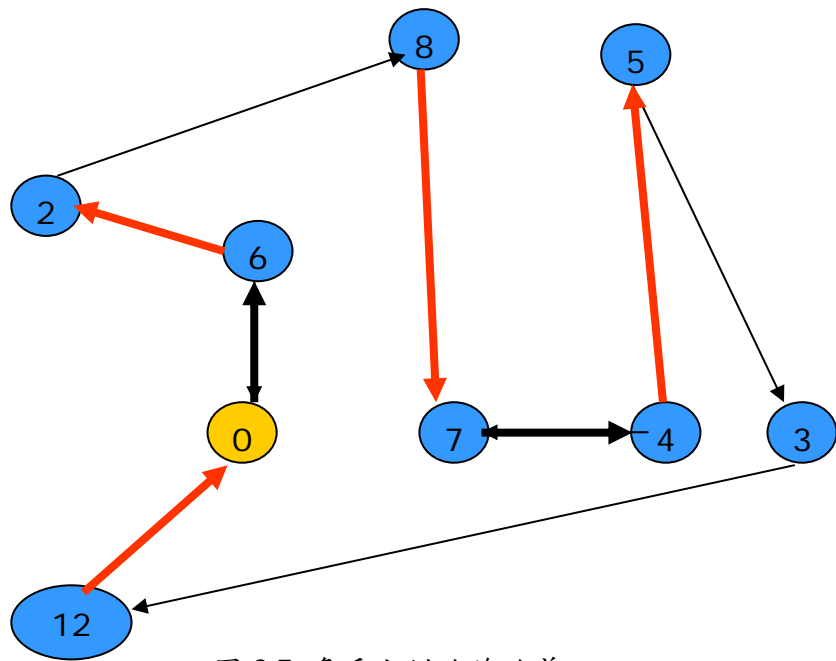


圖 3.7 多重交錯路線改善

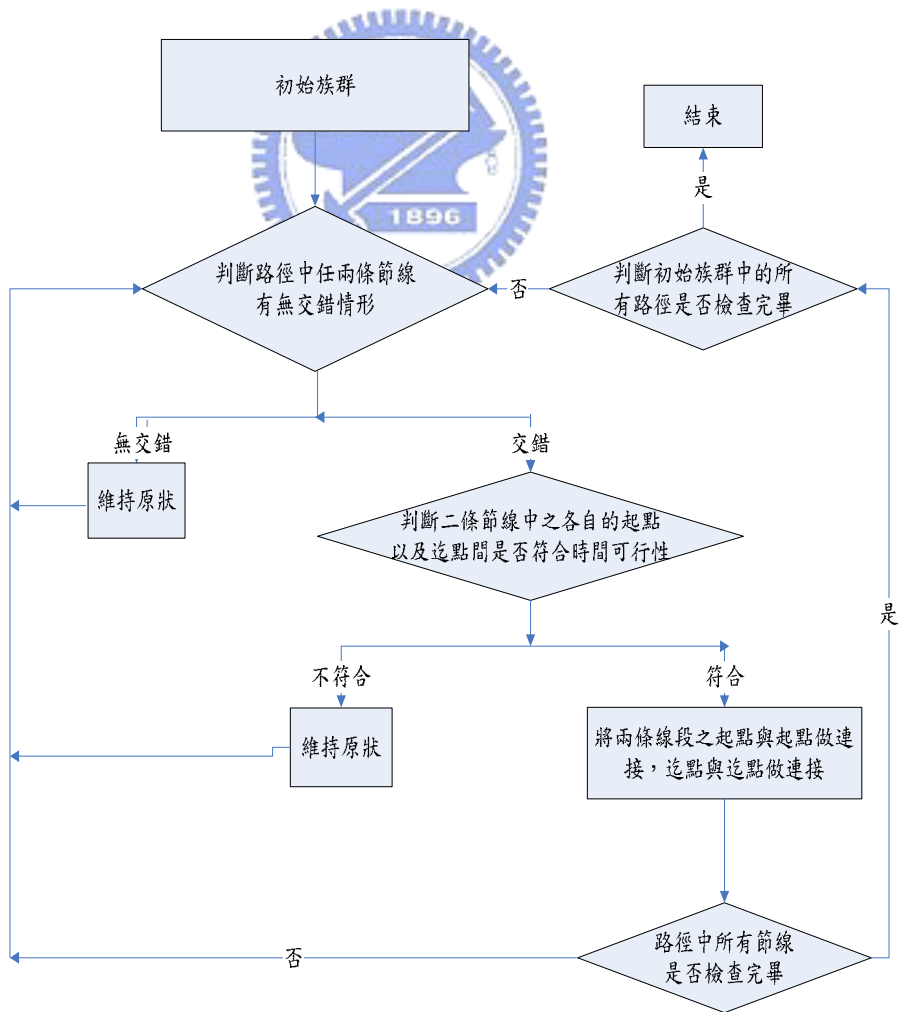


圖 3.8 改善交錯流程圖

### 三、適合度函數(Fitness Function)

適合度函數為總成本(TD)的倒數。如式(21)：

$$TD = \sum s_t + \sum (e_i - a_i) + \sum t_{ij} + \sum \text{penalty}(a_i - l_i) \dots\dots\dots(20)$$

$$\text{fitness} = \frac{1}{TD} \dots\dots\dots(21)$$

參數說明：

$s_t$ ：服務時間

$e_i$ ：顧客 i 的時間窗下界

$l_i$ ：顧客 i 的時間窗上界

$a_i$ ：到顧客 i 的時間

$\text{penalty}$ ：違反時間窗的懲罰值，設為 100

### 四、選擇(Selection)

使用競爭法，隨機選 3 個染色體，從中挑選適合度函數值愈高者，此過程直到交配槽的數目到達族群數目時結束。如圖 3.9 所示。

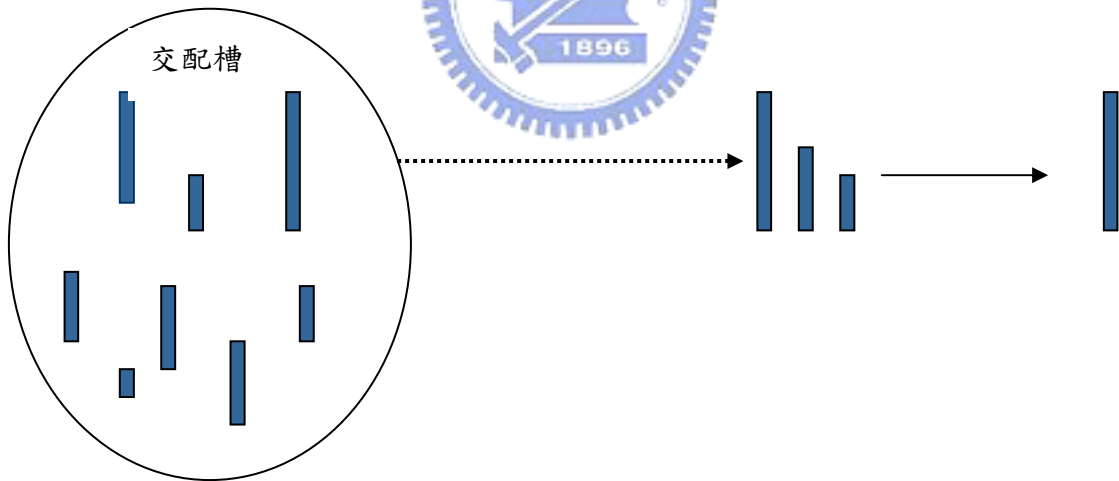


圖 3.9 選擇示意圖

### 五、交配(Crossover)

交配步驟如下：

- (1) 從交配槽中選擇兩個母代，母代一和母代二。
- (2) 判斷母代中是否還有未考慮的路徑，若是，到步驟(2)；若否，到步驟(6)。
- (3) 在母代一尚未被考慮的路徑中挑選旅行成本最小的路徑。
- (4) 判斷路徑在子代中是否有重覆的顧客點，若是，則繼承此路徑到子代中，到第(2)步驟；若否，則到步驟(3)。



