

國立交通大學

交通運輸研究所

碩士論文

探討不完整資訊成對比較矩陣之 AHP 問題

Solving the AHP Problem of Pairwise Comparison Matrix in
Incomplete Information

研 究 生：羅英嫻

指導教授：曾國雄

民國九十二年六月

中華民國九十二年六月九日

探討不完整資訊成對比較矩陣之 AHP 問題

Solving the AHP Problem of Pairwise Comparison Matrix in Incomplete Information

研究生：羅英嫻

Student: Ying-Ing Lo

指導教授：曾國雄

Advisor: Dr. Gwo-Hshiung Tzeng

國立交通大學
交通運輸研究所
碩士論文

A Thesis
Submitted to Institute of Traffic and Transportation
College of Management
National Chaio Tung University
In Partial Fullfillment of the Requirement
For the Degree of
Master of Engineering
in
Traffic and Transportation

June 2003
Taipei, Taiwan, Republic of China

中華民國九十二年六月

探討不完整資訊成對比較矩陣之 AHP 問題

Solving the AHP Problem of Pairwise Comparison Matrix in Incomplete Information

研究生：羅英嫻

指導教授：曾國雄

國立交通大學交通運輸研究所

摘要

在實際回收層級分析法(AHP)問卷時，常遇到問卷有漏填(miss)的現象，即產生不完整比較矩陣(incomplete pairwise comparison matrix)。而本研究之目的為提出一最佳化工具之基因演算法來估計遺漏值，並且與其相關方法做討論比較，以作為當 AHP 問卷發生遺漏時，能夠提供解決不完整資訊之 AHP 問題的參考。對於此問題，過去已有相關的文獻做過探討，是以建立在數學理論的基礎而發展出求解的方法，或是以線性規劃的方式求解，都是為了求出最佳之遺漏值。但未採用基因演算法求解，因此本論文提出另一求解最佳化方法之一的基因演算法來解決此問題。首先，先討論各方法之方法特性，以了解各方法之求解範圍。最後，經由本實驗模擬驗證的結果，可說明與闡釋本論文所提出之各方法在於解決此問題之估計誤差、命中率的績效與程式執行時間，並提供有用的資訊，以作為解決不完整成對比較矩陣之 AHP 問題的參考，以提升 AHP 問卷之有效回收率。基於在填回問卷之遺漏值時，估計誤差應越小越好，而基因演算法的估算誤差是小於其他方法，故提出基因演算法在求解不完整成對比較矩陣之 AHP 問題上的績效是較其他方法兼具有效性與可行性。並期望此研究能作為主要使用 AHP 問卷之決策者的參考。

關鍵字：不完整資訊成對比較矩陣、層級分析法(AHP)、基因演算法、線性規劃、乘冪法(power method)、特徵多項式

Solving the AHP Problem of Pairwise Comparison Matrix in Incomplete Information

Student: Ying-Ing Lo

Advisors: Dr. Gwo-Hshiung Tzeng

Institute of Traffic and Transportation

National Chaio Tung University

Abstract

In practice, when we retrieve AHP questionnaires, they often run into miss case, i.e. arise incomplete pairwise comparison matrices. The purpose of this study is to address one optimize tool based upon genetic algorithms to approximate missing pairwise comparison value, and we can discuss and compare with related literatures to support the reference of missing problem. With regards to the problem, in the passed, some related researches were based on mathematic theorem or linear programming, which all want to derive the best missing value. But genetic algorithms is another way to deal with the problem, we employ genetic algorithms to automatically find the best missing value. First, in order to understand individual methods of solving domain, we discuss with features of methods. Second, through numerical simulation results, we illuminate that individual performance proposed methods in estimate error , recovered rate and programs running time to provide useful information to refer to solving the AHP problem of pairwise comparison matrix in incomplete information. In conclusion the genetic algorithms is the best way for estimate error and it would be more effective and practicable than others. We expect this research can be manipulated to assist every decision maker in using AHP questionnaires.

Keywords: Pairwise Comparison Matrix in Incomplete Information, the Analytic Hierarchy Process (AHP), Genetic Algorithms, Linear Programming, Power Method, Character of Polynomial

謝誌

很高興能夠進入曾國雄研究室，因為本研究室除了有曾老師的教導與照顧外，還有宜中學長、瑜淨學姊、嘉鴻學長、宗偉學長在研究及課業上的細心指導與幫助；再加上曦霞姐、麗敏學姊、伯豪學弟的關心，讓我有一個充滿溫馨的研究與求學的環境，本人在此獻上無限的感謝。

再者，瀛全是我唯一的依靠，他支持與關懷我無論在學業上或生活上的種種事情，他是我心中最重要的寶貝。回憶從小至今，從田中國小無憂無慮的童年；到田中國中的快樂時光；彰化女中的青春歲月；在逢甲大學的茫然與快速成長；乃至交通大學的獨立學習與研究，在成長過程中一直陪伴我的是摯愛的爸爸、媽媽，他們不斷的提供支持與鼓勵給我，我願將所有努力的成果榮耀獻給你們。

目 錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	v
圖目錄	viii

一、 緒論

1.1 研究背景與動機	1
1.2 研究目的	2
1.3 研究範圍與限制	2
1.4 理論架構與研究方法	2
1.5 研究內容與流程	3

二、 相關文獻回顧

2.1 層級分析法(Alytic Hierarchy Process, AHP)：	6
2.2 AHP 之不完整成對比較矩陣相關文獻	8
2.3 AHP 相關發展文獻	12
2.4 基因演算法(Genetic Algorithms, GA)	13

三、 不完整資訊之成對比較矩陣遺漏值之估計方法

3.1 不完整資訊成對比較矩陣基因演算法之基因操作	20
3.2 方法討論	24
3.3 方法之求解範圍限制	27
3.3.1 Har 法	27
3.3.2 Non-LP 與 LP 兩方法	28
3.3.3 Polynomial 法	30
3.3.4 基因演算法	30
3.4 方法之績效衡量	31

