

國立交通大學

運輸科技與管理學系碩士班

碩士論文

具時間窗限制提送貨問題之研究

**Solving a Pickup and Delivery Problem
with Time Windows Constraints**

研究生：周韻佳
指導教授：王晉元

中華民國九十六年六月

具時間窗限制提送貨問題之研究

Solving a Pickup and Delivery Problem

with Time Windows Constraints

研究生：周韻佳

Student：Yunn-Jia Chou

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

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

具時間窗限制提送貨問題之研究

學生姓名：周韻佳

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

貨運業者為滿足客戶需求，又面臨同業競爭壓力，以在分秒必爭之微利時代從中勝出。如何提供此一迅捷、確實之貨運品質以滿足客戶之貨運需求，即為各貨運公司之核心價值所在，其中最為關鍵的部份乃為貨車提送貨路徑選擇。良好的路徑規劃可減少諸如車輛油耗、司機工作時間等因素所反應之營運成本，亦可增加公司整體貨運量與貨運時間之彈性，以滿足多元化的貨運需求，並進一步提高公司聲譽和顧客忠誠度，對於貨運公司之永續經營乃極為重要之因素。本研究之目的在滿足所有貨運需求、允許提送貨時間窗、貨物材積等限制因素下產生一組最小成本為目標之車輛繞行路徑。同時藉由路徑規劃作業之自動化，提供較佳車輛繞行路徑，降低人工調度負擔。

本研究採用基因演算法，求解具時間窗限制提送貨問題。目標為最小化所有車輛之總旅行距離，以及違反時窗限制的懲罰成本。本研究設計十個大小不同的演算範例進行驗證，同時針對各項因子，如人口數、車輛數、車容量、突變率、突變比例、懲罰乘數以及停止條件為連續演化固定世代，對於求解品質及速度之影響，故針對以上因子進行一系列之敏感度分析。經研究結果可知，本研究可在滿足允許執行時間窗、提送貨任務配對、提送貨優先次序、司機下班時間與車輛載重容量等提送貨特性條件下進行求解，並可適用於各種不同規模的問題。

關鍵詞：時間窗、基因演算法、提送貨

Solving a Pickup and Delivery Problem with Time Windows Constraints

Student : Yunn-Jia Chou

Advisor : Jin-Yuan Wang

Department of Transportation Technology & Management

ABSTRACT

Under the highly competitive circumstance, providing good quality service is the core value of all freight transportation companies. An effective routing plan for pickup and delivery is the most important part contributing to reducing fuel consumption and operation costs. The purpose of this research is proposing an efficient solution algorithm for a real world pickup and delivery problem with time window constraints while minimizing the total vehicles used.

This research adopts Genetic Algorithm for solving the pickup and delivery problem with time windows constraints. The goal is minimizing the total traveling distance and the degree of violating time windows constraints. Ten testing cases with different sizes are generated for testing purpose. We also conduct sensitivity analysis for various factors, such as population, vehicle numbers, vehicle capacity, mutation rate, mutation percentage, penalty weights, and stopping rules. Our testing results show that the proposed algorithm is efficient and sound.

Key Words: Time Windows, Genetic Algorithm, Pickup and Delivery

誌 謝

此時此刻能提筆寫誌謝，要感謝的人真的太多了！首先要感謝指導老師——王晉元老師，每個星期總是撥出些時間細心指導學生的論文，長時間來總是扮演亦師亦友的角色，時時刻刻為學生打氣。同時也要謝謝論文口試老師——中華大學蘇昭銘老師與逢甲大學李克聰老師，於口試前撥冗審閱學生的碩士論文，口試期間提出寶貴建議與未來可繼續研究的方向，使得本論文能夠更加完善與完整。

接著要感謝我摯愛的雙親，從出生至今，父母親無論在精神與物質都是我努力向前衝的最大後盾，使學生能夠安心的挑戰各種難題與困境。在求學期間常因論文或功課，沒有常伴父母左右，所以也要特別感謝父母的全力支持與包容體諒。

最後感謝在交大這六年來陪伴我求學的所有人，不管是老師、學長姐、同學與學弟妹們，有你們的陪伴，讓學生在求學的過程中充滿更多的歡笑與樂趣，這是我一輩子不可抹滅的美好回憶。

謝謝我認識的所有人，有你們的祝福、鼓勵與叮嚀，我才能信心滿滿並堅持的完成論文，希望能與你們一起分享這甜美的碩果，並表達我真摯的愛與謝意。最後僅將此篇論文獻給所有我認識的至親與好友們，再一次由衷的感謝你們！

周韻佳 謹誌

中華民國九十六年六月 于風城交大

目 錄

目 錄.....	i
表 目 錄.....	ii
圖 目 錄.....	iii
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目的.....	2
1.3 研究範圍.....	2
1.4 研究流程.....	2
第二章 文獻回顧.....	5
2.1 具時間窗限制的提送貨問題簡介.....	5
2.2 相關啟發式解法.....	8
2.3 基因演算法之流程.....	11
2.4 小結.....	18
第三章 研究方法.....	20
3.1 本研究所探討的PDPTW問題.....	20
3.2 數學模式.....	21
3.3 演算法設計.....	24
第四章 範例測試.....	34
4.1 產生測試範例.....	34
4.2 測試過程.....	35
4.3 測試結果.....	36
4.4 敏感度分析.....	41
4.5 小結.....	55
第五章 結論與建議.....	57
5.1 結論.....	57
5.2 建議.....	57
參考文獻.....	58
附 錄.....	60
簡 歷.....	91

表 目 錄

表 1.1 研究範圍表.....	2
表 4.1 任務資訊範例表.....	35
表 4.2 距離矩陣表.....	35
表 4.3 範例測試參數設定表.....	36
表 4.4 節點範例總表.....	36
表 4.5 敏感度分析參數設定表.....	41
表 4.6 人口數變動敏感度分析表.....	42
表 4.7 車輛數變動敏感度分析表.....	44
表 4.8 車容量變動敏感度分析表.....	46
表 4.9 突變率變動敏感度分析表.....	48
表 4.10 突變比例變動敏感度分析表.....	50
表 4.11 懲罰乘數變動敏感度分析表.....	52
表 4.12 終止機制變動敏感度分析表.....	54



圖目錄

圖 1.1 研究流程圖.....	4
圖 2.1 混合式提送貨情況一.....	6
圖 2.2 混合式提送貨情況二.....	6
圖 2.3 混合式提送貨情況三.....	7
圖 2.4 單一提貨.....	7
圖 2.5 單一送貨.....	8
圖 2.6 基因演算法流程.....	13
圖 2.7 單點交配法.....	16
圖 2.8 雙點交配法.....	17
圖 2.9 均勻交配法.....	17
圖 2.10 遮罩交配法.....	17
圖 2.11 突變示意.....	18
圖 3.1 巡迴路徑示意圖.....	20
圖 3.2 染色體長度示意.....	25
圖 3.3 染色體簡例示意.....	25
圖 3.4 起始解簡例 (a).....	26
圖 3.5 起始解簡例 (b).....	27
圖 3.6 起始解簡例 (c).....	27
圖 3.7 起始解簡例 (d).....	28
圖 3.8 交配示意 (a).....	29
圖 3.9 交配示意 (b).....	29
圖 3.10 交配示意 (c).....	30
圖 3.11 交配示意 (d).....	30
圖 3.12 交配示意 (e).....	31
圖 3.13 交配示意 (f).....	32
圖 4.1 節點 10 之平均成本趨勢圖.....	37
圖 4.2 節點 10 之最佳成本趨勢圖.....	37
圖 4.3 節點 100 之平均成本趨勢圖.....	38
圖 4.4 節點 100 之最佳成本趨勢圖.....	38
圖 4.5 世代數變化趨勢圖.....	39

圖 4.6 CPU時間變化趨勢圖	39
圖 4.7 起始成本變化趨勢圖	40
圖 4.8 最佳成本變化趨勢圖	40
圖 4.9 CPU時間/世代數變化趨勢圖	41
圖 4.10 人口數變動趨勢圖	43
圖 4.11 車輛數變動趨勢圖	45
圖 4.12 車容量變動趨勢圖	47
圖 4.13 突變率變動趨勢圖	49
圖 4.14 突變比例變動趨勢圖	51
圖 4.15 懲罰乘數變動趨勢圖	53
圖 4.16 終止機制變動趨勢圖 (a)	55
圖 4.17 終止機制變動趨勢圖 (b)	55



第一章 緒論

1.1 研究動機

現今為消費者意識高漲的年代，對於貨物運輸產業而言，客戶必然希冀得到安全、準時、迅捷之貨運服務品質。貨運業者為滿足客戶需求，又面臨同業競爭之壓力，以便在分秒必爭之微利時代從中勝出。如何提供此一迅捷、確實之貨運品質以滿足客戶之貨運需求，即為各貨運公司之重點核心價值所在。其中最為關鍵的部份乃為貨車提貨 (Pickup) 與送貨 (Delivery) 之路徑選擇，此亦為貨運業者最需審慎考量與重視之環節。良好的路徑規劃可減少諸如車輛油耗、司機工作時間等因素所反應之營運成本，亦可增加公司整體貨運量與貨運時間之彈性，以滿足多元化的貨運需求，並進一步提高公司聲譽和顧客忠誠度，對於貨運公司之永續經營乃極為重要之因素。

貨物運送乃為貨運公司利用各種可能之運輸工具 (Transportation Mode) 將客戶所託運之貨物由甲地運送至乙地之一種過程。一般而言，我們通稱甲地為提貨點 (Pickup Node)，乙地為送貨點 (Delivery Node)。貨運公司在考量安全、迅捷等前提下，於所要求之提送貨時間內完成運輸之動作。為此，貨運公司在規劃執行提送貨任務之路徑時，常需考量因素包括提送貨位置、允許提送貨之時間窗、貨物材積等等。同時，貨運公司常會利用混合提送貨之方式以減少不必要之車輛繞行路徑，即滿足客戶服務時允許交叉進行提送貨。目前貨運業者之調度方式多為調度人員人工作業，依據往常之調配經驗決定貨車繞行之路徑，而在無法完善考量諸多繁雜之限制因素的情況下，做出不公平，甚至於不合理的路徑組合，造成資源過度浪費，進而影響公司之信譽以及獲利能力。

一般而言，替一組車輛規劃繞行路徑之問題，稱為車輛途徑問題 (Vehicle Routing Problem, VRP)。每輛車由起始場站 (Depot) 出發，滿足所有需求點，最後再行回到起始場站。一般來說，VRP問題不僅只適用於上述之貨運業，諸如垃圾車、宅配業者、救護車、消防車、計程車等皆適用。傳統VRP問題在國內外已有相當廣泛之研究，但通常皆不考量提送貨時間窗與駕駛人員下班時間等限制因素。有鑑於此，將上述限制納入傳統VRP問題後，即成為具時間窗限制之提送貨問題 (Pickup and Delivery Problem with Time Windows, PDPTW)。本研究將針對PDPTW問題建構一數學模式，並設計一演算法進行求解，以產生一組可行車輛路徑，同時滿足貨運需求與限制，降低車輛繞行之距離與油耗，並提升整體運送效率。

1.2 研究目的

本研究之目的在於考量貨物運送的限制與特性之前提下，針對貨運業之提送貨問題建構數學模式，並設計一套演算法進行求解。所建構之演算法期望能達到以下目的：

1. 在滿足所有貨運需求、允許提送貨時間窗、貨物材積等限制因素下產生一組最小成本為目標之車輛繞行路徑。
2. 藉由路徑規劃作業之自動化，提供較佳車輛繞行路徑，降低人工調度負擔。

1.3 研究範圍

本研究求解貨運業之提送貨問題，其研究範圍整理如表 1.1：

表 1.1 研究範圍表

項目	型態
模式目標	最小化車輛總旅行距離
場站數目	單一場站
車輛種類	多車種，各車容量皆可不同
需求型態	需求點已知且不變
車輛數目	多部車輛，且車輛數目已知且固定
時間窗型態	每一提、送貨點均有允許執行的軟性時間窗限制
提送貨型態	<ol style="list-style-type: none">1. 以空車狀態從場站出發，亦以空車狀態回到場站2. 每一提貨點皆有一相對應之送貨點3. 允許車輛混合提送貨

1.4 研究流程

本研究的研究流程如圖 1.1，茲將流程圖中各步驟詳細說明如下：

1. 描述與界定問題
由研究動機與目的針對貨運車提送貨之路徑問題做完整的描述與範圍界定。
2. 蒐集與回顧文獻

蒐集並回顧國內外探討提送貨問題，與問題解決方法之相關文獻，包括最佳解解法（Exact Solution Method）與啟發式解法（Heuristics Algorithm）等，分析回顧解法之優缺點與適用性，以挑選出較符合本研究提送貨問題範圍之解法，並做適當之改良和修正。

3. 建構車輛提送貨路徑模式

根據現況了解、研究範圍，與相關文獻之蒐集、回顧，建構足以描述車輛提送貨路徑之問題特性與其相關限制條件之數學模式。

4. 設計求解模式之演算法

考量本研究需求，設計一套演算法，以求解車輛提送貨路徑的問題。

5. 撰寫程式

將演算法撰寫成電腦程式語言，以利後續測試與修正。

6. 進行測試範例求解

設計若干測試範例，以驗證演算法設計是否正確且有效率。

7. 分析與評估測試結果

分析求解結果並評估演算法是否良好且有效率。

8. 修正演算法

分析測試結果，若演算法正確且能在可容忍之時間內求解則不需進行修正；反之則須修正甚至重新設計求解模式之演算法。

9. 結論與建議

對本研究之過程與結果提出結論與建議。



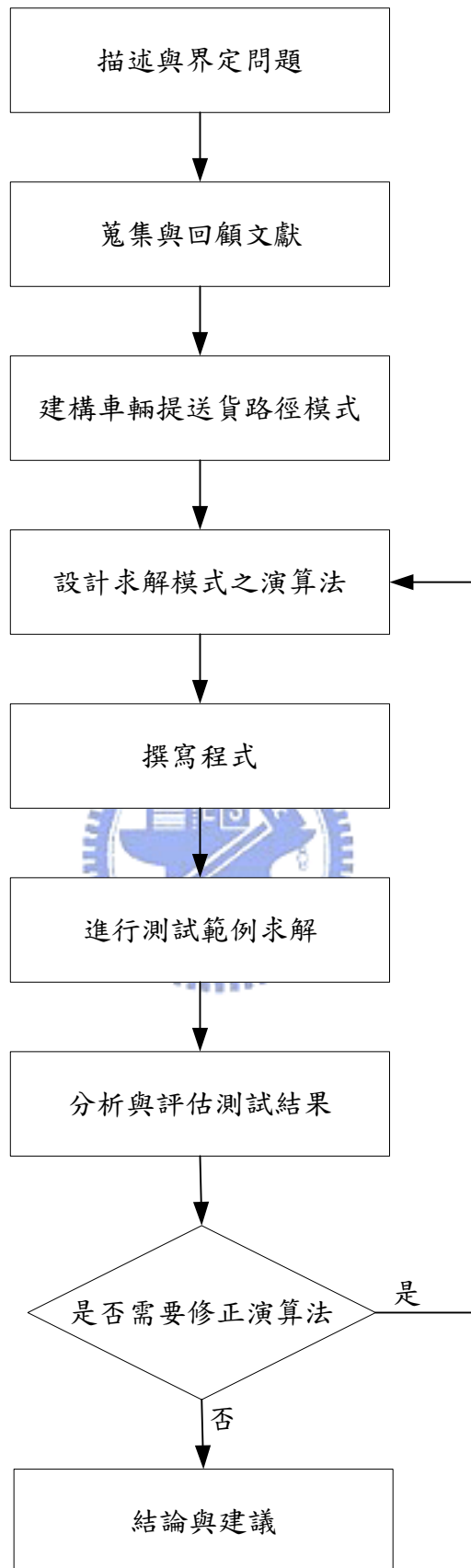


圖 1.1 研究流程圖

第二章 文獻回顧

由於本研究所探討的具時間窗限制的提送貨問題，常面臨需在短時間內快速產生車輛之巡迴路徑，此類型問題其複雜度高，屬於NP-Hard問題，要求出所有車輛路徑以找出最佳解是非常耗費時間。大多數研究針對PDPTW問題皆採用啟發式解法求解，故本研究所進行之文獻回顧乃以啟發式解法為主。2.1 節首先介紹具時間窗限制的提送貨問題簡介；2.2 節為其相關啟發式解法文獻回顧；2.3 節對本研究所欲採用的基因演算法之流程作一詳細介紹；2.4 節為小結。

2.1 具時間窗限制的提送貨問題簡介

具時間窗限制的提送貨問題 (Pickup and Delivery Problem with Time Windows, PDPTW) [1] 乃針對一隊貨車做路徑規劃，在不違反貨車容量的限制下，於指定的時間窗內，讓每輛車行經一連串配對的提貨點與送貨點，以滿足客戶的託運需求。因此，每一提貨點皆有一相對應配對之送貨點，故於PDPTW問題中通常會滿足兩種限制：一為優先限制 (Precedence Constraints)，針對每一貨運需求，提貨任務皆應發生於送貨任務之前；另一為聯結限制 (Coupling Constraints)，每一貨運需求之提送貨任務均由同一輛車執行。

PDPTW問題可分為好幾類，若依貨運需求已知與否，則可分為動態 (Dynamic) 與靜態 (Static) 之提送貨問題[2]；若依車輛種類之異同則可分為單一車種 (Homogeneous Vehicles) 與異種車種 (Heterogeneous Vehicles) 之提送貨問題；若依場站數之多寡則可分為單一場站 (Single Depot) 與多場站 (Multi-Depot) 之提送貨問題。此外，亦有「混合式提送貨問題」與「單一提送貨問題」之分別。混合式提送貨乃在貨車離開場站至回到場站之路途中，允許交叉進行提貨、送貨任務，但前提是同一訂單之提貨任務需在送貨任務之前執行，且貨車離開場站時，須尚有容量方可直接執行提貨任務，如圖 2.1、2.2 所示；單一提送貨則限制貨車在離開場站至回到場站的路途中，對於提、送貨之任務僅能擇一執行，故貨車離開場站時之狀態可能包括尚有容量或載貨等兩種狀態，如圖 2.3、2.4 所示。本研究探討之提送貨問題乃為混合式提送貨，貨車離開場站時皆假設為空車狀態，經執行一連串之提送貨任務，最後再以空車狀態回到場站，如圖 2.1、2.2 所示。

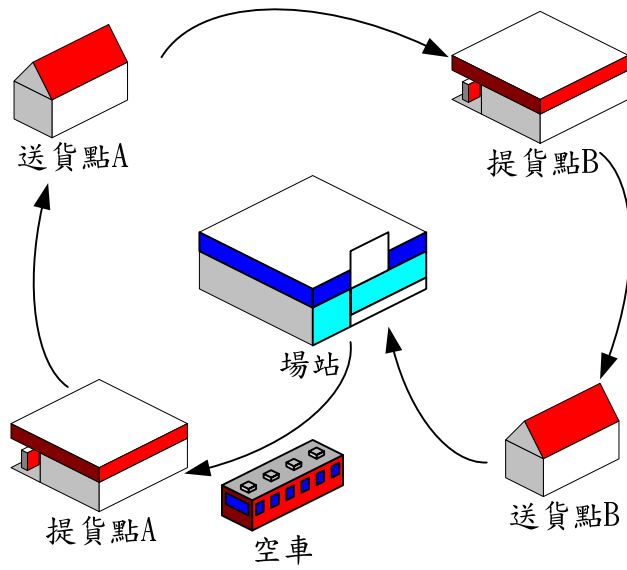


圖 2.1 混合式提送貨情況一

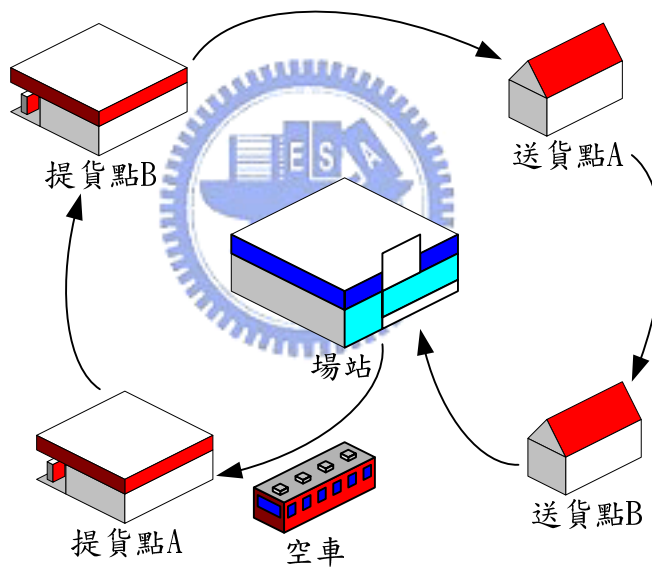


圖 2.2 混合式提送貨情況二

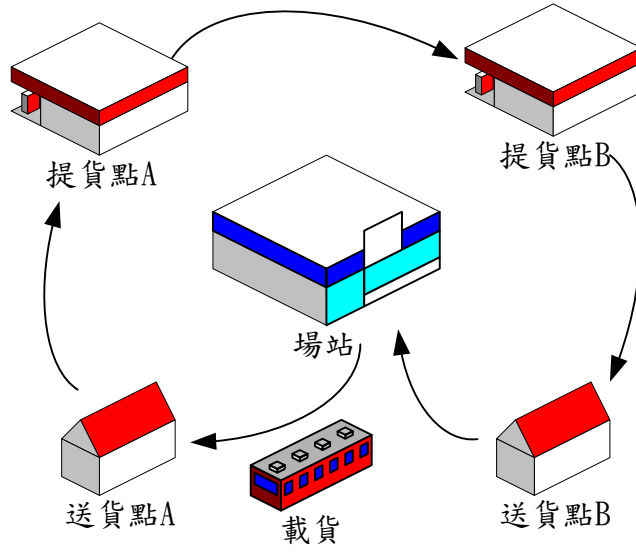


圖 2.3 混合式提送貨情況三

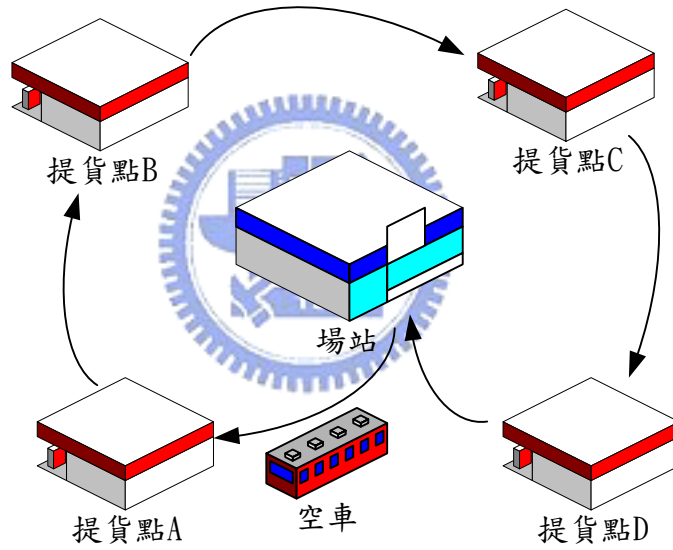


圖 2.4 單一提貨

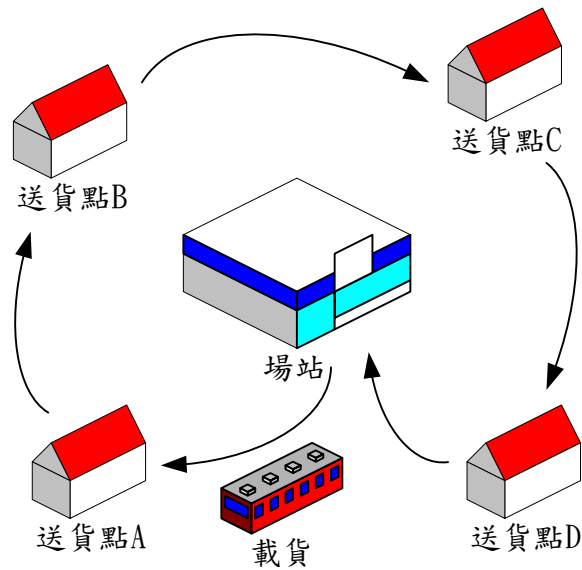


圖 2.5 單一送貨

2.2 相關啟發式解法

近年來啟發式解法在求解應用上之作法，大多先行產生一可行初始解，在其鄰近區域 (Neighborhood) 搜尋可改善目標值之解，此步驟稱為區域搜尋 (Local Search)，為避免僅找到區域最佳解 (Local Optimum)，啟發式解法常有其特殊機制跳脫區域最佳解，至其他未搜尋的解空間進行搜尋，此步驟稱為全域搜尋 (Global Search)，進而增加找到全域最佳解 (Global Optimum) 之機率。

在面臨大型問題或具有時效性問題時，採用啟發式解法雖無法確保必能找到全域最佳解，但若設計得宜，通常可求得品質不錯之近似最佳解。以下針對使用啟發式解法來解決PDPTW問題的文獻作一討論說明：

Nanry and Barnes[1]提出以適應性禁忌搜尋法 (Reactive Tabu Search) 解決PDPTW問題，在滿足優先限制 (Precedence Constraints) 與聯結限制 (Coupling Constraints) 之情況下，使用三種不同的移動交換方式獲得新鄰近解，並且設計一套選擇鄰近解之階級策略，可在三種鄰近解中動態選擇並修正搜尋方向。檢測是否掉入區域最佳解，並以有系統之方式跳出區域最佳解。為驗證其效度，以索羅門 (Slomon) [3]具時間窗限制之車輛路徑問題 (Vehicle Routing Problem with Time Windows, VRPTW) 標竿例題為基礎，改造適用於PDPTW問題之標竿例題進行測試，由測試結果可得解的品質及求解速度皆有一定之水準。

Li and Lim[4]提出以禁忌嵌入式模擬退火法 (Tabu-Embedded Simulated Annealing

Algorithm) 解決PDPTW問題，此為一巨集啟發式解法。目標為最小化使用車輛數、總旅行成本、工作時間與司機人員等待時間的總和，上述四個考量因素均有其相對應權重值，以差異因素之間之相對重要性。為驗證其效度，同樣以索羅門(Slomon)[3]之VRPTW標竿例題為基礎進行測試，由測試結果可得解的品質及求解速度皆有不錯之表現。

Lau and Liang[5]提出以兩階法(Two Phase Method)解決PDPTW問題，目標為最小化使用車輛量與總旅行距離的總和。第一階段提出融合傳統插入法(Insertion Heuristic)與掃描法(Sweep Heuristic)優點之新式構建演算法產生初始解，第二階段使用禁忌搜尋法(Tabu Search)改善目標值，透過三種不同鄰近區域移動方式找尋最佳解。同樣以索羅門(Slomon)[3]之VRPTW標竿例題為基礎進行測試，由測試結果可得解的品質及求解速度皆有優異之水準。

Gronalt et al.[6]探討貨車滿載情況下之PDPTW問題，以空車繞行對於貨物運送上無任何附加價值之角度，企求最小化空車移動距離。首先建構一數學模式，再依據網路流量關係放鬆，即可計算出目標式之下限值(Lower Bound)。提出四種以節省法(Savings Algorithm)為基礎之新式啟發式解法，使用一小型範例詳細說明其步驟流程，並以隨機產生之範例測試新式演算法，由測試結果得知均可快速求解品質良好的解。

Snežana et al.[2]探討動態之PDPTW問題，快遞業者需滿足同天內所有提送貨要求，且貨運需求並非可利用隨機模式所預測。動態PDPTW之標準解法是利用由Psaraftis[7]所提出之滾動時間軸(Rolling Time Horizon)，面臨指派新任務時，需同時考量對於短期與長期時間軸之影響。提出修改插入法(Insertion Heuristic)之雙時間軸(Double-Horizon)啟發式解法解決動態PDPTW問題，其中雙時間軸即同時考量短期與長期。測試結果顯示使用雙時間軸啟發式解法解決動態PDPTW問題可有效求解。

Renaud et al.[8]探討提送貨之旅行銷貨員問題(Pickup and Delivery Traveling Salesman Problem, PDTSP)。提出三種不同擾動啟發式演算法(Perturbation Heuristics)跳脫區域最佳解，分別為Instance Perturbation(IP)、Algorithmic Perturbation(AP)與Solution Perturbation(SP)。測試結果顯示SP解之品質優於AP，AP解之品質又優於IP。

楊馥豪[14]構建一時窗限制下混合收送貨的路線規劃模式，目標為最小化成本。考慮單一物流中心，假設配送車隊為自營、多部車且單一容量車種，需求點座標、收送貨量及服務時間皆為已知。構建模式參考收送貨車輛途程問題(Pick-up and Delivery Vehicle

Routing Problem, PDVRP)，並將時窗限制納入考慮。採用最近鄰點法 (Nearest Neighbor, NN) 及掃描法 (Sweep Algorithm) 求出起始解，配合模擬退火法 (Simulated Annealing, SA) 避免區域最佳解發生。

楊雅斐[15]提出改良式基因演算法求解單一場站、單一車種、無車輛數限制、固定需求點且均加入時間限制之車輛途程問題，其時間窗限制參照宅配業者分為三個時段進行配送服務。使用掃描法和最鄰近法，代入基因演算法之選擇方式，再行重新將路線中節點編碼進行求解；可在短暫時間內找到一較佳解，且確實符合車容量限制。

黃信翔[16]提出以變數產生法 (Column Generation) 求解單一場站、單一車種、無車輛數限制、固定需求點之PDPTW問題，目標為最小化車輛巡迴路徑組合的總成本。將主問題建構為放鬆整數限制之集合分割問題，故視為一線性規劃問題，而子問題則建構為考慮多種提送貨特性之最短路徑問題，以產生可改善主問題目標式之變數，並提出修正後之Dijkstra's演算法來求解子問題。

Jih 和 Hsu[9]發展一混合式基因演算法求解 PDPTW 問題，然其僅能以單一車輛方式進行求解。Jung 和 Haghani[10]提出以基因演算法求解單一場站、多車種且有車輛數限制、固定需求點之 PDPTW 問題。目標為最小化總成本，其中包括車輛固定成本、繞行成本，以及造成客戶不便成本，即違反時窗限制之懲罰成本。套用在非常小規模之虛擬範例中求解，並將求解結果與最佳解加以比較，發現其演算效率極佳。

Schonberger[11]以基因演算法求解提送貨選擇問題 (Pickup and Delivery Selection Problem, PDSP)，其乃將 PDPTW 問題加以衍伸，可選擇接受或拒絕一顧客，目標為最大化整體利潤。基於此一選擇機制，問題焦點已著重在利潤獲得之多寡，與 PDPTW 問題已有差異。

Barrie 和 Ayechev[12]混合基因演算法和鄰域搜尋法 (Neighborhood Search Method) 求解單一場站和旅行距離限制之 VRP 問題。與傳統基因演算法、禁忌搜尋法、模擬鍛鍊法 (Simulated Annealing) 等方法相互比較，發現在求解速度與品質等方面超越傳統基因演算法，並與後兩者不相上下。

Giselher[13]發展以群組化之基因演算法 (Grouping Genetic Algorithm, GGA) 求解單一場站、單一車種、無車輛數限制、固定需求點之 PDPTW 問題。目標為最小化車輛總

旅行距離。有別於其他研究中單一基因代表單一顧客之表示方法，改以單一基因表示一組顧客，即單一染色體表示單一解之方式編碼。將 GGA 建構在兩套標竿試題中，並與 Nanry[1]、Li[4]所發展之模式加以比較，演算結果顯示 GGA 與上述兩者之模式皆非常有效率。

2.3 基因演算法之流程

本小節首先針對基因演算法中所使用之名詞作一解釋，後續則針對演算法中各流程進行說明與圖解。

1. 染色體 (Chromosome)：演算法求解單位。
2. 基因 (Genetic)：構成染色體之單位。
3. 族群 (Population)：目前所有染色體之集合。
4. 世代 (Generation)：演化的次數別。
5. 母代 (Parents)：為候選交配之染色體。
6. 子代 (Offspring)：為交配後之染色體。
7. 交配點 (Crossover Point)：為進行基因值交換之染色體位置。
8. 適合度 (Fitness Function)：用以計算染色體品質之數值。

基因演算法 (Genetic Algorithms) 之理論基礎源自於達爾文進化論「物競天擇」、「適者生存，不適者淘汰」(Survive of the first) 之觀念。解釋自然界中的基本現象，物種在不斷變遷或惡劣的環境中為了生存及適應環境，而不斷競爭且進化，繁衍出生存力及適應力更強的下一代。生物在繁殖的過程中染色體會進行交配及突變來改變基因的組成，使得子代和親代及子代之間產生差異性。生物就是利用這種繁殖的機制，造成無數的變異發生，使得每一個個體有著不同的特性，而在自然環境的考驗及彼此之間的競爭下，將不適生存者淘汰，而生存者繼續繁殖，不僅將優良的基因延續，還有可能使下一代擁有更好的基因。

基因演算法 (Genetic Algorithms) 簡稱 GA，有人稱之遺傳演算法或者基因遺傳演算法，可以追溯到 1950 年代，生物學者與電腦學者合作，嘗試在電腦上模擬出基因的

運作。緊接著在 1960 年代初期，密西根大學的 John Holland 和他的同事們將電腦化的遺傳學—染色體、基因、配對基因、適配函數等—應用到其他許多領域。1967 年，一位 John Holland 的學生 J.D. Bagley 在其畢業論文中首度創用了「基因演算法」這個名詞，來描述這種最適化的技術。然而，許多學者質疑，因為基因演算法相當依賴隨機選擇，而這些選擇是任意且不可預測的。開啟遺傳演算法大門的 Dr. John Holland，以 1975 年在密西根大學發表闡述自然以人工系統的適應問題著作，該書總結了賀藍博士二十年來對於學習、演化與創作的研究心得，此文奠定了遺傳演算法的發展基礎與思考架構。基因演算法可選擇物種中較為優良之母代，透過交配隨機交換位元資訊，以期產生更加優異之子代，再藉由突變機制，預防流失優良因子與不良因子常駐，如此重複演化，逐漸逼近適應度最強之物種，其演化流程如圖 2.6 所示。



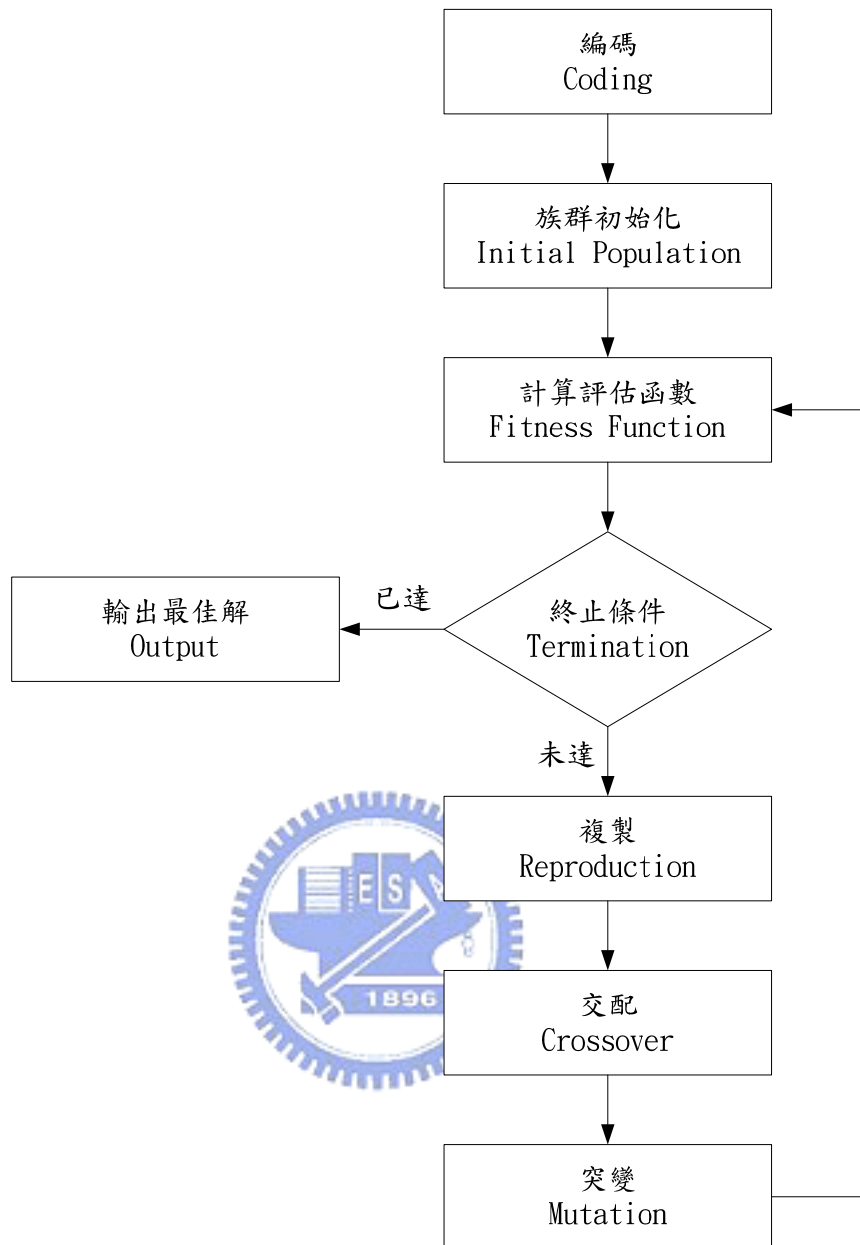


圖 2.6 基因演算法流程

以下就各步驟相關內容加以探討並分別說明：

1. 編碼 (Coding)：

編碼係將求解問題中之變數轉換為字串，此字串即為生物學中的染色體。其資料型態可以包含二元 (Binary) 編碼、整數 (Integer) 編碼、實數 (Real-Number) 編碼、文字編碼及符號編碼等多種型態，在一個染色體中通常只單獨使用一種資料型態，但也可以混合使用各種型態。各種資料型態中又以二元編碼最為常用，假若染色體之編碼長度為 L ，則所形成之編碼解之表達空間為 2^L 。但事實上其他各種資料型態也都可以轉換為