- (5) 在母代二尚未被考慮的路徑中挑選距離最小的路徑。
- (6) 判斷路徑在子代中是否有重覆的顧客點，若是，則繼承此路徑到子代中，到第(2)步驟；若否，則到步驟(4)。
- (7) 尚未被服務的顧客，在考量車容量限制與時間可行性之下，若可行則嘗試將其插入到子代原有的路徑中。
- (8) 最後，若尚有未被服務的顧客，再進行隨機 PFIH 構建新路徑。

#### 六、保留菁英(Elitism)

將初始族群的適合度函數值最高者保留下來，保留菁英的好處是可以確保新世代的適合度函數值不會降低。將最佳的個體複製為兩份，避免因突變、交配而使個體的品質退化，進行演化時最佳的個體維持不變，另一複製的最佳個體仍可做交配、突變。

#### 八、突變(Mutation)

在突變之前先產生一個介於 0~1 間之隨機亂數  $P$ ，若  $P$  小於突變機率 0.05 則進行突變。

突變有以下三個步驟，如圖 3.10 所示：

- (1) 在染色體中隨機挑選一個顧客點  $i$ 。
- (2) 在染色體中隨機挑選一條路徑  $r$ 。
- (3) 若顧客點  $i$  符合路徑  $r$  之車容量限制與時間可行性，則將顧客點  $i$  插入路徑  $r$  中。

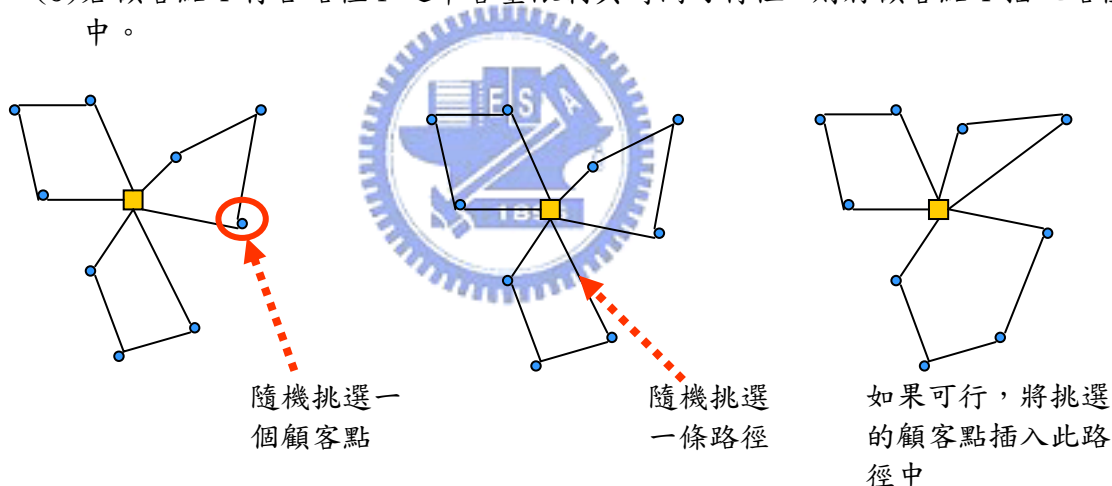


圖 3.10 突變示意圖

#### 九、終止條件

Zhu(2000)對於 VRPTW 問題，建議以演化 1000 世代為設定停止法則，本研究則採用此法則作為終止條件，若世代數到達 1000，則終止；反之，則繼續演化。終止則將目前世代輸出產生結果，否則繼續進行交配、突變與總距離計算等程序。

### 3.2 第二階段演算法設計

在第一階段演算法構建完路線後，其最後結果仍然有錯的服務順序，像是違反時窗

限制的顧客點，或是在路線中兩位顧客點間的節線距離比同路線內的其他顧客點的節線距離長的情況。在車容量與時間窗的條件限制下，為了能夠降低目標值，本研究在第二階段中提出一套改善方法並且在本節中詳細說明。

此階段演算法主要目的是改善第一階段演算法的結果。演算法的設計分為二個部份，違反時間窗成本最小與車輛旅行成本最小。

### 一、違反時間窗顧客數最小

在第一階段所求得的路徑集合解中可能會有些違反時間窗限制的顧客點，本研究的解決方法是先將違反時間窗限制的顧客點刪除，接著嘗試將此顧客點插入到路線內的符合時間窗限制位置，若有多個可行位置可插入，則選擇時間窗最接近的位置插入；若不可行，則將此顧客點嘗試插入到其他路徑中，若找到符合車容量與時間窗限制的位置則插入，否則將其恢復原狀。此步驟直到每條路線都檢查為止。