四、 數值例模擬與模擬結果之討論

4.1 實驗步驟與模擬實驗一：隨機遺漏一個元素	32
4.2 實驗步驟與模擬實驗二：隨機遺漏二個元素	38
4.3 實驗步驟與模擬實驗三：十二種遺漏狀況	42

4.4 結果與討論	53
五、結論與建議	
參考文獻：	55
附錄一：	57

表目錄	頁次
表 2-1 BGA 與 RGA 比較	19
表 3-1 方法符號對照表	24
表 3-2 方法之比較	26
表 3-3 Har 法之一個與兩個遺漏值之求解範圍	27
表 3-4 Non-LP 與 LP 兩方法之求解範圍	28
表 3-5 Polynomial 法之求解範圍	30
表4-1 實驗一之GA運算子設定	33
表 4-2 矩陣大小為 4 之各方法結果	36
表 4-3 矩陣大小為 5 之各方法結果	36
表 4-4 矩陣大小為 6 之各方法結果	36
表 4-5 矩陣大小為 4-6 之各方法程式執行時間	36
表 4-6 實驗二之 GA 運算子設定	38
表 4-7 矩陣大小為 4 之各方法結果	40
表 4-8 矩陣大小為 5 之各方法結果	40
表 4-9 矩陣大小為 6 之各方法結果	40
表 4-10 矩陣大小為 4-6 之各方法程式執行時間	41
表 4-11 十二種遺漏狀況與解決方法	42
表 4-12 實驗三之 GA 運算子設定	43
表 4-13 十二種遺漏狀況各方法之估計誤差	43

表 4-14 十二種遺漏狀況各方法之命中率	44
表 4-15 十二種遺漏狀況各方法之程式執行時間	44

圖目錄	頁次
圖 1-1 研究方法與架構	3
圖 1-2 研究流程	5
圖 2-1 基因演算法之演化流程圖	14
圖2-2 輪盤法	17
圖2-3 單點交配機制	18
圖2-4 單點突變機制	18
圖 3-1 基因演算法之演化流程	20
圖 4-1 實驗模擬內容	32
圖 4-2 矩陣大小為 4 之適應值收斂圖	32
圖 4-3 矩陣大小為 5 之適應值收斂圖	33
圖 4-4 矩陣大小為 6 之適應值收斂圖	33
圖 4-5 三方法之估計誤差	36
圖 4-6 三方法之命中率	37
圖 4-7 矩陣大小為 4 之適應值收斂圖	39
圖 4-8 矩陣大小為 5 之適應值收斂圖	39
圖 4-9 矩陣大小為 6 之適應值收斂圖	40
圖 4-10 二方法之估計誤差	41
圖 4-11 二方法之命中率	41
圖 4-12 狀況一適應值收斂圖	45

圖 4-13 狀況二適應值收斂圖	45
圖 4-14 狀況三適應值收斂圖	46
圖 4-15 狀況四適應值收斂圖	46
圖 4-16 狀況五適應值收斂圖	47
圖 4-17 狀況六適應值收斂圖	47
圖 4-18 狀況七適應值收斂圖	48
圖 4-19 狀況八適應值收斂圖	48
圖 4-20 狀況九適應值收斂圖	49
圖 4-21 狀況十適應值收斂圖	49
圖 4-22 狀況十一適應值收斂圖	50
圖 4-23 狀況十二適應值收斂圖	50
圖 4-24 十二種遺漏狀況各方法之估計誤差	51
圖 4-25 十二種遺漏狀況各方法之命中率	51
圖 4-26 十二種狀況之整體平均估計誤差	52
圖 4-27 十二種狀況之整體平均命中率	52

一、緒論

1.1 研究背景與動機

層級分析法(Alytic Hierarchy Process, 簡稱 AHP)之理論基礎是由 Saaty 在 1970 所提出, 主要應用於不確定性(uncertainty)之情況, 即具有多個評估準則之決策問題, 且能同時擷取多數專家意見, 甚具實用性, 因此是現今決策分析法中最廣泛被使用的方法。AHP 的主要部分是在每一層級之成對比較(pairwise comparison)為一完整的 $n \times n$ 矩陣 $A = (a_{ij})$ 。矩陣 A 是 $a_{ij} = 1/a_{ji}, a_{ij} > 0$ 的形式, 即 A 是一正的倒值矩陣(reciprocal matrix)。AHP 理論基礎是建立在 a_{ij} 是考慮 n 個方案之相對權重的近似值的事實。在執行 AHP 分析方法時, 需要透過問卷的方式, 來獲取每一決策層級之成對比較矩陣, 即每一完整成對比較矩陣(complete pairwise comparison matrix)需回答 $n(n-1)/2$ 題成對比較, 之後才能夠算出優勢權重。然而根據現實中, 決策者對於問卷所要判斷之屬性 (或方案), 常有不確定性或不明確(vagueness)或資訊不足(incomplete information)的情況, 極可能造成決策者在填答問卷時, 很難去做比較或者根本無法做比較。且加上填寫者在回答成對比較矩陣時, 必須符合一致性的要求($C.I. \leq 0.1$), 所以當 n 越大時, 相對上要回答的題目越多, 要能夠符合一致性就更困難了。也因此實際在回收問卷時, 常會遇到問卷不符合一致性, 或是有漏填(miss)的現象, 即產生不完整比較矩陣(incomplete pairwise comparison matrix), 使得有效問卷回收率低的情況, 因此需要再重新發問卷, 不僅是時間、金錢的浪費, 也會讓重複填寫者覺得麻煩。