二元編碼，因為在電腦中無論任一編碼型態，內部最終一定會轉換成二進位來處理。只是若以二進位資料型態取代其他資料型態時需考慮是否會改變解答空間的大小。

事實上很多狀況下解答空間中可行解的數量未必是二的整指數次方個，此時就會因編碼表達空間 > 解答空間，也就是編碼解數量 > 可行解數量，此時會造成不可行解的產生。基因演算法在處理非法解時主要有兩類方式，其一透過可行解加分之獎勵方式，或為不可行解扣分之懲罰方式。其次則是將多個編碼解對應到同一個可行解，讓編碼表達空間等於解答空間以消除非法解。兩者皆會有副作用，必需謹慎使用。

基因演算法在編碼及解碼上耗費相當之時間，尤其當系統參數增加時，編碼及解碼之運算將嚴重減緩搜尋程式執行的速度。編碼所使用之字串長度若不夠時將可能使基因演算法雖然搜尋到系統整體最佳解出現的區域，但是由於精確度不足的關係只能搜尋到最佳解附近，而無法真正搜尋到最佳解。因此可知位元長度將影響解析度，位元數越多則所求之參數值越精確，但所需之編碼與解碼時間也相對提高，故應視問題需求而決定位元長度。二元編碼主要是將十進制變數轉換為 0 與 1 的二位元字串組合，整數編碼則視問題需求將問題中具表徵性之數值作為編碼依據。然而使用整數編碼須特別留意資料間之關連性，以避免發生資料無法有效比對之情形，而造成演算上的錯誤產生。

2. 族群初始化 (Initial Population):

初始族群為第一次演化之親代，通常初始族群的成員為隨機產生，其族群數量亦為演化過程中的族群數量。族群中染色體亦稱為個體 (Individual)。初始族群之意義就是在基因演算法的運算之初，利用隨機的選擇讓族群中的基因能就有多樣性，同時使得搜尋能具有隨機和多點的特性。

族群大小則可依據問題之複雜程度而定，族群數量之大小影響演化搜尋之範圍，族群大代表有較大機會找到最佳解，唯需較長之搜尋時間與空間；反之族群小則搜尋時間短，雖然求解快，但有可能產生提早收斂導致搜尋結果不佳之情形，即容易陷入區域最佳解之陷阱。

3. 評估函數 (Fitness Function):

評估函數用來評估族群中每一個體之適應力，依系統所設定目標值之最大化或最小化所得適合度值來評估子代優劣與否之依據。在達爾文的進化論中提到「物競天擇，適

者生存」的基本概念，將其應用到基因演算法中，個體的評估函數值越高，表示個體的適應能力越高，競爭力越強，則生存的機會也愈大，而可能將本身的基因遺傳到下一代身上之機率愈高。在演化的過程中所生存下來的個體特性會隨著所選擇的評估函數不同而變化，因此可藉由設定不同之評估函數達到控制演化生態環境的方式。

常見適合度函數有目標函數 (Objective Function)、成本函數 (Cost Function)、性能指標 (Performance of Index; PI)、誤差平方函數等等。為了將基因演算法應用到實際層面的問題上，必須將最佳化的問題及目標適當的以評估函數來表達。當最佳化問題是屬於有條件限制時，通常會使用懲罰函數 (Penalty Function) 之概念來使最佳化結果滿足限制條件。懲罰函數之原理在於處罰落入不合理區域的解，將其評估函數值變小，在演化過程中自然淘汰。

4. 終止條件 (Termination):

終止條件為停止繼續演化之機制，一般而言分為以下三種：

- (1) 適合度值在演化過程中皆無法有效改進。
- (2) 演化已達預設最大世代數或指定代數時。
- (3) 適合度值已達預設需求目標。

因此若符合上述條件時，基因演算法即將目前世代輸出產生結果，否則繼續進行複製、交配、突變、與適合度值計算等程序。

5. 複製 (Reproduction):

由於生物演化之自然法則為具有適應力較強之物種，其優良基因容易持續繁衍下去。因此，基因演算法當中的複製步驟即為天擇之角色。根據每一個體的評估函數值高低來決定該個體被複製之機率。評估函數值愈高之個體有愈大的機會被選擇，進而複製出子代的新個體。淘汰不適合之個體，使得下一世代的族群評估函數值會提高。在此需注意設計使演算法適時收斂之機制，避免收斂太慢而降低在指定世代內求得最佳解之機率；或收斂過快而陷入區域最佳解之陷阱。一般常用之選擇機制如下：

- (1) 輪盤法 (Roulette Wheel Selection): 係將目前族群之適合度值予以加總，並視各染色體適合度值佔總值之比例切割輪盤面積大小，最後於輪盤上隨機決定一

點，該點落入何者面積範圍內即被選擇，因此適合度值越高者所佔面積亦越大，所以被選到機率愈高。

- (2) 競爭法 (Tournament Selection)：係隨機選取兩個或多個染色體，將其中具有最高評估函數值者複製至下一代，評估函數值越高者愈容易被選中。競爭法其計算量少，並可透過所選取數量控制競爭速度。
- (3) 排序法 (Rank Selection)：使用排序法可避免輪盤法常因某幾條染色體其評估函數值過大，占去大部分輪盤面積，造成其選取機率過大，導致過早收斂於局部最佳解之窘境。排序法是依染色體評估函數值優劣名次進行排序，被複製機率是依據排序之順序，而非染色體本身評估函數值之大小比例，故排序結果呈線性關係。

6. 交配 (Crossover)：

交配係進行兩組染色體間基因值交換之動作，隨機選擇兩組染色體互換基因，進而組成子代染色體的運算過程。並非所有被選中之母代個體都要進行交配，而是由預先設定之交配機率 (Probability of Crossover) 來決定是否進行資訊交換。

高交配率可使演算法容易獲得新的子代，因此可增加求得最佳解之機率，不過可能導致族群中原優良之親代被取代速度比新子代演化速度快。換言之，交配率低，將使演化速度緩慢，不過由於所交配之染色體係經過選擇程序所篩選，因此族群中所留下之親代本來就有較佳之適合度值，如此才使基因演算法之效率有別於一般隨機搜尋法。一般交配方式有單點交配 (Single-Point Crossover)、雙點交配 (Two-Point Crossover)、均勻交配 (Uniform Crossover) 與遮罩交配 (Mask Crossover) 四種，茲其內容說明如下：

- (1) 單點交配：如圖 2.7 所示，係於候選交配親代之染色體，隨機決定一個交配點 (Crossover Point)，並直接交換該點相對應位置上之基因資訊值。

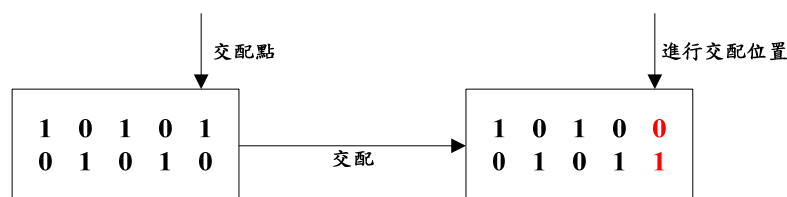


圖 2.7 單點交配法

- (2) 雙點交配：如圖 2.8 所示，係於候選交配親代之染色體，隨機決定兩個不同之交配點，並交換包含兩位置與其中間所有位置上之基因資訊值。

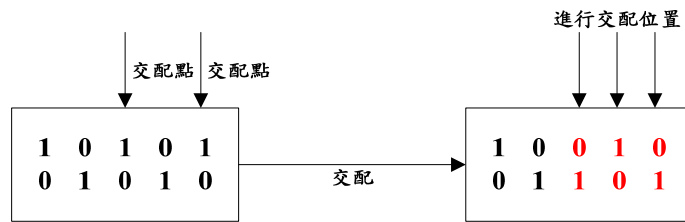


圖 2.8 雙點交配法

- (3) 均勻交配：如圖 2.9 所示，係指候選交配親代之染色體，每個基因都是交配點，且其交配機率獨立且均等。均勻交配法在交配時雖不受到染色體編碼位置影響，但可能使母代染色體架構受到大量破壞，而無法保留母代染色體之特性。

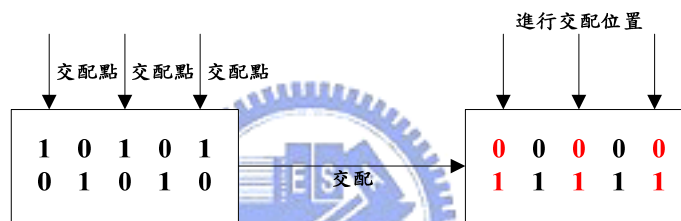


圖 2.9 均勻交配法

- (4) 遮罩交配：如圖 2.10 所示，係隨機產生一組與親代染色體長度相同之二元字串，稱之為遮罩 (Mask)。檢查遮罩字元值為 1，則交換親代相對位置上之基因資訊值；遮罩字元值為 0，則不須進行任何動作。

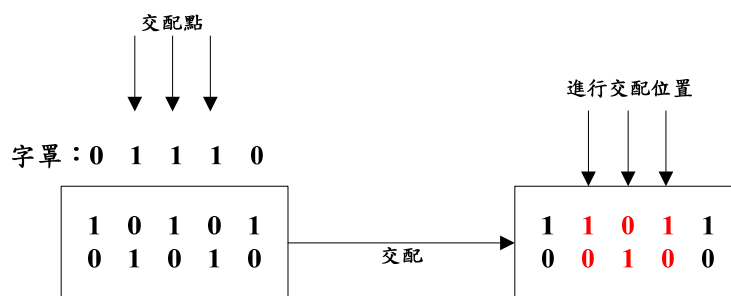


圖 2.10 遮罩交配法

7. 突變 (Mutation)：

僅靠交配與複製並不能夠使得演化造出一個具有新特性之個體。因為這些個體僅繼

承親代的種種特性，所以使得演化速度過於緩慢，同時使得演化結果侷限在特定少數區域上。在自然界中，生物藉著突變來造出新的特種。在高等生物中，由於結構愈趨複雜，因此突變往往對於高等生物是有害的。然而，在基因演算法中，仍然藉著自然界中的這種運算，來增加族群之多樣性，同時避免發生過早收斂於次佳解或得到區域最佳解的情形。

突變係隨機直接改變某個子代染色體任一位置之基因值，目的在產生新型態的染色體，藉以跳脫目前的現解空間避免得到區域最佳解，增加新的搜尋區域以增加尋得全域最佳解之機率。突變程序於演算法過程中以突變率（Mutation Rate）控制其操作，通常其設定值都很低。運作情形如圖 2.11 所示。

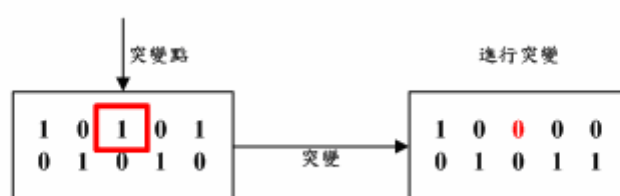


圖 2.11 突變示意

2.4 小結

本研究探討的PDPTW問題，貨運業者常需在極短時間內即時產生車輛巡迴路徑供調度人員使用，且此類型問題其複雜度高，屬於NP-Hard，若要求出所有車輛路徑找出最佳解非常耗費時間。據所收集之文獻可知，大多數研究針對PDPTW問題皆採用啟發式解法求解，以求得品質優良之近似最佳解。故本研究採用啟發式解法以有效率且快速的針對大規模之問題求得可行解。

使用單純之啟發式解法，在解空間中要找到全域最佳解機率甚低，並且容易陷入區域最佳解之陷阱，若欠缺完善之跳脫機制則將造成嚴重影響缺失。近年來，眾多研究領域皆利用基因演算法進行求解，皆有不錯之成果與表現。為此，本研究選擇以基因演算法進行PDPTW問題之求解。

以基因演算法求解PDPTW之問題，藉由文獻回顧可得知以往只能對單一車輛方式進行求解[9]；或是可對多車輛進行求解，卻無限制車輛數上限之前提[13]；或是只有單一車種，無法支援多車種容量之假設[13]；或是只能應用在小規模之問題才会有演算效率優良之情形[10]。基於以上所述，本研究採用基因演算法求解PDPTW問題，並針對

前人研究之部分限制條件予以放鬆，使其更加適配於現實狀況，以符合營運業者之規劃使用。



第三章 研究方法

基因演算法具有深度搜尋的特性，並且具有跳脫區域最佳解之突變機制，以期在演化中提高納入較多可能組合數之機率。近年來許多應用基因演算法為基礎之排班問題亦有不錯的結果呈現，故本研究擬以基因演算法為基礎，進行求解具時間窗限制的提送貨問題。3.1 節簡介本研究所探討之 PDPTW 問題；3.2 節為數學模式；3.3 節為演算法設計。

3.1 本研究所探討的 PDPTW 問題

PDPTW 問題常會使用以下名詞，茲說明如下：

1. 任務：貨運公司接獲託運者之需求訂單後，即產生提貨與送貨之任務配對。貨運公司必須派車前往提貨地點，並在指定之提貨時間窗內執行完畢其提貨任務，之後再行前往送貨地點並在指定之送貨時間窗內執行完畢其送貨任務，如此即完成此筆需求訂單。在位置表示上，提貨點即表示一提貨任務，送貨點即表示一送貨任務[5]。
2. 巡迴路徑：每輛貨車從場站出發後執行一連串提送貨任務，最終再回到場站的任務鏈即稱為巡迴路徑，簡稱為路徑。如圖 3.1 所示，貨車從場站出發後其行走之巡迴路徑為場站 → 提貨任務 A → 送貨任務 A → 提貨任務 B → 送貨任務 B → 場站 [5]。

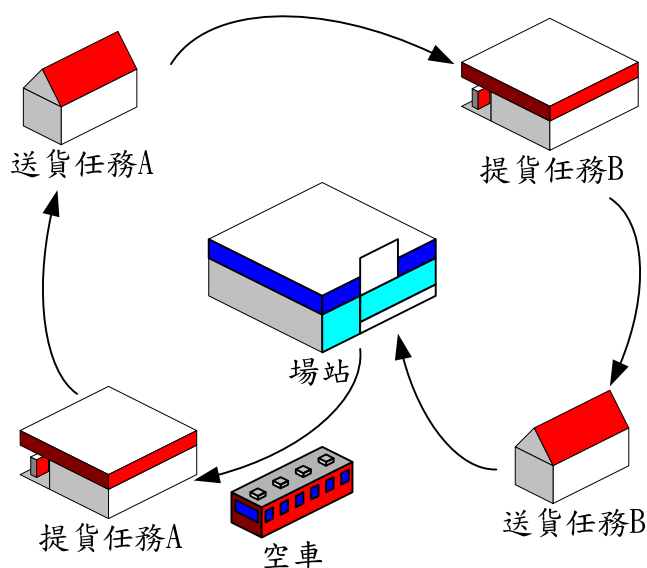


圖 3.1 巡回路徑示意圖

3. 優先限制 (Precedence Constraints): 針對每一貨運需求, 提貨任務皆應發生於送貨任務之前。

4. 聯結限制 (Coupling Constraints): 每一貨運需求之提送貨任務均由同一輛車執行。

本研究所探討的 PDPTW 問題具有以下特性, 茲將歸納如下所示:

1. 本 PDPTW 問題之成本僅考慮距離, 故求解目標為最小化車輛總旅行距離。
2. 僅考慮單一場站。每輛車均以空車狀態從場站出發, 亦以空車狀態回到場站
3. 多車種, 各車容量可有所差異。
4. 多部車輛, 且車輛數上限為已知。
5. 每輛車均為混合式提送貨, 如 2.1 小節中所描述。
6. 司機人員有預定下班時間, 此即為貨車可執行任務之最晚時間。
7. 貨運需求已知且不變, 亦即提貨任務與送貨任務為已知。
8. 每個貨運需求即代表一組提貨與送貨的配對, 故每一提貨任務皆有一相對應之送貨任務。
9. 每個任務均有貨物載重材積, 若為提貨任務則其貨物載重材積為正值; 若為送貨任務則其貨物載重材積為負值。
10. 每一提、送貨點均有允許執行的時間窗限制, 車輛僅能在此時間窗內執行任務。若於時間窗前到達, 則須等待至最早允許執行任務之時間點, 如此將產生等待時間。反之, 若於時間窗內到達, 則可直接執行任務而不產生任何等待時間。

3.2 數學模式

本研究求解 PDPTW 問題, 其時窗限制乃為軟時窗, 針對太早或太晚滿足需求點皆給予一定之懲罰成本。本研究可指定所求解最多可使用之車輛數或根據固定的車輛數求解, 所發展之模式如下:

$$\text{Min } Z = \sum_{i \in TS} \sum_{j \in TE} (x_{ij} \times d_{ij}) + Pw \sum_{i \in Td} w_i + Pd \sum_{i \in TD} v_i$$

$$\text{S.T. } \sum_{i \in TS} x_{ij} = 1 \quad ; \quad j \in TD \quad (1)$$

$$\sum_{j \in TE} x_{ij} = 1 \quad ; \quad i \in TD \quad (2)$$

$$p_i^k - p_j^k \leq M(1 - l_{ij}) \quad ; \quad i \in PL, j \in PD, k \in NV \quad (3)$$

$$p_i^k - p_j^k \geq M(l_{ij} - 1) \quad ; \quad i \in PL, j \in PD, k \in NV \quad (4)$$

$$p_i^k - p_j^k \leq M(1 - x_{ij}) \quad ; \quad i \in TS, j \in TE, k \in NV \quad (5)$$

$$p_i^k - p_j^k \geq M(x_{ij} - 1) \quad ; \quad i \in TS, j \in TE, k \in NV \quad (6)$$

$$\sum_{k \in NV} p_i^k = 1 \quad ; \quad i \in TD \quad (7)$$

$$t_i \leq t_j + M(1 - x_{ij}) \quad ; \quad i \in TS, j \in TE \quad (8)$$

$$t_i \leq t_j + M(1 - l_{ij}) \quad ; \quad i \in PL, j \in PD \quad (9)$$

$$w_i \geq a_i - t_i \quad ; \quad i \in TD \quad (10)$$

$$v_i \geq t_i - b_i \quad ; \quad i \in TD \quad (11)$$

$$qq_i + q_i \leq qq_j + M(1 - x_{ij}) \quad ; \quad i \in TS, j \in TE \quad (12)$$

$$qq_i \leq \sum_{k \in K} (p_i^k \times Q_k) \quad ; \quad i \in TT \quad (13)$$

$$t_i \leq JOB \quad ; \quad i \in TT \quad (14)$$

符號定義：

決策變數：

t_i ：需求點 i 之到達時間， $i \in TT$

w_i ：需求點 i 之等待時間， $i \in TD$

v_i ：需求點 i 之延遲時間， $i \in TD$

qq_i : 至需求點 i 前之累積載運量, $i \in TT$

p_i^k : 若需求點 i 被車輛 k 服務, 則其值為 1, 反之為 0。 $i \in TD, k \in NV$

x_{ij} : 若車輛先後服務需求點 i 及 j , 則其值為 1, 反之為 0。 $i \in TS, j \in TE$

參數 :

Pw : 早到懲罰乘數

Pd : 晚到懲罰乘數

d_{ij} : 需求點 i 至 j 之距離, $i \in TS, j \in TE$

l_{ij} : 若節點 i 與 j 為相對應之提送貨點, 則其值為 1, 反之為 0。 $i \in PL, j \in PD$

q_i : 節點 i 之需求量, $i \in TE$

Q_i : 車輛 i 之車容量, $i \in NV$

JOB : 工作結束時間

(a_i, b_i) : 需求點 i 之時間窗, $i \in TD$

M : 極大值



集合 :

NV : 車輛集合

PL : 提點集合

PD : 送點集合

TD : 需求點集合

TE : 除起點外, 所有點之集合

TS : 除迄點外, 所有點之集合

TT : 所有點集合

目標式在於最小化所有車輛之總旅行距離, 以及造成客戶不便 (違反時窗限制時) 懲罰成本。式 (1) 和式 (2) 確認每個需求點僅被一輛車繞行過一次; 式 (1) 限制只能有一輛車進入該需求點, 式 (2) 限制只能有一輛車離開該需求點。基於聯結限制之前提, 式 (3) 和式 (4) 確認同一任務之提送點皆由同一輛車負責。式 (5) 和式 (6)

確認若兩需求點相鄰，則此二點須由同一輛車負責，否則將會產生矛盾之情形。因每筆訂單皆需被滿足，故以式(7)來限制每一個需求點一定要被一輛車服務過。式(8)確認若兩需求點相鄰，到達前方節點之時間需早於後方節點，否則將會產生時間上的矛盾情形。基於優先限制之前提，式(9)確認每一提貨點必須在相對應之送貨點之前被服務。本研究為軟時窗限制，故需針對違反時窗限制之情形給予懲罰；式(10)針對較最早時間滿足需求點給予造成客戶不便(違反時窗限制時)懲罰成本；式(11)針對較最晚時間滿足需求點給予造成客戶不便(違反時窗限制時)懲罰成本。式(12)和式(13)為車輛容量之限制，以確保各車輛在滿足需求點時，不致產生超過容量限制之情形。式(14)確認每一輛車皆須在工作天結束前回到場站。

3.3 演算法設計

2.3 小節已針對基因演算法中各流程進行說明與圖解，在此列出以基因演算法求解 PDPTW 問題須特別考量之處，並逐一討論如下：

1. 評估函數 (Fitness Function)

本研究之評估函數為計算總車輛之總旅行距離。若違反時間窗之限制，則以懲罰函數 (Penalty Function) 之概念來表示。針對此類型違反時間窗限制之解，將其評估函數值變大，使其在演化過程中淘汰，進而使最佳化結果滿足限制條件。

本研究求解 PDPTW 問題，其時窗限制乃為軟時窗，針對太早或太晚滿足需求點皆給予一定之懲罰成本。評估函數之目的在於最小化所有車輛之總旅行距離，以及造成客戶不便(違反時窗限制時)懲罰成本。

2. 編碼原則 (Coding)

本研究採用整數 (Integer) 編碼，其架構為一染色體代表一組解，即一染色體表示多車之提送貨繞行路徑，一基因代表單一提(送)貨點。假若目前有 N 台服務車輛，為每一車輛都先預留 M 個點之繞行路徑，故染色體之長度為 $N \times M$ ，如圖 3.2 所示。

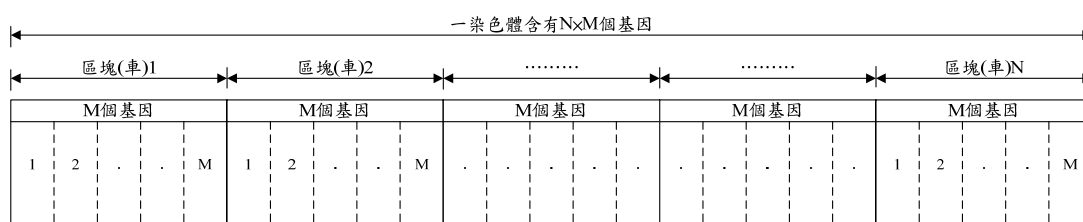


圖 3.2 染色體長度示意

本研究以基因位置判斷該車之繞行服務提送貨點之順序，在基因值的部份填入提送貨點之節點編號。舉一簡例而言，現今有五對提送貨點，其節點編號分別為 1 到 10，並有三輛服務車輛 A、B、C。由下圖 3.3 所示，此染色體代表一組解，其意義為 A 車繞行路徑順序為節點 1、3、5、7；B 車繞行路徑順序為節點 2、4、6、8；C 車之繞行路徑順序為節點 9、10。

A車				B車				C車									
1	3	5	7	0	0	2	4	6	8	0	0	9	10	0	0	0	0

圖 3.3 染色體簡例示意

3. 產生起始解 (Initial Population)

由於 PDPTW 問題須滿足優先限制 (Precedence Constraints) 與聯結限制 (Coupling Constraints)，意即同一訂單的提貨任務須優先於送貨任務執行，且每筆訂單的提貨與送貨任務均由同一輛車執行。故在取決起始解時，就需考慮以上前提限制。

(1) 各提貨任務必有一相對應之送貨任務，故將提送貨點先行配對。假設有 N 輛服務車輛，故每輛車服務第一對提送貨點之機率皆為 $\frac{1}{N}$ 。

(2) 機率 P 值為介於 0 到 1 之均勻分配隨機亂數，若 $0 < P \leq \frac{1}{N}$ ，該對提送貨點由第一車服務；若 $\frac{1}{N} < P \leq \frac{2}{N}$ ，該對提送貨點由第二車服務；同理，以此類推， $\frac{(N-1)}{N} < P \leq 1$ ，該對提送貨點由第 N 車服務。

(3) 假若第一對提送貨點由第一車進行服務，該車機率變為 0，其他車之機率變為 $\frac{1}{(N-1)}$ ，再如上述之步驟決定第二對提送貨點交由何車服務，此時若 $0 < P \leq \frac{1}{(N-1)}$ ，該對提送貨點由第二車服務；若 $\frac{1}{(N-1)} < P \leq \frac{2}{(N-1)}$ ，該對提送貨點由第三車服務； $\frac{(N-2)}{(N-1)} < P \leq 1$ ，該對提送貨點由第 N 車服務。

- (4) 重複以上步驟直至每車皆服務一組提送貨點，則每台車之機率皆恢復至 $\frac{1}{N}$ ，此循環過程會結束於每一對提送貨點都分派至其服務車輛。
- (5) 當每一對提送貨點都分派至其服務車輛，即產生一組起始解，重複上述步驟，因 P 值為介於 0 到 1 之隨機亂數，故可產生多組起始解。

舉一簡例而言，現今有四對提送貨點以及三輛台服務車輛 A、B、C。當第一對提送貨點進行分派時，三車之服務機率皆為 $\frac{1}{3}$ 。假若其隨機亂數為 0.2，則第一對提送貨點交由 A 車進行服務，如圖 3.4 所示。

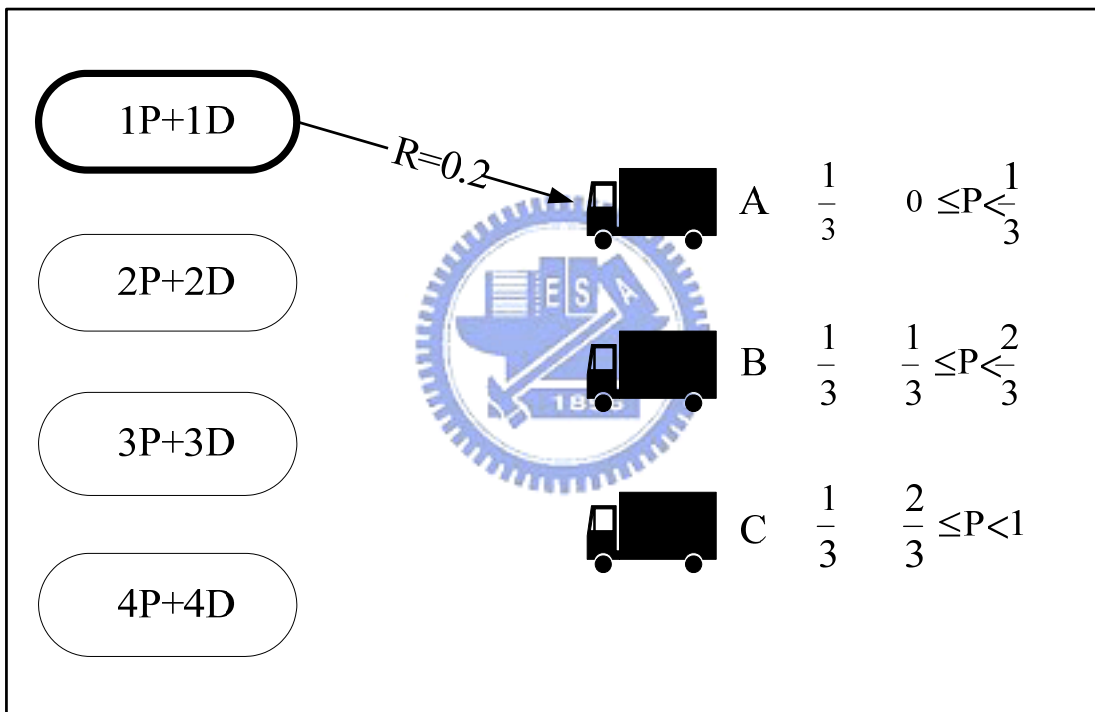


圖 3.4 起始解簡例 (a)

此時，A 車之服務機率降為 0，B、C 兩車之服務機率提高為 $\frac{1}{2}$ 。假若其隨機亂數為 0.7，則第二對提送貨點交由 C 車進行服務，如圖 3.5 所示。

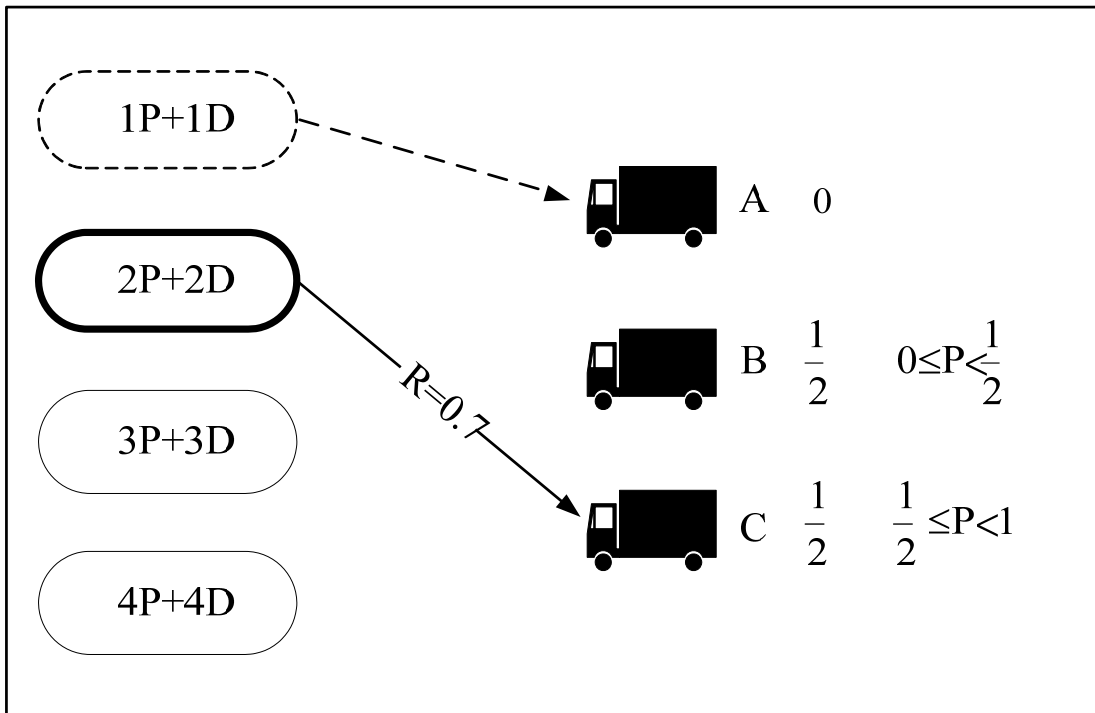


圖 3.5 起始解簡例 (b)

此時，C 車之服務機率降為 0，B 車之服務機率提高為 1，如圖 3.6 所示。

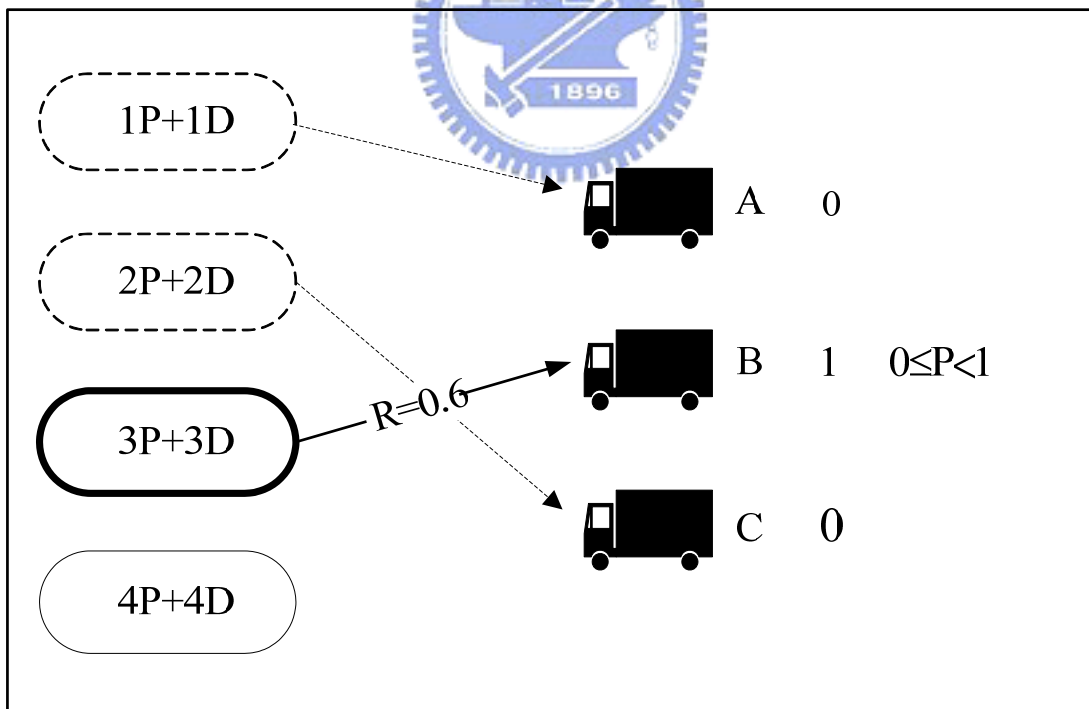


圖 3.6 起始解簡例 (c)

因三輛車皆服務過一對提送貨點，故三車之服務機率恢復至 $\frac{1}{3}$ 以重新接受分派。假若其隨機亂數為 0.1，則第四對提送貨點交由 A 車進行服務，如圖 3.7 所示。

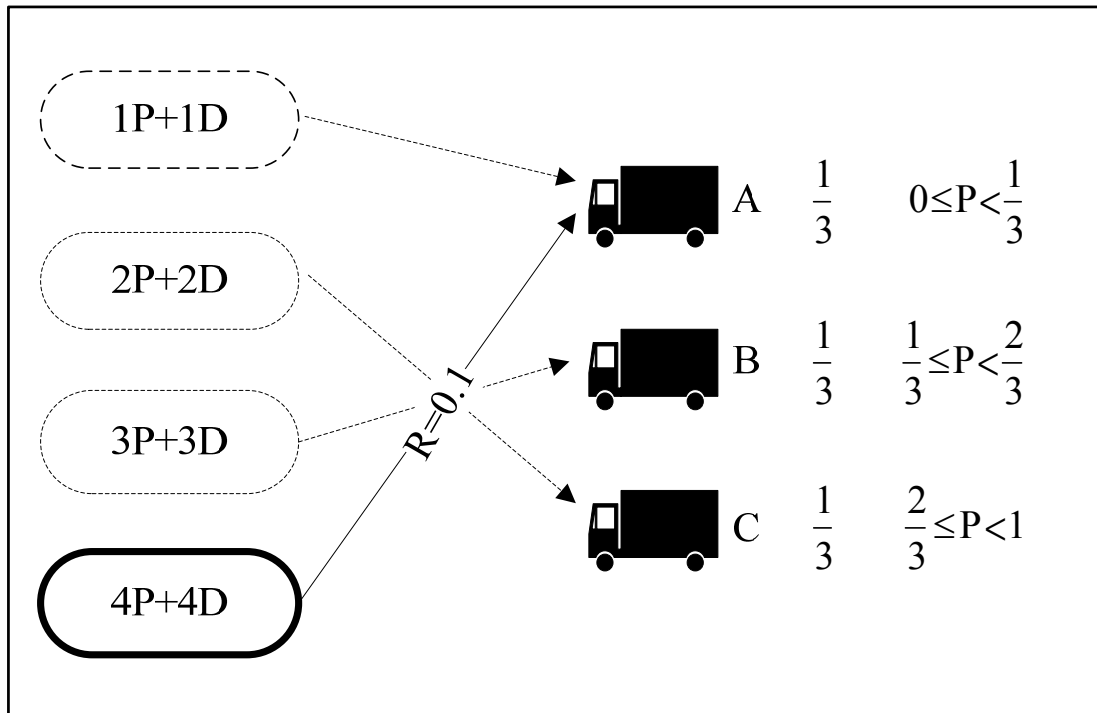


圖 3.7 起始解簡例 (d)

4. 交配方式

詳細交配步驟如下：

- (1) 由於本研究之評估函數為最小化所有車輛之總旅行距離，故適合度值越小，則有較高被挑選之機率。首先將各染色體適合度值轉為倒數，並將其倒數比化簡為最簡整數比，進而採用輪盤法 (Roulette Wheel Selection) 選擇欲進行交配之兩條染色體 Parent1 和 Parent2。

A. 舉例來說，三條染色體之適合度值分別為 1、2、3，其倒數比為

$$\frac{1}{1} : \frac{1}{2} : \frac{1}{3} = 6 : 3 : 2。採用輪盤法，三條染色體分別被選取之機率為 \frac{6}{11}、\frac{3}{11}、\frac{2}{11}。$$