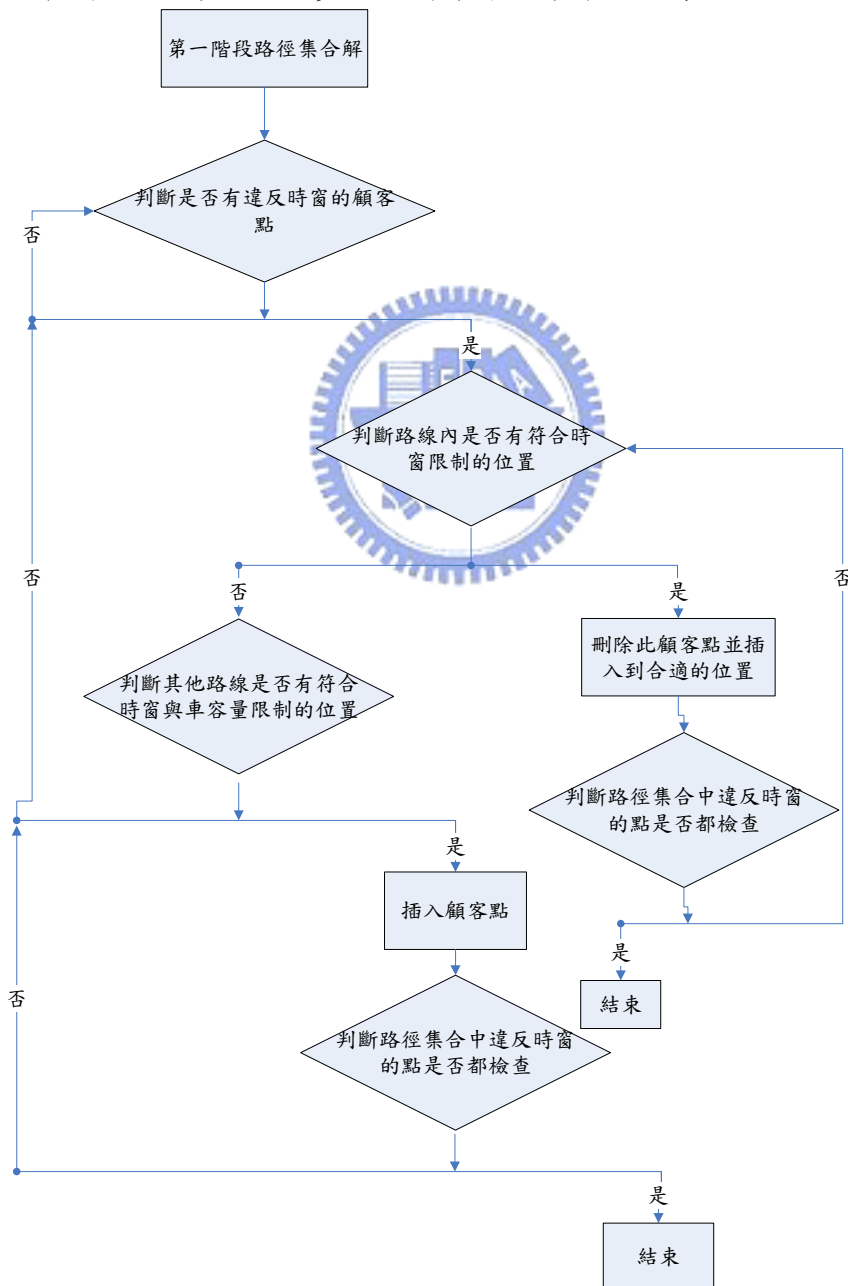


圖 3.11 改善違反時間窗顧客點流程圖

## 二、車輛旅行成本最小

在第一階段所求得的路徑集合解中有些路徑可能會服務到「不佳的顧客」(同一條路線中兩顧客點的節線距離超過總旅行成本的 15%的迄點顧客),使得整條路線的總旅行成本增加。為解決此問題,必須檢查路徑集合解的每條路線中是否存在不佳的顧客點,若有,將之刪除並且嘗試插入其他符合時間窗與車容量限制條件下的路徑中,若找不到符合限制條件下的位置,則將此顧客回復原狀。此步驟直到每條路線兩點顧客間的條節線都檢查為止。

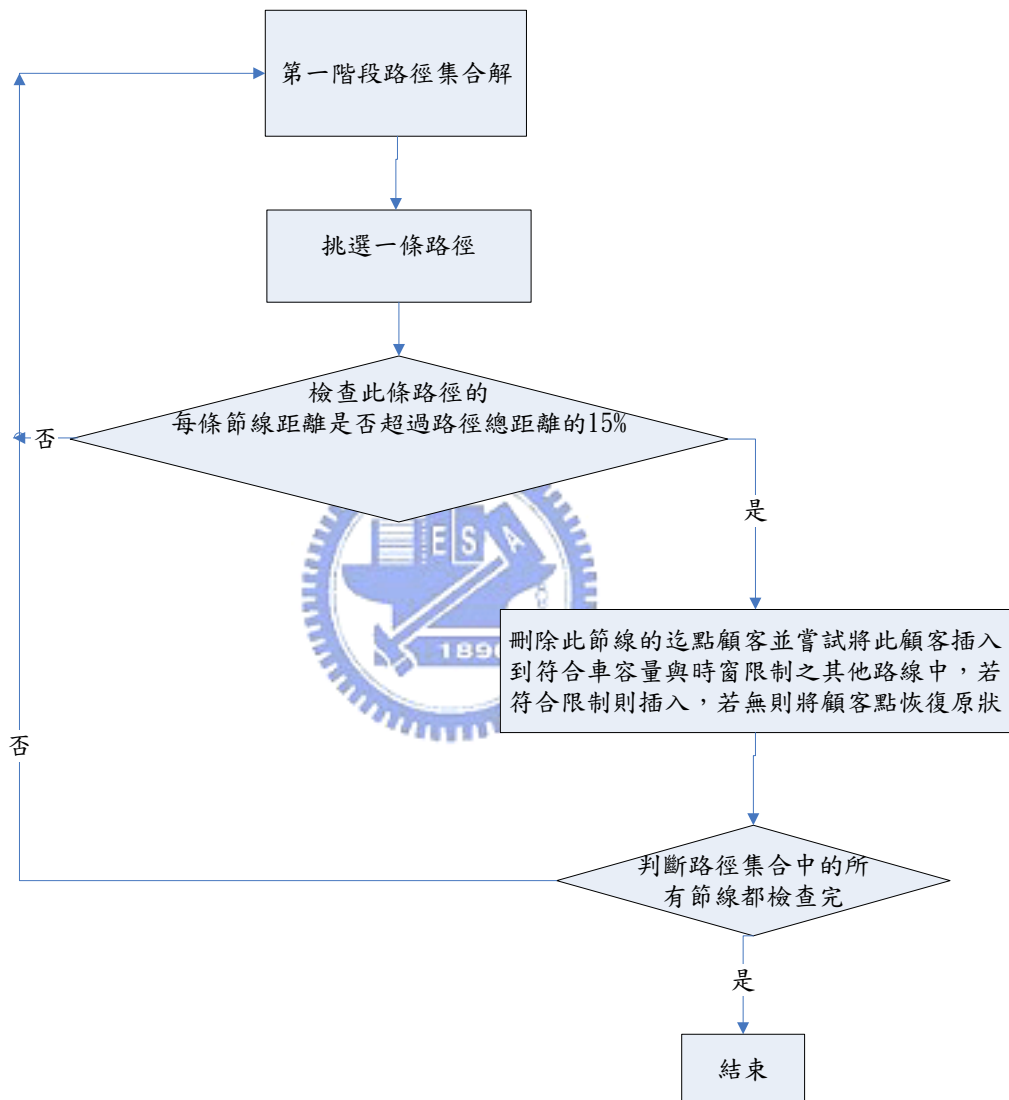


圖 3.12 車輛途程成本改善流程圖

## 第四章 測試範例

本研究採用 Solomon(1983)針對具時間窗限制車輛路線問題所設計的測試例題。Solomon 的 VRPTW 各例題依顧客點空間分佈方式分為三類：群聚(C)，完全隨機分佈(R)與混合完全隨機分佈與群聚式(RC)。將其分為六小類：C1、C2、R1、R2、RC1、RC2，共 56 題。其中 1 為時窗區間與車容量較小的問題；2 為時窗區間與車容量較大的問題。

### 4.1 測試過程

本研究所有架構的執行都是採用 C++ 進行程式撰寫，且在 Microsoft Visual Studio 2005 中編譯，測試環境的作業系統為 Microsoft Windows XP，Intel® Pentium® 4 CPU 1.80GHz 1.00GB RAM 的個人電腦進行運作。

表 4.1 範例測試參數設定表

初始族群數目	30
突變率	0.5%
懲罰乘數	100
停止條件	世代數達到 1000 時

本研究依據表 4.1 的各項參數設定，初始族群數目的設定是參照[12]。開啟 Solomon 的範例執行程式，首先產生起始解，接著開始進行基因演算法程序，最後進行改善，達到終止條件為止。4.2 節為本研究測試結果與文獻中已知最佳解之比較；4.3 節為測試改善交錯效果；4.4 節針對參數做敏感度分析，依程式執行的結果編製趨勢圖，並且對於程式演算結果做分析比較。

### 4.2 測試結果

表 4.2 為 Solomon(1983)56 題標竿例題文獻中已知最佳解。本研究將針對 R1 與 RC1 題型做測試，測試在不同的車輛數下，未違反時間窗個數的結果為何，並與文獻最佳解的車輛數做比較。表 4.3 與表 4.4 分別為 R1 與 RC1 的測試結果，表 4.5 為本研究測試結果與軟性時窗限制文獻最佳解之比較。

表 4.2 Solomon 56 題測試題庫文獻已知最佳解

題號	車輛數	總旅行成本	最佳解資料來源
C101	10	827.3	Desorchers set al.(1992)
C102	10	827.3	Desorchers set al.(1992)
C103	10	828.06	Rochat and Taillard(1995)
C104	10	824.06	Rochat and Taillard(1995)
C105	10	824.78	Rochat and Taillard(1995)
C106	10	828.94	Desorchers set al.(1992)
C107	10	827.3	Desorchers set al.(1992)
C108	10	827.3	Desorchers set al.(1992)
C109	10	828.94	Rochat and Taillard(1995)
C201	3	591.56	Rochat and Taillard(1995)
C202	3	591.56	Rochat and Taillard(1995)