對於此不完整矩陣的問題, 過去已有相關的文獻做過探討, 是以建立在數學理論的基礎而發展出求解的方法, 或是以線性規劃的方式求解, 都是為了求出最佳的遺漏值。但因有完整的理論架構, 所以相對上就會有許多的條件限制, 即不完整矩陣需符合方法假設條件, 才可估算遺漏之比較值。因此本研究提出另一求解最佳化的方法之一為以基因演算法來解決此問題。基因演算法基本精神為生物界中「物競天擇、優勝劣敗」的自然進化法則, 有擇優、交配及突變的能力, 故可有效率的求解。在實際應用上, 過去的相關文獻都有其解決不完整矩陣情況的適用範圍, 有相當的貢獻在且過去文獻方法中之適用範圍, 基因演算法也可求, 而對於適用範圍以外的情況, 基因演算法也可解決。總結為不同的方法解決遺漏值的效力是不同的, 且對於估計遺漏值誤差方面與命中率的討論亦較少深入討論, 因此本研究的目的為嘗試以基因演算法來求解遺漏值, 並且評估其預測能力還有比較相關已發展的方法做一整合性的預測力比較時間的, 在實驗模擬之結果顯示, 基因的估算誤差是比其他方法小, 雖然命中率方面不一定是最好的, 但在一般

填回遺漏值的主要觀念為估計誤差越小越好，因此 GA 法在求解不完整成對比較矩陣之 AHP 問題上是較其他方法有較好的績效。

1.2 研究目的

基於前述之背景與動機，本研究之目的為提出一最佳化工具之基因演算法來估計遺漏值，並且與其相關方法做一比較，以作為當 AHP 問卷發生遺漏時，能夠提供解決不完整資訊之 AHP 問題的參考。

1.3 研究範圍與限制

- (1)本研究主要是探討 4-6 的成對比較矩陣（AHP 法成對比較之屬性項，以不超過 7 項為原則，Saaty, 1980）。
- (2)假設原本完整之成對比較 AHP 矩陣為一致的矩陣，因某些外在或內在因素而漏填了某些成對比較，因而導致不完整之 AHP 成對比較矩陣，不為本文所考量。

1.4 理論架構與研究方法

AHP 法普遍使用於決策分析中，其需先透過問卷收集每一層級之成對比較矩陣，然後透過特徵值法求算出屬性（或方案）之權重與等級順序。但在收集 AHP 矩陣時，因決策者填寫問卷時，常會對屬性（或方案）有不確定或不明確或資訊不完全的困擾，極可能造成決策者在填答問卷時，很難去做比較或者根本無法做比較。且填寫問卷時，若所要評估的屬性越大，要回答的題目越多，易造成問卷不一致與發生漏填的情況，即產生了不完整比較矩陣的問題，使得有效問卷回收率低的情況，需要重新發放問卷，造成時間、金錢、人力的浪費。

對於本研究之整體架構與方法，首先在於對本研究主題的確認與了解，因此本研究將回顧相關文獻，包括：蒐集層級分析法的發展相關文獻，以期能對層級分析法更加的了解；過去有關解決層級分析法不一致的問題或遺漏之問題等文獻，將作為本研究之主要比較的方法；以及本研究提出利用基因演算法來求出不完整成對比較矩陣之遺漏值，與基因演算法之精神與操作等文獻或書籍，將有助於本研究為解決 AHP 之問題，為解決本研究 AHP 問題而建立基因演算法。最後以實驗模擬的方式，來證實比較基因演算法與其他相關方法對求解遺漏值的預測力與命中率。本研究結構及方法之詳述內容如下：

- 1.文獻評析法：藉由回顧層級分析法之原理與相關解決不完整成對比較之文獻，有助於對不完整資訊成對比較之 AHP 問題有深入的認識；而由於解決不完整成對比較矩陣之

方法不同，故所能解決此問題的效果和範圍也會不同，需針對各方法進行整理比較。

2. 基因演算法：由於本研究之目的是希望以最佳化工具之一的基因演算法，在求解問題上，沒有任何的條件限制，可以很直觀的解決有遺漏成對比較之 AHP 問題，且基因演算法是利用生物進化原理求解，且運算過程中只使用適應函數的資訊而不需要其他輔助資訊，可避免繁複的數學運算，有效的求出最佳解，加上其是一演算法，不受限於數學理論，因此適用於任何情況之不完整矩陣。
3. 最後以實驗模擬的方式來證實各方法對於估算遺漏值的估計能力，以此驗證本研究所提出之基因演算法其估計遺漏值的能力是較其他方法好。