- (2) 本研究一次選擇兩條染色體 Parent1 和 Parent2，進行單點交配。隨機產生一交配點，交配點將兩條母代染色體分為前後兩段，再分別將其後段染色體互換，亦即 Parent1 保留交配點前段之基因值，配上 Parent2 交配點後段之基因值成為一條新染色體 Child1。同理，Parent2 保留交配點前段之基因值，配上 Parent1 交配點後段之基因值成為一條新染色體 Child2。

(3) 承上，又可將其交配點分成以下兩種情況來討論，如圖 3.8 中所示。交配點落在區塊內之虛線，亦或是區塊與區塊之實線。在本研究中，一輛車代表一區塊；實線代表車輛與車輛間之間格；虛線代表單輛車中載運需求點之間格。

A. 若交配點落在區塊內之虛線，如圖 3.8 所示：

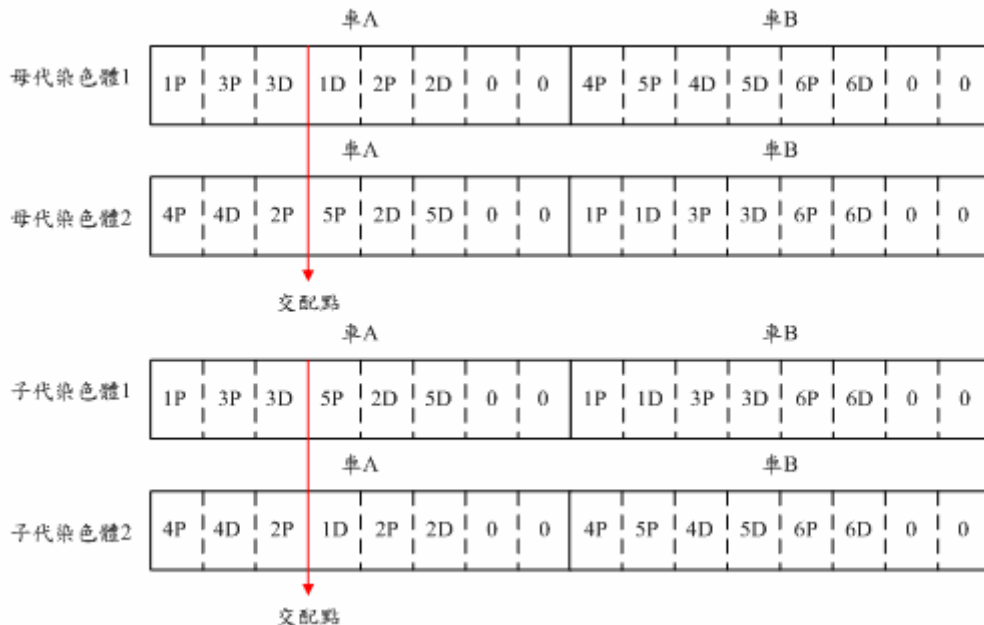


圖 3.8 交配示意 (a)

a. 檢視其區塊內，交配點後若出現與交配點前重複之基因，將後方重複之基因刪除。以圖 3.9 為例，子代染色體 2 之交配區塊車 A 中，其交配點後出現與交配點前重複之基因 2P，將交配點後方 2P 之刪除。

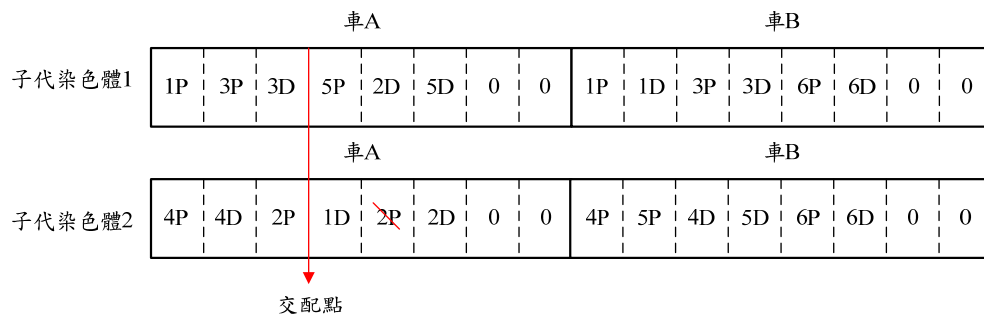


圖 3.9 交配示意 (b)

b. 再行檢視其區塊內，交配點前是否皆含成對之提送貨點。若有落單

之提點，在優先限制 (Precedence Constraints) 與聯結限制 (Coupling Constraints) 之前提下，找出於其提點後且在同車範圍內，符合容量限制且為最佳評估函數值之所在，加上該落單提點之對應送點。若有不只一落單提點，則依序補上其對應之送點。以圖 3.10 為例，子代染色體 1 之交配區塊車 A 中，檢驗交配點前是否皆含成對提送貨點，發現落單提點 1P，故在提點 1P 之後且在車 A 範圍內，找出最佳位置安插相對應送點 1D。

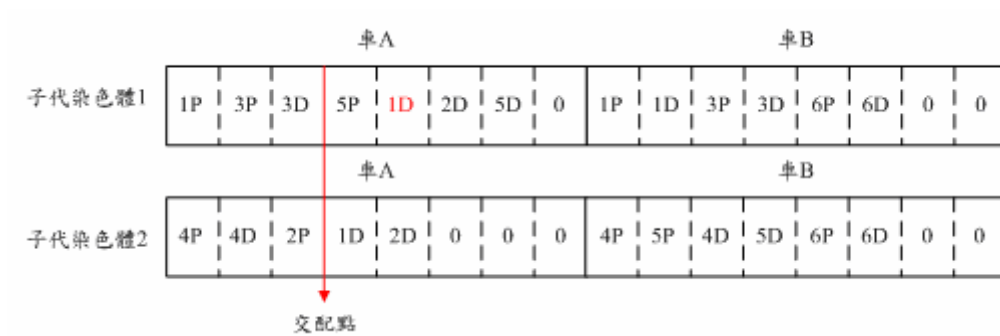


圖 3.10 交配示意 (c)

- c. 子代染色體中，在交配點後若出現與交配點前重複之基因，將後方重複之基因刪除。以圖 3.11 為例，子代染色體 1 在交配點後方出現與交配點前重複之提送貨點 1P、1D、3P 以及 3D，將其刪除。

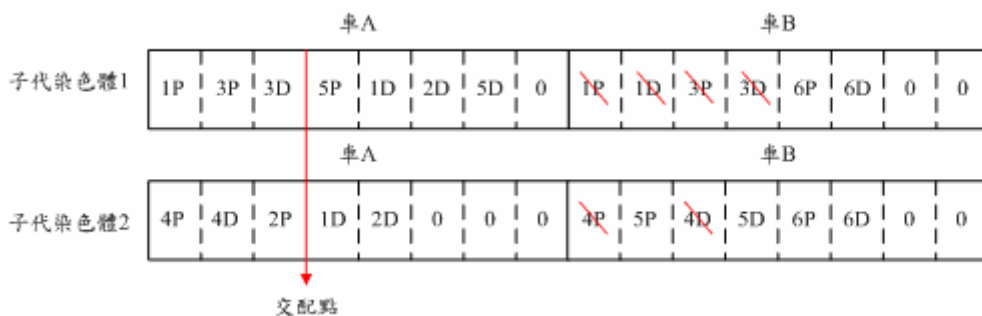


圖 3.11 交配示意 (d)

- d. 檢視交配點後，是否皆含成對之提送貨點。若有落單之送點，在優先限制 (Precedence Constraints) 與聯結限制 (Coupling Constraints) 之前提下，找出於其送點前且在同車範圍內，符合容量限制且為最佳評估函數值之所在，加上該落單送點之對應提點。若尋無適當地

點，則將此送點於該車刪除，重新檢視所有車輛，將其對應提點插入符合容量限制且為最佳評估函數值之所在。再者，因優先限制與聯結限制之前提下，找出於其提點後且在同車範圍內，符合容量限制且為最佳評估函數值之所在，加上該提點之對應送點。若有不只一落單送點，則依序補上其對應之提點。以圖 3.12 為例，檢驗子代染色體 1 之交配點後是否皆含成對提送貨點，發現落單送點 2D，故在送點 2D 之前且在車 A 範圍內，找出最佳位置安插相對應提點 2P。

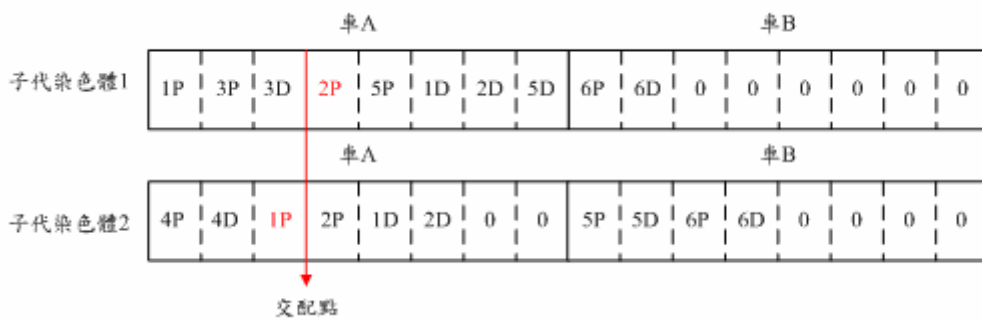


圖 3.12 交配示意 (e)

- e. 再行檢視子代染色體其基因值是否出現所有提送貨點，若有缺漏未服務之成對提送貨點，先將其提點在所有車中找出符合容量限制且為最佳評估函數值之所在，加上該提點；再將其送點依優先限制（Precedence Constraints）與聯結限制（Coupling Constraints）之前提下，找出於其提點後且在同車範圍內，符合容量限制且為最佳評估函數值之所在，加上該送點。以圖 3.12 為例，檢驗子代染色體 1 是否出現所有提送，發現缺漏未服務之成對提送貨點 4P 及 4D。在車 A 及車 B 中找尋最佳位置安插提點 4P，並在 4P 之後且在同車範圍內，安置相對應送點 4D。

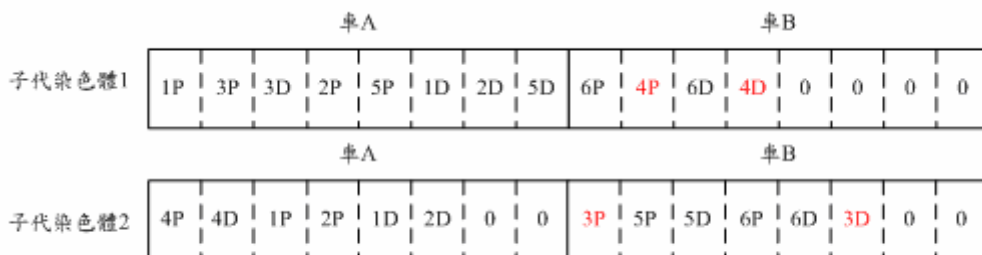


圖 3.13 交配示意 (f)

B. 交配點落在區塊與區塊之實線：

- a. 在子代染色體中，在交配點後若出現與交配點前重複之基因，將後方重複之基因刪除。
- b. 檢視子代染色體其基因值是否出現所有之提送貨點，若有缺漏未服務之成對提送貨點，先將其提點在所有車中找出符合容量限制且為最佳評估函數值之所在，加上該提點；再將其送點依優先限制（Precedence Constraints）與聯結限制（Coupling Constraints）之前提下，找出於其提點後且在同車範圍內，符合容量限制且為最佳評估函數值之所在，加上該送點。

5. 族群替換

在進行交配步驟時，皆考量容量、優先、聯結限制等因素，故可保證每代產生之子代染色體皆為可行解。本研究採用「每世代」皆進行族群替換，此方式與「僅替換最佳解」之差異，在於可藉此跳脫目前現解空間，以避免陷於區域最佳解，並增加新的搜尋區域以得到全域最佳解。因此，本研究採用每世代族群替換，並另外紀錄且更新最佳解。

6. 突變機制

每次進行交配步驟時，先行產生一介於 0 到 1 間之隨機亂數 P ，若 P 值小於突變率，則進行突變。先隨機選取此世代一半之染色體，並找尋每個染色體中車輛服務提送貨點最少之車輛，若有兩者以上之車輛發生此情形，以較後方之車輛進行突變。將該車輛之提送貨點，以成對之方式依序分派至符合容量限制並以違反時間窗限制最小之車輛內，直至該車輛已無任何基因值。藉此突變機制，減少現行服務車輛數。以利藉由新型態之染色體，跳脫目前現解空間必免得到區域最佳解，增加新的搜尋區域以增加尋得全域最佳解之機率。

7. 終止條件 (Termination)

本研究之終止條件為評估函數值在演化過程中皆無法有效改進。判斷無法有效改進的條件為此世代最佳評估函數值與當前最佳評估函數值 (Incumbent Solution) 兩者之差，對於當前最佳評估函數值之改善幅度若連續 100 世代皆小於 0.01，即將目前世代輸

出產生結果並終止演算，否則繼續進行交配、突變與適合度計算等程序。



第四章 範例測試

本研究為測試和求證以基因演算法求解具時間窗提送貨問題之速度與品質，設計十種包含不同節點數的網路進行測試。4.2 節將會說明詳細的測試過程；4.3 節列示並說明範例測試結果；4.4 節列示並說明敏感度分析結果；4.5 節則為小結。

4.1 產生測試範例

依據 3.1 節的問題特性描述，設定所有提送貨任務、貨車載重材積與最早最晚服務時間等資訊，作為演算法讀取資料來源。本研究分別建構十個具有不同節點數量的網路，分別為包含 10、20、30 直到 100 個節點的網路，每個網路均包含一個起點與一個迄點，其餘節點均為任務節點，其中一半為提貨節點，另一半則為送貨節點。因此，包含 10 個節點的網路中，是由 4 筆訂單所產生 4 個提貨節點與 4 個送貨節點，再加上起點與迄點所組成；同理，包含 20 個節點的網路中，是由 9 筆訂單所產生 9 個提貨節點與 9 個送貨節點，再加上起點與迄點所組成；同理其他網路亦依此方式所組成。最後設定每個節點的允許執行時間窗，意即最早時間與最晚時間，此部份由本研究針對各個節點設定一對合理的時間窗。

以 10 個節點的網路來做說明，每一筆任務所包含的資訊如表 4.1 所示。其中「節點編號」為此任務在網路中所代表的節點號碼；「訂單編號」為此任務所屬的訂單號碼；「任務類型」則表示此任務為提貨任務 P、送貨任務 D、或起迄任務 W；「配對節點」為同一訂單編號中互為提貨任務 P、送貨任務 D 之相對應節點編號，若為起迄任務 W 則配對節點即為其自身節點編號；「最早時間」為允許執行該任務的最早時間；「最晚時間」為允許執行該任務的最晚時間；「載重材積」為此任務的貨物載重材積，當此任務為提貨時，則載重材積為正值，當此任務為送貨時，則載重材積為負值。訂單編號相同的提貨任務與送貨任務稱為一任務配對，任務配對之載重材積加總為零。

表 4.1 任務資訊範例表

節點編號	訂單編號	配對節點	任務類型	載重材積	最早時間	最晚時間
1	0	1	W	0	0	0
2	1	6	P	25	3	6
3	2	7	P	10	12	15
4	3	8	P	15	9	9
5	4	9	P	20	7	10
6	1	2	D	-25	33	38
7	2	3	D	-10	21	22
8	3	4	D	-15	41	44
9	4	5	D	-20	21	24
10	0	10	W	0	10000	10000

表 4.2 記載點與點之間的距離，同樣以 10 個節點的網路來進行說明，每一個點與點之間的距離皆可查表所得知。例如點 5 至點 8 的距離為 77，點 8 至點 3 之距離為 66。除自身點其距離為 0，其他值為常態分配下 1-100 之隨機亂數。

表 4.2 距離矩陣表

	1	2	3	4	5	6	7	8	9	10
1	0	87	70	63	29	28	95	56	31	0
2	87	0	66	95	91	79	52	70	23	87
3	70	66	0	81	50	44	22	66	35	70
4	63	95	81	0	32	82	50	23	49	63
5	29	91	50	32	0	24	19	77	80	29
6	28	79	44	82	24	0	24	12	96	28
7	95	52	22	50	19	24	0	27	41	95
8	56	70	66	23	77	12	27	0	15	56
9	31	23	35	49	80	96	41	15	0	31
10	0	87	70	63	29	28	95	56	31	0

4.2 測試過程

本研究之測試過程是藉由 AMD Athlon(tm) 1.09GHz, 1.00GB RAM 的個人電腦進行運作，作業系統為 Microsoft Windows XP Professional Version 2002 SP2，使用 Microsoft Visual Basic 6.0 編譯器撰寫演算法。

表 4.3 範例測試參數設定表

人口數	50
車輛數	3
車容量	150、200、250
突變率	0.1%
突變比例	人口數之 50%
懲罰乘數	100
停止條件	連續 100 代改善幅度未達 1%

於 4.1 節建構完成網路拓撲，開啟並執行程式，依照表 4.3 完成各項參數設定，並讀取欲進行計算的範例資訊，如表 4.1 以及表 4.2。程式產生起始解，並開始進行交配演化、族群替代等過程，中間參雜突變機制，直到達成終止條件。最後，依程式計算結果繪製趨勢圖，並分析演算結果。

4.3 測試結果

本研究將十個分別包含 10、20、30 到 100 個節點的網路依照 4.2 節所說明的測試過程進行測試，以得到求解結果如表 4.4 所示。以下先列舉節點數最少（10）以及節點數最多（100）之網路，分別詳述其成本趨勢。

表 4.4 節點範例總表

節點數	世代數	CPU 時間	起始成本	最佳成本	CPU 時間/世代數
10	234	0.18	1758	473	0.00076923
20	970	2.85	49137	625	0.00293814
30	380	3.22	65696	1083	0.00847368
40	814	16.14	84733	1440	0.01982801
50	946	34.41	126559	1904	0.03637421
60	393	26.33	166801	2289	0.06699746
70	312	32.16	205046	2710	0.10307692
80	1447	198.63	215617	2889	0.13727021
90	849	185.39	234380	3217	0.21836278
100	1032	262.03	250338	3872	0.25390504

網路包含 10 個節點的範例，僅演化 234 個世代，耗費 CPU 時間 0.18 秒即可求解出總成本為 473 的可行路徑組合，如圖 4.1 和圖 4.2 所示。一開始的起始最佳成本為 1758，到第 3 世代時，最佳成本降低為 509；到第 59 世代時，最佳成本降低為 473。第 134 代到第 234 代被視為終止條件的連續 100 代改善幅度未達 1%。世代總人口之平均呈現下

滑趨勢；世代內最佳解亦呈現下滑趨勢。

以 Lingo 8.0 求解此 10 個節點範例之數學模式（相關資料參照表 4.1 和表 4.2），找出最佳解 473，此數據和本研究利用基因演算法所找出之最佳解完全吻合。本研究進行範例測試時，預設最大使用車輛數為 3，如表 4.3 所示。不論 Lingo 8.0 或本研究演算之結果，皆為使用兩輛車，即可找到最佳解 473。其中一車輛之繞行節點路徑為 1→2→3→7→6→10，另一車輛繞行節點路徑為 1→4→5→9→8→10。由上可知，本研究可找出預設最大使用車輛數以下之最佳實際所需車輛數。

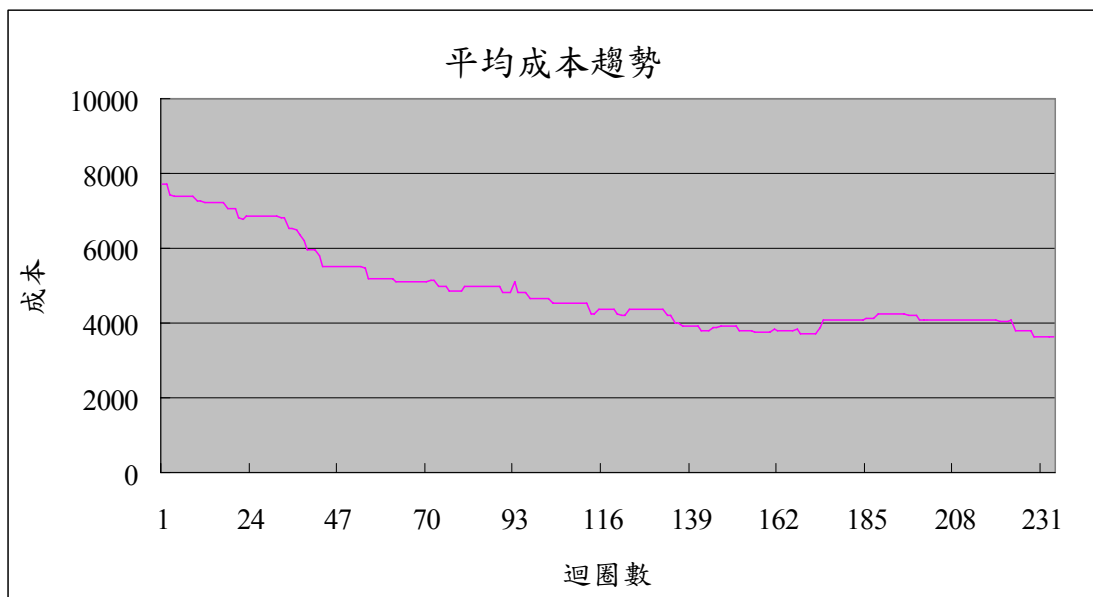


圖 4.1 節點 10 之平均成本趨勢圖

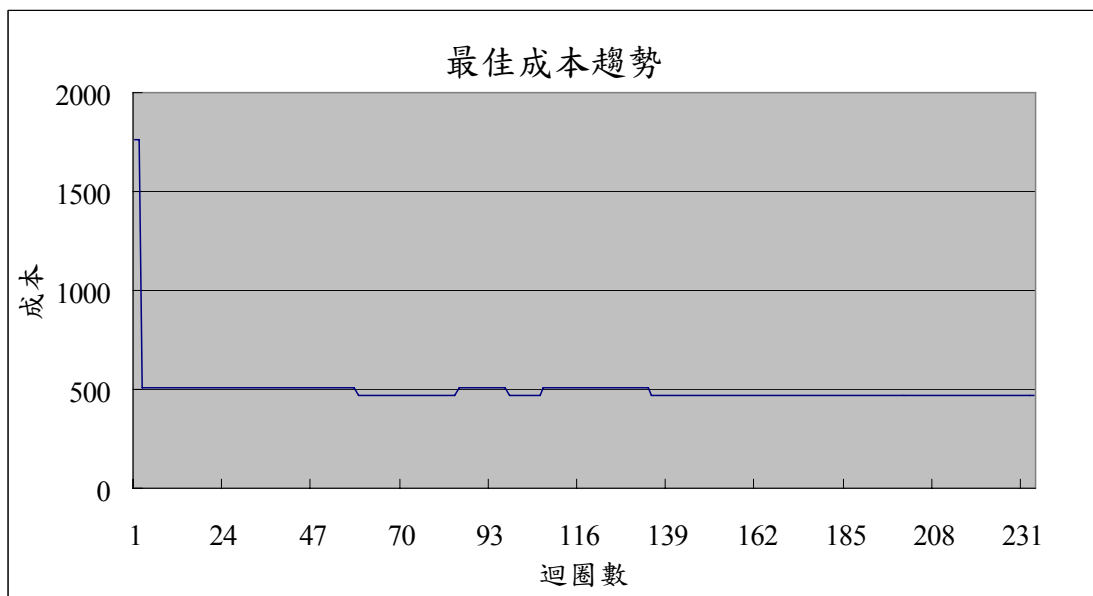


圖 4.2 節點 10 之最佳成本趨勢圖

網路包含 100 個節點的範例，演化 1032 個世代，耗費 CPU 時間 262.03 秒；求解出總成本為 3872 的可行路徑組合，如圖 4.3 和圖 4.4 所示。一開始的起始最佳成本為 250338，中間歷經 2 次突變，分別在第 224 以及 709 世代。造成 2 次世代全體人口平均條地直線上升，且引發其最佳成本突破區域最佳解，進而找到成本更低之改善解。突變之後，其後續平均成本亦先大幅改善呈現下滑趨勢，但最終仍持平，呈現小幅不甚明顯之波動。最佳成本在一開始呈現大幅降低狀態，但後繼為時有起伏之貌，最終在第 932 代到第 1032 代視為終止條件的連續 100 代改善幅度未達 1%，並持續維持 3872 之成本。

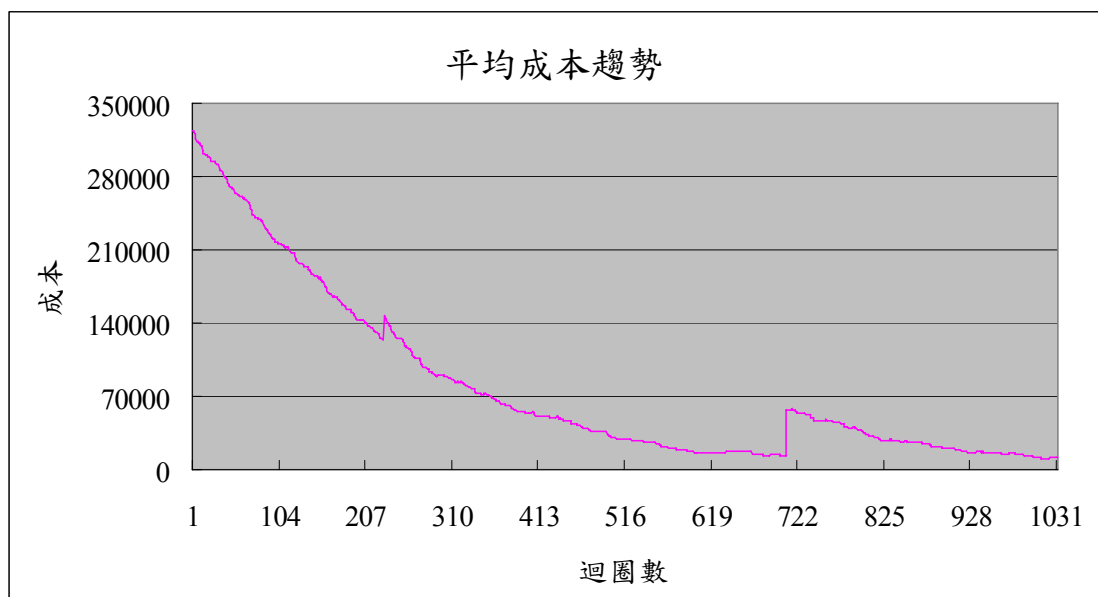


圖 4.3 節點 100 之平均成本趨勢圖

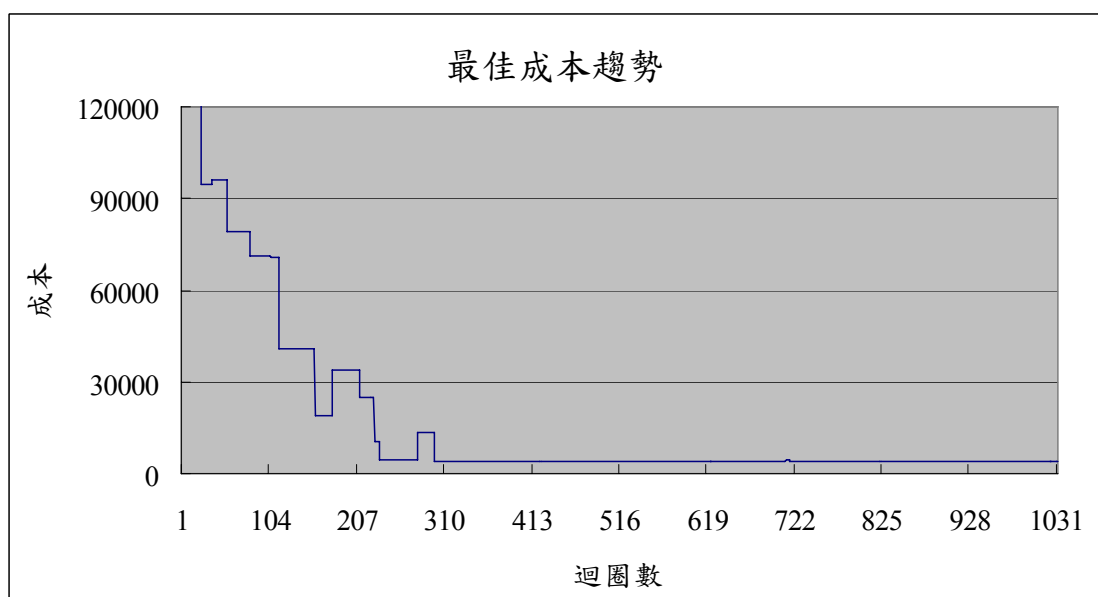


圖 4.4 節點 100 之最佳成本趨勢圖

由表 4.4 可察得以下現象：

1. 欲求得品質不錯之近似最佳解，當面臨範例節點數愈多，不代表其演化世代數及 CPU 時間相對應成長，如圖 4.5 和圖 4.6 所示。基因演算法會由於機率因素，而去影響其演化世代數，如範例 80 演化世代數為 1447、CPU 時間為 198.63 秒，範例 90 演化世代數為 849、CPU 時間為 185.39 秒。

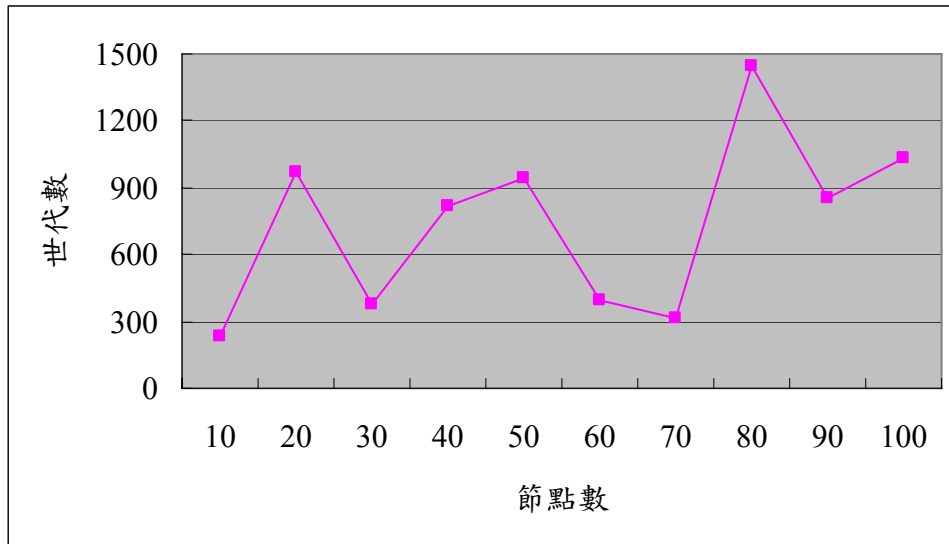


圖 4.5 世代數變化趨勢圖

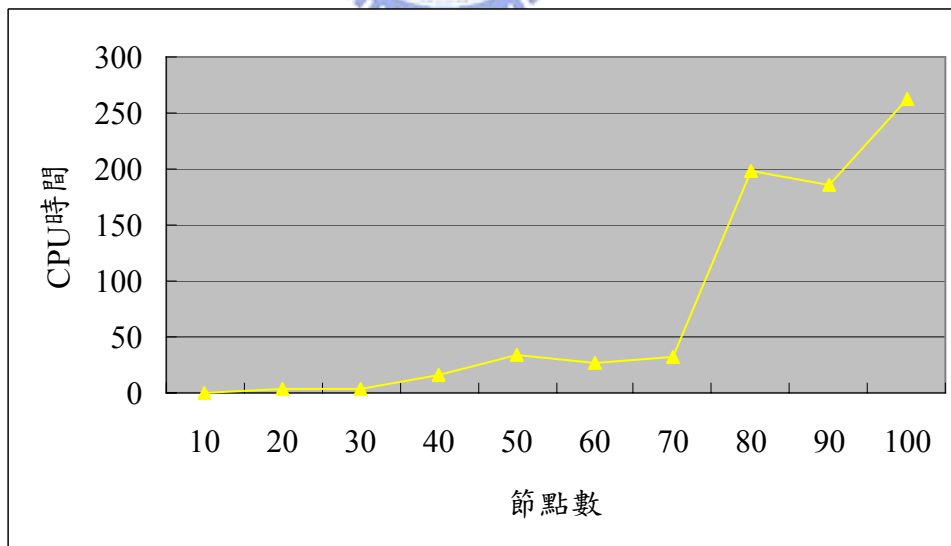


圖 4.6 CPU 時間變化趨勢圖

2. 由圖 4.7 和 4.8 所示，起始成本與最佳成本之變化趨勢，大體上皆為正向成長，即隨著節點數愈多，其成本愈大。

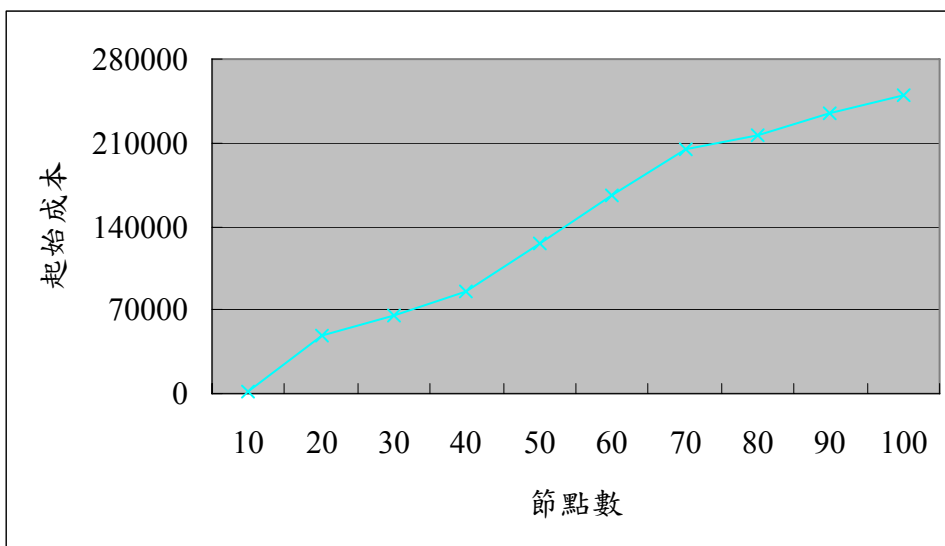


圖 4.7 起始成本變化趨勢圖

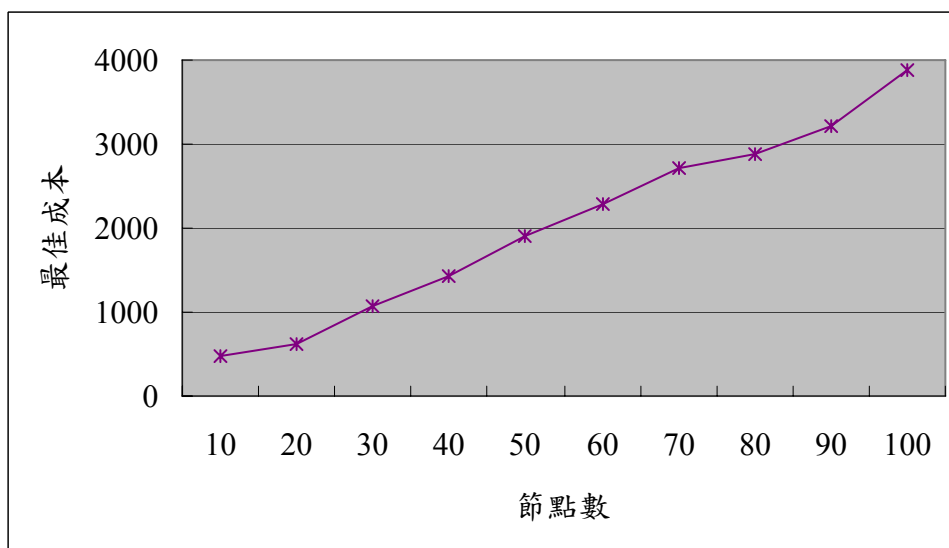


圖 4.8 最佳成本變化趨勢圖

3. 範例節點數愈多，每演化一世代所耗費之 CPU 時間愈長，如圖 4.9 所示。

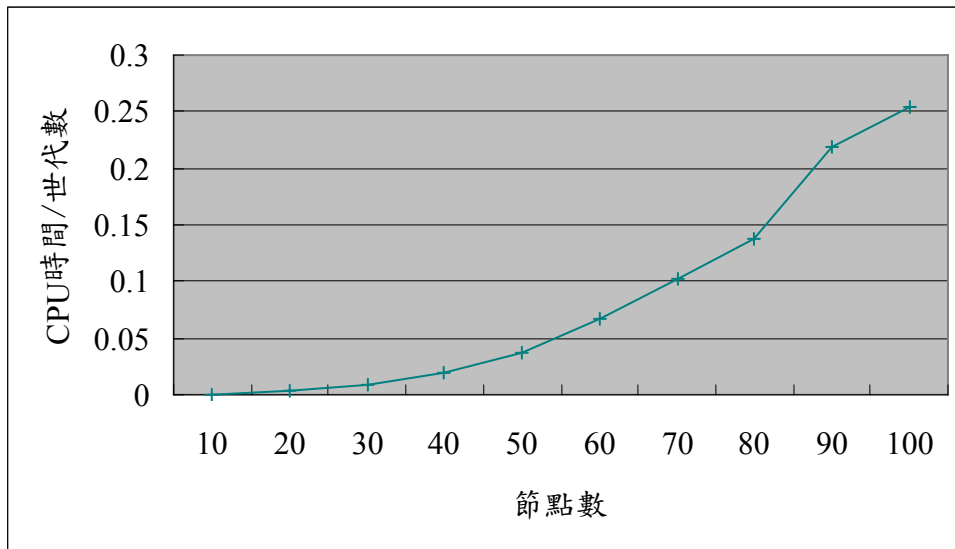


圖 4.9 CPU 時間/世代數變化趨勢圖

4.4 敏感度分析

本研究為了解各項因子，如人口數、車輛數、車容量、突變率、突變比例、懲罰乘數以及停止條件為連續演化固定世代，對於求解品質及速度之影響，故針對以上因子進行一系列之敏感度分析。