C203	3	591.17	Rochat and Taillard(1995)
C204	3	590.60	Rochat and Taillard(1995)
C205	3	588.88	Rochat and Taillard(1995)
C206	3	588.49	Rochat and Taillard(1995)
C207	3	588.29	Rochat and Taillard(1995)
C208	3	588.32	Rochat and Taillard(1995)
題號	車輛數	總旅行成本	最佳解資料來源
R101	18	1604.5	Desrochers et al.(1992)
R102	17	1434.0	Desrochers et al.(1992)
R103	13	1207	Thangiah et al. (1004)
R104	9	982.01	Rochat and Taillard (1995)
R105	14	1377.11	Rochat and Taillard (1995)
R106	12	1251.98	Mester et al. (2007)
R107	10	1104.66	Shaw (1997)
R108	9	960.88	Berger and Barkaoui (2004)
R109	11	1194.73	Homberger and Gering (1999)
R110	10	1118.59	Mester et al. (2007)
R111	10	1006.72	Rousseau et al. (2002)
R112	9	953.63	Rochat and Taillard (1995)
R201	4	1252.37	Homberger and Gering (1999)
R202	3	1191.70	Rousseau et al. (2002)
R203	3	939.54	Tangiah et al. (1994)
R204	2	825.52	Ropke and Pisinger (2005)
R205	3	994.42	Mester et al. (2007)
R206	3	833	Thangiah et al. (1994)
R207	2	890.61	Ropke and Pisinger (2005)
R208	2	725.75	Mester et al. (2007)
R209	3	855	Thangiah et al. (1094)
R210	3	939.34	Mester et al. (2007)
R211	2	892.71	Bent and Van Hentenryck (2007)
題號	車輛數	總旅行成本	最佳解資料來源
RC101	14	1669	Thangiah et al. (1994)
RC102	12	1554.75	Thangiah et al. (1997)
RC103	11	1110	Thangiah et al. (1994)
RC104	10	1135.48	Cordeau et al. (2001)
RC105	13	1629.44	Berger and Bardkaoui (2004)
RC106	11	1424.73	Berger and Bardkaoui (2004)
RC107	11	1230.54	Taillard et al. (1997)
RC108	10	1139.82	Taillard et al. (1997)
RC201	4	1406.91	Mester et al. (2007)
RC202	3	1368.645	Debudaj-Grabysz and Czech (2004)
RC203	3	1049.62	Czech and Czarnas (2002)
RC204	3	798.41	Mester et al. (2007)
RC205	4	1297.19	Mester et al. (2007)
RC206	3	1146.32	Homberger and Gering (2005)
RC207	3	1061.14	Ben and Van Hentenryck (2001)
RC208	3	828.14	Ibaraki et al. (2002)

資料來源：王春鑑(2007)

表 4.3 本研究與文獻最佳解之比較(R1 型)

題型										最佳解
R101	車輛數	21	20	19	18	17	16	15	14	18
	未違反時窗顧客數	98	98	94	92	85	70	62	47	
R102	車輛數	19	18	17	16	15	14	13	12	17
	未違反時窗顧客數	98	97	92	88	82	77	70	58	
R103	車輛數	14	13	12	11					13
	未違反時窗顧客數	100	95	92	77					
R104	車輛數	11	10	9						9
	未違反時窗顧客數	100	94	77						
R105	車輛數	14	13	12						14
	未違反時窗顧客數	92	78	67						
R106	車輛數	12	11	10						12
	未違反時窗顧客數	92	77	52						
R107	車輛數	10								10
	未違反時窗顧客數	85								
R108	車輛數	10	9							9
	未違反時窗顧客數	98	80							
R109	車輛數	13	12	11						11
	未違反時窗顧客數	95	92	81						
R110	車輛數	11	10							10
	未違反時窗顧客數	98	80							
R111	車輛數	10								10
	未違反時窗顧客數	83								
R112	車輛數	10	9							9
	未違反時窗顧客數	87	80							

我們從表 4.3 中可明顯看出，當測試車輛數愈多，違反時間窗顧客數愈少。舉例：在 R103 題型中，將車輛數分別設為 14、13、12 與 11 去做測試，在文獻已知最佳解中的車輛數為 13，而本研究使用 13 輛車的結果，會有 5 個違反時間窗顧客，若車輛數小於 13，違反時間窗的顧客會遞增。

表 4.4 本研究與最佳解之比較(RC1 型)

題型						最佳解
RC101	車輛數	16	15	14	13	14
	未違反時窗顧客數	100	96	85	78	
RC102	車輛數	14	13	12	11	12
	未違反時窗顧客數	100	95	84	75	
RC103	車輛數	13	12	11	10	11
	未違反時窗顧客數	100	96	90	74	
RC104	車輛數	11	10			10
	未違反時窗顧客數	98	87			
RC105	車輛數	13	12	11		13
	未違反時窗顧客數	90	73	65		
RC106	車輛數	13	12	11		11
	未違反時窗顧客數	100	90	81		
RC107	車輛數	11	10			11
	未違反時窗顧客數	92	82			
RC108	車輛數	11	10			10
	未違反時窗顧客數	98	85			

表 4.4 為測試 RC1 例題的，從測試結果中，我們亦可得到車輛數愈多，違反時間窗顧客數愈少之結論。

表 4.5 本研究與文獻最佳解之比較(R1 型)

題型		BAL	1896	PGSH			本研究			
R101	車輛數	17	16	14	17	16	14	17	16	14
	未違反時窗顧客數	72	55	44	89	75	52	85	70	47
R102	車輛數	19	13		13			13		
	未違反時窗顧客數	100	63		74			70		
R103	車輛數	13	12		13	12		13	12	
	未違反時窗顧客數	86	68		97	93		95	92	
R109	車輛數	13	12	11	12	11		12	11	
	未違反時窗顧客數	100	90	67	99	87		92	81	

表 4.6 本研究與文獻最佳解之比較(RC1 型)

題型		BAL			PGSH			本研究		
RC101	車輛數	16	15	14	16	15	14	16	15	14
	未違反時窗顧客數	100	96	68	100	100	92	100	96	85
RC102	車輛數	14	13		14	13		14	13	
	未違反時窗顧客數	100	88		100	98		100	95	
RC103	車輛數	13	12		13	12		13	12	
	未違反時窗顧客數	100	92		100	100		100	96	
RC106	車輛數	13	12		13	12		13	12	
	未違反時窗顧客數	100	71		100	99		100	90	

我們參考 Balakrishnan [26]與 George et al.[27]的軟性時間窗文獻，其方法分別以 BAL 與 PGSH 來表示。表 4.5 與 4.6 為本研究與此二篇文獻做車輛數與未違反時窗顧客數的比較。此兩篇文獻針對 R101、R102、R103、R109、RC101、RC102、RC103 與 RC106 例題來做測試。在表 4.4 中本研究的結果皆優於 BAL，但比 PGSH 差。在表 4.5 中的 RC1 測試例題中，當 RC101、RC102、RC103 與 RC106 的車輛數分別為 16、14、13 與 13 時，未違反時窗顧客數皆為 100，至於其他車輛數而言，PGSH 表現最好，其次為本研究。

### 4.3 改善交錯的效果

本節為測試第三章的改善交錯的效果，分別針對表 4.7 為聚分佈例題(C 型)、表 4.8 為隨機分佈例題(R 型)，表 4.9 為隨機與群聚分佈例題(RC 型)來測試改善交錯前後的距離差值。

表 4.7 改善交錯前後測試比較結果(群聚分佈例題)