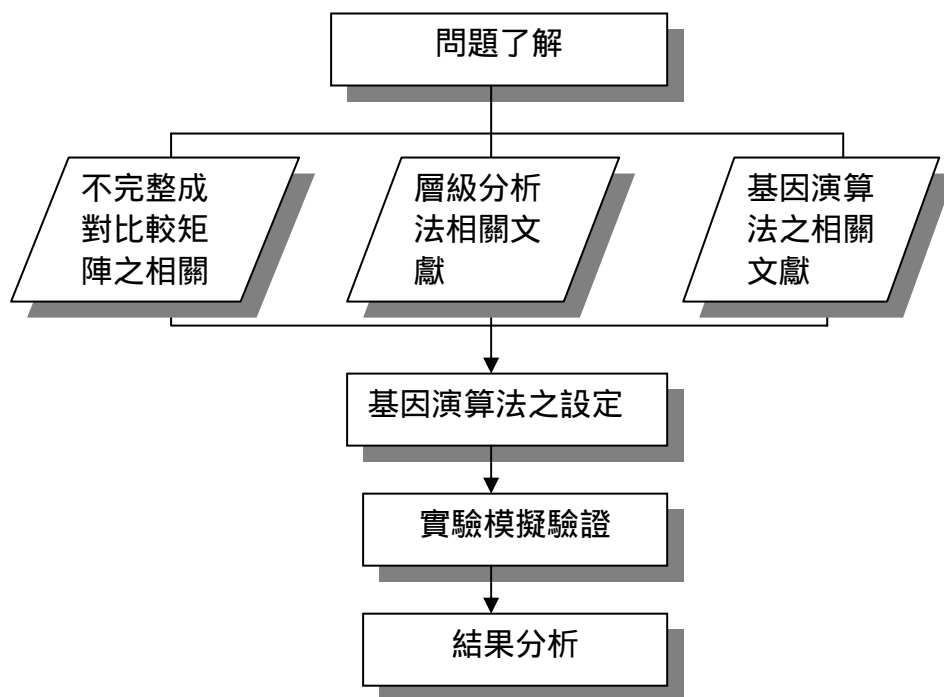


圖 1-1 研究方法與架構

1.5 研究內容與流程

本研究之研究內容主要為六部分，涵蓋內容如下：

1. 確立研究主題與研究範圍

包含研究動機與目的之界定、研究範圍與限制，及研究方法之擬定。

2. 資料收集與分析

收集 AHP 理論方法上相關研究發展的文獻，且針對與本研究有密切相關的相關主題之文獻內容，作摘要整理與討論各發展方法之優缺點，然後在根據全部收集的文獻

的綜合下，決定出本研究所要採取的解決方法。

3.研究方法與模式建構

基因之適應函數設定為以一致性指標值越小越好，來求出最佳之遺漏值，並說明求解步驟。再者，討論其他過去相關文獻的方法特性與條件。

4.實驗模擬驗證

由於個人電腦之流行，近年來相關之各種套裝軟體大量出籠。Maltlab 為其中一種數學套裝軟體，尤其它在矩陣與向量上的運算能力特別強，故選擇使用 Maltlab6.1 作為研究的分析工具來進行模擬。以 Matlab 程式語言來執行各方法的實驗模擬與隨機矩陣的產生，而實驗模擬範圍為 4-6 階的矩陣。接著進行模擬不同遺漏情況的實驗模擬，以驗證各方法之估計誤差與命中率（即救回問卷的比率），以此評估比較各方法解決遺漏問題的效力。最後，將實驗模擬結果進行綜合分析與討論。

5.結論與建議

彙整研究之結果，以作為在進行 AHP 問卷研究者，當回收問卷時產生遺漏狀況時之參考。另外說明研究限制與不足，以供他人進行後續研究。依據上述內容，本研究之流程圖如圖 1-2 所示。

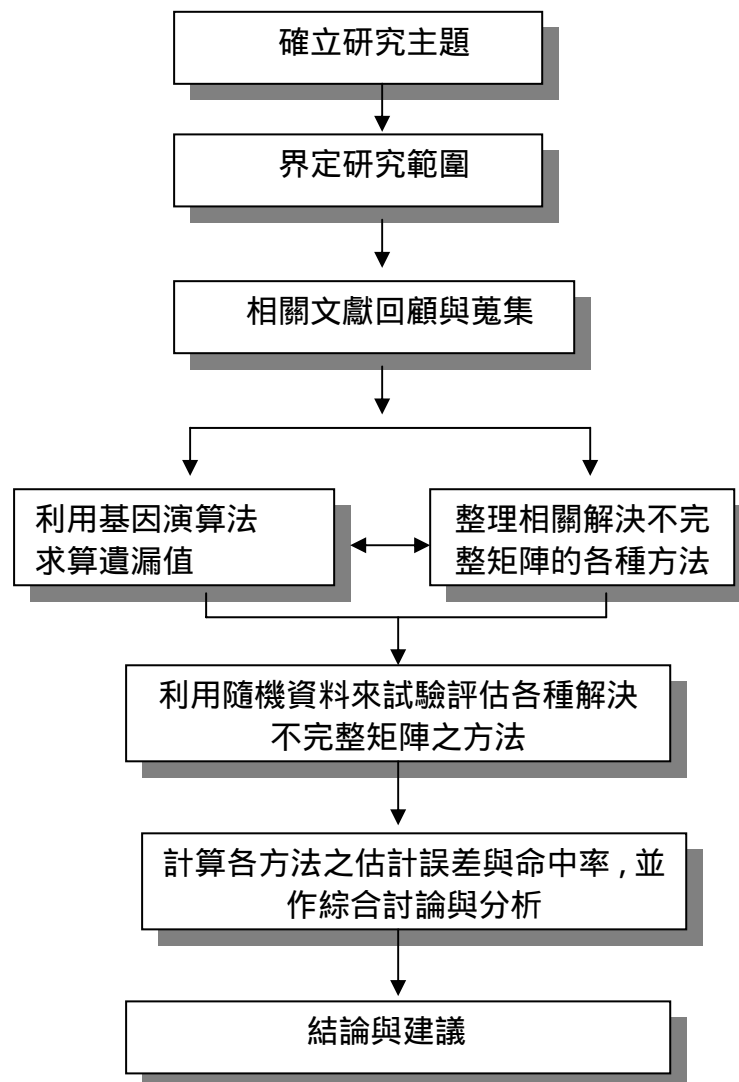


圖 1-2 研究流程

二、相關文獻回顧

由於本研究所探討的問題為不完整資訊成對比較矩陣之 AHP 問題，因此，文獻回顧方面首先對層級分析法之理論與相關發展文獻做探討；其次回顧相關解決不完整資訊成對比較之 AHP 問題的方法文獻及其方法的操作與內容。而由於本研究提出以基因演算法來解決此問題，所以最後敘述基因演算之概念與操作等相關文獻。

2.1 層級分析法(Analytic Hierarchy Process, AHP)：

層級分析法是 1971 年由 Thomas L. Saaty 所發展出來，主要應用於不確定 (uncertainty) 情況及具有多個評估準則之決策問題。當時係應用該理論從事埃及國防部之應變計劃問題上，至 1973 年已發展成熟，翌年(1974 年)開始廣為各界應用。1980 年後，AHP 的整體理論更臻完備 (Saaty, 1980, 1989, 1994; Saaty and Vargas, 1987)。