以 30 個節點之網路進行測試，並以表 4.5 之參數設定為標準狀態，即人口數 50、車輛數 3、車容量 200、突變率 0.1%、突變比例 50%、懲罰乘數 100 以及停止條件為連續演化 1000 世代。表 4.6 至 4.11 標示為灰底之欄位，即為本節定義之標準狀態。

表 4.5 敏感度分析參數設定表

人口數	50
車輛數	3
車容量	200
突變率	0.1%
突變比例	總人口數之 50%
懲罰乘數	100
停止條件	連續演化 1000 世代

以上述七個項目進行測試，除停止條件此項目外，其他六個項目皆為進行 1000 個演化世代即終止。測試數據如以下所示，人口數分別為 25、50、100、200、300 以及 400；車輛數分別為 1、2、3、4、5 以及 6；車容量分別為 35、50、75、100、150 以及 200；突變率分別為 1/5000、1/1000、1/500、1/100、1/50 以及 1/10；突變數分別為 1%、10%、

30%、50%、70%以及 90%；懲罰乘數分別為 5、15、25、50、75 以及 100；停止條件分別為 25、50、75、100 以及 125；其結果分析將於下列各點分別說明。

1. 人口數變動

人口數即為族群個體數。由表 4.6 和圖 4.10 可觀察得知，當人口數過大或過小時，其平均最佳成本之趨勢皆為上揚。目標值一樣之優良解，在人口數少時，較人口數多時有更高之機率被挑選到，故其解會更有效率之進行改善。但若人口數過少時，造成搜尋之對象與可參考之路徑組合過少，無法有效進行廣度搜尋。

表 4.6 人口數變動敏感度分析表

編號	人口數					
	25	50	100	200	300	400
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
1	1083	1084	1124	1176	1159	1156
2	1120	1090	1132	1123	1154	1222
3	1107	1111	1106	1128	1170	1176
4	1092	1093	1117	1060	1202	1147
5	1047	1098	1093	1151	1133	1161
6	1090	1104	1088	1161	1119	1166
7	1099	1115	1083	1149	1216	1181
8	1099	1111	1112	1117	1166	1143
9	1091	1102	1091	1122	1179	1199
10	1102	1100	1087	1161	1114	1160
11	1116	1078	1113	1124	1151	1167
12	1093	1079	1059	1091	1128	1176
13	1091	1087	1104	1120	1091	1080
14	1096	988	1079	1111	1164	1145
15	1094	1054	1124	1122	1110	1214
16	1089	1119	1071	1110	1164	1197
17	1073	1062	1077	1143	1202	1146
18	1044	1100	1070	1156	1166	1084
19	1104	1060	1116	1094	1188	1137
20	1077	1094	1088	1134	1105	1123
平均	1090.35	1086.45	1096.7	1127.65	1154.05	1159

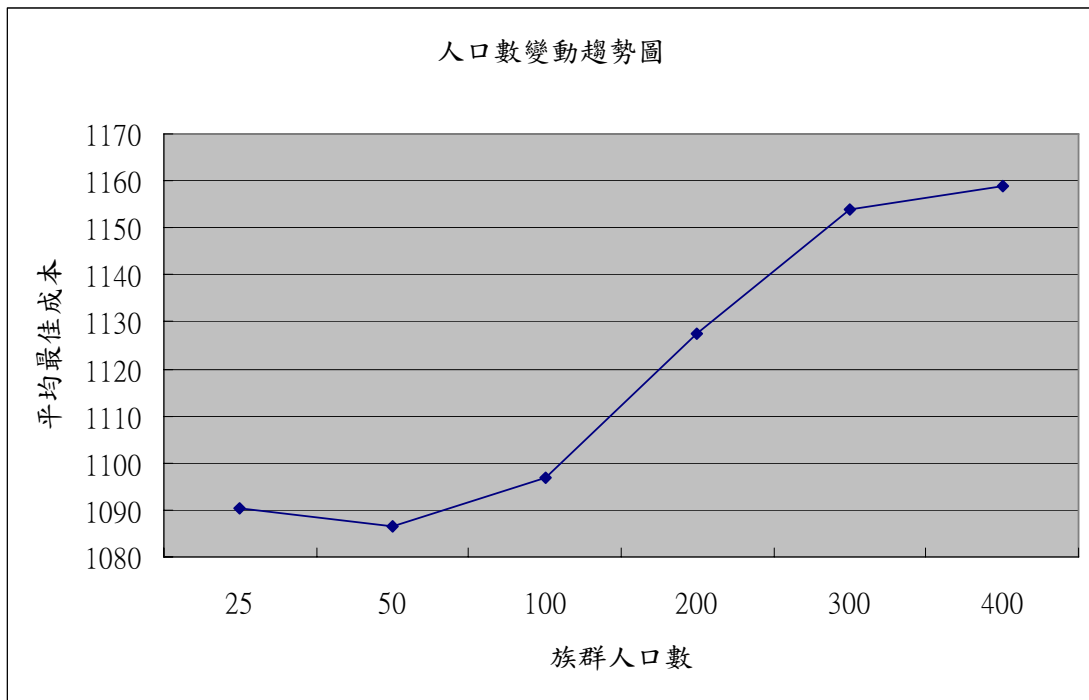


圖 4.10 人口數變動趨勢圖

2. 車輛數變動

由表 4.7 和圖 4.11 可觀察得知，在以此狀況下，演算法會自行找尋出符合範例狀況條件下之最佳服務車輛數，不管低於或高於此車輛數，皆會造成其平均最佳成本上升。

表 4.7 車輛數變動敏感度分析表

編號	車輛數					
	1	2	3	4	5	6
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
1	1056	1052	1084	1163	1189	1263
2	1064	1031	1090	1180	1186	1214
3	1040	1043	1111	1136	1189	1245
4	1062	1016	1093	1129	1185	1250
5	1070	1000	1098	1164	1190	1190
6	1046	1016	1104	1167	1187	1217
7	1073	1000	1115	1136	1184	1245
8	1062	1008	1111	1147	1181	1259
9	1090	1018	1102	1170	1173	1235
10	1082	1040	1100	1164	1218	1217
11	1062	998	1078	1141	1190	1238
12	1066	1000	1079	1152	1187	1209
13	1100	1060	1087	1141	1216	1283
14	1091	1056	988	1173	1198	1215
15	1078	1032	1054	1170	1173	1251
16	1078	1022	1119	1126	1200	1241
17	1056	1016	1062	1153	1160	1223
18	1067	1062	1100	1159	1170	1248
19	1057	1023	1060	1136	1219	1249
20	1070	1039	1094	1117	1210	1262
平均	1068.5	1026.6	1086.45	1151.2	1190.25	1237.7

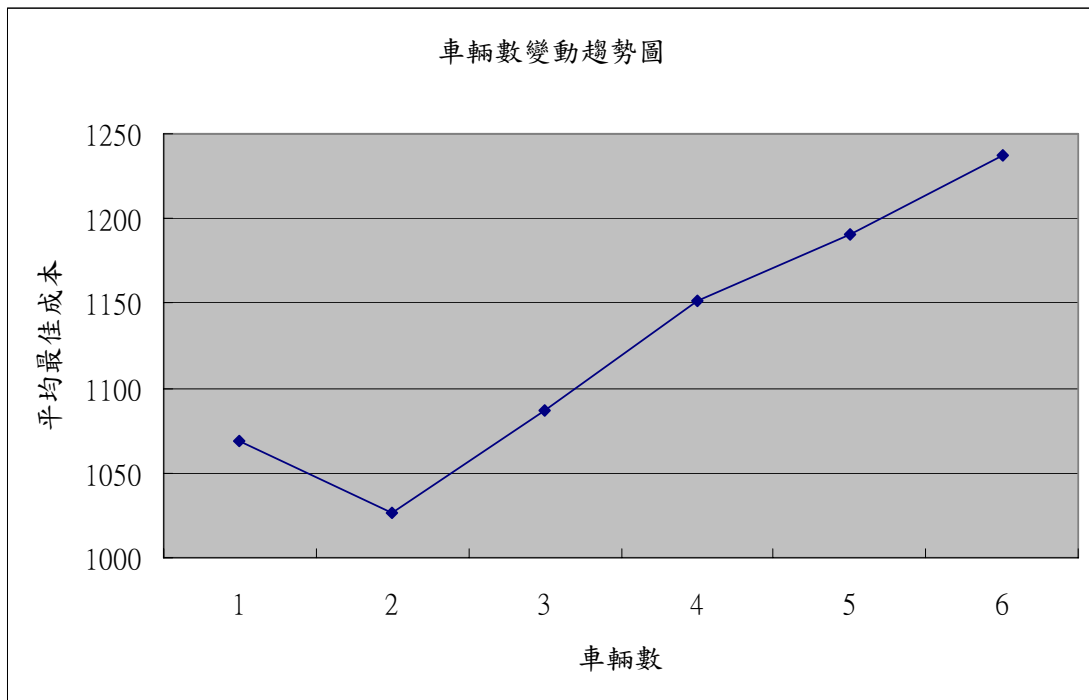


圖 4.11 車輛數變動趨勢圖

3. 車容量變動

由表 4.8 和圖 4.12 可觀察得知，當車容量增加時，其平均最佳成本之趨勢呈現反向關係下降。在車容量 35 的狀況下，由於範例關係使然，造成其車容量呈現明顯不足的狀態，所以使其平均最佳成本狀態居高不下。車容量 50 的狀況下，已有紓解之情形。車容量 75 的狀況下，又因範例關係使然，車容量已顯足夠之狀態，所以造成車容量再行增添，亦不會使平均最佳成本有大幅下滑之趨勢，呈現細微的降低波動。

表 4.8 車容量變動敏感度分析表

編號	車容量					
	35	50	75	100	150	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
1	2989	1400	1123	1074	1049	1084
2	4305	1708	1132	1118	1103	1090
3	3781	1540	1091	1070	1113	1111
4	4478	1315	1081	1135	1089	1093
5	4260	1256	1111	1073	1077	1098
6	5599	1299	1107	1092	1097	1104
7	6206	1414	1085	1084	1100	1115
8	3927	1430	1129	1073	1053	1111
9	4268	1435	1108	1104	1093	1102
10	6136	1574	1101	1048	1052	1100
11	3721	1460	1089	1118	1069	1078
12	4658	1223	1132	1105	1087	1079
13	5086	1712	1112	1090	1115	1087
14	4613	1308	1087	1090	1107	988
15	6213	1375	1107	1097	1112	1054
16	5637	1395	1128	1081	1121	1119
17	4547	1405	1064	1053	1077	1062
18	4284	1358	1118	1133	1118	1100
19	4528	1698	1148	1093	1053	1060
20	7209	1517	1100	1101	1102	1094
平均	4822.25	1441.1	1107.65	1091.6	1089.35	1086.45

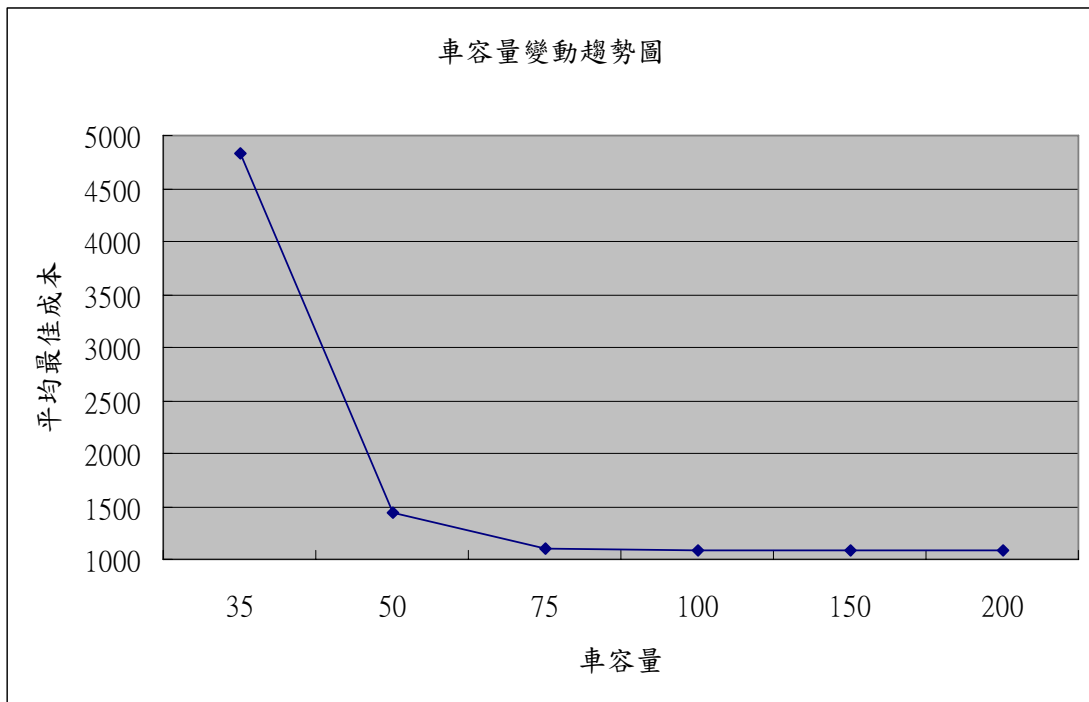


圖 4.12 車容量變動趨勢圖

4. 突變率變動

由表 4.9 和圖 4.13 可觀察得知，當突變率過大或過小時，其平均最佳成本之趨勢皆為上揚。突變之作用在於跳脫區域最佳解，進而搜尋到全域最佳解。然而，當突變率過高，造成突變發生過於頻繁，使得解無法有效在同一區塊進行深度搜尋。當突變率過低，造成突變發生過於少見，使得解無法有效跳脫區域進行廣度搜尋。

表 4.9 突變率變動敏感度分析表

編號	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
1	1074	1084	1112	1090	1095	1101
2	1117	1090	1116	1093	1059	1101
3	1117	1111	1082	1100	1129	1119
4	1100	1093	1101	1075	1122	1112
5	1103	1098	1088	1095	1119	1134
6	1107	1104	1053	1127	1101	1108
7	1073	1115	1053	1101	1072	1144
8	1076	1111	1117	1080	1109	1088
9	1092	1102	1110	1074	1081	1110
10	1105	1100	1107	1092	1117	1116
11	1093	1078	1119	1061	1101	1091
12	1098	1079	1057	1110	1137	1127
13	1090	1087	1072	1110	1102	1091
14	1048	988	1094	1124	1077	1112
15	1120	1054	1060	1111	1092	1117
16	1059	1119	1025	1080	1080	1141
17	1098	1062	1119	1148	1104	1140
18	1091	1100	1134	1063	1102	1124
19	1093	1060	1102	1094	1099	1107
20	1076	1094	1111	1131	1131	1111
平均	1091.5	1086.45	1091.6	1097.95	1101.45	1114.7

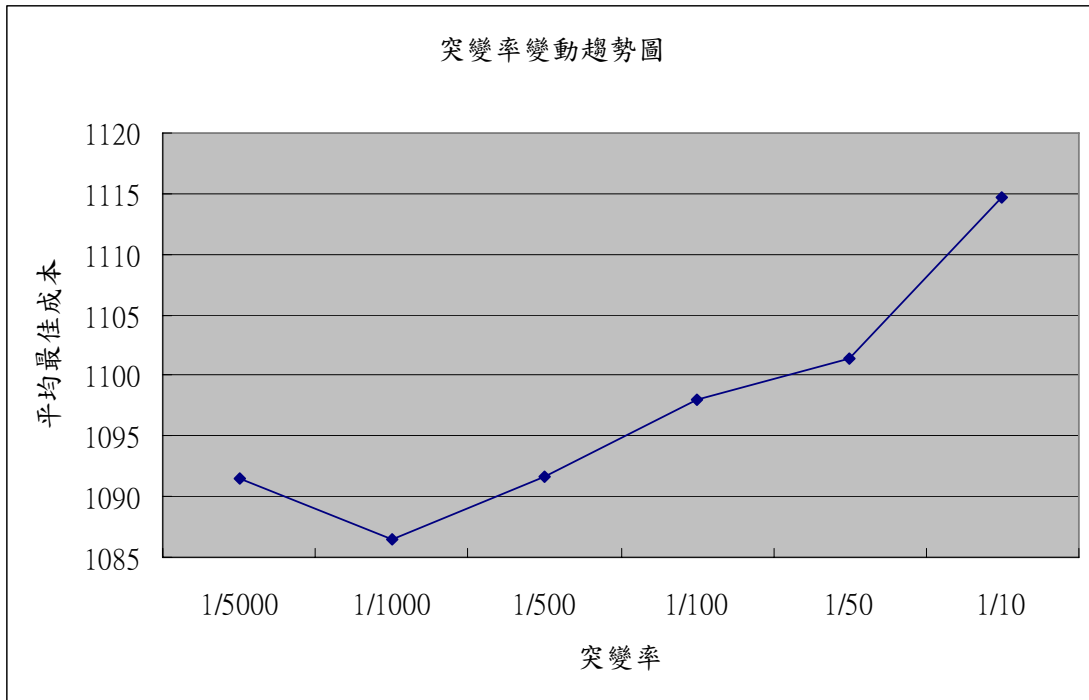


圖 4.13 突變率變動趨勢圖

5. 突變比例變動

由表 4.10 和圖 4.14 可觀察得知，當突變比例增加時，其平均最佳成本之趨勢沒有固定上升或下降之變化，呈現擺盪不定之狀態。分析其中可能原因為在此設定突變機率為 1/1000，且停止條件為演化 1000 個世代，故造成其突變次數不多之情形。又因突變次數不多，所以突變比例變動就相對不敏感，造成趨勢擺盪不定，沒有明顯走向。

表 4.10 突變比例變動敏感度分析表

編號	突變比例					
	1%	10%	30%	50%	70%	90%
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
1	1120	1156	1100	1084	1101	1106
2	1056	1072	1120	1090	1127	1084
3	1085	1096	1116	1111	1056	1077
4	1066	1072	1080	1093	1101	1108
5	1087	1080	1110	1098	1141	1089
6	1096	1115	1105	1104	1122	1081
7	1105	1077	1060	1115	1108	1079
8	1114	1102	1074	1111	1087	1022
9	1053	1109	1063	1102	1096	1066
10	1066	1111	1087	1100	1106	1117
11	1092	1085	1101	1078	1109	1090
12	1080	1109	1097	1079	1060	1072
13	1063	1111	1076	1087	1075	1097
14	1060	1092	1053	988	1113	1135
15	1076	1110	1084	1054	1097	1103
16	1054	1098	1110	1119	1129	1059
17	1077	1104	1084	1062	1096	1088
18	1043	1077	1047	1100	1072	1084
19	1102	1096	1123	1060	1128	1063
20	1100	1099	1076	1094	1076	1090
平均	1079.75	1098.55	1088.3	1086.45	1100	1085.5

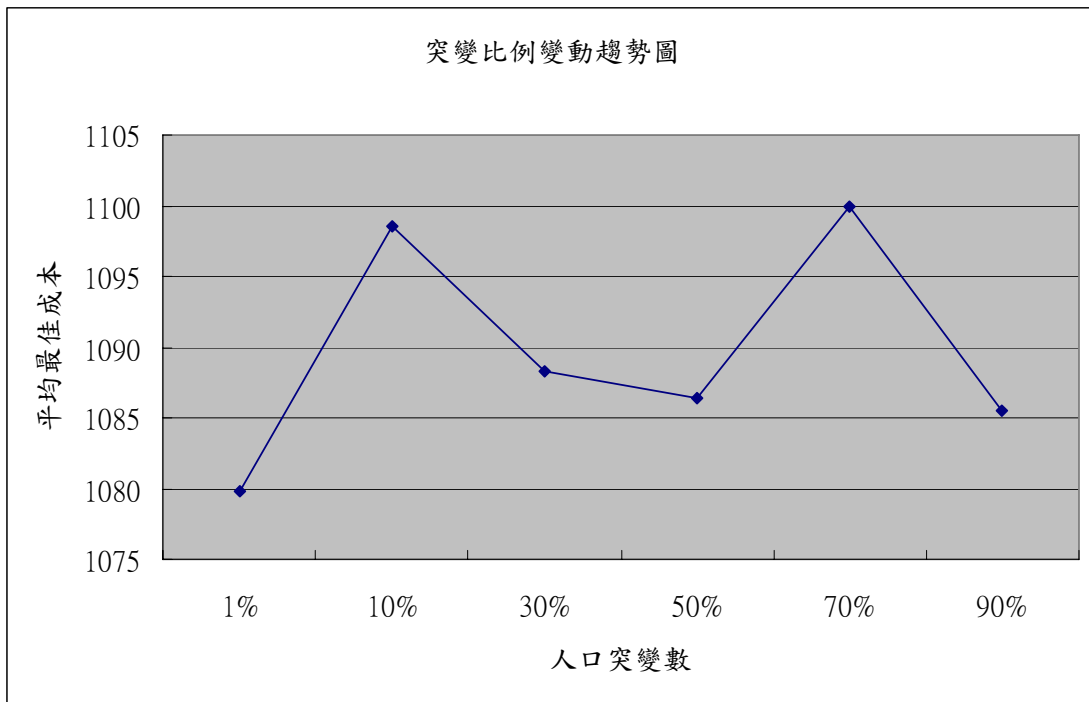


圖 4.14 突變比例變動趨勢圖

6. 懲罰乘數變動

由表 4.11 和圖 4.15 可觀察得知，當懲罰乘數增加時，其平均最佳成本之趨勢呈現反向關係下降。但當懲罰乘數大至一定程度之後，其平均最佳成本之趨勢呈現持平略有起伏之貌。

懲罰機制之作用在於避免違反時間窗，以去滿足顧客需求。當懲罰乘數過小時，其懲罰機制作用效果不明顯，同時造成平均最佳成本較懲罰乘數適當時為高。由以上，平均最佳成本會隨懲罰乘數增加，呈現負相關之下降趨勢。若懲罰乘數依範例狀況條件下已呈現過大之情形，其趨勢即會停止下降，呈現小幅度之上下擺盪。

表 4.11 懲罰乘數變動敏感度分析表

編號	懲罰乘數					
	5	15	25	50	75	100
	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本	最佳成本
1	1075	1129	1113	1073	1104	1084
2	1134	1099	1076	1083	1085	1090
3	1061	1117	1088	1073	1090	1111
4	1107	1116	1106	1092	1031	1093
5	1062	1061	1081	1068	1079	1098
6	1157	1090	1116	1122	1103	1104
7	1108	1016	1094	1085	1079	1115
8	1078	1096	1114	1088	1096	1111
9	1093	1133	1122	1063	1075	1102
10	1145	1088	1073	1124	1092	1100
11	1093	1153	1049	1101	1096	1078
12	1155	1071	1100	1085	1080	1079
13	1096	1113	1130	1093	1106	1087
14	1123	1103	1062	1093	1075	988
15	1089	1090	1106	1060	1126	1054
16	1126	1111	1124	1116	1076	1119
17	1082	1072	1117	1110	1089	1062
18	1045	1118	1088	1093	1091	1100
19	1135	1092	1114	1043	1071	1060
20	1138	1149	1099	1071	1081	1094
平均	1105.1	1100.85	1098.6	1086.8	1086.25	1086.45

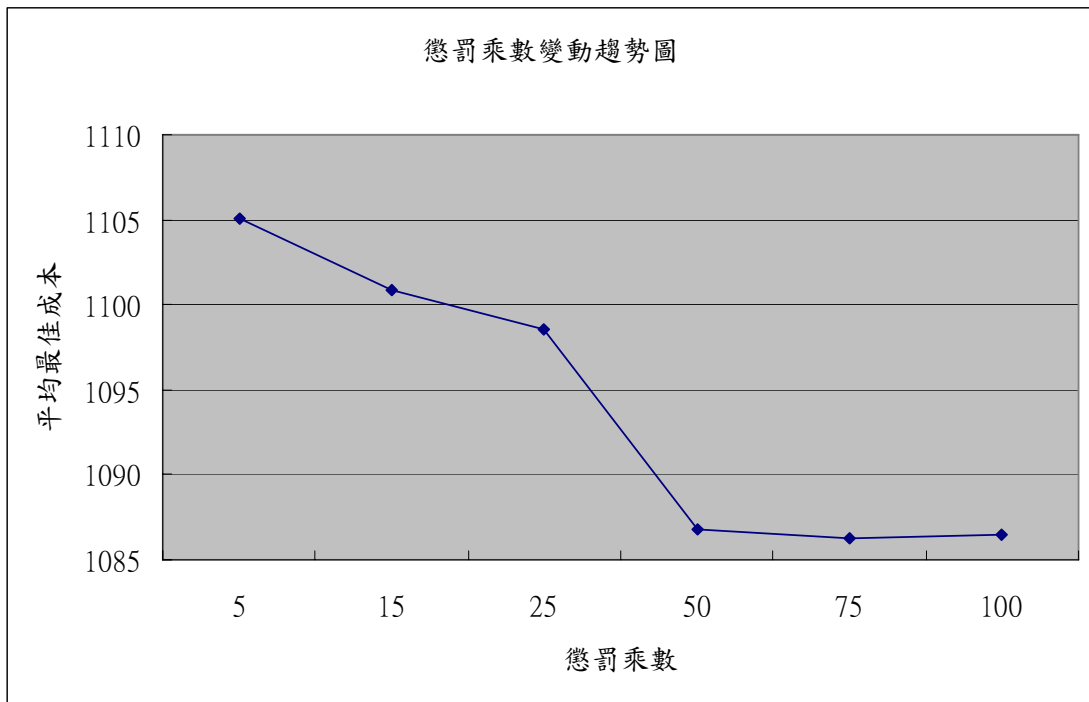


圖 4.15 懲罰乘數變動趨勢圖

7. 終止機制變動

由表 4.12、圖 4.16 和圖 4.17 可觀察得知，當終止機制連續代數增加時，其平均最佳成本之趨勢呈現反向關係下降，但與演化迴圈數之趨勢呈現正向關係上升。由於連續代數增加時，表示須停止演化之狀況較連續代數少時愈趨於穩定，故其平均最佳成本值亦較低。但因為連續代數增加，亦代表其終止條件愈顯嚴苛，故其演化迴圈數亦發上升。

表 4.12 終止機制變動敏感度分析表

編號	終止機制									
	25		50		75		100		125	
	最佳成本	迴圈數	最佳成本	迴圈數	最佳成本	迴圈數	最佳成本	迴圈數	最佳成本	迴圈數
1	1388	57	1321	67	1048	2498	1148	448	1150	367
2	1436	120	1244	172	1196	276	1146	513	1131	697
3	1145	272	1130	150	1139	607	1122	655	1161	323
4	1098	362	1229	165	1077	375	1122	402	1182	345
5	1173	240	1094	634	1100	1838	1099	940	1112	700
6	1298	76	1141	310	1092	2164	1220	242	1060	3582
7	1319	43	1084	410	1129	501	1174	311	1063	2258
8	1243	147	1206	220	1101	500	1127	538	1081	1437
9	2521	49	1525	94	1198	248	1078	328	1088	822
10	1359	82	1109	377	1151	184	1169	380	1137	655
11	1474	56	1108	455	1048	1854	1212	193	1094	544
12	1161	169	1104	627	1271	187	1170	375	1072	2048
13	2953	71	1077	798	1182	263	1087	2685	1125	589
14	1906	66	1111	276	1079	1950	1097	2197	1084	1549
15	1362	122	1100	443	1078	1013	1092	794	1103	626
16	1209	145	1144	250	1088	1174	1102	3467	1064	457
17	1196	228	1233	166	1232	141	1062	1427	1112	885
18	1230	111	1048	1151	1195	186	1102	450	1077	1827
19	1190	309	1129	166	1183	339	1048	2496	1097	1327
20	1226	96	1105	522	1100	1206	1100	1308	1053	2841
平均	1444.35	141.05	1162.1	372.65	1134.35	875.2	1123.85	1007.5	1102.3	1194

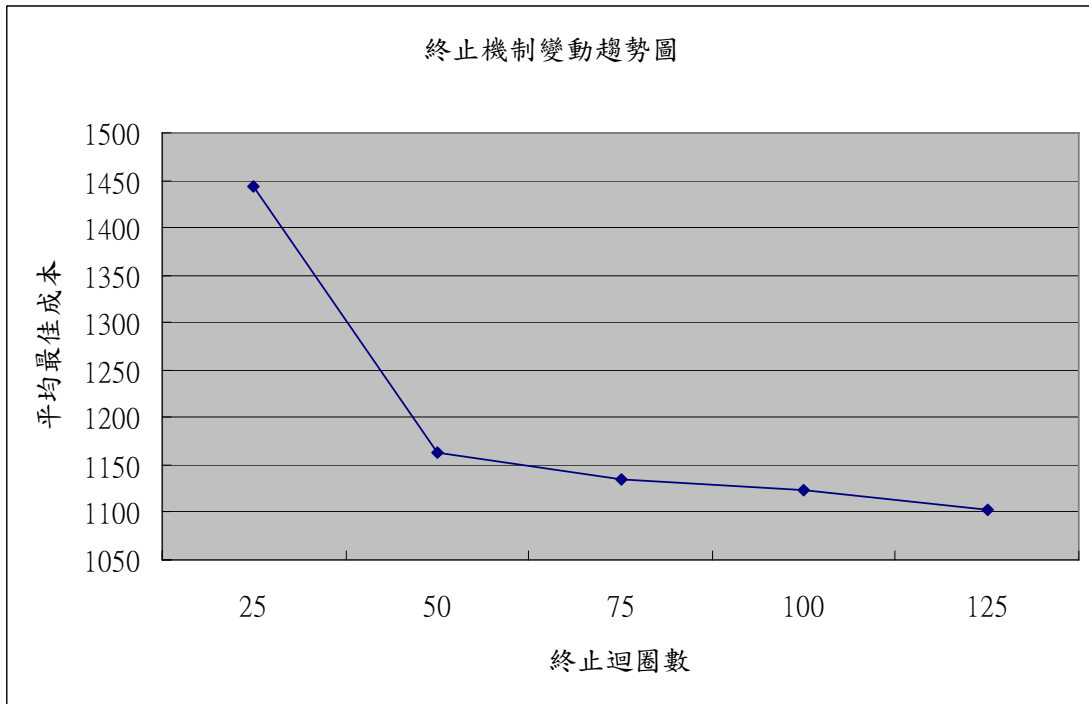


圖 4.16 終止機制變動趨勢圖 (a)

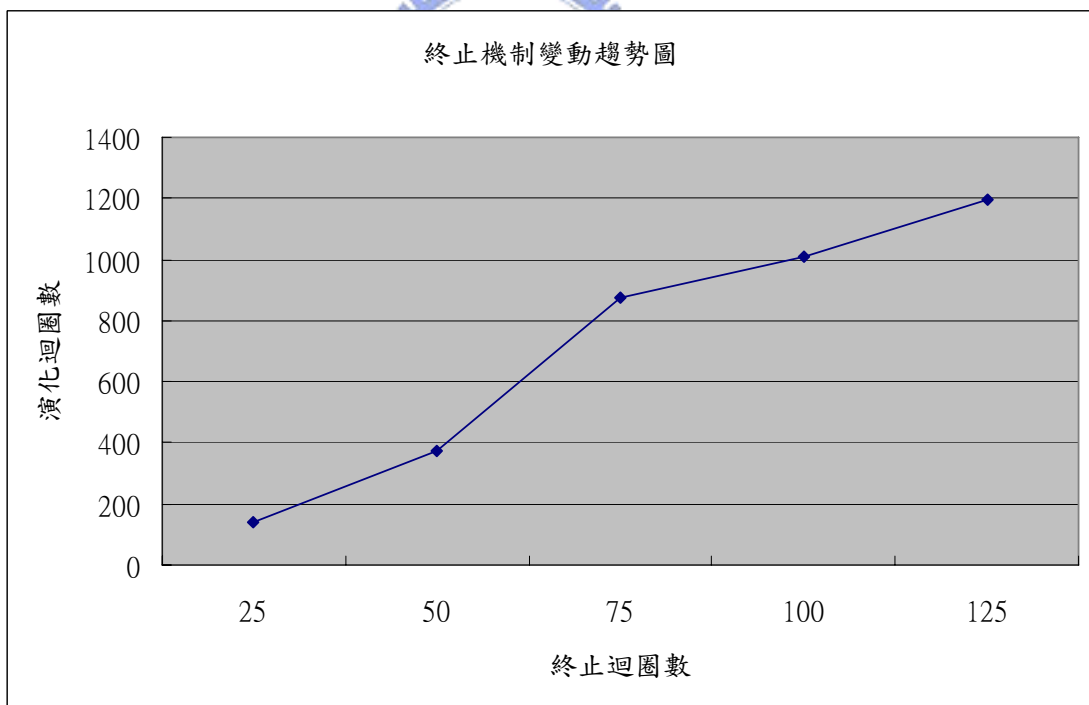


圖 4.17 終止機制變動趨勢圖 (b)

4.5 小結

本研究對於範例測試之整體趨勢，由圖 4.1 到圖 4.9 可作出以下結論：

1. 整體平均成本其趨勢為漸進式的持續減緩，到達一定程度之後呈現略有差異之持平狀態。由此，可知道本研究所發展之演算法可達到有效收斂的效果。
2. 每世代最優成本皆有急速下滑，但逐漸減緩的情形，亦可證明本演算法可有效降低成本，以達收斂之效。
3. 每世代最優成本歷經突變之後，可能發生找到更為優良之解，此可表示突變機制發揮其效用，跳脫區域最佳解進而找到全域最佳解。

本研究對於敏感度分析之整體趨勢，由圖 4.10 到圖 4.17 可作出以下結論：

1. 不同範例狀況條件下，會有其最適人口數。若高於或低於此人口數，將會造成其最佳成本上升。
2. 本研究之演算法會依不同範例狀況條件下，自行找出最佳服務車輛數。若高於或低於此車輛數，將會造成其平均最佳成本上升。
3. 依範例狀況之不同，在一定車容量之內，最佳成本會隨車容量增加，呈現負相關之下降趨勢。但若車容量依範例狀況條件下已呈現過大之情形，其趨勢即會停止下降，呈現小幅度之上下擺盪。
4. 不同範例狀況條件下，會有其最適突變率。若高於或低於此突變率，將會造成其平均最佳成本上升。
5. 最佳成本會隨懲罰乘數增加，呈現負相關之下降趨勢。若懲罰乘數依範例狀況條件下已呈現過大之情形，其趨勢即會停止下降，呈現小幅度之上下擺盪。
6. 最佳成本會隨連續代數增加，呈現負相關之下降趨勢；演化迴圈數會隨連續代數增加，呈現正相關之上升趨勢。

第五章 結論與建議

5.1 結論

1. 本研究之演算結果，證實可在滿足允許執行時間窗、提送貨任務配對、提送貨優先次序、司機下班時間與車輛載重容量等提送貨特性條件下進行求解。
2. 本研究無論為 10 個節點之小範例（耗費 0.18 秒 CPU 時間，求得最佳解），或是 100 個節點之大範例（耗費 262.03 秒 CPU 時間，求得近似最佳解）皆可有效求解，可適用於各種不同規模的問題。
3. 本研究可將車容量以及車輛數多寡視為可調整之變數，符合業者實際營運之需求與狀況，更增添其使用彈性。
4. 本研究進行求解具時間窗提送貨問題，能在合理運算時間下求出不錯的路徑組合，有效提升貨車調度人員的派遣作業與降低貨物運送的成本，亦可滿足業界需要在短時間內快速產生車輛之巡迴路徑之需求。
5. 基因演算法具有廣度搜尋的特性，幫助本研究在短暫合理時間內，可找出不錯的路徑組合。
6. 基因演算法之突變機制，亦可跳脫區域最佳解，將所有可能組合均納入演化中考量。

5.2 建議

1. 本研究之需求型態為已知且不變，建議未來研究可結合即時動態需求，以利增加營運效率。
2. 本研究之場站型態僅為單一，建議後續研究可擴大為多場站，以更加貼近業者之實際營運狀態。
3. 未來研究可以考慮結合 Local Search 進行深度搜尋，彌補基因演算法有可能因交配、突變所造成跳出此區域搜尋之契機，以進行更為有效率之演算。
4. 後續研究可加入即時路況與車輛旅行時間之考量，使營運業者可於線上即時更新車輛巡迴路徑，針對多變之實際路況即時進行適當更新，以利增加營運效率。

參考文獻

1. William P. Nanry, J. Wesley Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research Part B*, 34 (2000) 107-121.
2. Snežana Mitrovič-Minič, Ramesh Krishnamurti, and Gilbert Laporte, "Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows," *Transportation Research Part B*, 38 (2004) 669–685.
3. Solomon, M., "Algorithms for the vehicle routing and scheduling problems with time window constraints." *Operations Research* 35, 254-265.
4. Haibing Li and Andrew Lim, "A Metaheuristic for the Pickup and Delivery Problem with Time Windows," *Department of Computer Science National University of Singapore*
5. Hoong Chuin LAU, Zhe LIANG, "Pickup and Delivery with Time Windows :Algorithms and Test Case Generation," *13th IEEE International Conference on Tools with Artificial Intelligence*
6. Manfred Gronalt, Richard F. Hartl, and Marc Reimann, "New savings based algorithms for time constrained pickup and delivery of full truckloads," *European Journal of Operational Research* ,151 (2003) 520–535.
7. Psaraftis, H. N., "Dynamic vehicle routing: status and prospects," *Annals of Operations research* 61, 143-164.
8. Jacques Renaud, Fayez F. Boctor, and Gilbert Laporte, "Perturbation heuristics for the pickup and delivery traveling salesman problem," *Computers & Operations Research* ,29 (2002) 1129-1141.
9. Jih W-R, Hsu Y-J, "Dynamic Vehicle Routing Using Hybrid Genetic Algorithms," *1999 IEEE International Conference on Robotics & Automation*, (2000) pp 453–458.
10. Jung SJ, Haghani A., "Genetic algorithm for a pickup and delivery problem with time windows," *Transportation Network Planning Transportation Research Record* (2000)1-7 2000
11. Schöonberger J, Kopfer H, Mattfeld DC, "A combined approach to solve the Pickup and Delivery Selection Problem," *Operations Research Proceedings*, (2002) pp 150–155.
12. Barrie M. Baker, M.A. Ayechev, "A genetic algorithm for the vehicle routing problem," *Computers & Operations Research*, (2003) 787–800
13. Giselher Pankratz, "A grouping genetic algorithm for the pickup and delivery problem with time windows," *OR Spectrum* (2005) 27: 21–41
14. 楊馥豪,「時窗限制下混合收送貨之配送路線規劃」,立德管理學院應用資訊所碩士論文,民國九十五年六月。
15. 楊雅斐,「使用改良式遺傳演算法求解車輛途程問題」,立德管理學院應用資訊所碩士論文,民國九十四年六月。