題型	改善前車輛數	改善前總旅行成本	改善後車輛數	改善後總旅行成本	改善交錯前後距離差百分比
C101	10	1167.3	10	943.1	19.2%
C102	10	1172.1	10	947.2	19.1%
C103	10	1195.1	10	984.6	17.6%
C104	10	1180.0	10	964.1	18.3%
C105	10	1151.2	10	937.7	18.5%
C106	10	1179.0	10	963.6	18.3%
C107	10	1120.5	10	903.0	19.4%
C108	10	1200.1	10	989.2	17.6%
C109	10	1182.1	10	969.5	18.0%
C201	3	954.2	3	762.1	20.1%
C202	3	940.1	3	758.2	19.3%
C203	3	932.0	3	743.0	20.3%
C204	3	945.3	3	762.0	19.4%
C205	3	940.6	3	752.0	20.1%
C206	3	947.1	3	755.6	20.2%
C207	3	945.0	3	752.9	20.3%
C208	3	960.7	3	759.8	20.9%



表 4.8 改善交錯前後測試比較結果(隨機分佈例題)

題型	改善前車輛數	改善前總旅行成本	改善後車輛數	改善後總旅行成本	改善交錯前後距離差百分比
R101	18	1874.1	18	1639.7	12.5%
R102	18	1745.0	18	1522.9	12.7%
R103	15	1556.8	15	1387.1	10.9%
R104	10	1367.2	10	1183.5	13.4%
R105	15	1685.8	15	1457.8	13.5%
R106	13	1573.1	13	1338.6	14.9%
R107	11	1542.4	11	1255.9	18.6 %
R108	11	1312.8	11	1097.7	16.4%
R109	12	1584.7	12	1231.2	22.3%
R110	12	1600.3	12	1275.5	20.3%
R111	13	1328.2	13	1171.7	11.8%
R112	10	1273.6	10	1005.4	21.1 %
R201	4	1490.2	4	1253.8	15.9 %
R202	6	1375.4	6	1195.3	13.1%
R203	5	1320.7	5	1084.6	17.9%
R204	5	1217.3	5	986.5	19.0%
R205	4	1345.4	4	1027.6	23.6%
R206	5	1137.2	5	903.2	20.6%
R207	5	1200.5	5	957.3	20.3%
R208	4	1025.6	4	839.7	18.1%
R209	5	1247.2	5	998.0	20.0%
R210	5	1315.8	5	1104.4	16.1%
R211	5	1361.4	5	1041.6	23.5%

表 4.9 改善交錯前後測試比較結果(隨機與群聚分佈例題)

題型	改善前車輛數	改善前總旅行成本	改善後車輛數	改善後總旅行成本	改善交錯前後距離差百分比
RC101	14	1824.1	14	1680.9	7.9%
RC102	12	1752.5	12	1584.9	9.6%
RC103	12	1413.2	12	1192.8	15.6%
RC104	11	1463.4	11	1213.6	17.1%
RC105	14	1851.1	14	1719.0	7.1%
RC106	12	1723.5	12	1580.1	8.3%
RC107	11	1458.6	11	1230.7	15.6%
RC108	11	1512.5	11	1285.6	15.0%
RC201	4	1658.1	4	1412.1	14.8%
RC202	3	1572.1	3	1368.7	12.9%
RC203	3	1241.8	3	1049.9	15.5%
RC204	4	1187.4	4	920.8	22.5%
RC205	4	1503.7	4	1298.0	13.7%
RC206	3	1375.8	3	1148.1	16.6%
RC207	3	1342.4	3	1117.7	16.7%
RC208	3	1124.1	3	902.3	19.7%

從表 4.7、4.8 與 4.9 的改善交錯前後距離差百分比一欄中，我們可以觀察到改善交錯方法能使總距離有明顯的減少，證明改善交錯為一有效降低距離的方法。

#### 4.4 敏感度分析

本研究為了解各項參數對於求解品質等影響，針對以下參數做敏感度分析。包括突變率、懲罰乘數、初始母體族群與兩顧客點的節線距離超過總旅行成本的百分比。

以表 4.10 的參數設定為標準，初始族群數目為 30、突變率為 0.5%、懲罰乘數為 100，兩顧客點間節線距離超過總旅行成本的百分比為 12%。

表 4.10 敏感度分析參數設定表

突變率	0.5%
懲罰乘數	100
初始母體族群	30
兩顧客點間節線距離超過總旅行成本的百分比	12%

以上述四項參數進行測試，除了停止條件參數外，其他三項參數皆進行 1000 個演化世代即終止。測試的數據分別為，初始母體族群：30、50、100、150 以及 200；突變率：1/5000、1/1000、1/500、1/100、1/50 以及 1/10；懲罰乘數：25、50、75、100、125 以及 150；兩顧客點間節線距離超過總旅行成本的百分比：6%、9%、12%、15% 與 18%。以下將對於各參數結果分別做詳細說明。

##### 1. 突變率

從表 4.11 和圖 4.1 中可以觀察到，突變率為 1/1000 時的平均最佳成本最低，當突變率增加時，平均最佳成本也隨之增加。突變的設計，最主要是目的為避免在尋優的過程中，落入區域最佳解(local optimum)，藉以跳脫出區域最佳解，再搜尋鄰近的區域，能有機會找到全域最佳解(global optimum)。因此突變的發生機率對於尋優的影響很大，如果機率太大的話，會過度提高母體的異樣程度，造成難以求得較佳的解；若機率低的話，又難有跳脫的機會。

表 4.11 突變率變動敏感度分析表

例題	突變率					
	1/5000	1/1000	1/500	1/100	1/50	1/10
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
RC101	1691.7	1661.64	1623.6	1614.5	1640.4	1538.8
RC102	1550.2	1530.9	1569.5	1529.1	1546.9	1542.8
RC103	1515.1	1540.2	1522.8	1529.1	1533.1	1553.5
RC104	1316.1	1314.1	1369.1	1328.4	1317.9	1320.0
RC105	1815.3	1818.8	1883.8	1840.2	1869.1	1816.3
RC106	1742.6	1758.1	1748.5	1754.2	1739.0	1743.5
RC107	1102.5	1042.7	1030.7	1047.7	1102.5	1098.9
RC108	1540.1	1562.3	1564.4	1551.57	1534.6	1563.0
RC201	1130.5	1130.5	1132.9	1138.2	1132.9	1329.0
RC202	1120.8	1086.7	1162.4	1145.3	1152.0	1193.5
RC203	1033.4	1042.8	1042.8	1098.7	1045.4	1042.8
RC204	1025.3	1025.1	1025.3	1078.6	1025.3	1096.8
RC205	1122.1	1122.1	1122.1	1122.1	1150.6	1122.1
RC206	1088.5	1122.1	1100.2	1098.4	1122.7	1088.5
RC207	1128.4	1056.9	1117.7	1117.7	1117.7	1126.0
RC208	1051.6	1057.9	1014.9	1079.5	1082.1	1088.2
平均	1310.8	1304.55	1314.4	1317.0	1319.5	1328.9