AHP 之理論簡單 (AHP 的理論基礎如附件)，操作容易，同時能擷取多數專家之意見，在實務上甚具有實用性。國內有關 AHP 之研究與應用，也日益普遍，曾國雄、鄧振源(民國 78 年 6 月)對 AHP 的應用範圍與程序及 AHP 的問題點均有詳細之探討。

AHP 的進行步驟：利用 AHP 進行決策問題之分析時，主要包括以下三個階段：

第一階段：建立層級結構

處理複雜問題時，利用層級結構加以分解。基於人類無法同時對 7 種以上進行比較之假設下，每一層級的要素不宜超過 7 個。假若複雜之問題有 n 個要素，利用成對比較而獲得之比率尺度，總共需作 $n(n-1)/2$ 個判斷；在最大要素個數為 7 個下，則可進行合理之比較，同時可以保證其一致性。因此，有效之層級數可用 $n/7$ 估計；如此之層級結構，較易進行有效之成對比較，與能獲得較佳之一致性。

第二階段：各層級要素間權重的計算

此一階段可區分為三個步驟：

(1) 建立成對比較矩陣

某一層級之要素，以上一層級某一要素作為評估基準下，進行要素間之成對比較。若有 n 個要素時，則需進行 $n(n-1)/2$ 個成對比較。成對比較時使用之數值，分別為 $1/9, 1/8, \dots, 1/2, 1, \dots, 8, 9$ ，將 n 個要素比較結果之衡量，置於成對比較矩陣 A 的上三角形部分 (主對角線為要素自身之比較，故均為 1)，而下三角形部分之數值，為上三角部分相對位置數值之倒數，即 $a_{ij} = 1/a_{ji}$ 。有關成對比較矩陣的元素，如下所示：

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{bmatrix} \quad (1)$$

(2)計算特徵值與特徵向量

成對比較矩陣得到後，即可求取各層級要素之權重。使用數值分析中常用之特徵值(eigenvalue)解法，找出特徵向量或稱優勢向量(priority vector)；由於成對比較矩陣為正倒值矩陣，而不是對稱矩陣，因此可用之特徵值主要有乘幂法(power method)與Householder 法，而後者之計算速度又較前者快許多。

(3)一致性的檢定

若成對比較矩陣為正倒值矩陣，要求決策者在成對比較時，能達到前後一貫性，這是相當困難的。因此需進行一致性之檢定，作成一致性指標(Consistency Index, C.I.)，檢查決策者回答所構成之成對比較矩陣，是否為一致性矩陣。一致性指標之提出，主要告訴決策者在評估過程中，所作判斷之合理程度如何？是否不太一致？或有矛盾現象？以及時修正，避免作成不良之決策。

一致性之檢定，除用於評量決策者之判斷外，尚可用於整個層級結構。由於各層級間之重要性不同，所以要測試整個層級結構是否具一致性。一致性指標值，不論在決策者判斷之評量或是整個層級結構之測試，Saaty 建議宜在 0.1 左右（一般採 $C.R. < 0.1$ ），如此一致性能獲得保證。

第三階段：整體層級權重的計算

各層級要素間之權重計算後，再進行整體層級權重之計算。最後依各替代方案之權重，以決定最終目標的最適替代方案。若為群體決策時，各替代方案之權重可以加以整合，其方法如前所述。

以上對於層級分析法的說明，我們可發現以下之優點：

- (1)操作簡單：只要兩兩成對比較各屬性（準則或要素）間的偏好值後，透過特徵向量法計算後，即可求出最佳之方案。
- (2)加入權重值表示：以不同的權重值來表現，對於不同屬性之偏好。
- (3)可幫助決策者處理不確定性的決策問題，且可同時擷取多數專家的意見。

層級分析法之缺點如下：

- (1)每一層級間的要素接必須具備完整性，即需完成全部之成對比較。
- (2)當要素個數多時，要使每一個成對比較矩陣符合一致性是很困難的一項工作。

2.2 AHP 之不完整成對比較矩陣相關文獻

許多學者曾深入探討 AHP 含有不完整成對比較矩陣知情況，亦提出各種方法去解決是與本研究主題相關且具代表性者如下：

Harker (1987)最早提出解決不完整矩陣方法，主要理由為其發現 AHP 執行時，時常會發生的三大缺點為：

- (1)完成 $n(n-1)/2$ 比較之時間問題：因當所評估的 n (準則或方案數很大時)，易造成決策者填寫的意願；
- (2)不願意直接比較某方案：是因為某些比較是比較敏感的議題，因此決策者不想直接正面回答問題；
- (3)不確定某些比較：因某些比較決策者並不完全瞭解，所以不知道如何回答。

由於上述的問題，很容易造成 AHP 問卷回收時，常有不完整的成對比較矩陣資料，為了解決此問題，因此 Harker 提出一方法來處理在不完整矩陣下之優勢權重求取。此方法是乘幂法(power method 理論基礎如附錄)之一延伸，其做法為若 (i,j) 成份是遺漏的，則以人工變數 w_i/w_j 來代替，完成一完整正倒值矩陣 $A(w)$ ，接著為計算其特徵值，公式如下：

$$A(w)w = \lambda w$$

給定一不完整矩陣 $A = (a_{ij})$ ，定義一相當推論的正倒值矩陣 $\tilde{A} = (\tilde{a}_{ij})$

$$a_{ij} = \begin{cases} 1+m_i & \text{if } i = j \\ 0 & \text{if } i \neq j \text{ and } (i,j)\text{-element miss} \\ a_{ij} & \text{otherwise} \end{cases}$$