16. 黃信翔，「解決具時間窗限制的提送貨問題」，國立交通大學運輸科技與管理學系碩士論文，民國九十五年六月。

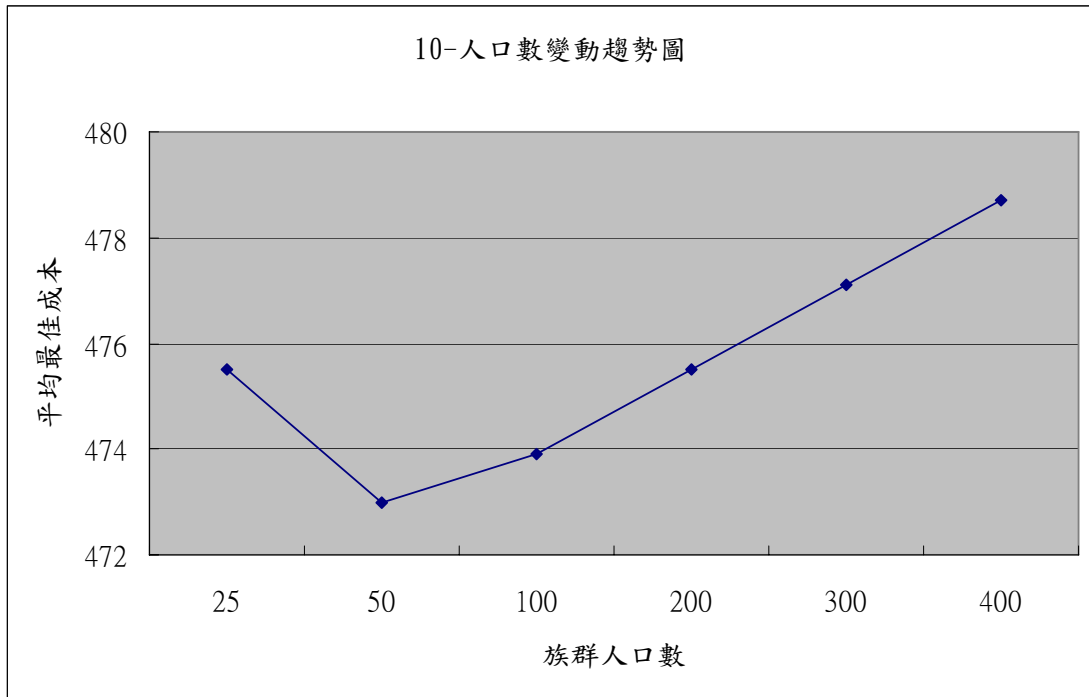


附 錄

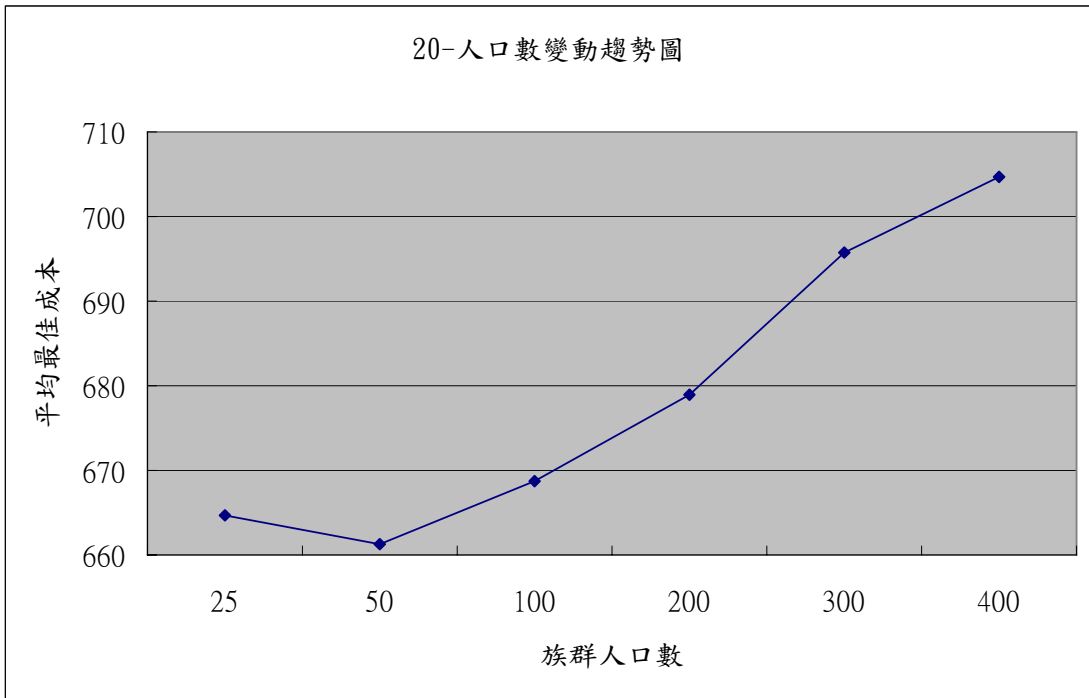
敏感度分析，以 10 個節點為例。

1. 人口數變動趨勢圖

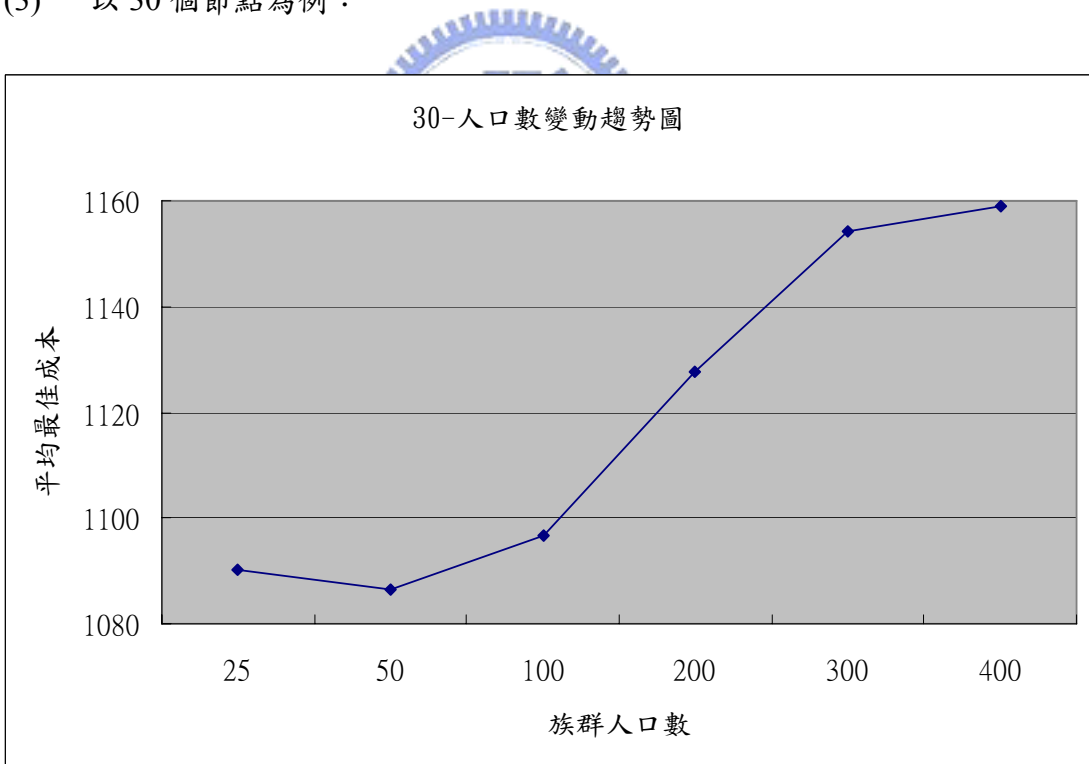
(1) 以 10 個節點為例：



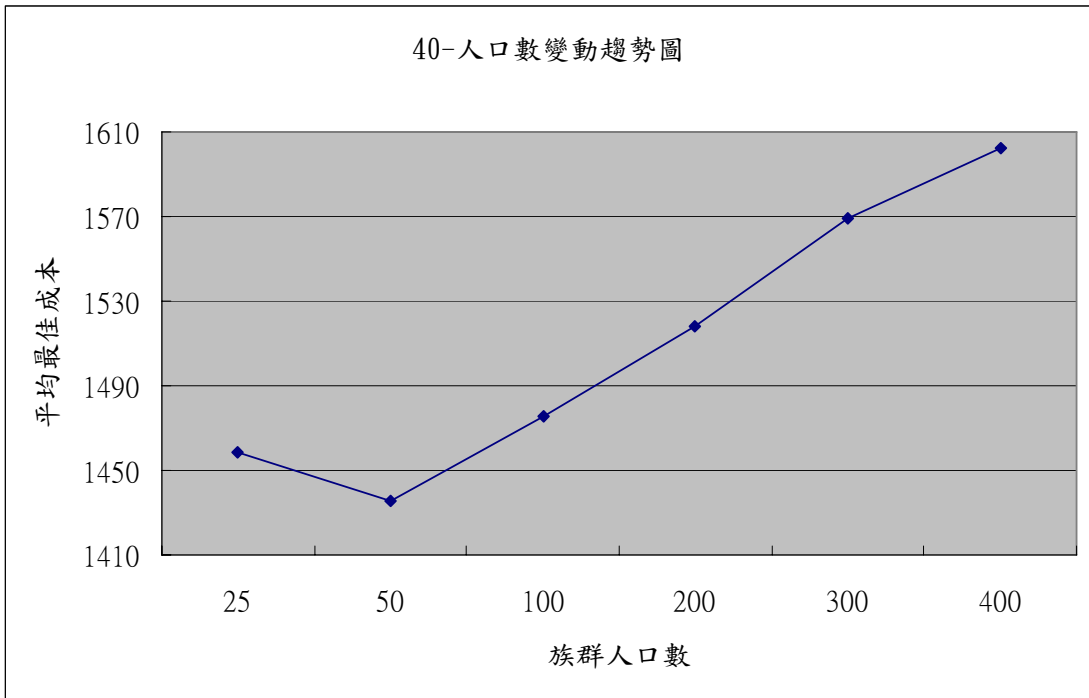
(2) 以 20 個節點為例：



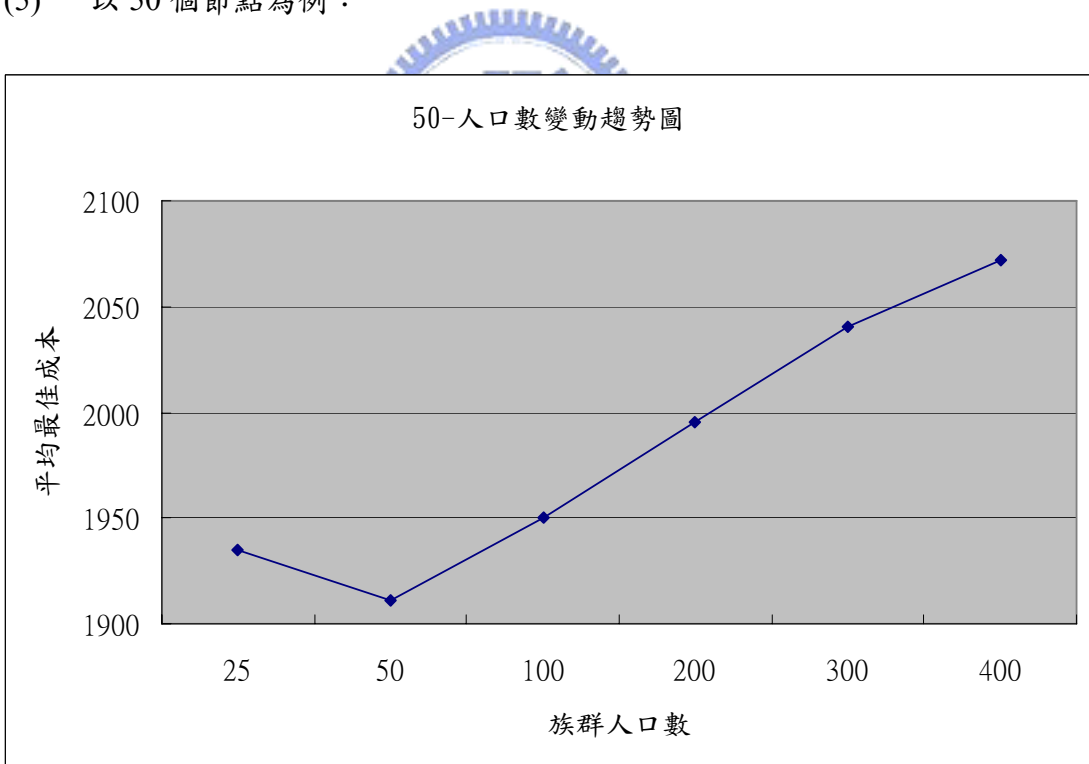
(3) 以 30 個節點為例：



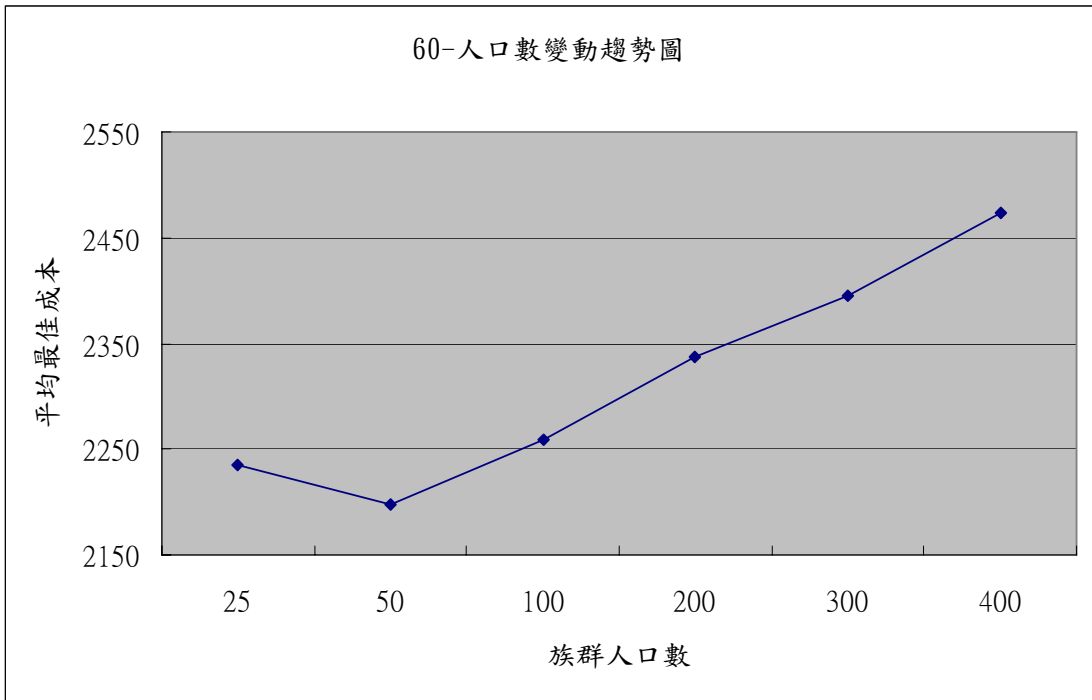
(4) 以 40 個節點為例：



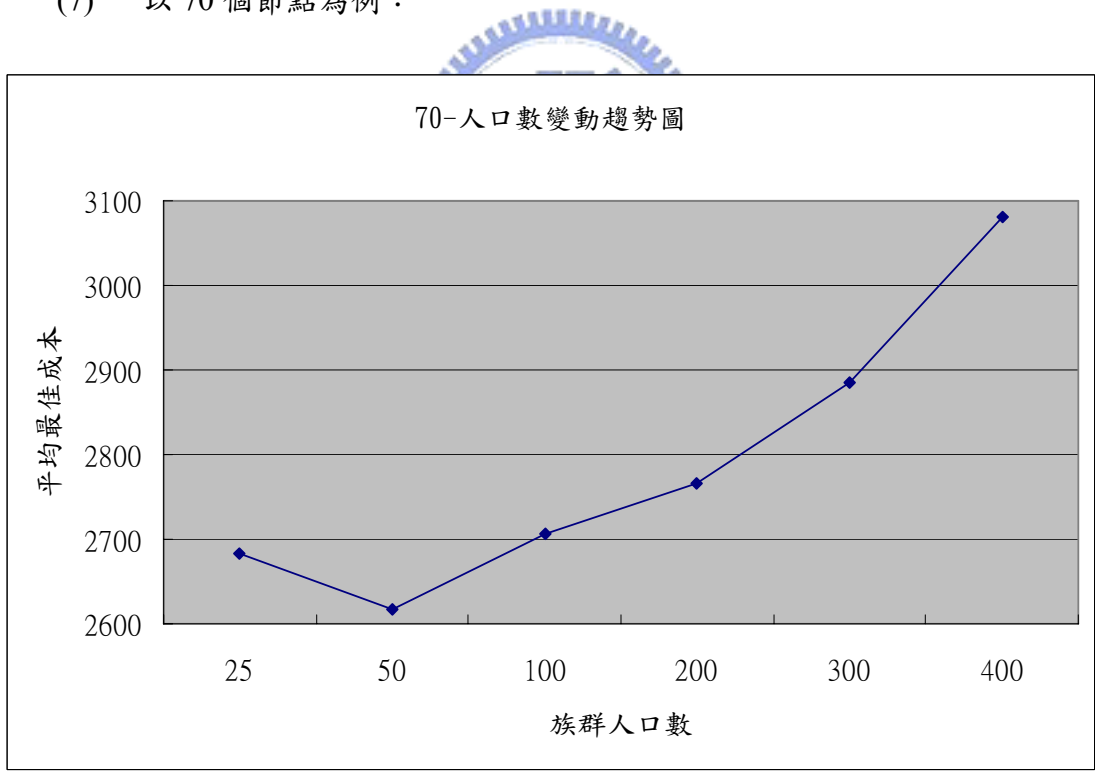
(5) 以 50 個節點為例：



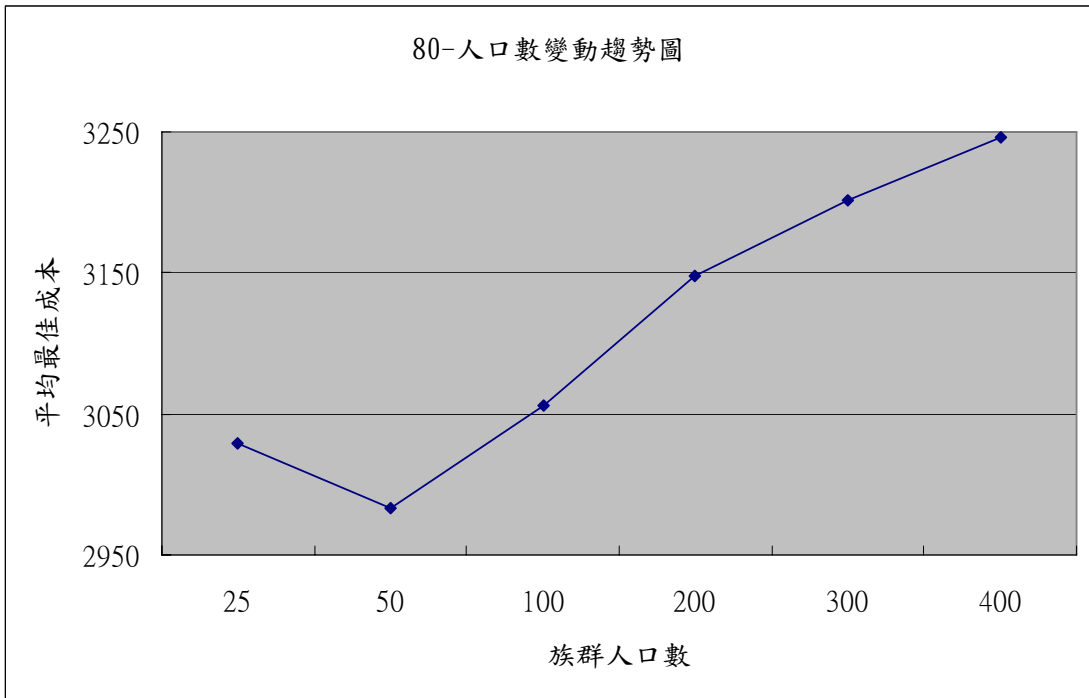
(6) 以 60 個節點為例：



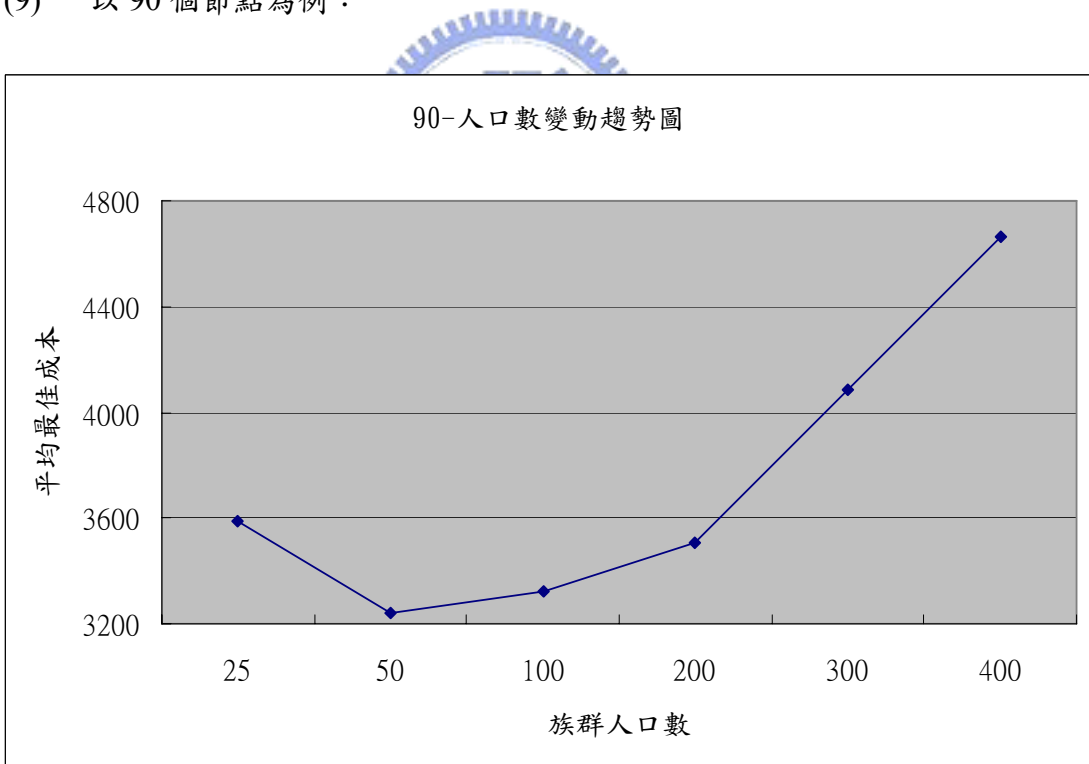
(7) 以 70 個節點為例：



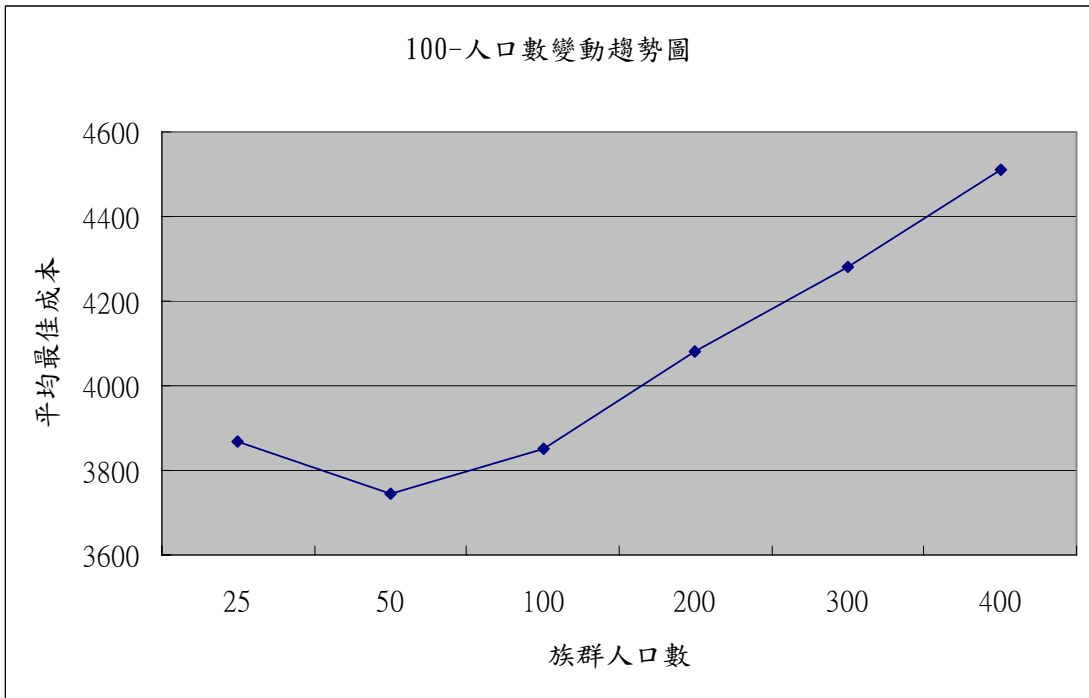
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：