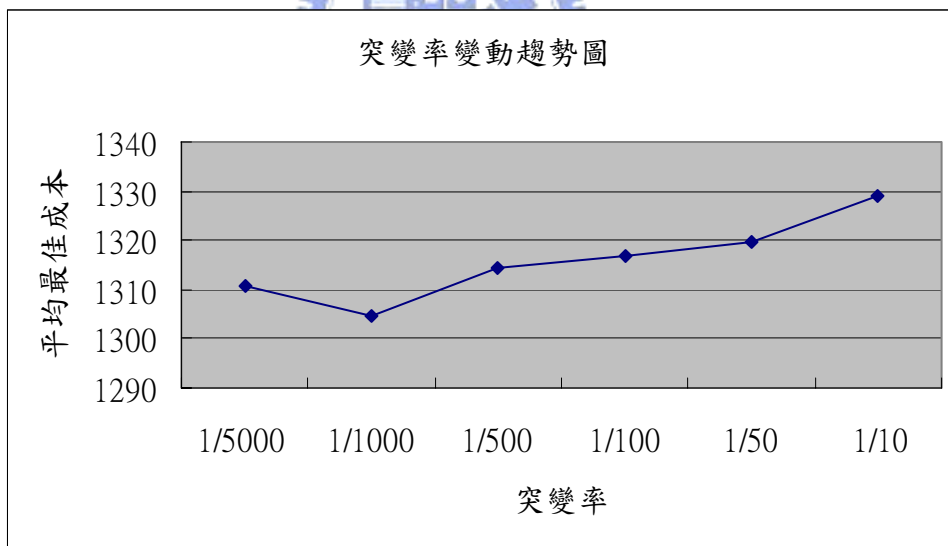


圖 4.1 突變率變動趨勢圖

## 2. 懲罰乘數

從表 4.12 和圖 4.2 中可以觀察到，當懲罰值介於 25 到 100 時的平均最佳成本趨勢略為持平的狀態，當懲罰值增加到一定程度呈現下滑的狀態。由此可知，平均最佳成本和懲罰乘數為負相關。

懲罰乘數是針對遲到或早到，給予一個固定的成本，目的為總違反時窗成本最小，以滿足顧客的需求。隨著懲罰乘數值增加，將促使車輛盡量在時窗限制內服務。相對而言，若懲罰乘數過小，時間價值相對較小，效果較不明顯，違反時窗程度也較嚴重。

表 4.12 懲罰乘數變動敏感度分析表

例題	懲罰乘數							
	25	50	75	100	125	150	175	200
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
RC101	1790.5	1782.4	1769.5	1720.2	1712.3	1680.2	1685.7	1681.5
RC102	1734.3	1726.1	1713.6	1696.9	1670.2	1590.8	1590.3	1584.9
RC103	1292.3	1176.2	1254.4	1220.7	1205.6	1195.4	1193.1	1192.9
RC104	1423.4	1410.5	1387.5	1342.1	1315.5	1250.6	1257.9	1215.5
RC105	1827.0	1805.2	1785.5	1763.4	1743.2	1725.0	1728.8	1719.0
RC106	1672.2	1650.0	1630.4	1607	1597.7	1588.4	1588.2	1585.1
RC107	1314.5	1302.2	1287.6	1260.7	1252.5	1238.7	1233.5	1230.3
RC108	1326.6	1315.9	1304.6	1300.2	1291.4	1288.6	1285.6	1285.6
RC201	1537.2	1518.2	1488.1	1452.4	1429.3	1421.2	1417.4	1413.1
RC202	1497.1	1478.4	1460.4	1423.2	1415.6	1387.4	1377.2	1369.7
RC203	1149.2	1125.7	1102.3	1081.5	1076.7	1061.3	1054.3	1049.9
RC204	1020.6	1002.2	978.7	970.3	953.2	943.8	933.5	924.8
RC205	1335.1	1310.3	1312.6	1302.0	1302.6	1298.4	1298.0	1298.0
RC206	1258.9	1237.9	1215.1	1187.2	1167.1	1158.5	1152.3	1148.1
RC207	1247.6	1228.2	1216.2	1280.2	1250.7	1140.4	1132.7	1127.5
RC208	1005.7	994.3	972.8	964.3	951.4	910.3	910.1	905.3
平均	1402.0	1378.9	1367.4	1348.2	1333.4	1304.9	1302.4	1295.7

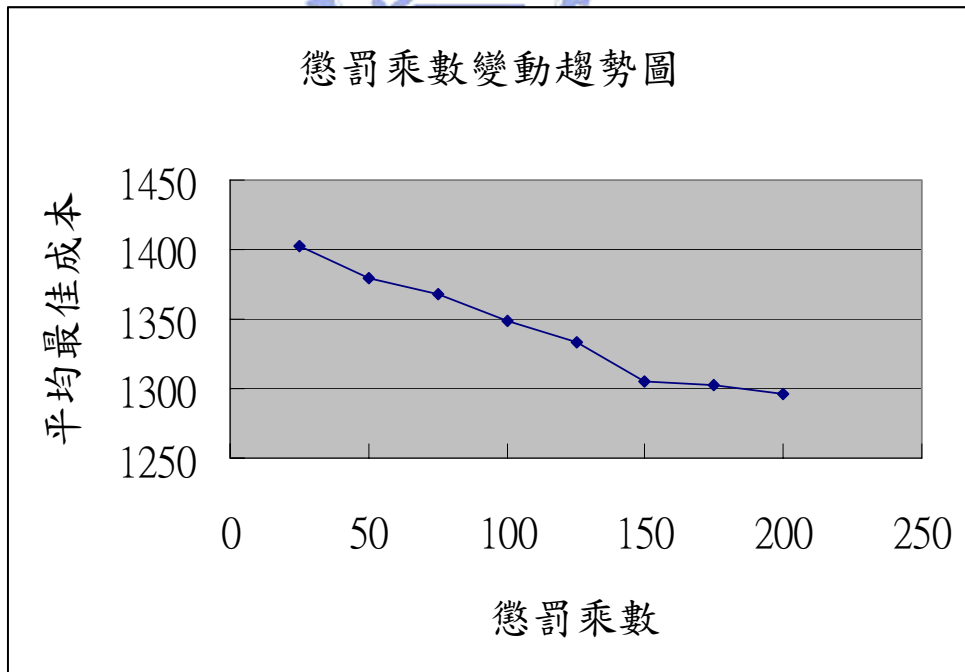


圖 4.2 懲罰乘數變動趨勢圖

### 3. 初始族群

從表 4.13 和圖 4.3 中可觀察到，平均最佳成本隨著初始族群的數量增加而呈現下降的趨勢，兩者為負相關。從圖 4.3 與圖 4.4 中亦可觀察出，雖然初始族群增加其平均最佳成本是下降的，但其 CPU 執行時間卻是增加。

初始族群大小對於演算法的求解品質有直接的關係，若初始族群過小，會容易提早收斂；若初始族群過大，則易造成搜尋時間過長。因此本研究訂 30 為初始族群，期能求解更有效率。

表 4.13 初始族群變動敏感度分析表

例題	初始族群				
	30	50	100	125	150
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
RC101	1447.5	1447.4	1453.7	1456.6	1440.9
RC102	1473.9	1462.4	1497.6	1495.0	1494.9
RC103	1373.3	1392.9	1385.6	1424.8	1372.8
RC104	1373.4	1377.0	1368.9	1353.4	1313.6
RC105	1716.7	1720.1	1756.1	1732.1	1719.0
RC106	1693.3	1685.4	1690.3	1706.1	1680.1
RC107	1097.1	1149.8	1121.4	1152.9	1094.7
RC108	1500.4	1494.6	1516.1	1534.7	1485.6
RC201	1232.9	1230.3	1249.5	1232.9	1212.1
RC202	1112.6	1108.1	1145.5	1063.3	1106.2
RC203	1045.4	1086.3	1028.4	1089.1	1080.1
RC204	1186.7	1108.0	1028.3	1143.5	1096.8
RC205	1119.9	1130.5	1230.9	1100.6	1155.2
RC206	1165.1	1133.1	1088.5	1088.5	1122.1
RC207	1117.7	1123.8	1119.1	1059.8	1169.0
RC208	1226.4	1222.0	1152.9	1072.6	1087.3
平均	1305.1	1304.4	1302.0	1294.1	1288.7