其中 m_i 表示第 i 列遺漏要素。

Harker 方法求解步驟：

第一步驟：建立出一推論倒值矩陣。

第二步驟：計算出 \tilde{A} 之最大特徵值 $\tilde{\lambda}_{\max}$ 和其對應之特徵向量值(eigenvector)。

第三步驟：正規化特徵向量而得出優勢向量。

Harker 方法條件限制：條件一：任一不完整成對比較矩陣（未知比較以 0 表示）必須為不能縮減矩陣(irreducible)。

文章最後舉一 6×6 之矩陣進行實驗，完整矩陣共需回答 15 題比較，而結果顯示現在只需回答 10-11 題比較就可以有一精確公平地權重值，且也顯示回答 5-15 題分別求出的權重值不同但權重之排序都一樣。

Triantaphyllou (1995)年提出減少收集資料過程中過多的過程，起因為他發現 AHP 一個嚴重應用上的缺點為若在分析 n 個課題時，是需要完成 $n(n-1)/2$ 之成對比較集合，所以當要素 n 太大時，比較集合會太大，因此提出一方法來減少比較數，此在實際應用上是很重要的。作者表示不需要完成所有之成對比較，即包含有未知的比較相當於遺漏的比較，所以提出兩個方法來估計未知的比較：(1)為簡單易懂法；(2)為使誤差最小之線性規劃法。方法基本想法為任一不完整 $n \times n$ 矩陣，都可分割成兩完整之方正矩陣，分別為 $n_1 \times n_1$ 矩陣與 $n_2 \times n_2$ 矩陣，並且必須符合 $n_1 + n_2 > n$ 才可求解。

1.簡單易懂法(simple method, non-LP)：

在完全一致性之成對比較倒值矩陣滿足以下關係：

$$a_{ij} = a_{ik} a_{kj} \quad \text{for } i, j, k = 1, 2, \dots, n \quad (1)$$

從(1)的關係，在完全一致性的情況下，未知比較（如以 x_{ij} 表示）可求得以下之關係：

$$x_{ij} = a_{ik} / a_{jk} \quad \text{for } i = 1, 2, \dots, n - n_2; \quad j = (n_1 + 1), \dots, n; \quad k = (n - n_2 + 1), \dots, n_1 \quad (2)$$

在不一致性情況下，式（2）之關係不在是完全為真的，然而未知之 x_{ij} 我們仍可期望去求出越接近的值，因此綜合所有可能的值來合理衡量出未知 x_{ij} ，關係如下：

$$x_{ij} = \frac{\sum_{k=n-n_2+1}^n \frac{a_{ik}}{a_{jk}}}{(n_1 + n_2 - n)} \quad \text{for } i = 1, 2, \dots, n - n_2; \quad j = (n_1 + 1), \dots, n \quad (3)$$

上述之平均法卻沒有考慮以下關係：

$$X_{ij} \approx (a_{ii'} \times a_{jj'}) \times X_{i'j'} \quad \text{for } i, i' = 1, 2, \dots, n - n_2; \quad j, j' = (n_1 + 1), \dots, n \quad (4)$$

因此發展另一線性規劃的方法，是結合（3）式與（4）式結果如下：

$$x_{ij} \approx \left[\sum_{k=N-N_2+1}^{N_1} (a_{ik}/a_{jk}) \right] / (n_1 + n_2 - n) \quad \text{for } i=1, 2, \dots, n-n_2; j=n_1+1, \dots, n \quad (5)$$

由上式可看出左邊幾乎接近右邊，架設期望兩邊的關係式相等，則式(5)與式(4)有一誤差項分別以 e_{ij} 與 $e_{ij}^{i'j'}$ 來使式(4)與式(5)兩邊相等，其關係表示如下：

$$x_{ij} = \left[\sum_{k=N-N_2+1}^{N_1} (a_{ik}/a_{jk}) \right] / (n_1 + n_2 - n) + e_{ij} \quad \text{for } i=1, 2, \dots, n-n_2; j=n_1+1, \dots, n \quad (6)$$

2. 為使誤差最小之線性規劃法 (linear programming, LP)：

$$\text{目標式：Minimize } f = \sum_{i=1}^{n-n_2} \sum_{j=n_1+1}^n |e_{ij}| + \sum_{i=1}^{n-n_2} \sum_{j=n_1+1}^n \sum_{i'=1}^{n-n_2} \sum_{j'=n_1+1}^n |e_{ij}^{i'j'}|$$

限制式：

$$x_{ij} = \frac{\sum_{k=n-n_2+1}^n \frac{a_{ik}}{a_{jk}}}{(n_1 + n_2 - n)} + e_{ij} \quad \text{for } i=1, 2, \dots, n-n_2; j=(n_1+1), \dots, n$$

$$x_{ij} = (a_{ii'} \times a_{jj'}) \times X_{ij'} + e_{ij}^{i'j'} \quad \text{for } i, i'=1, 2, \dots, n-n_2; j, j'=(n_1+1), \dots, n$$

文章最後以 100 個隨機問題來檢測上述的兩方法。在考量電腦運算上的速度，所以只考慮 $n \leq 13$ 的矩陣情況，且整理出共同比較個數(number of common pairwise comparison, PC)，為兩分割矩陣(n_1 與 n_2)共同擁有的比較個數；和可得共同成對比較比率(percent of number of common pairwise comparison, PC%)，為共同成對比較個數佔可得比較個數的比率。結果顯示若矩陣共同比較率很低時，則兩方法所算的誤差率都很高。還可觀察出當矩陣大小越大時，誤差率越低。在可得共同成對比較率(PC%)介於 10%-60%間時，線性規劃法明顯比簡單易懂法好，而在 PC%在 10%以下兩方法的誤差率一樣大，在 60%以上時，兩方法所算的結果幾乎一樣好。

Shiraishi et al. (1998) 年提出正倒值矩陣之特徵多項式 (characteristic polynomial) 之某些性質與 AHP 成對比較矩陣之一致性有很大的關聯存在。基於此結果，作者提出一啟發式之方法來估計不完整成對比較矩陣之遺漏要素。