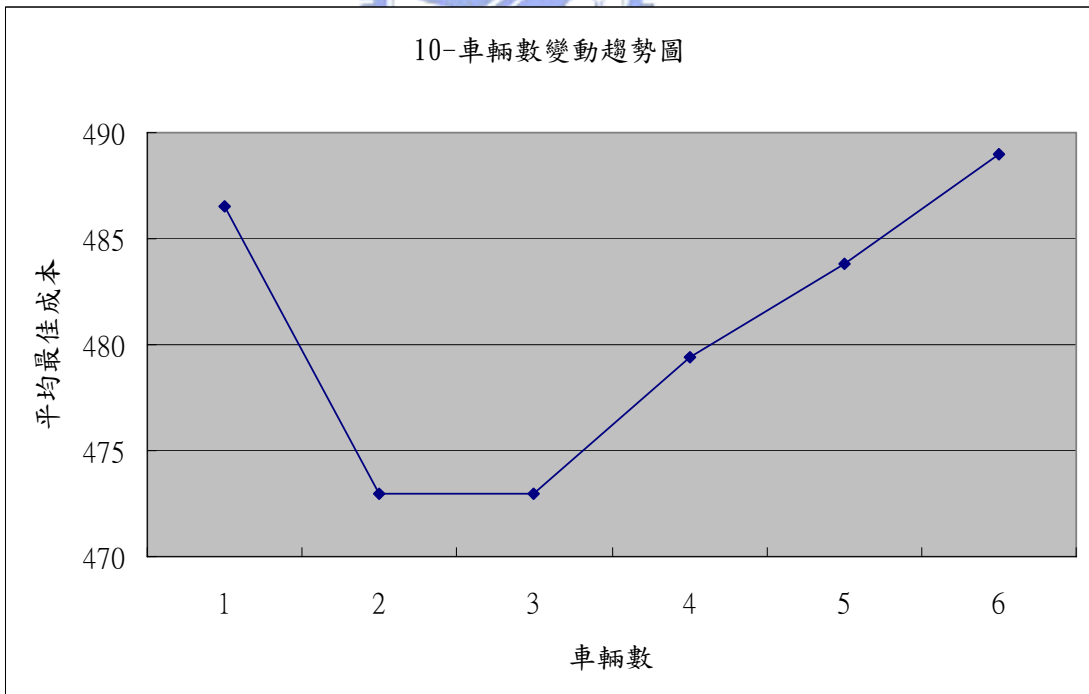


(10) 以 100 個節點為例：

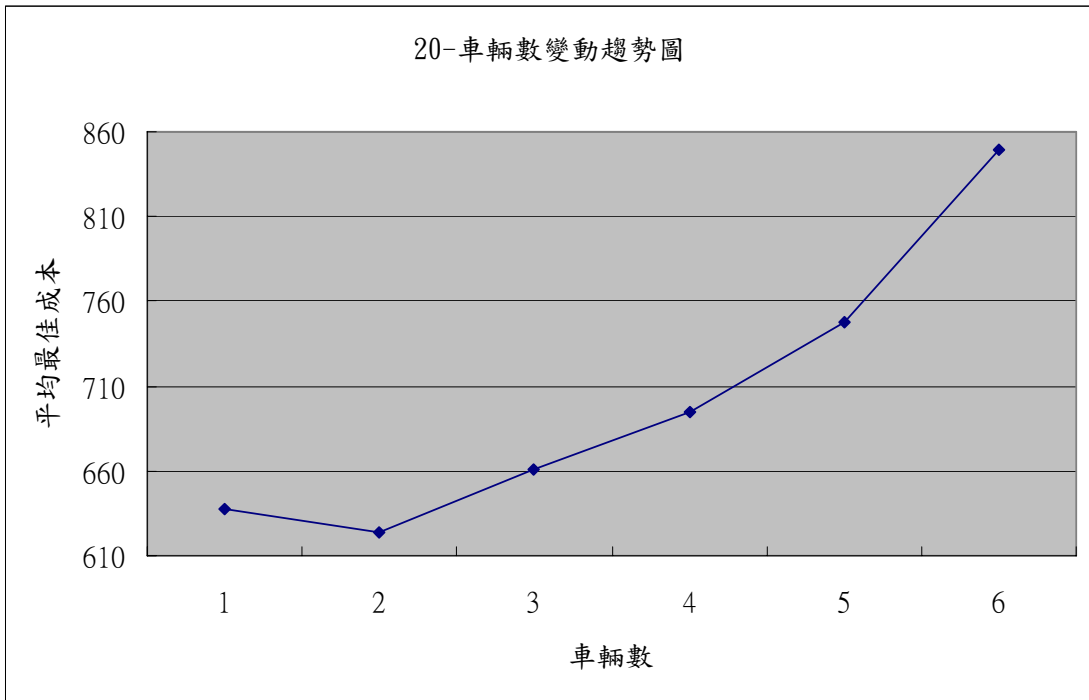


2. 車輛數變動趨勢圖

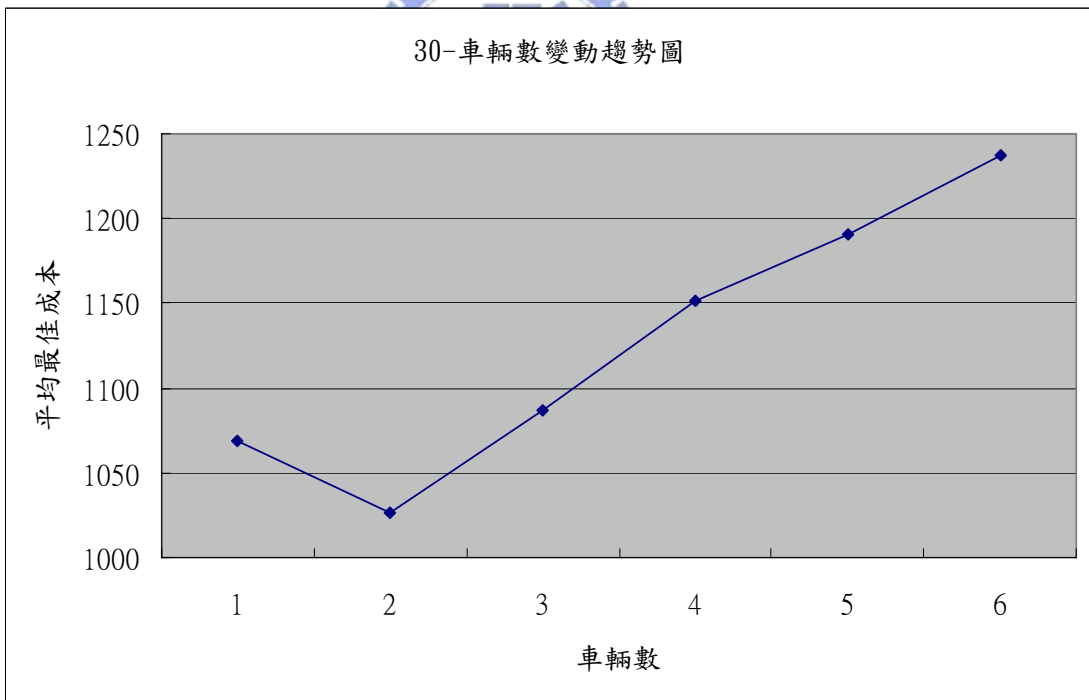
(1) 以 10 個節點為例：



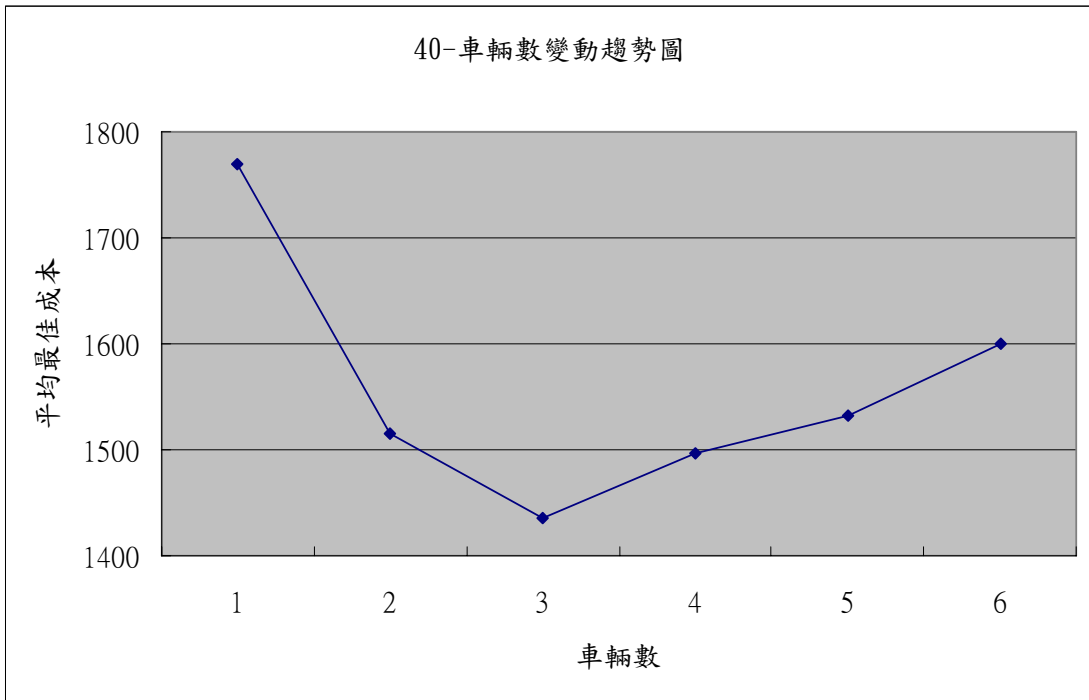
(2) 以 20 個節點為例：



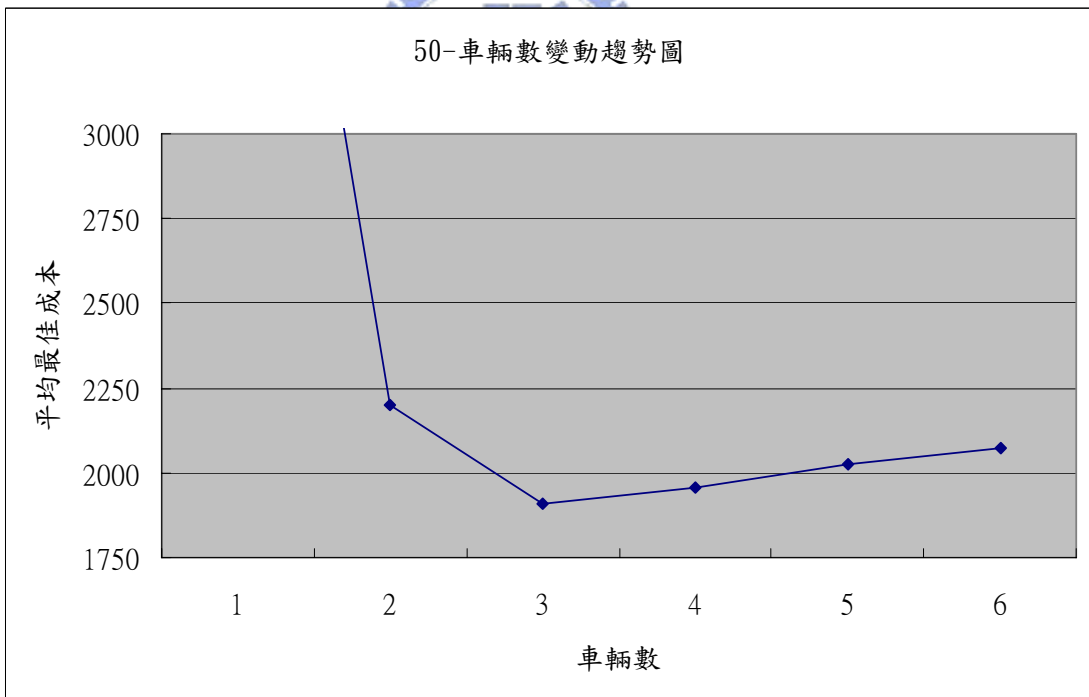
(3) 以 30 個節點為例：



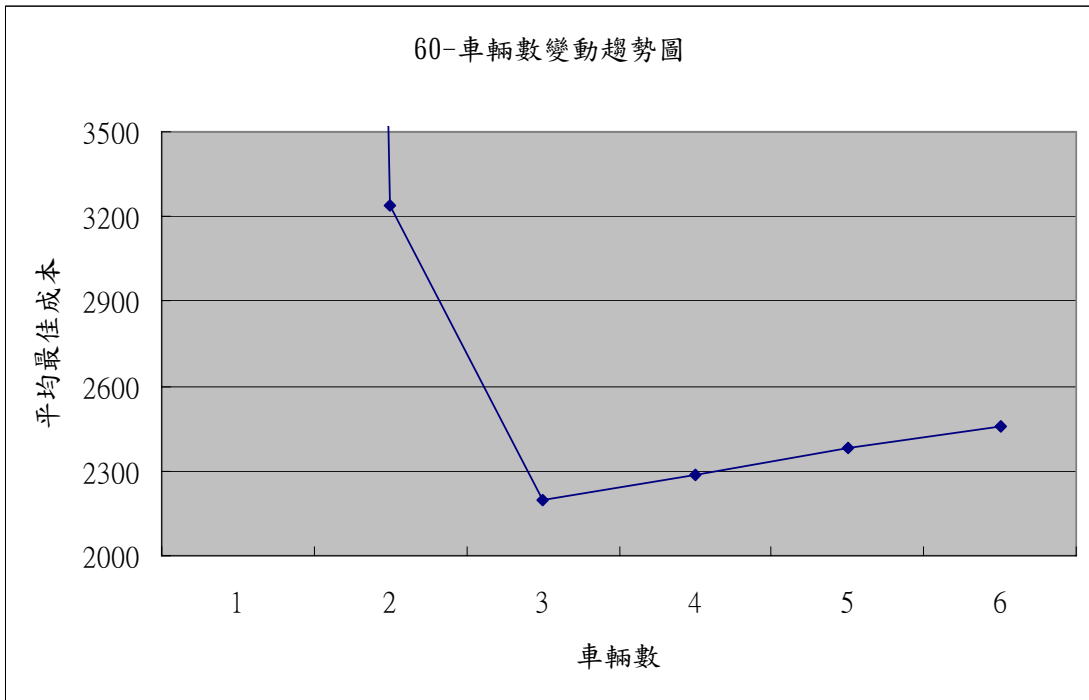
(4) 以 40 個節點為例：



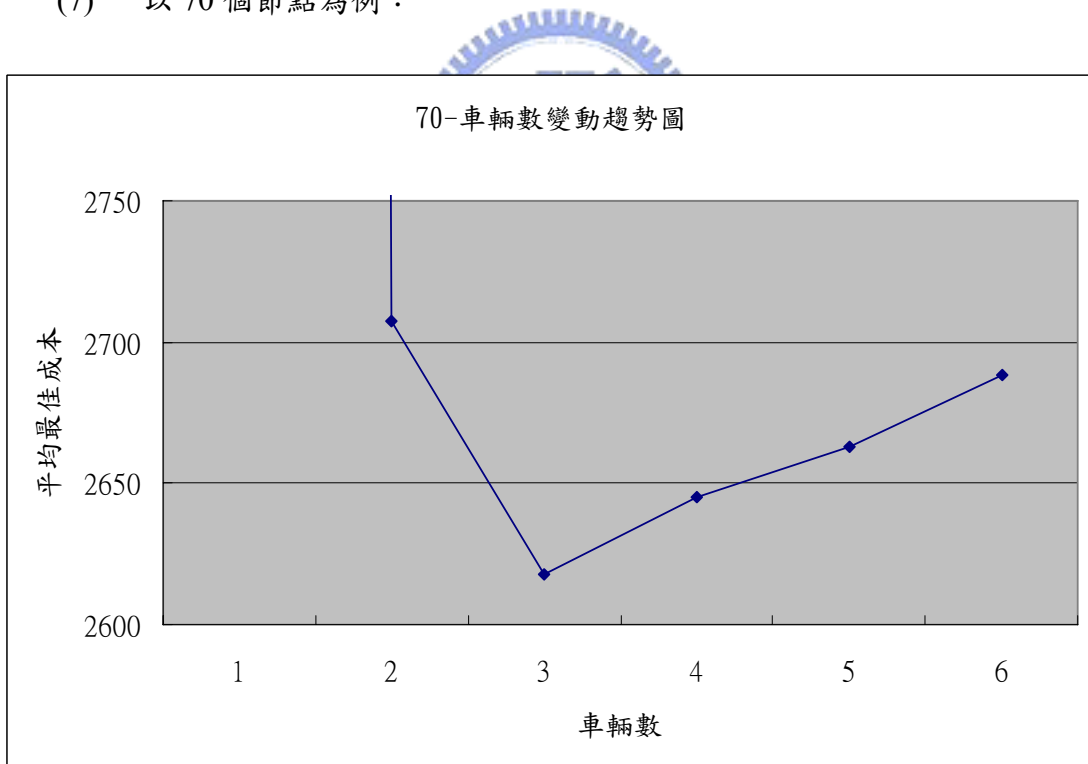
(5) 以 50 個節點為例：



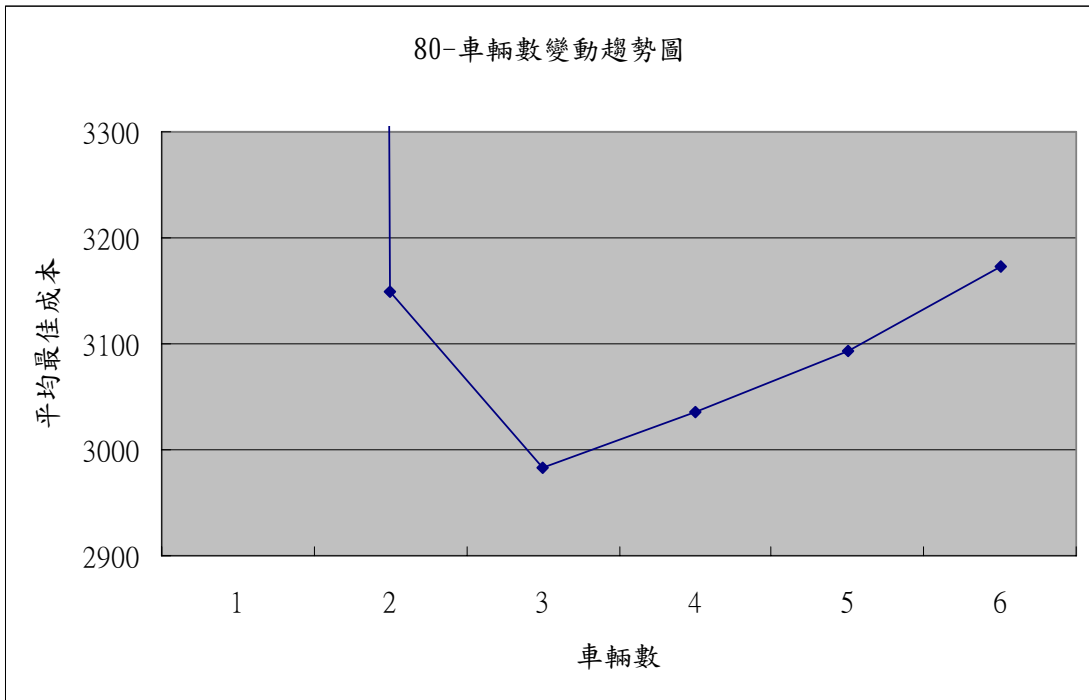
(6) 以 60 個節點為例：



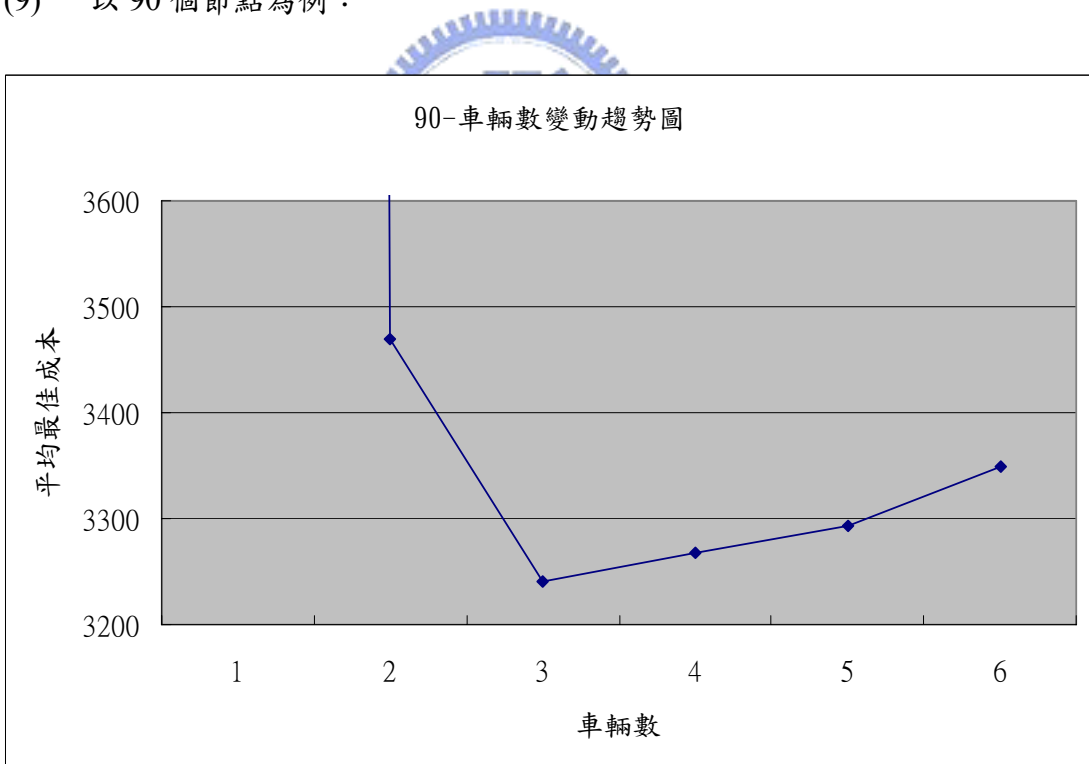
(7) 以 70 個節點為例：



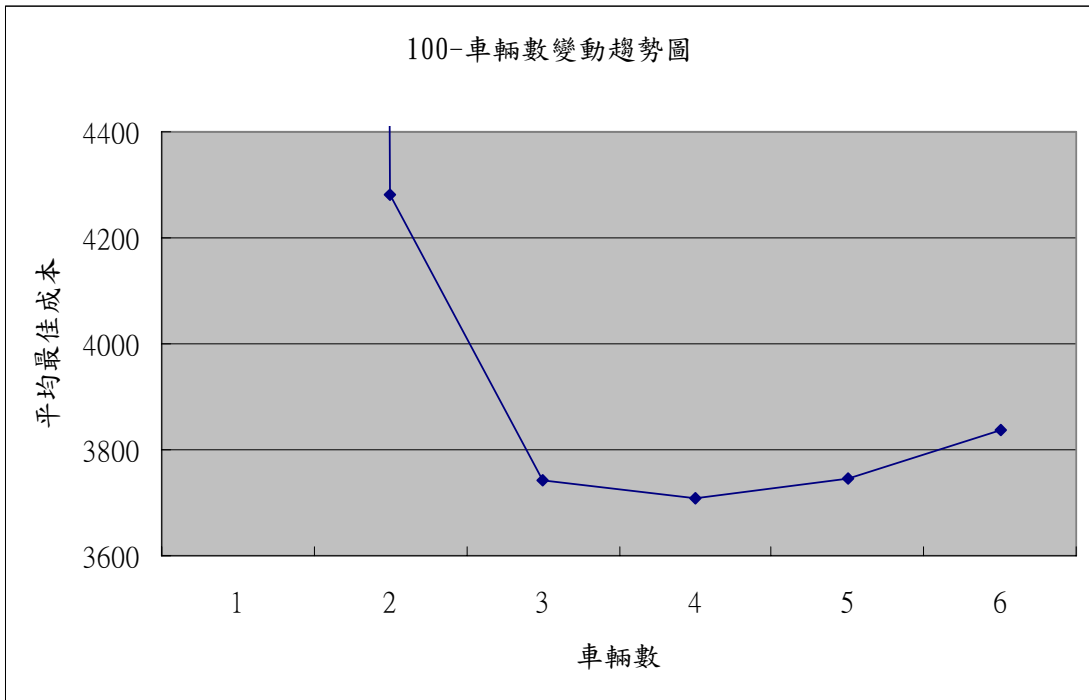
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：

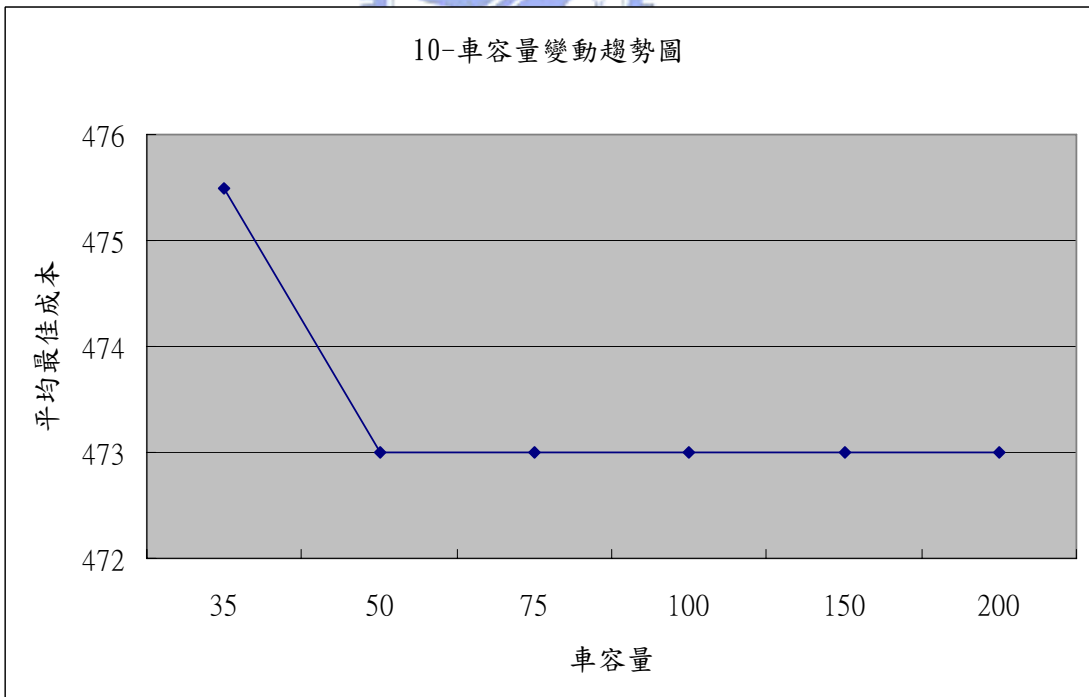


(10) 以 100 個節點為例：

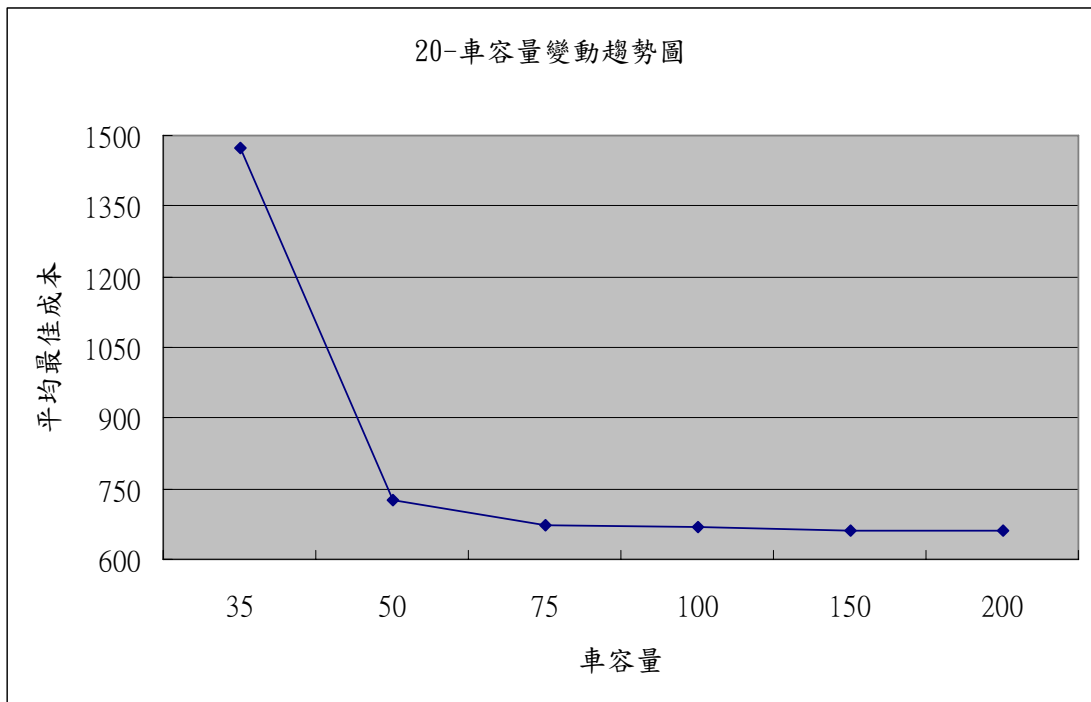


3. 車容量變動趨勢圖

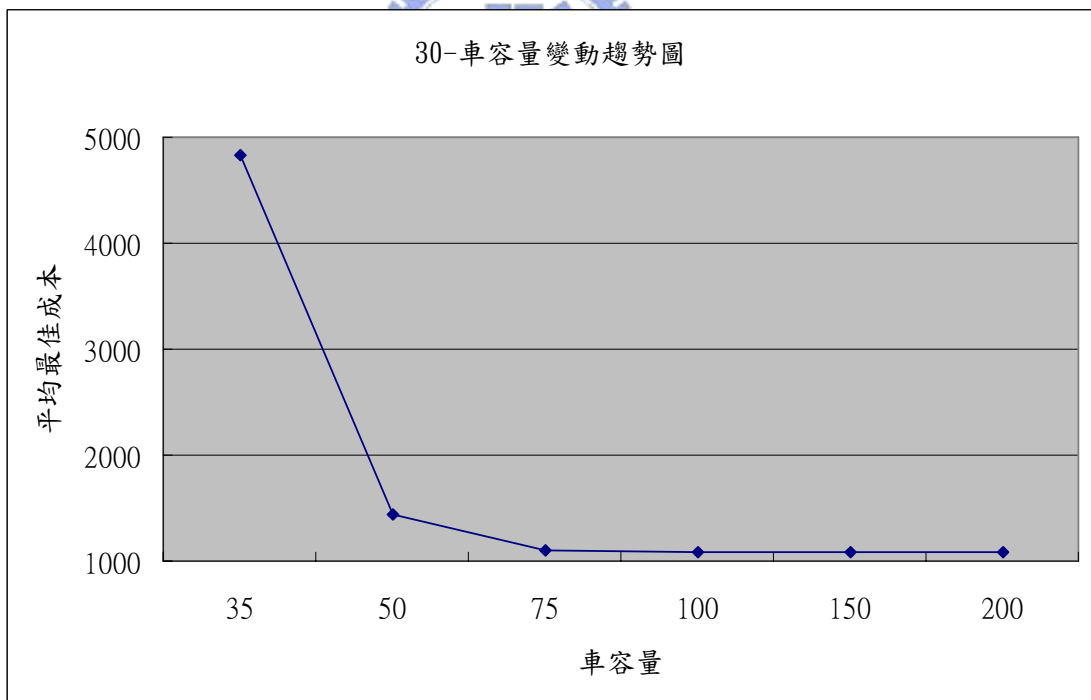
(1) 以 10 個節點為例：



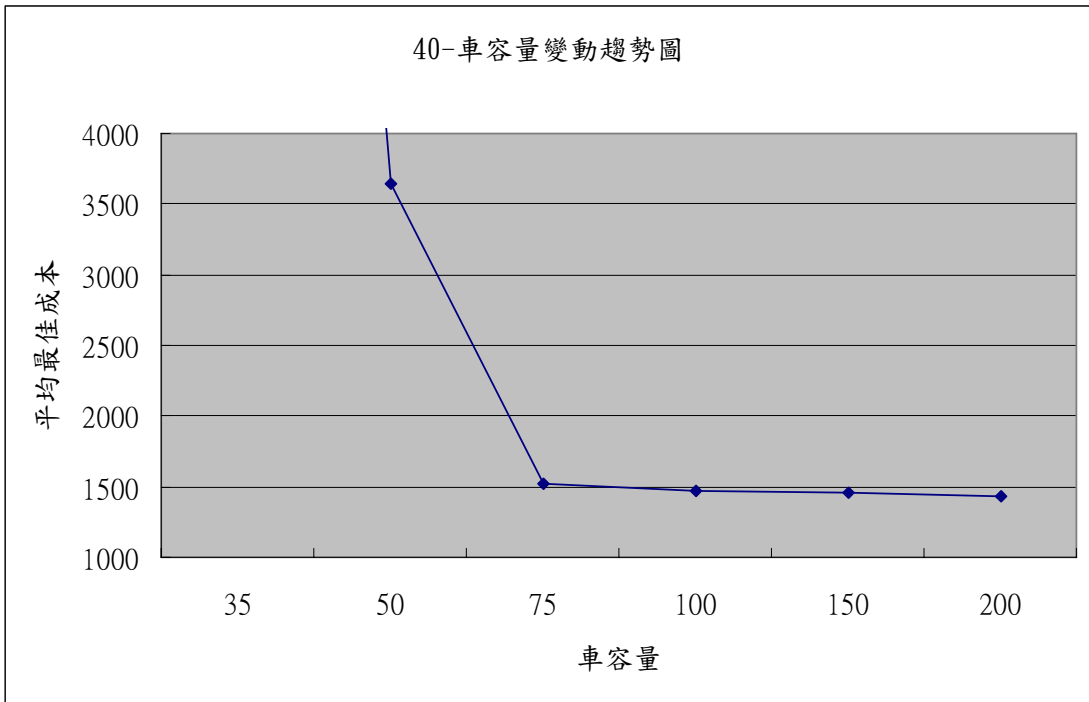
(2) 以 20 個節點為例：



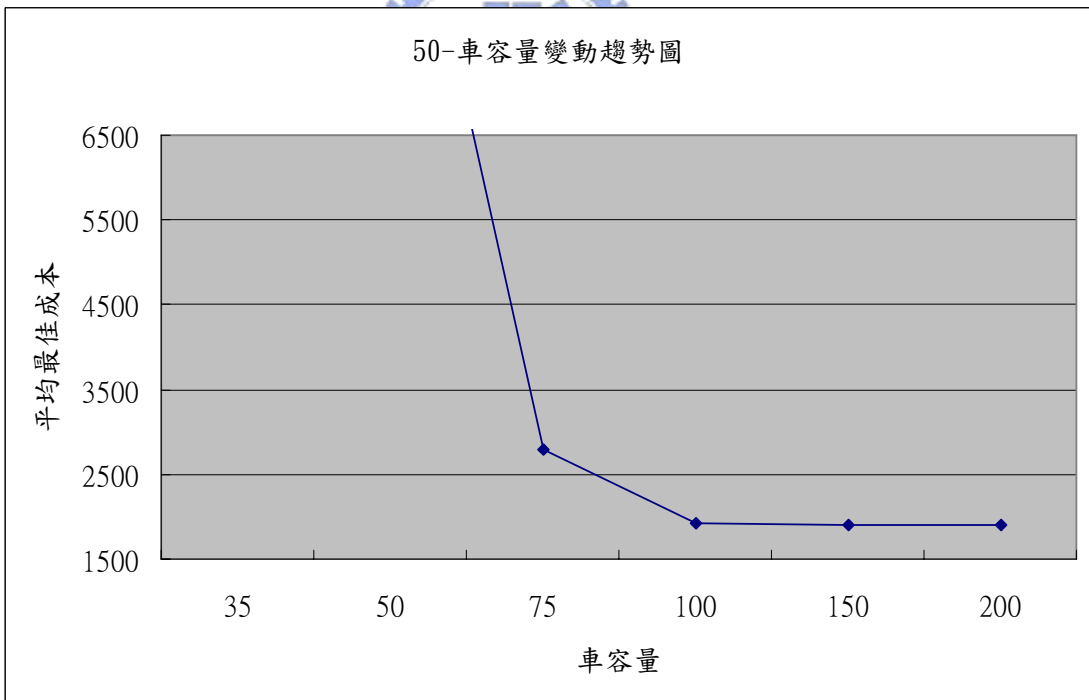
(3) 以 30 個節點為例：



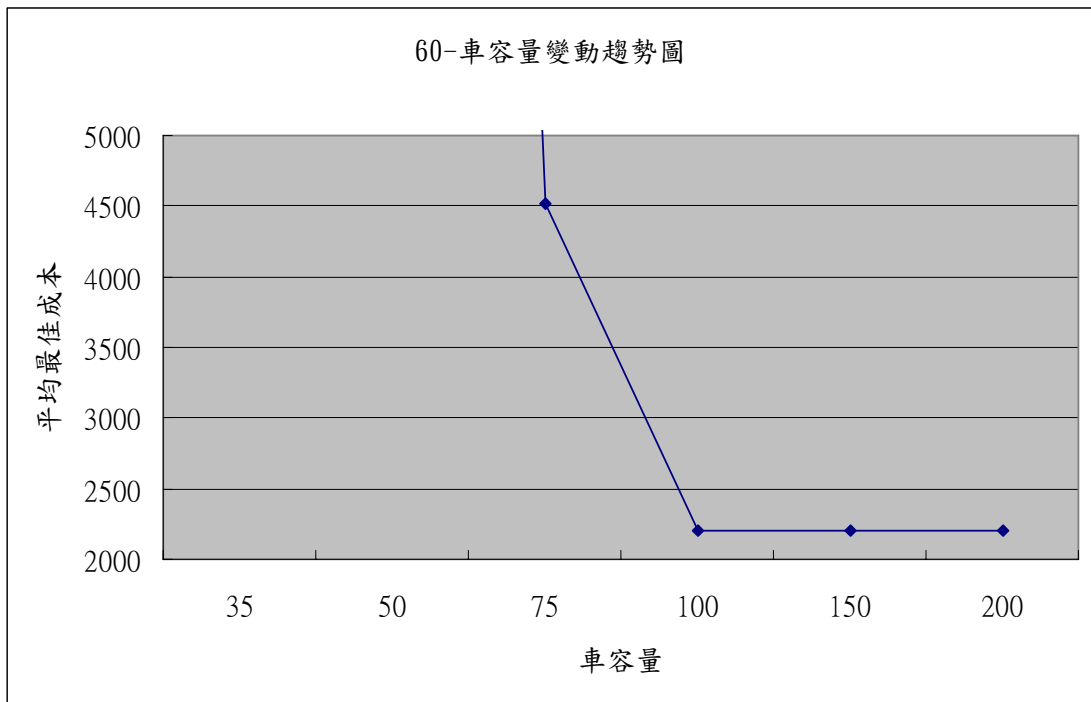
(4) 以 40 個節點為例：



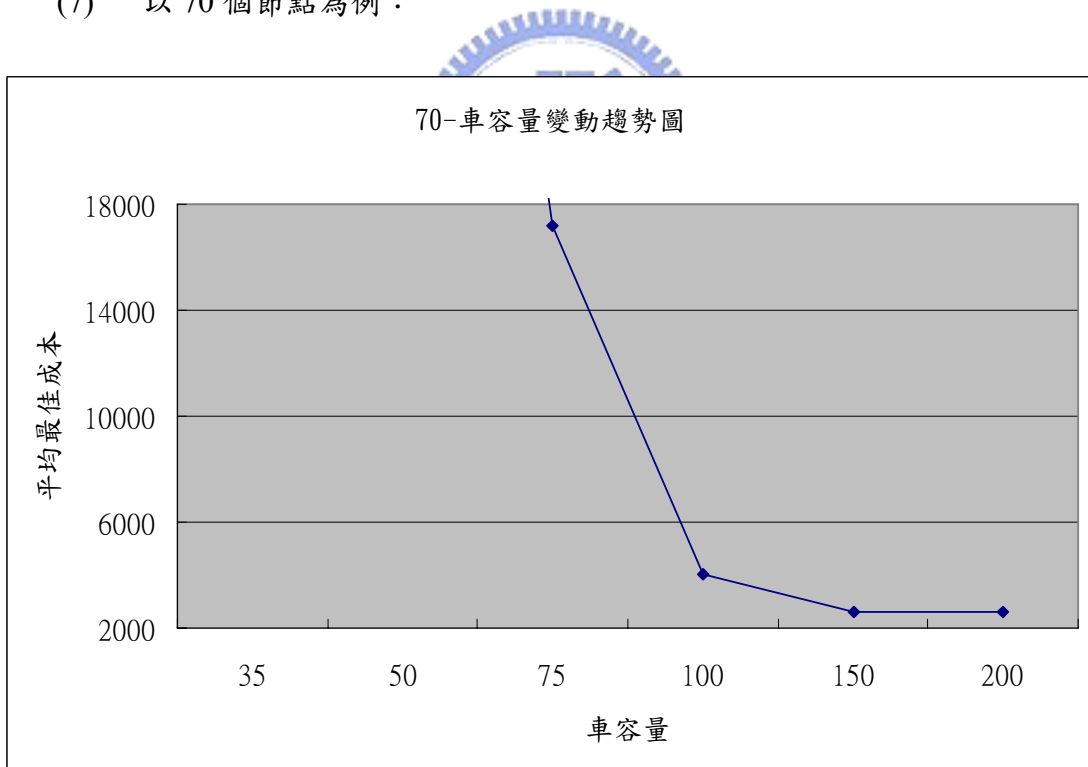
(5) 以 50 個節點為例：



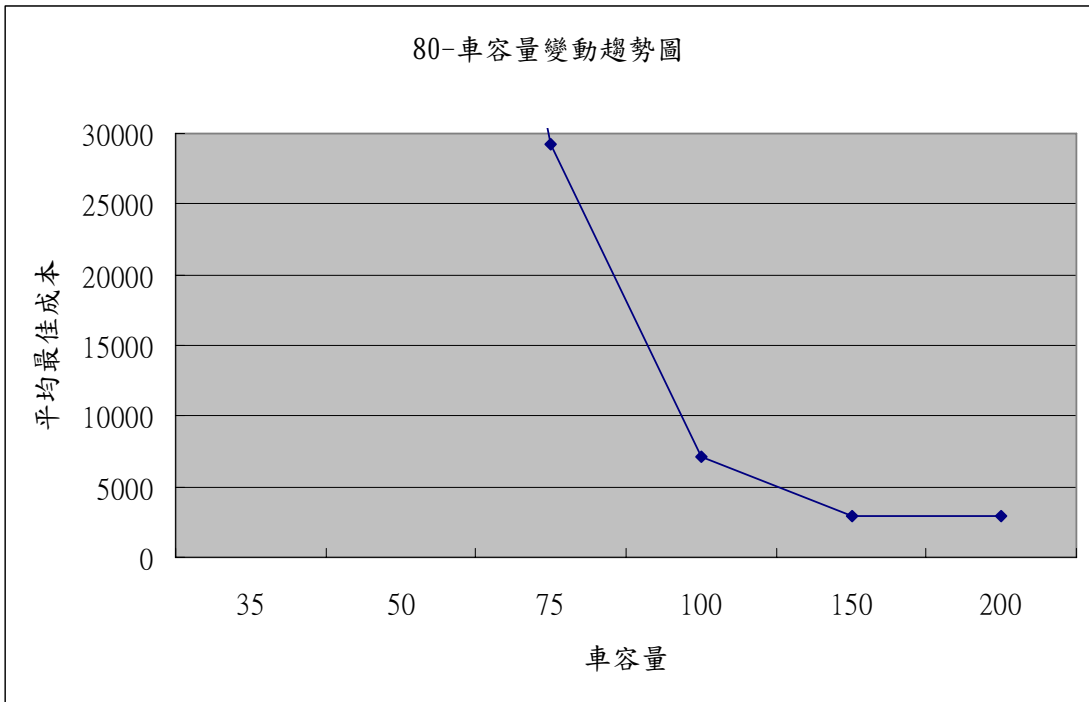
(6) 以 60 個節點為例：



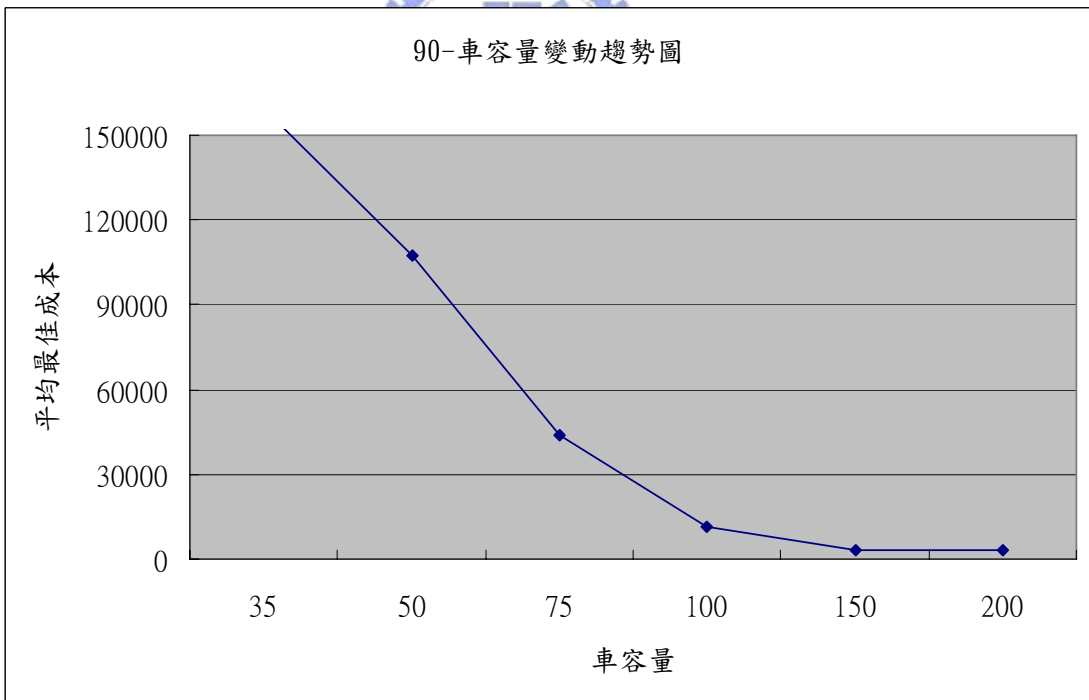
(7) 以 70 個節點為例：



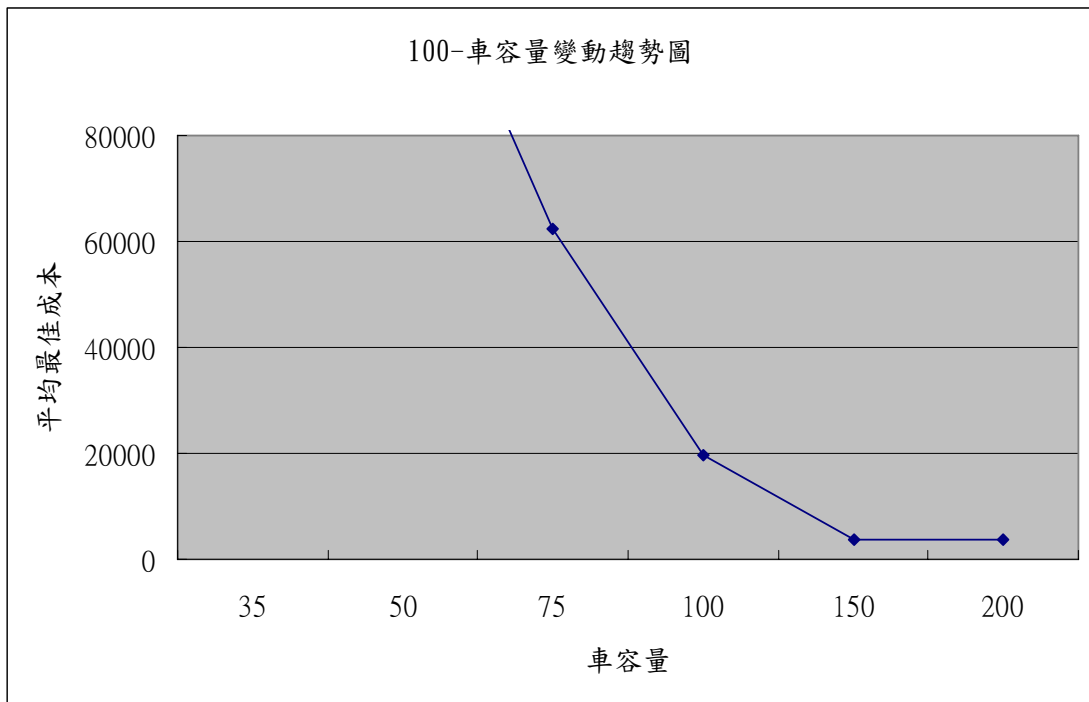
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：