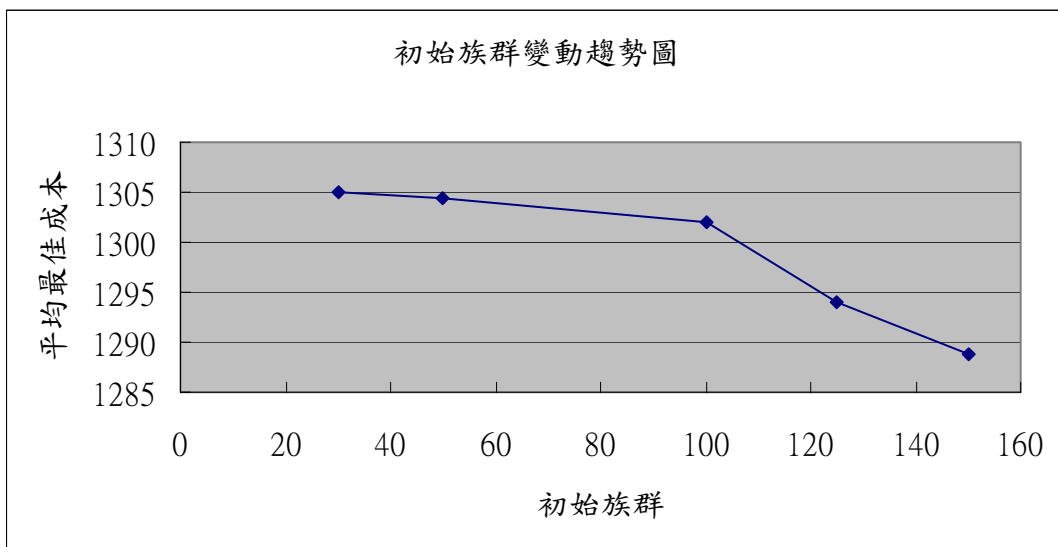


圖 4.3 初始族群變動趨勢圖

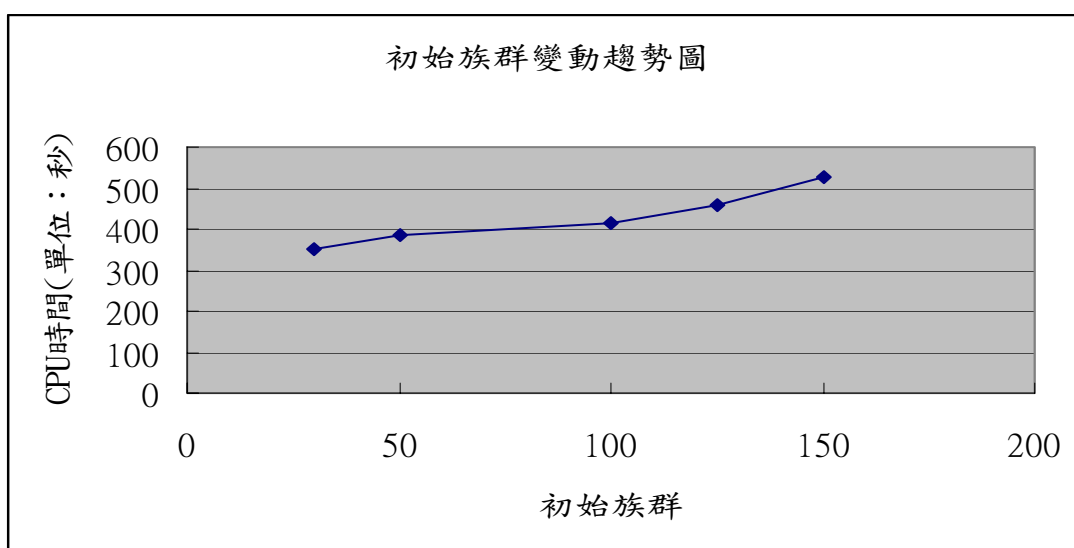


圖 4.4 初始族群與 CPU 時間趨勢圖

4. 兩顧客點間節線距離超過總旅行成本的百分比

在 3.2 節中第二階段之最小旅行成本演算法設計中，其中在同一條路線中兩顧客點間節線距離超過總旅行成本的百分比的參數設定上，本研究將測試當百分比參數改變時，總旅行成本的變化。從表 4.14 與圖 4.5 中可觀察到，當顧客點間節線距離超過總旅行成本的百分比為 15% 與 18% 時的平均最佳成本最小。因此在進行第二階段之最小旅行成本演算法檢查顧客點間節線距離超過總旅行成本的百分比，將採用 15% 作為參數。

表 4.14 顧客點間節線距離超過總旅行成本的百分比敏感度分析表

例題	顧客點間節線距離超過總旅行成本的百分比				
	6%	9%	12%	15%	18%
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
RC101	1792.5	1790.7	1785.2	1675.1	1675.1
RC102	1741.2	1726.5	1713.6	1590.2	1590.2
RC103	1252.3	1225.9	1204.4	1173.3	1173.3
RC104	1409.8	1342.6	1307.8	1266.1	1266.1
RC105	1823.0	1805.2	1785.2	1730.9	1730.9
RC106	1670.4	1652.3	1634.4	1584.7	1584.7
RC107	1318.3	1300.7	1281.9	1230.5	1230.5
RC108	1329.6	1317.5	1308.2	1285.4	1285.4
RC201	1537.1	1516.8	1478.2	1412.3	1412.3
RC202	1495.1	1473.7	1420.3	1370.4	1370.4
RC203	1149.2	1128.9	1112.2	1076.7	1076.7
RC204	1023.8	1000.7	968.2	928.9	928.9
RC205	1338.2	1330.8	1322.3	1292.3	1292.3
RC206	1258.5	1230.5	1197.1	1148.5	1148.5
RC207	1300.6	1268.8	1243.2	1225.1	1225.1
RC208	1005.7	991.6	952.8	900.2	900.2
平均	1402.831	1381.45	1357.188	1305.663	1305.663