給定一不完整矩陣 A 包含一個遺漏值 x ，以 $A(x)$ 來表示，如下所示：

$$A(x) = \begin{bmatrix} 1 & a_{12} & \cdots & x \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/x & 1/a_{2n} & \cdots & 1 \end{bmatrix}$$

其 $A(x)$ 之最大特徵值以 $\lambda_{\max}(x)$ 表示。接著計算以下最佳化的問題：

$$\min_x \lambda_{\max}(x) \quad (1)$$

因為期望求得最佳 $\lambda_{\max}(x)$ ，令 $c_3(x)$ 為 $A(x)$ 特徵多項式中 λ^{n-3} 之係數。從理論 1 與自然結果 2，可知 $c_3 \leq 0$ 且假設我們能夠使 $c_3(x) = 0$ 則 $A(x)$ 能變成一致性。因此，期望 $A(x)$ 能有較好的一致性，則 c_3 越接近 0 越好。基於此想法，所以提出以下方法：

$$\min_x c_3(x) \quad (2)$$

代替(1)式，開始啟發式方法之步驟：

步驟一：找出式(2)之解 x_0 。

步驟二：計算 $A(x_0)$ 最大特徵值與其對應之特徵向量。

步驟三：正規化特徵向量為優勢向量。

$$\text{從定理 3，得 } c_3 = \sum_{i < j < k} \left\{ 2 - \left(\frac{a_{ij}a_{jk}}{a_{ik}} + \frac{a_{ik}}{a_{ij}a_{jk}} \right) \right\}$$

因此式(2)的問題相同於式(3)。

$$\min_x \sum_{j=2}^{n-1} a_{1j}a_{jn} \frac{1}{x} + \sum_{j=2}^{n-1} \frac{1}{a_{1j}a_{jn}} x \quad (3)$$

由可簡單地導出式(3)之解 x_0 ，如下所示：

$$x_0 = \sqrt{\left(\sum_{j=2}^{n-1} a_{1j}a_{jn} \right) / \left(\sum_{j=2}^{n-1} \frac{1}{a_{1j}a_{jn}} \right)}$$

文章並以 10000 個只包含一個遺漏比較的隨機矩陣進行模擬，所比較的矩陣大小為階數(order) 4 至 15；並且與 Harker 的方法進行比較，在基於 AHP 內容下，顯示特徵值傾向越小越好，如此之比較矩陣才能越接近一致性，結果使明顯比 Harker 的方法好。

但是特徵多項式法只能解決不完整矩陣只缺一個遺漏，相對於 Harker 法能缺許多遺漏比較起來，當面臨有不完整矩陣含一個以上的遺漏，此方法是很難去求解。

詹德光(1993)提出傳統 AHP 法包含：各層級元素所提供之訊息必須完整的提供給決策者得知、決策者必須對任兩各評估準則間作成對比較，不允許有不給定權值的情形發生、於決策過程中沒有考慮決策者對方案的偏好習性之假設，與現實存在的情形相比對，發現有幾點問題：(1)現實常存在曖昧不明的資訊。(2)若資訊供應不足，決策者可能根本無法做成對比較；(3)Saaty 在處理成對比較矩陣之權重值時，沒有考慮人類心智能力之限制，即決策者對於所給定的權重值，是否有絕對的信心程度支持。基於上述問題，並且利用決策屬性權值的不一致性所可能產生的誤差，以數學規劃的方式進行研究。所以，作者之研究目的為改良 AHP 法在資訊不完整情況下之問題，考慮決策者心理因素，提出透過 0-1 線性規劃法來求解。

2.3 AHP 相關發展文獻

針對層級劃分及決策屬性間成對比較值上不確定性的探討，是經由機率統計的方法，推估決策屬性的精確度與可靠性，有以下文章：

Zahedi (1986)以隨機誤差項來衡量成對比較值所含的不確定性，將成對比較值配合不同分配的隨機誤差項，在依不同的方法求估決策屬性權重，利用模擬分析法 (simulation)，探討估計式的統計精確度和權重等級(rank order)的維護能力。

Saaty and Vargas (1987)將決策屬性及每一層級的成對比較值皆視為隨機變數，以一區間值取代原來的單一成對比較值，結合確定或不確定的決策屬性，找出各可選擇方案的相對權重，並求出各方案間會產生變換的等級(rank)的機率，作為選取最適決策方案的輔助。

Levary and Wan (1998)引進不確定性(uncertainty)的考量於 AHP，利用模擬結合 AHP 來分析在不確定情況下，如何做決策？作者認為所謂正確的決策方案後，是建立在對於已決策的方案，在未來時的績效是有多好。且 AHP 法中之比較的必然性與精確性大小，兩者都是依據決策者可得資訊的數量與對問題了解的深度而定。由上述原因，應當要考慮到未來決策環境特性，因此加入不確定性於 AHP 模式中，是較有用的決策模式。所以情境(scenario)分析已被使用於預期未來決策之環境與相關操控不確定性的方式。然而在大部分的例子中，成對比較矩陣是根據情境給定權重。且決策方案之等級順序是基於

兩種關聯不確定性：(1)考量未來決策環境特性之不確定性(2)考量成對比較時，對相關特定情境之決策判斷的不確定性。最利用模擬的方法來求解方案之權重與等級順序。

在決策與優先理論之基礎(Fundamentals of Decision Making and Priority Theory)中所談論到衡量不一致性與主要右特徵相量，不一致性是非常嚴重的問題，即人本身有邏輯性的問題，所以應該去找出不一致性，並且建議一最佳值來改善不一致。若是適合整體了解，則建議的值或某值是可被採用的。為了去確定不一致的判斷，衡量不一致與導出等級(scale)必須是建構上的連結。