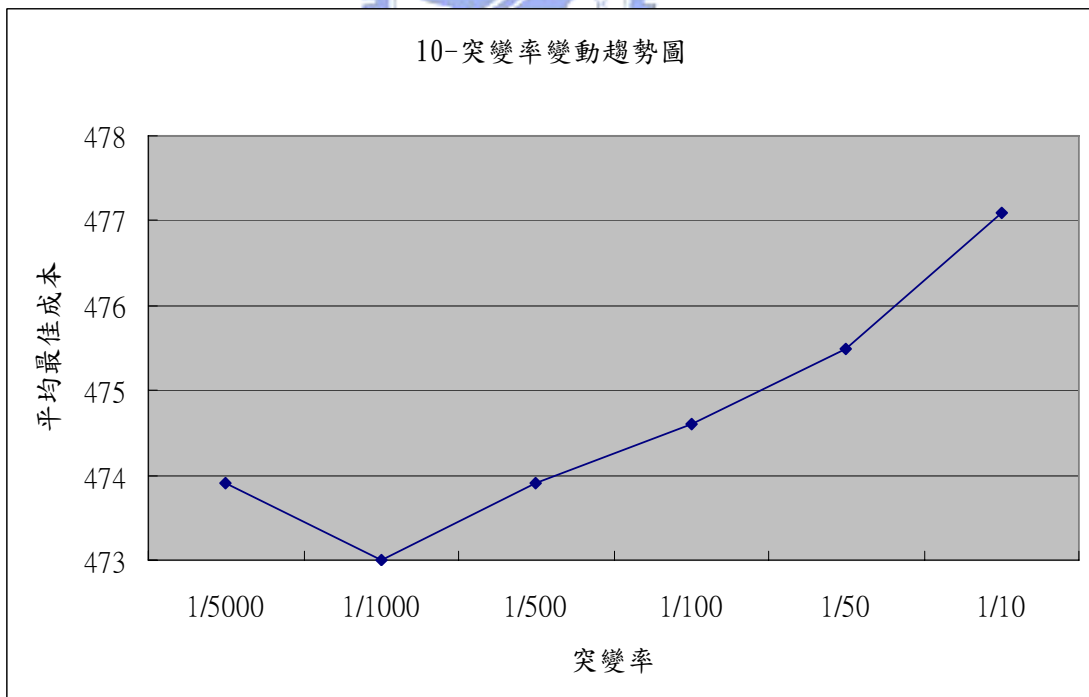


(10) 以 100 個節點為例：

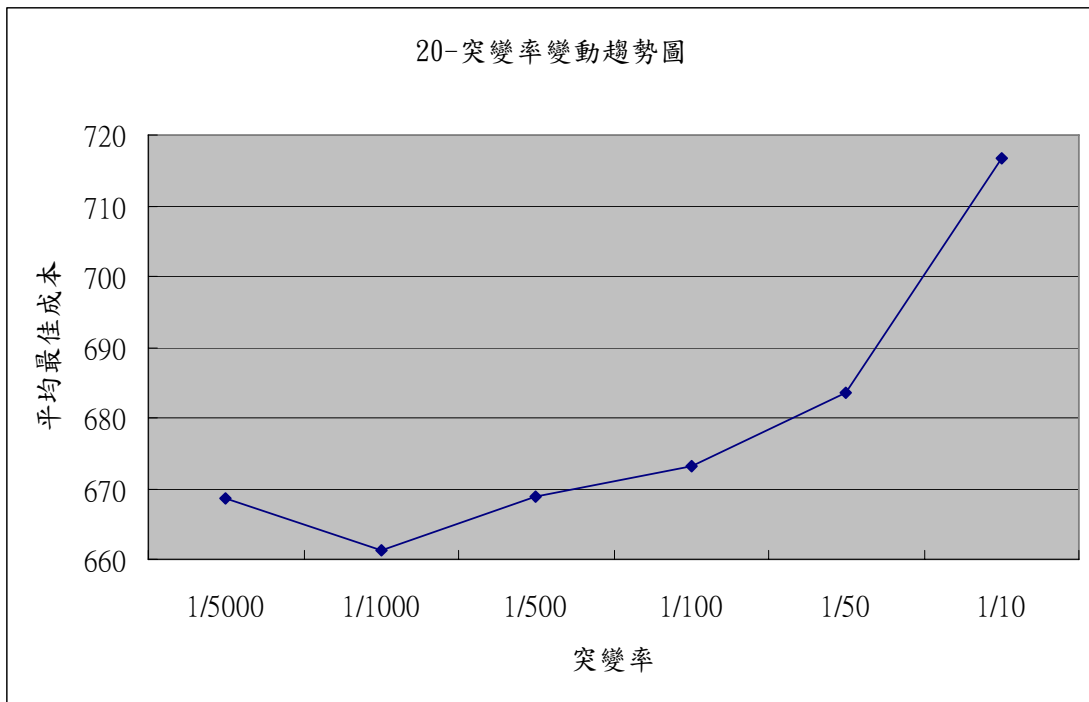


4. 突變率變動趨勢圖

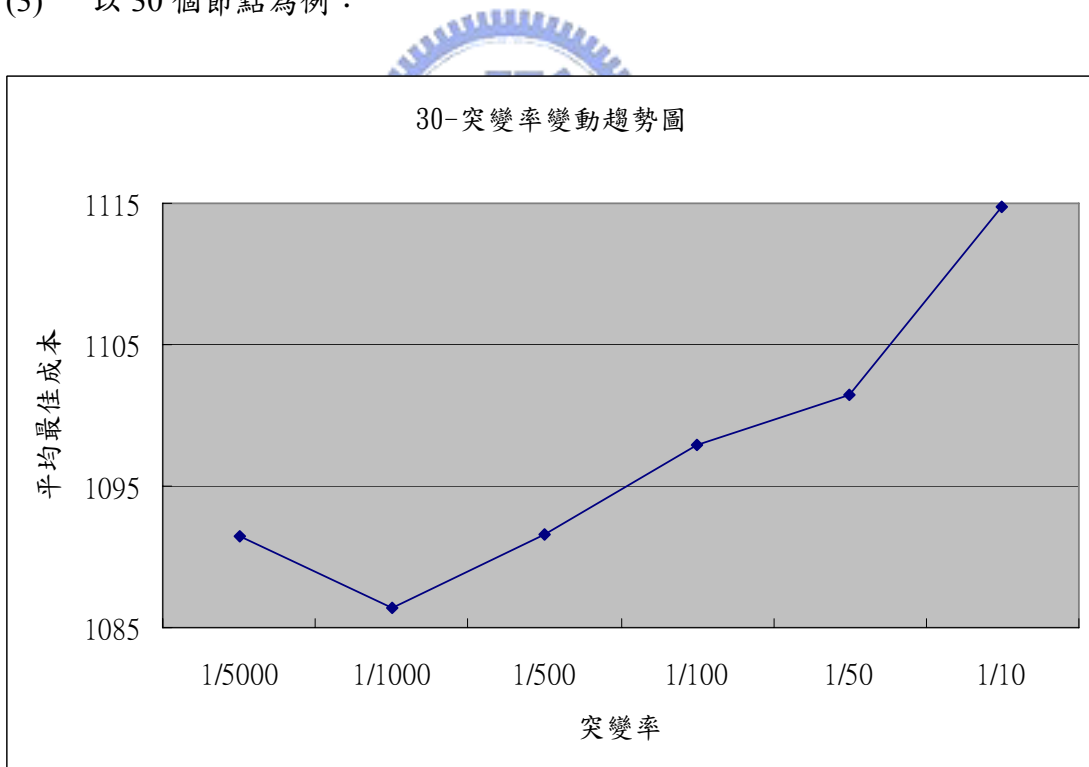
(1) 以 10 個節點為例：



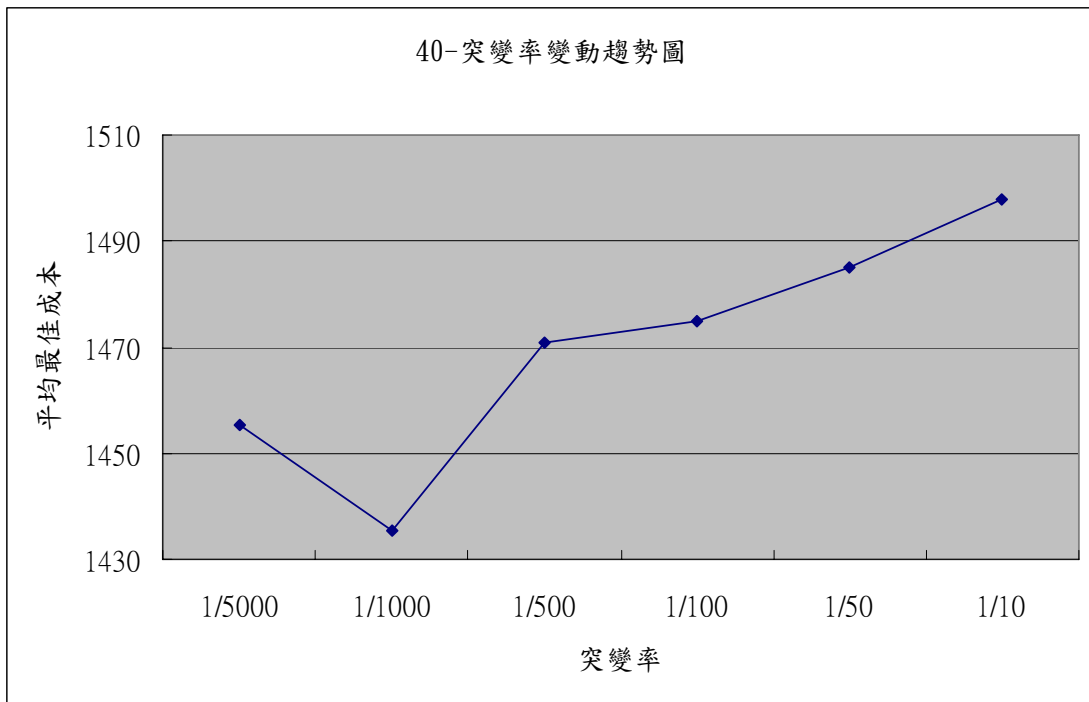
(2) 以 20 個節點為例：



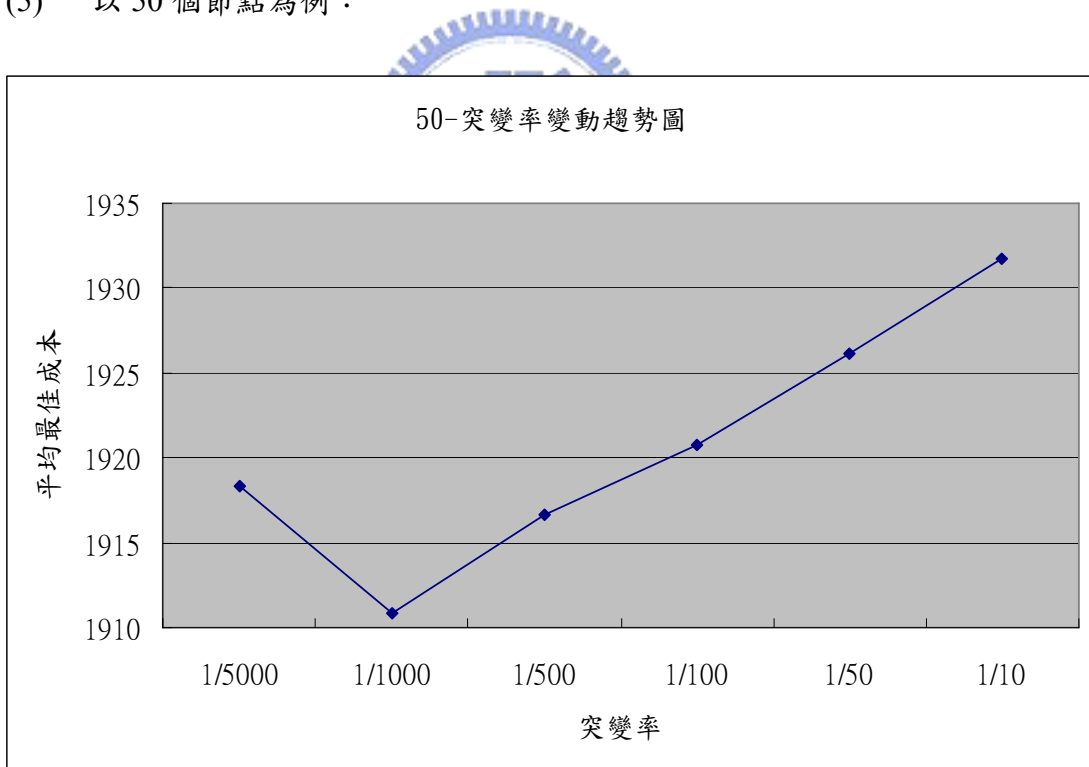
(3) 以 30 個節點為例：



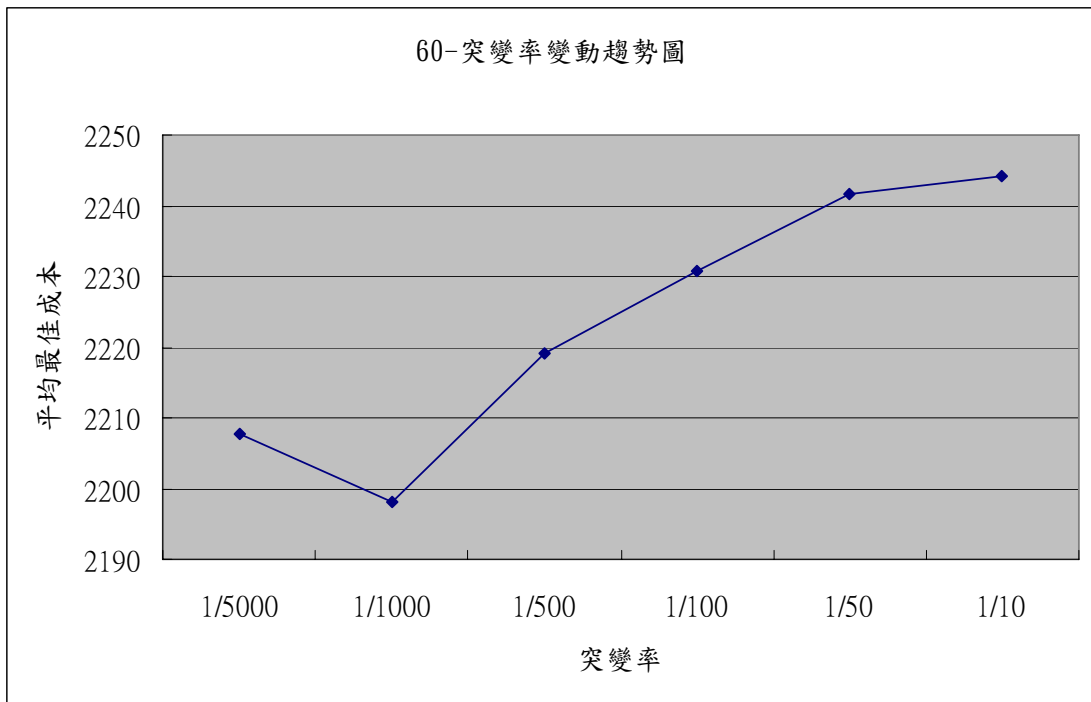
(4) 以 40 個節點為例：



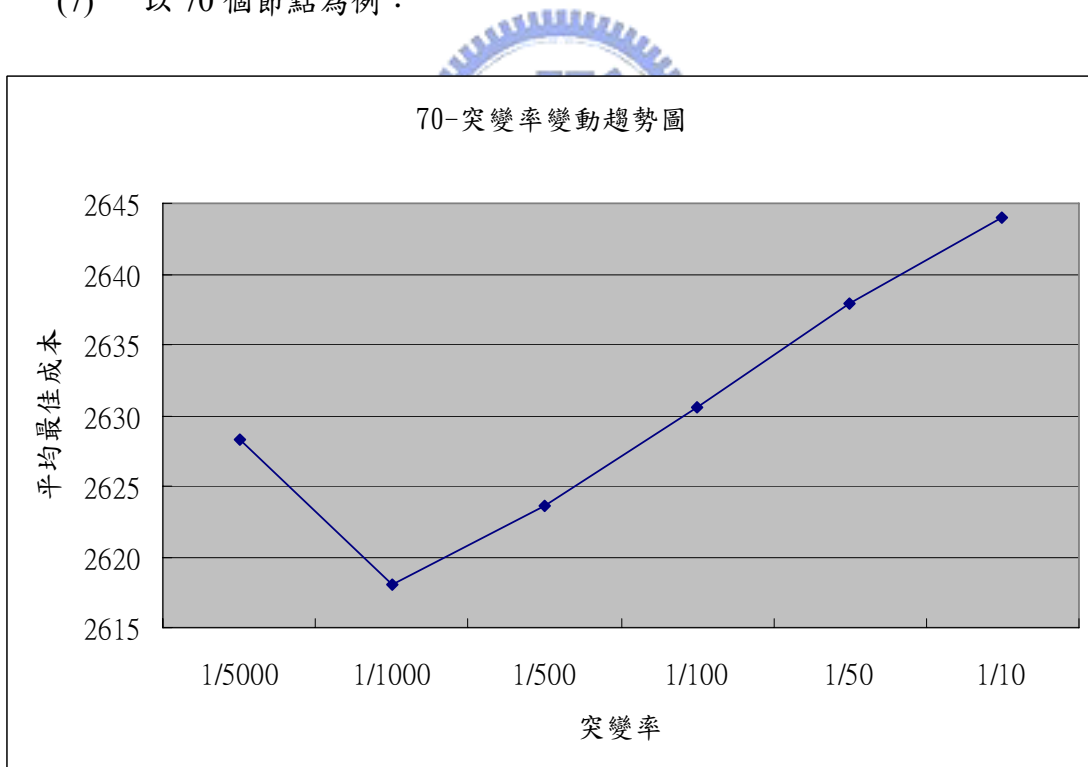
(5) 以 50 個節點為例：



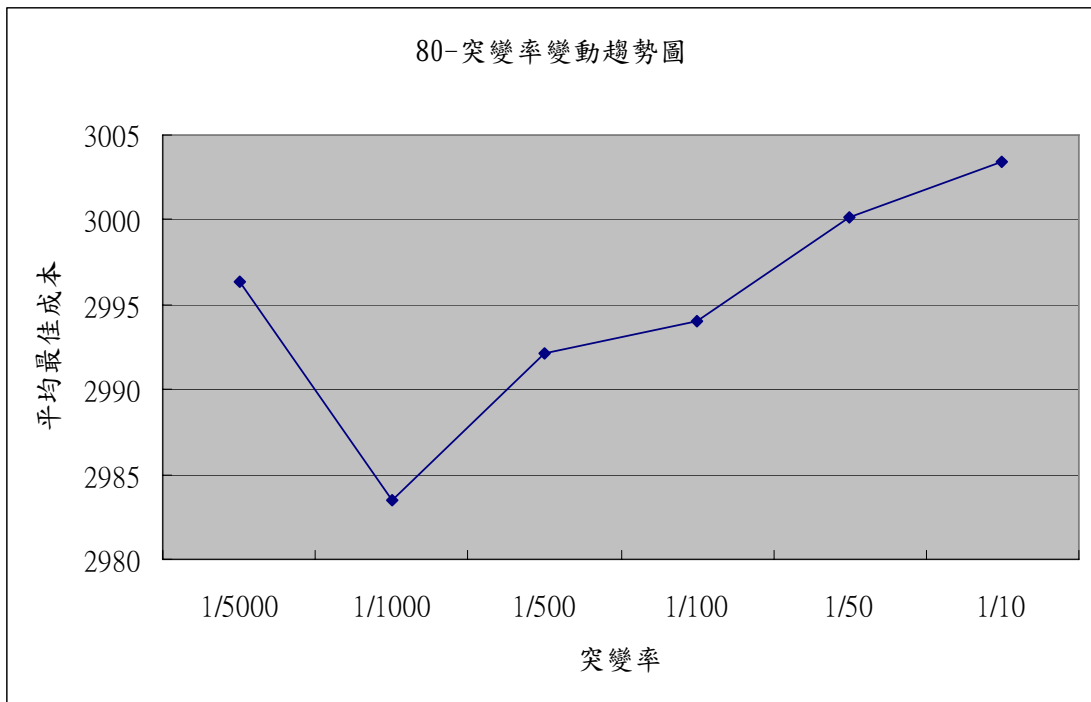
(6) 以 60 個節點為例：



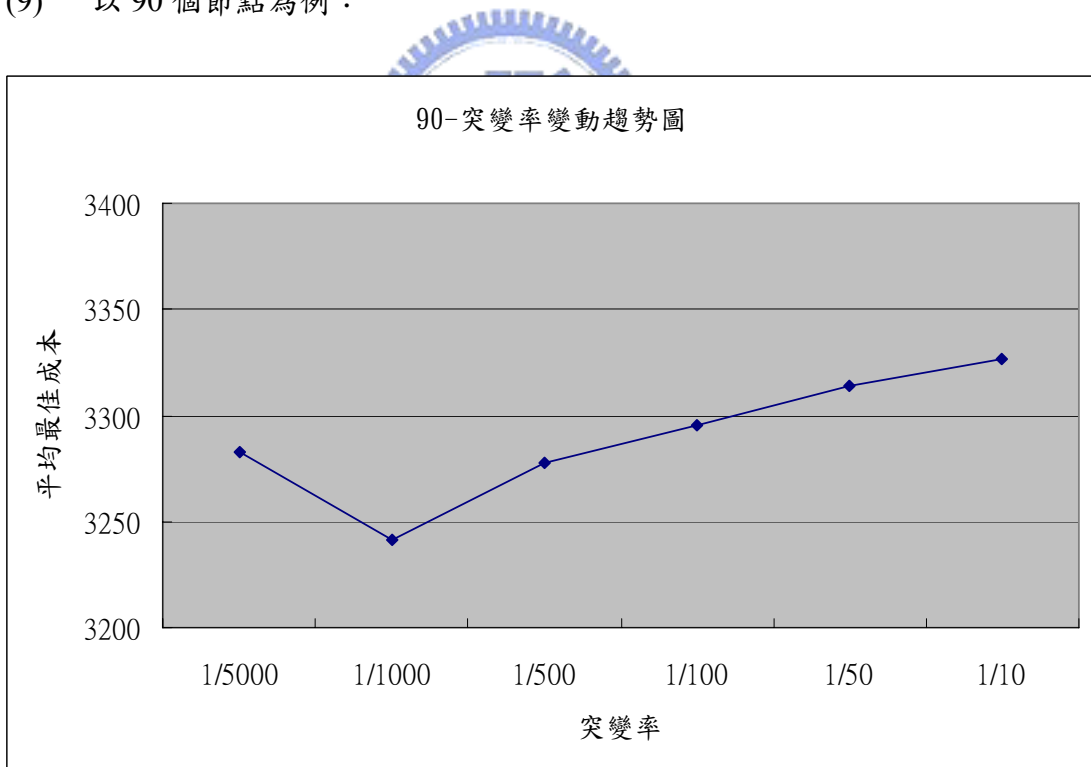
(7) 以 70 個節點為例：



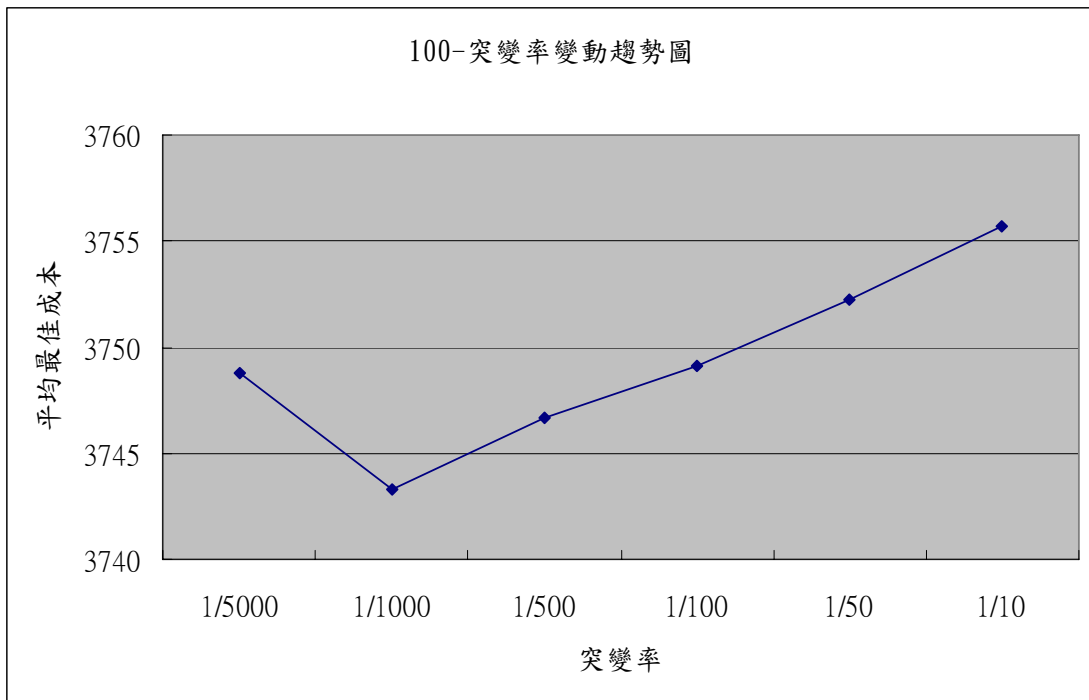
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：