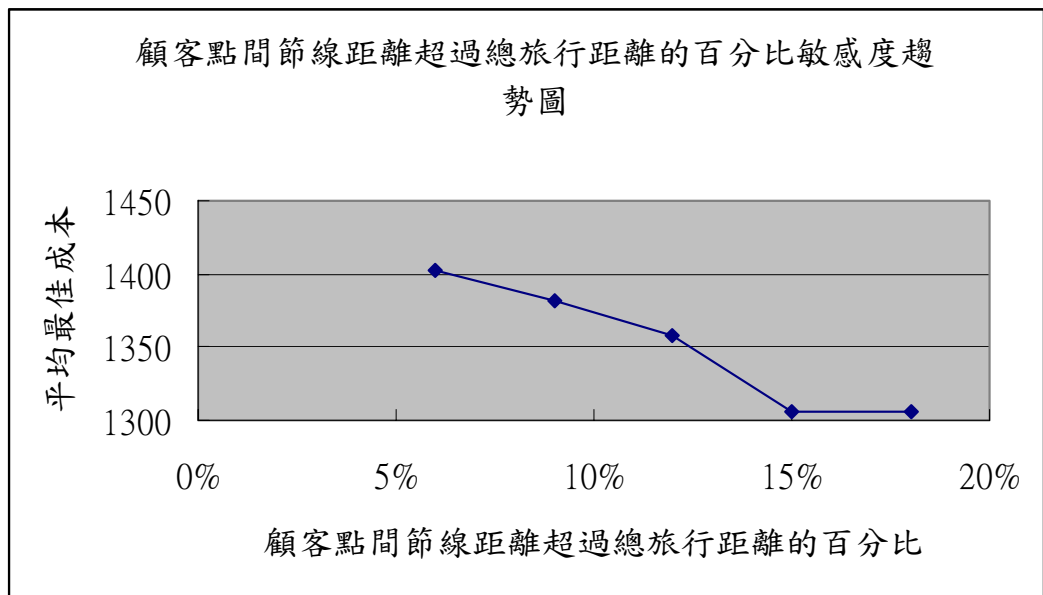


圖 4.5 顧客點間節線距離超過總旅行成本的百分比敏感度趨勢圖



## 第五章 結論與建議

### 5.1 結論

1. 本研究基因演算法使用許多隨機的概念，以機率做為演算法則的依據，做廣度的搜尋，能找出較好的路徑組合且增加其多樣化。
2. 本研究在建構完初始族群後，利用改善路線交錯的方式求得一個較好的母體，在時間範圍內能搜尋到不錯解。
3. 在敏感度分析上，證實懲罰乘數和初始母體族群的大小會影響求解的品質，且呈現正相關。
4. 使用基因演算法尋優，之後再利用路徑改善方法，可在合理的時間內找出不錯的路徑組合。
5. 本研究的改善交錯方法能明顯降低總旅行成本。
6. 在合理的車輛數下，能夠有效降低違反時間窗顧客數。

### 5.2 建議

1. 可針對基因演算法，加入其他相關的機制，使其能在合理時間內求得最佳解。亦可結合其他適合 VRPTW 的啟發式解法發展新的演算法，提高求解品質與效率。
2. 本研究假設距離和速度的比例是相同，但此情況和實際狀況不符，演算方法中若能考量到實際上的車速，在實務上會更有貢獻。
3. 在第二階段改善法上，後續的研究可加入 or-opt、2-opt、2-opt\*、1-1 節點交換或 1-0 交換等改善方法。
4. 未來研究可考慮即時路況系統與旅行時間，使物流業者可透過線上系統即時更新路線，以減少旅行時間成本且可增加效率。
5. 未來研究可考慮多場站，更能符合實務業者的需求。



## 參 考 文 獻

1. B.E. Gillett and L.R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem", Operations Research, Vol.22, pp.340-349, 1974.
2. J.Y Potvin and J.M Rousseau, "An Exchange Heuristic for Routing Problems with Time windows", Journal of the Operational Research Society Vol.46, pp.1433-1446, 1995.
3. M. Gendreau, A. Hertz, and G. Laporte, "New Insertion and Post optimization Procedures for the Traveling Salesman Problem" Operation Research Vol.40, pp.1086-1094, No.6, Nov-Dec, 1992.
4. B. L. Golden, J. S. DeArmon & E. K. Baker, "Computational Experiments with Algorithms for A Class of Routing Problems" , Computers and Operations Research Vol.10, No.1, pp.47-59, 1983.
5. Marshall L.Fisher & Ramchandran Jaikumar, "A Generalized Assignment Heuristic For Vehicle Routing", Networks, Vol.11, pp.109-124, 1981.
6. Dueck, G. and Scheuer, T., "Threshold Accepting: a General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," Journal of Computational Physics, Vol.90, pp.161-175, 1990.
7. Dueck, G., "New Optimization Heuristics: the Great Deluge Algorithm and the Record-to-Record Travel," Journal of Computational Physics Vol.104, pp.86-92, 1993.
8. Barrie M. Backer, M.A. Ayechev, "A genetic algorithm for the vehicle routing problem", Computers& Operations research Vol.30, pp.787-800, 2003.
9. Carter, A.E. and Ragsdale, C.T., "A new approach to solving the multiple traveling salesperson problem using genetic algorithms " European Journal of Operational Research ,Vol.175, pp. 246-257, 2006.
10. Tan K.C., Lee L.H. and Zhu K.Q., "Heuristic methods for vehicle routing problem with time windows", Artificial Intelligence Engineering Vol.15, pp.281-295, 2001.
11. Mak K.L. and Guo Z., "A Genetic Algorithm for Vehicle Problems With Stochastic Demand and Soft Time Windows" Proceedings of the 2004 Systems and Information Engineering Design Symposium.
12. G.B Alvarenga, G.R. Mateus, G.de Tomi, "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows" Computers &Operations research Vol.34 , pp.1561-1584, 2007.
13. Potvin, JY. And Laporte, G., "Genetic Algorithm for the Traveling Salesman Problem", Annals of Operations Research Vol.63, pp.339-370, 1996.
14. Koskosidis A., Powell B., and Solomon M., "An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints", Transportation Science, Vol.26, No.2, pp.69-85, 1992.
15. W. C. Chiang, R. A. Russell, "Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows", Annals of Operations Research Vol.63, pp.3-27, 1996.
16. W. C. Chiang, R. A. Russell, "A Reactive Tabu Search Metaheuristics for the Vehicle Routing Problem with Time Windows", Inform Journal on Computing Vol.9, No.4, pp.417-430, 1997.
17. W. R. Jin, J. Y. Hsu, "Dynamic Vehicle Routing Using Hybrid Genetic Algorithms",

IEEE Proceedings of International Conference on Robotics & Automation Vol.1, pp.453-458, 1999.

18. Thangiah S R, Osman I H, Sun T., "Hybrid genetic algorithm simulated annealing and tabu search methods for vehicle routing problem with time windows", Technical Report 27, Computer Science Department, Slippery Rock University, 1994.
19. Solomon MM., "Algorithms for the vehicle routing and scheduling problems with time window constraints." Operations Research Vol.35, No.2, pp.254-265, 1987.
20. B. Ombuki, B.J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows", Department of Computer Science, 2006.
21. K.C. Tan, T.H. Lee, K. Ou, and L.H. Lee, "A messy genetic algorithm for the vehicle routing problem with time window constraints", IEEE, 2001.
22. J. Berger, M. Barkaoui and O. Bräysy, "A route-directed hybrid genetic approach for the vehicle routing problem with time windows", Infor vol.41, No.2, pp179-194, 2003.
23. Jia Ma, Hao Zou, Li-qun Gao and Dan li, "Immune genetic algorithm for vehicle routing problem with time windows", IEEE, 2006.
24. Sam R, Tangiah, Kendall E. Nygard and Paul L. Juell, "Gideon: A genetic algorithm system for vehicle routing with time windows", IEEE, 1991.
25. Sam R. Tangiah and Ananda V. Gubbi, "Effect of genetic sectoring on vehicle routing problems with time windows", IEEE, 1993.
26. George Ioannou, Manolis Kritikos, Gregory Prastacos, "A problem generator-solver heuristic for vehicle routing with soft time windows", Omega Vol.31, pp.41-53, 2003.
27. Balakrishnan N., "Simple heuristics for the vehicle routing problem with soft time windows", Journal of the Operational Research Society, Vol.44, No.3, pp.279-87, 1993.
28. 陳昭榮, 「應用自我組織類神經網路於最長不交錯路徑問題」, 國立臺北科技大學電機工程系, 民國九十一年五月。
29. 卓裕仁, 「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」, 國立交通大學運輸工程與管理學系所博士論文, 民國九十年七月。
30. 黃春鎰, 「混合演化巨集啟發式解法應用於具時間窗車輛路線問題之研究」, 國立交通大學運輸科技與管理學系, 民國九十六年八月。
31. 「智慧化商用車隊資源管理系統整合之研究」計劃, 運輸研究所, 民國九十七年。

## 簡歷



姓 名：陳綠茵

籍 貫：桃園縣

出生日期：72年9月17日

電子郵件：[green.tem95g@nctu.edu.tw](mailto:green.tem95g@nctu.edu.tw)

學 歷：

國立交通大學運輸科技與管理學研究所

東吳大學經濟學系

德明技術學院