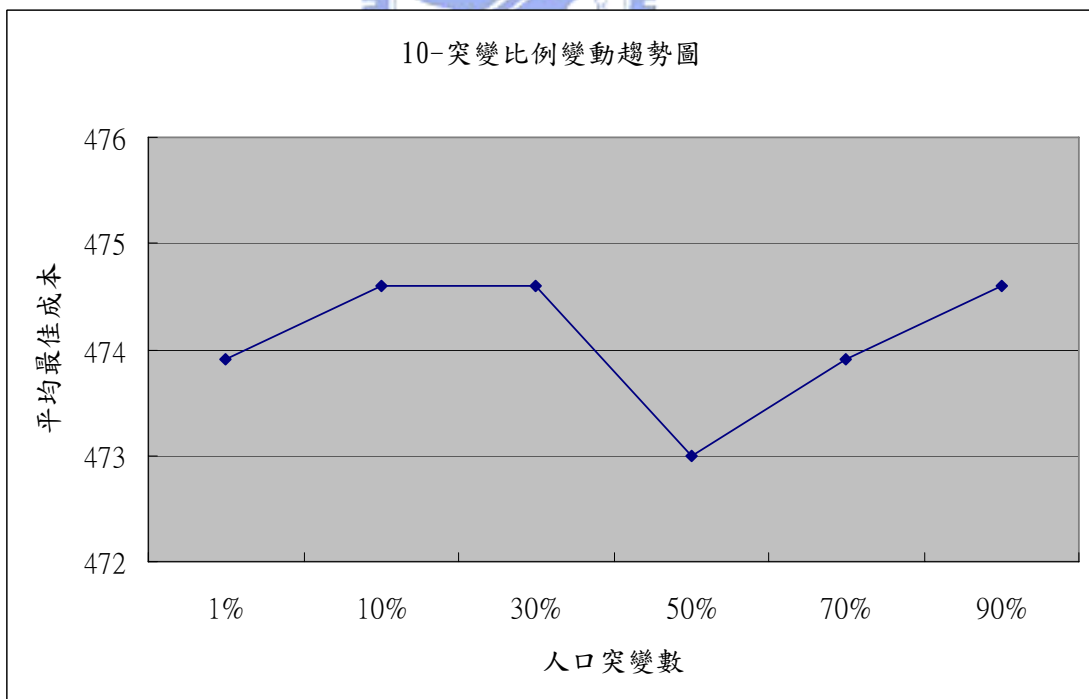


(10) 以 100 個節點為例：

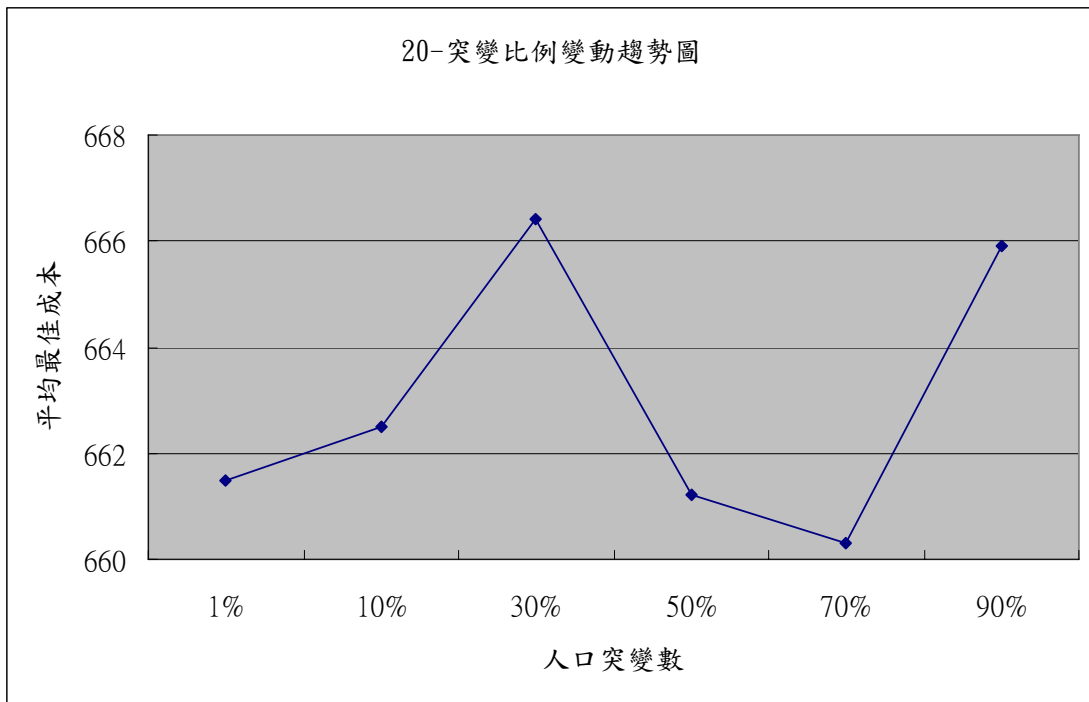


5. 突變比例變動趨勢圖

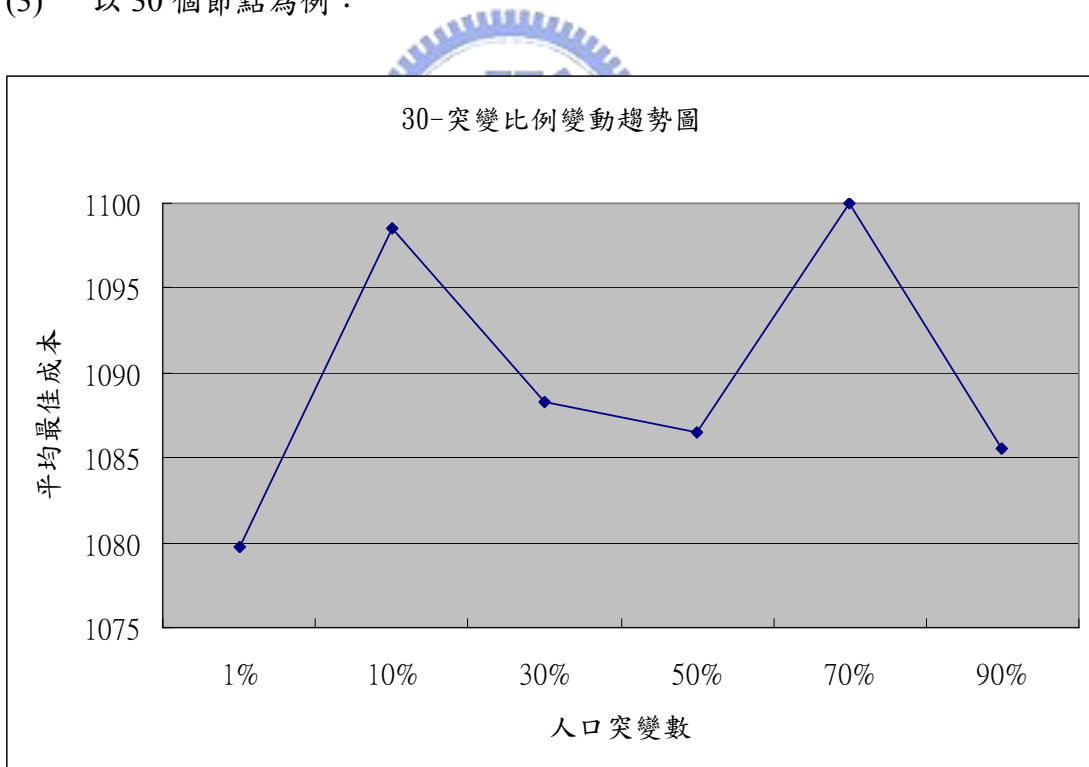
(1) 以 10 個節點為例：



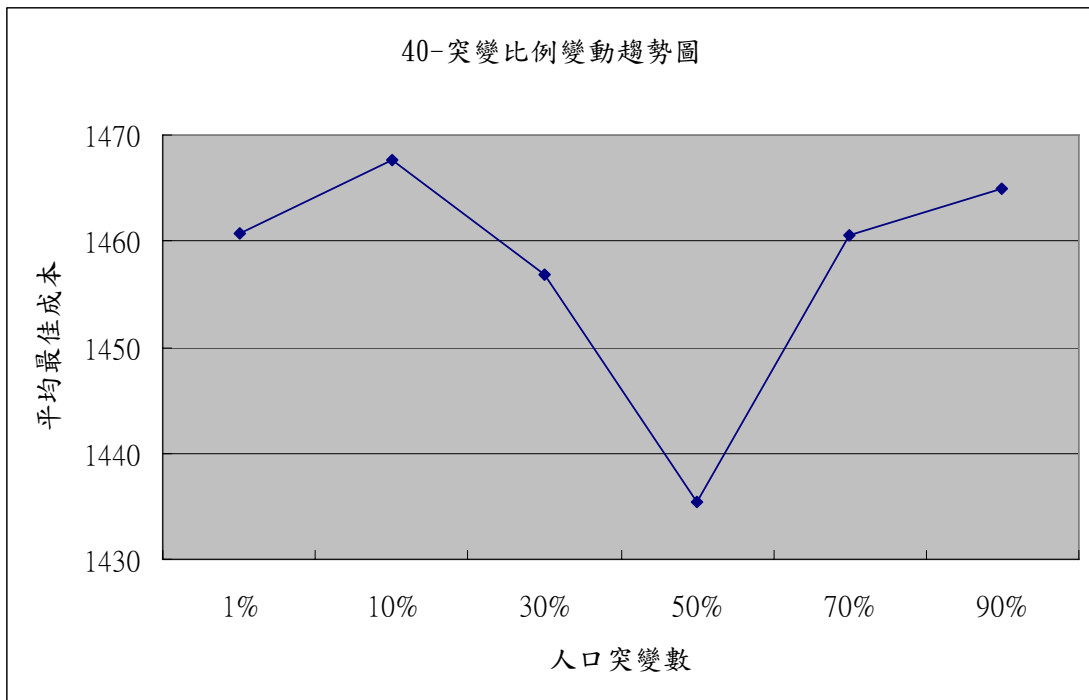
(2) 以 20 個節點為例：



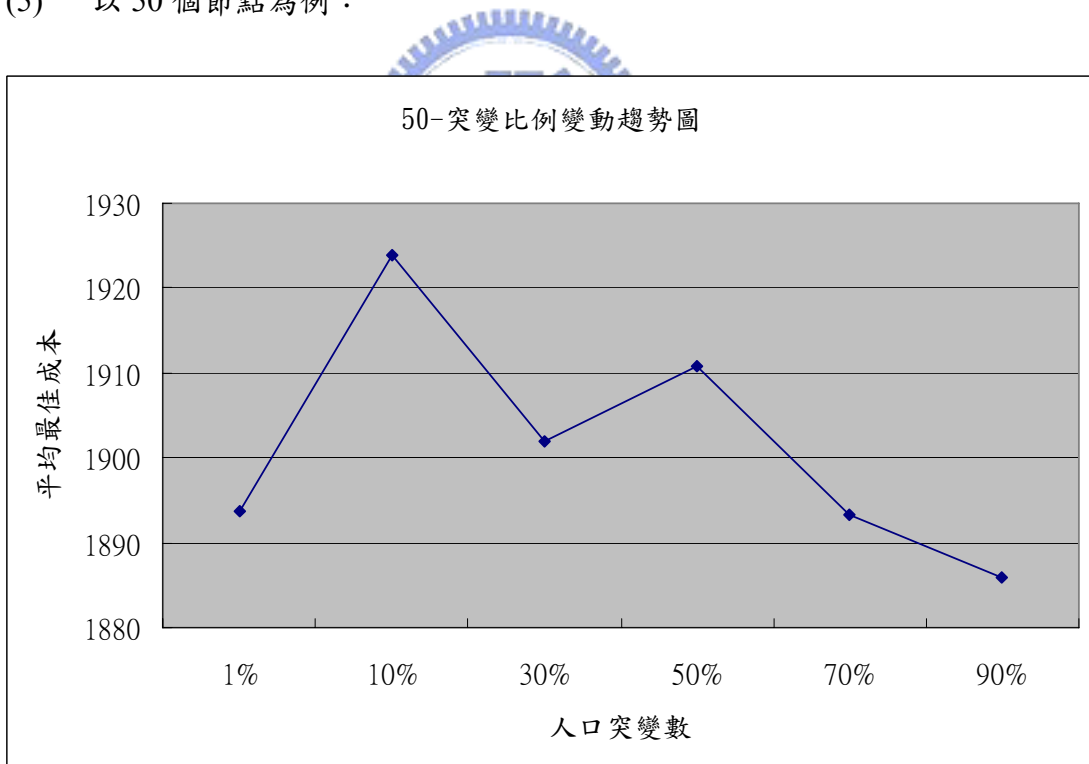
(3) 以 30 個節點為例：



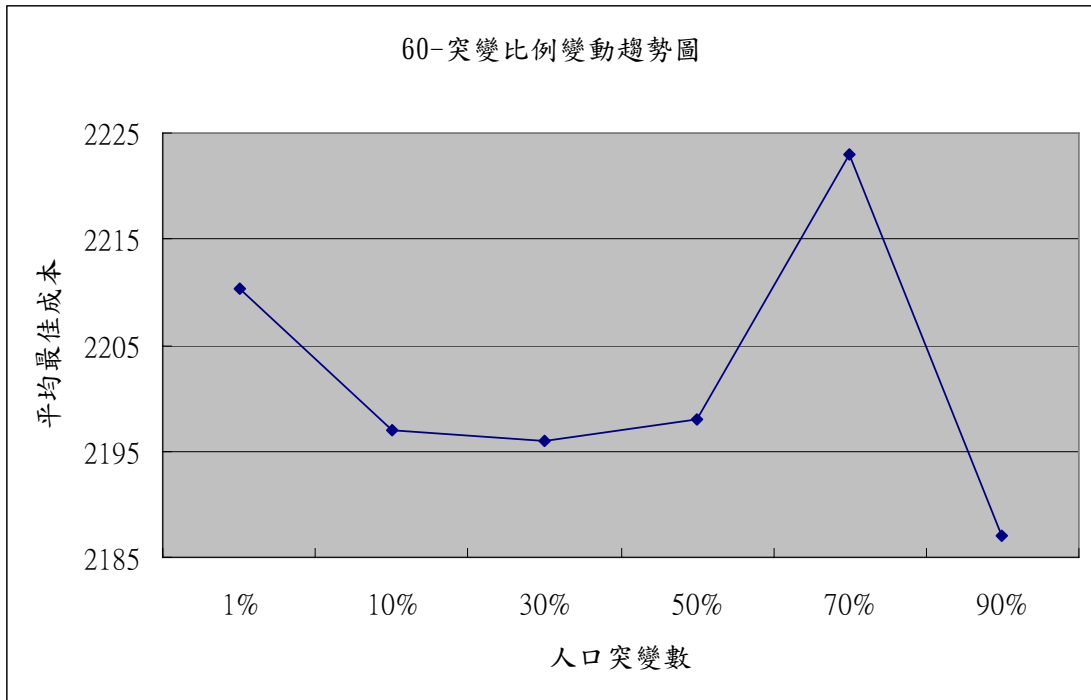
(4) 以 40 個節點為例：



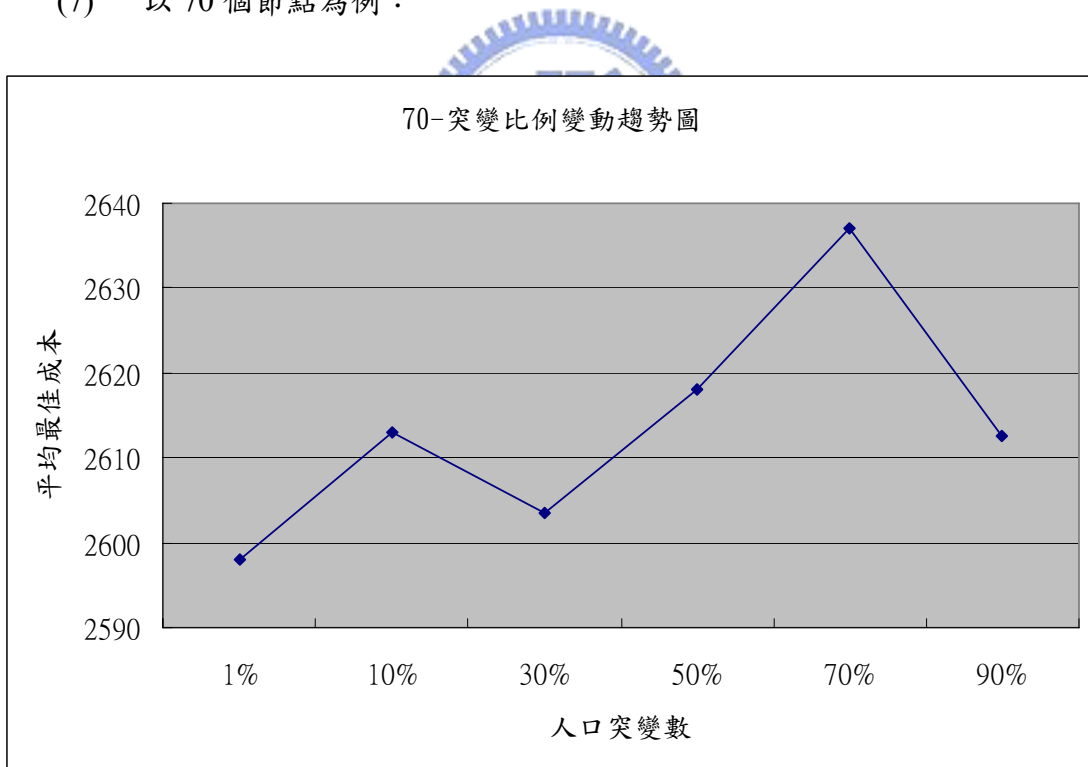
(5) 以 50 個節點為例：



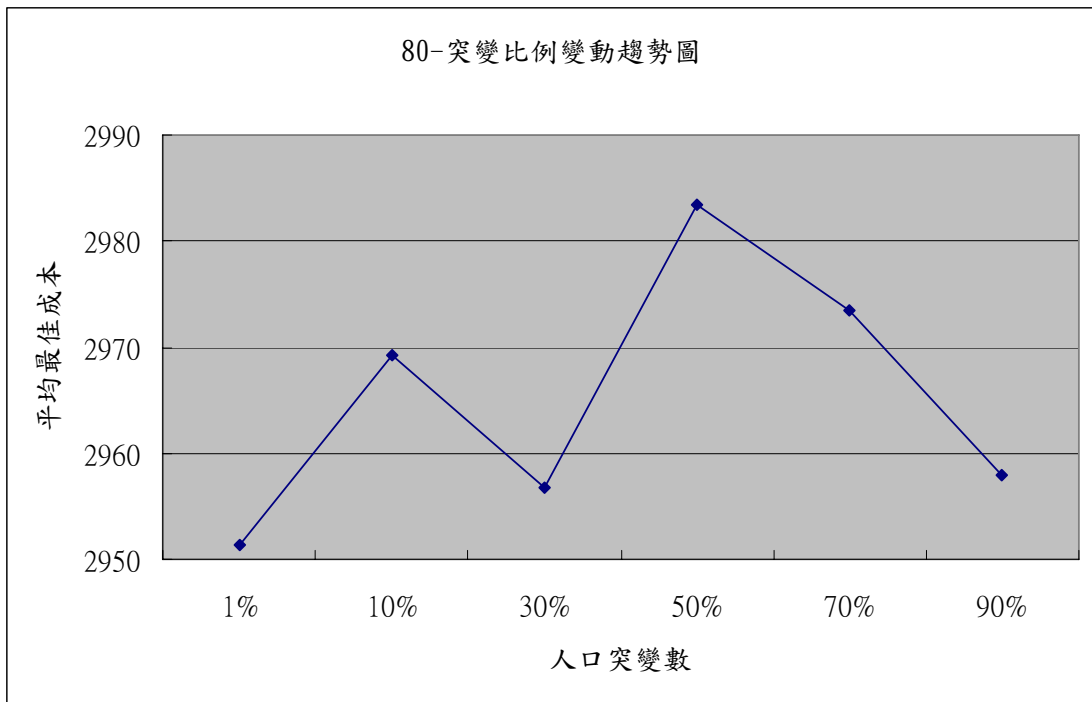
(6) 以 60 個節點為例：



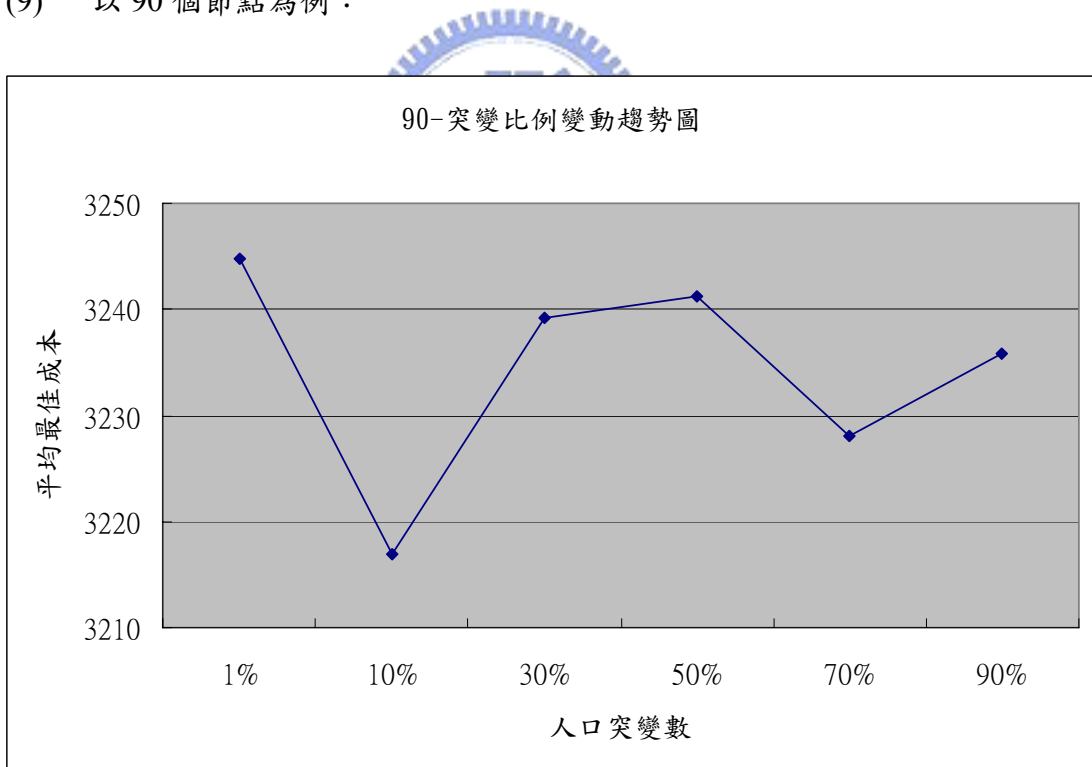
(7) 以 70 個節點為例：



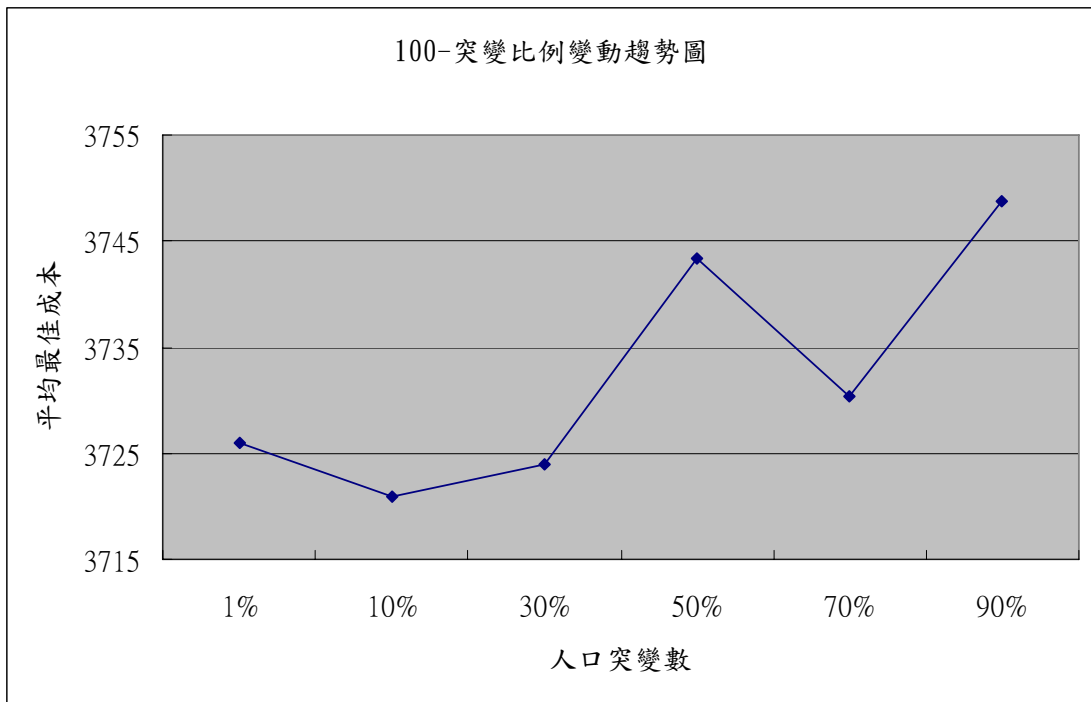
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：

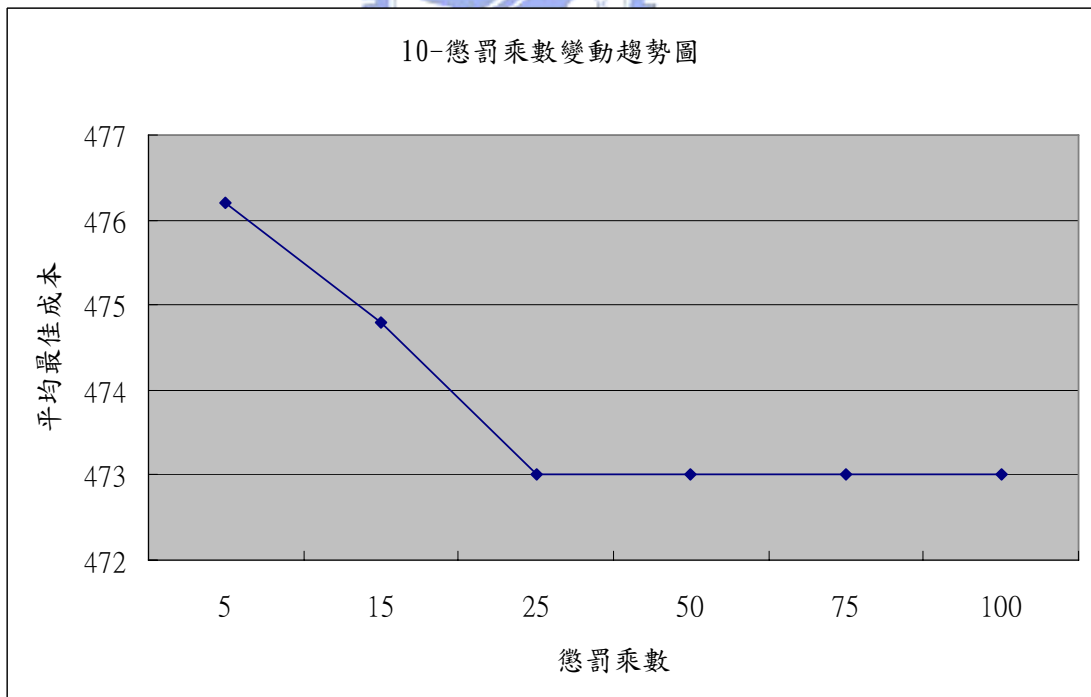


(10) 以 100 個節點為例：

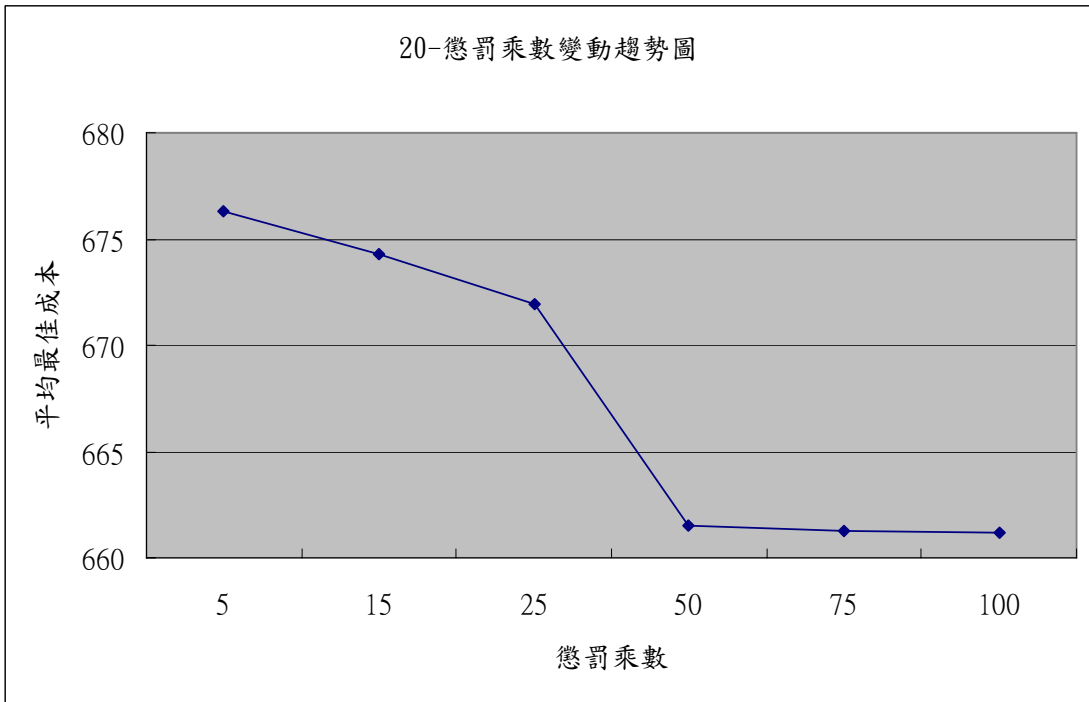


6. 懲罰乘數變動趨勢圖

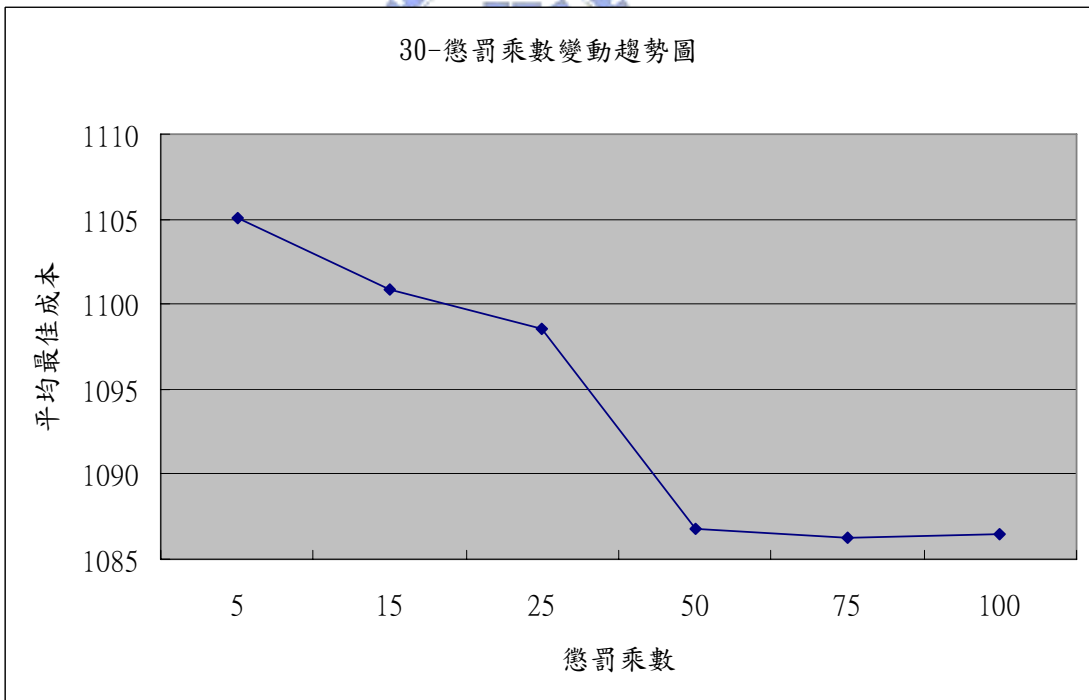
(1) 以 10 個節點為例：



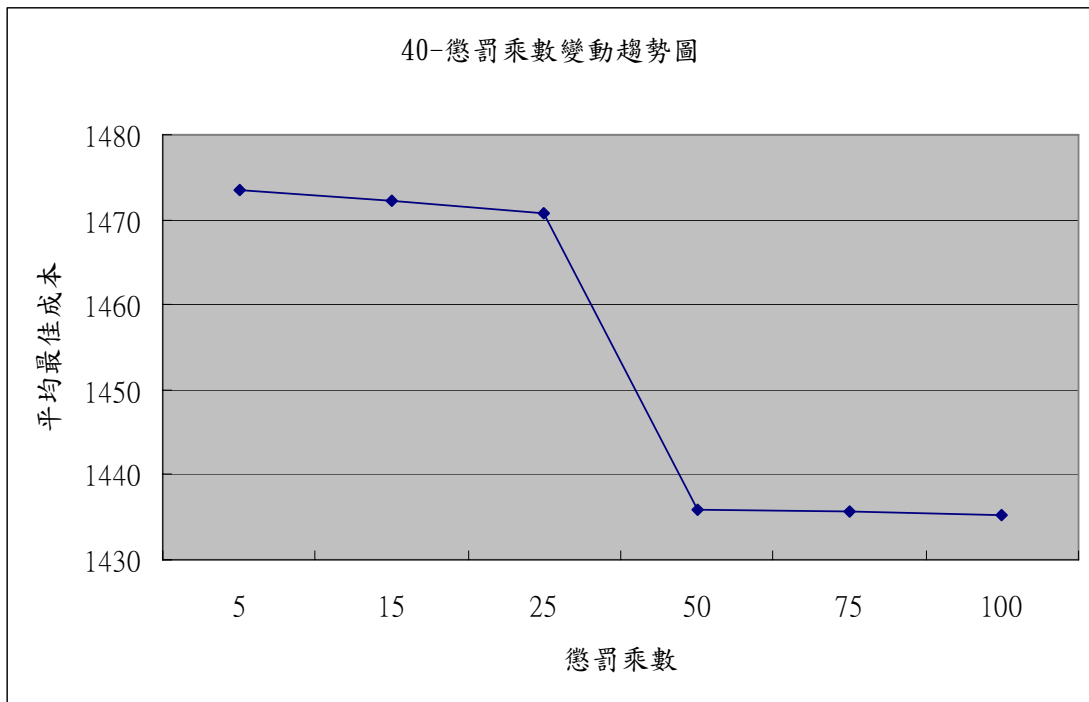
(2) 以 20 個節點為例：



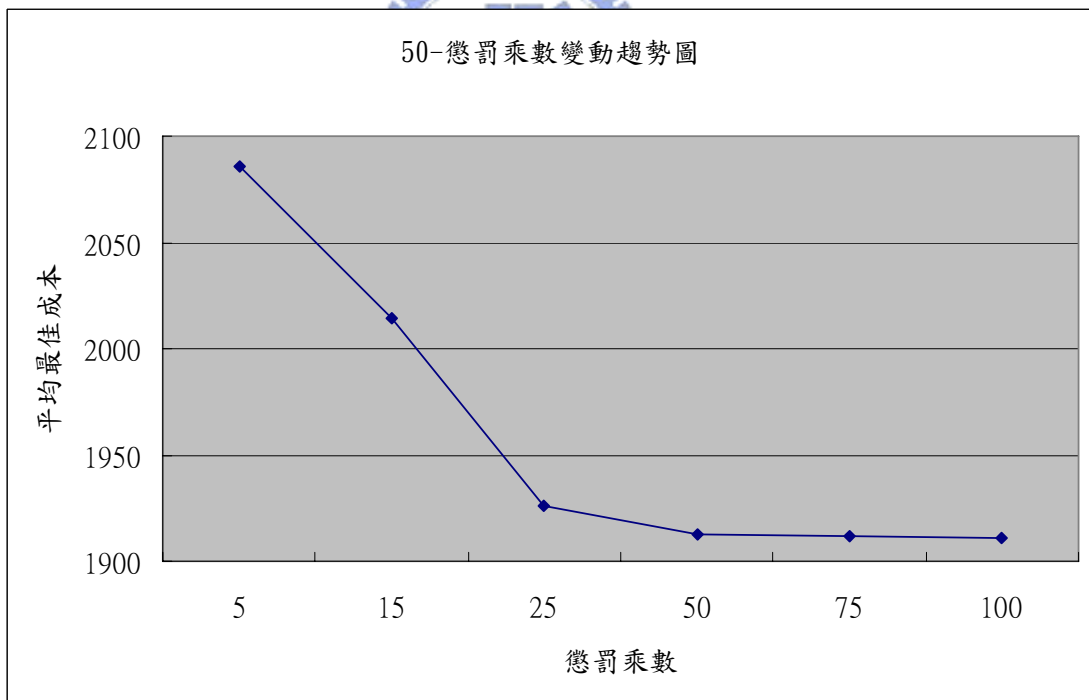
(3) 以 30 個節點為例：



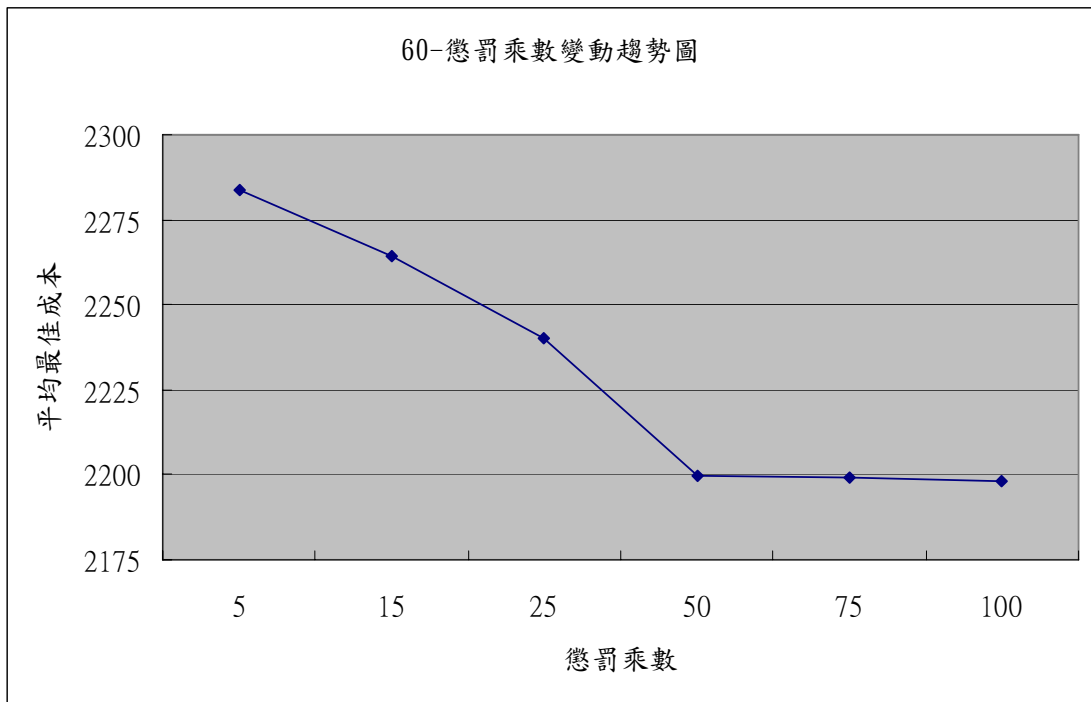
(4) 以 40 個節點為例：



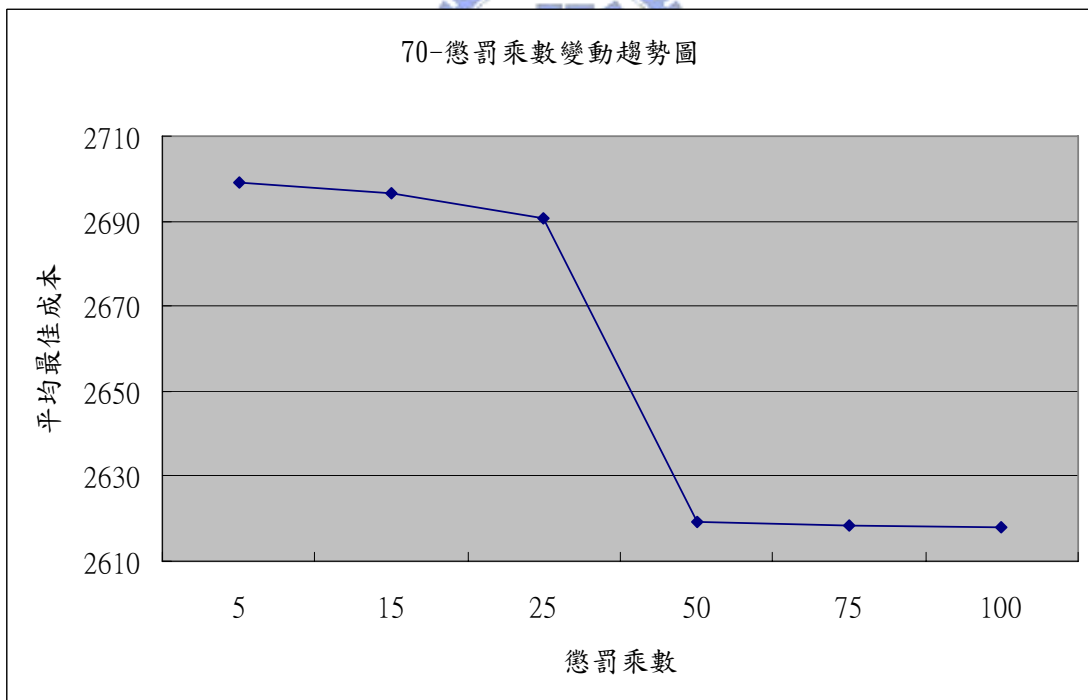
(5) 以 50 個節點為例：



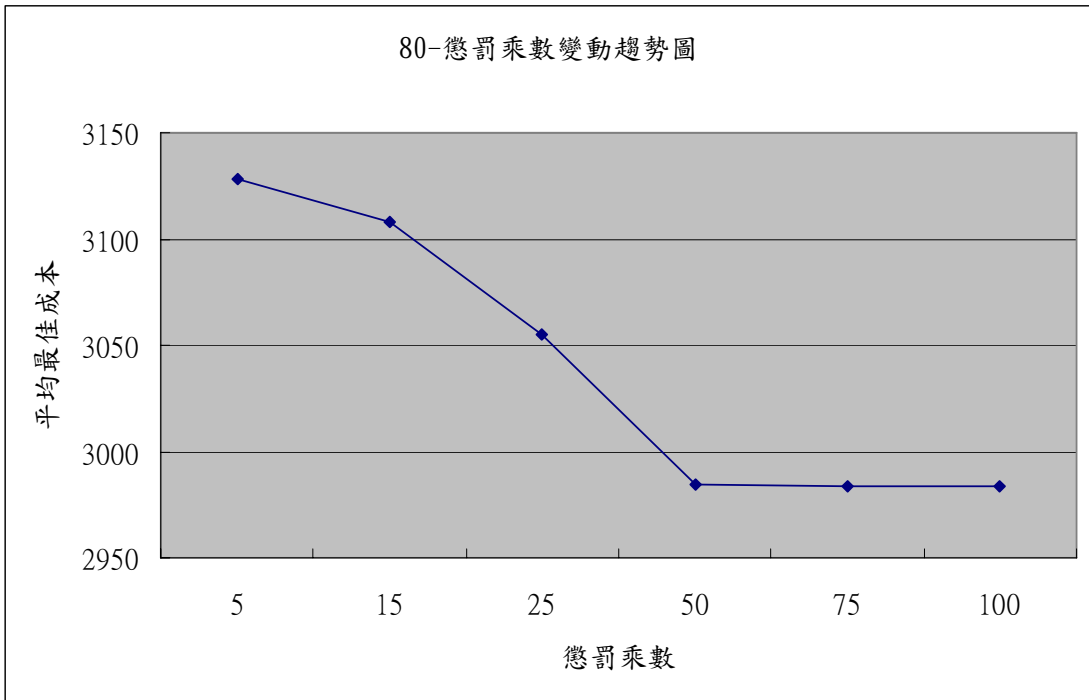
(6) 以 60 個節點為例：



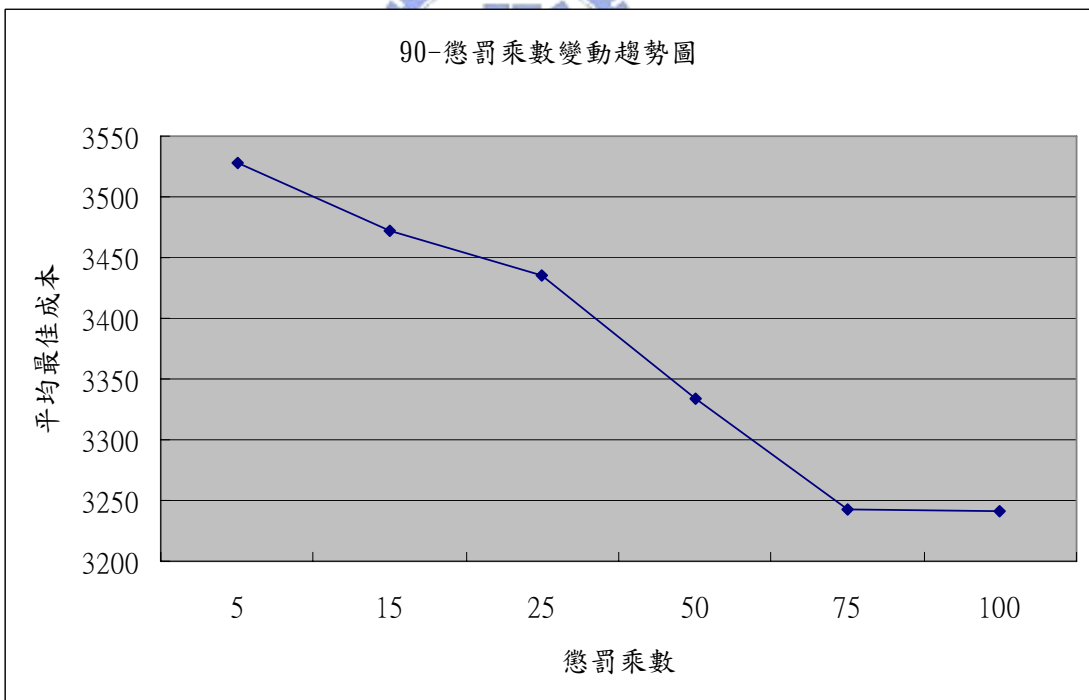
(7) 以 70 個節點為例：



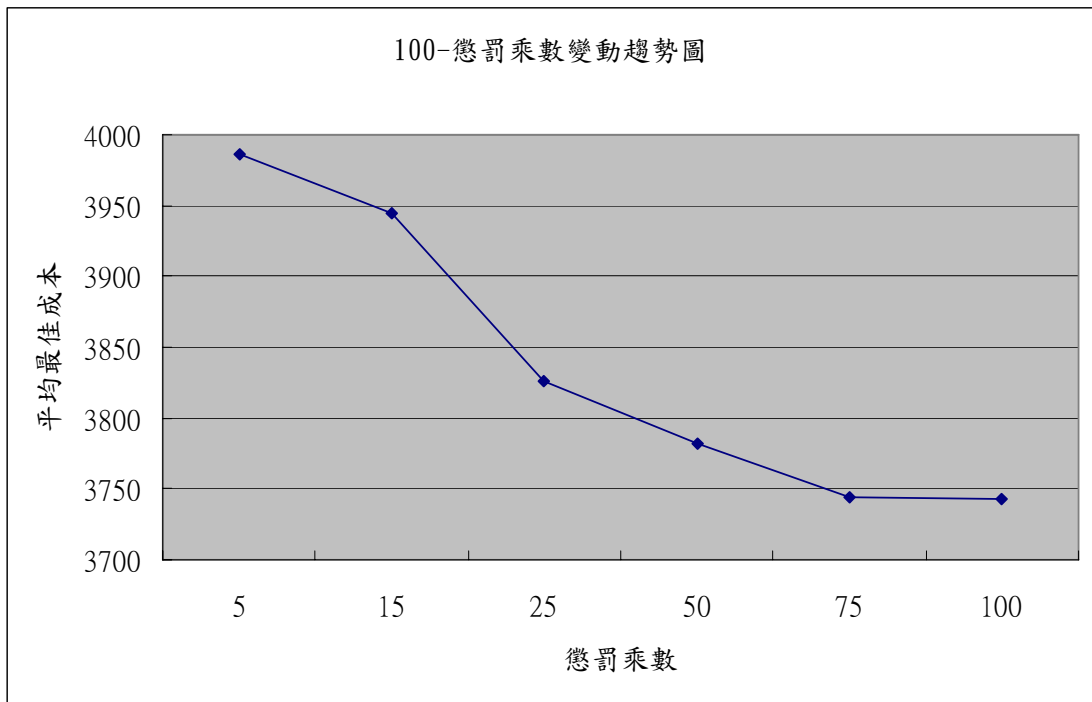
(8) 以 80 個節點為例：



(9) 以 90 個節點為例：



(10) 以 100 個節點為例：



簡 歷



姓 名：周韻佳

籍 貫：台北市

出生日期：71 年 6 月 9 日

電子郵件：yunnjia.tem94g@nctu.edu.tw

學 歷：

國立交通大學運輸科技與管理學系碩士

國立交通大學運輸科技與管理學系學士

台北市立景美女子高級中學

台北市私立復興國中

台北市私立復興國小