Obata, T. et al. (1999)，利用計算上的實驗，來比較特徵多項式法與 Harker 方法在處理不完整比較矩陣與不一致性兩者之有效性與績效，分別實驗 4 至 15 階之矩陣各 1000 次，其結果顯示，在改善一致性的事件上，特徵多項式法和 Harker 方法都一樣有用，兩者沒有明顯的差異。然而，特徵多項式法有以下優點，為當求解不完全比較矩陣的同時，其可求算出未知比較的比較值。在實際狀況下是比 Harker 方法有用。

2.4 基因演算法(Genetic Algorithms, GA)

基因演算法最初是由 John Holland 教授於 1960 與 1970 年代所發展，而其學生 David Goldberg 在其論文中使用此法解決了瓦斯管線的傳送，並證明此演算法在應用上的多途性後，此方法便開始廣受歡迎，每年世界上都有相當多的學者與研究投入其中，並以此法解決各種研究領域之問題。演算法(Algorithms)為計算機常用術語，意指一套經由有規則的數學運算或邏輯判斷後可以處理完成某一特定工作或任務之方法。此有規則的方法或流程稱之為演算法。基因演算法是利用基因遺傳的原則寫出的程式，即是基於自然選擇過程的一種最佳化搜尋機構。基本精神是在於仿效生物界中物競天擇、優勝劣敗的自然進化法則，它能夠選擇物種中具有較好特性的上一母代，並且隨機性的相互交換彼此的基因資訊，以期能產生較上一代更優秀的子代，如此重複下去以產生適應性最強的最佳物種。基因演算法則的三個主要運算子為複製(reproduction)、交配(crossover)、以及突變(mutation)。應用基因演算法則來解最佳化問題的基本精神為：將所要搜尋的所有參數編碼乘稱為染色體(chromosome)的離散(discrete)或二元(binary)字串(string)；其演化流程「參考：蘇木春與張孝德，民82」如下圖2-1所示：

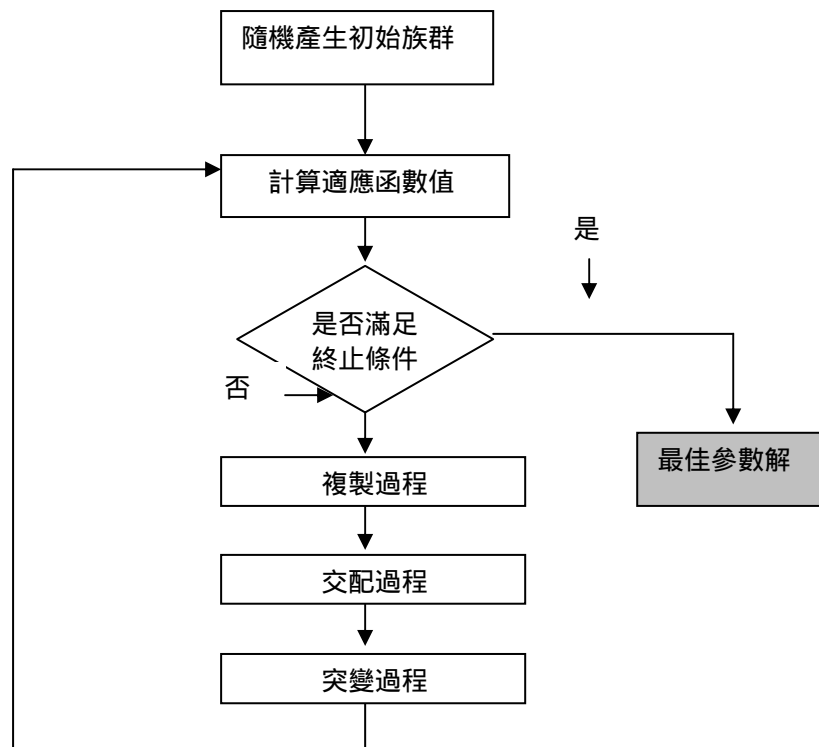


圖 2-1 基因演算法之演化流程圖

基因演算法則之主要特性：基因演算法則基本上有別於傳統之搜尋方式，即說明了為什麼找尋最佳解時需要基因演算法，說明如下：

- (1)基因演算法則是以參數集合之編碼進行運算而不是參數本身，因此可跳脫搜尋空間分析上的限制；
- (2)基因演算法則同時考慮搜尋空間上多個點而不是單一個點，因此可以較快地獲得整體最佳解(global optimum)，同時也可以避免陷入區域最佳解(local optimum)的機率，此項特性是基因演算法則的最大優點；
- (3)基因演算法則只使用適應函數的資訊而不需要其他輔助資訊（例如：梯度），因此可以使用各種型態的適應函數，並可節省計算機資源避免繁複的數學運算；
- (4)基因演算法則使用機率規則方式去引導搜尋方向，而不是用明確的規則，因此較能符合各種不同類型的最佳化問題。

然而，此演算法在使用上可能面臨的限制與缺點：

- (1)因此方法屬啟發式近似解，若求解函數為標準的凸性函數(convex function)，則求解效率必不若牛頓法等最佳化搜尋方法為佳；
- (2)須針對問題設計合適的編碼方式，編碼方式的優劣與否直接影響求解結果與求解品質；
- (3)除非模式能提供最佳解參考值，否則演算結果是否為最佳解不能確定；
- (4)由於本方法為群體同步搜尋，故會浪費電腦記憶體空間與演算速率。

基因演算法之假設條件：

- (1) 所有的解必須以字串來表示，且字串的長度必須為固定的；
- (2)一個解只能以一種字串來表示，依各字串只能代表一個解；
- (3)演算過程中，適應函數值(fitness)是唯一的回饋標準值；
- (4)演算過程中的時間的計算採離散區間衡量，稱為世代；
- (5)無論任何世代，群體大小(現行解之集合)是固定的。

基因演算法的演化與生物界的進化相類似，透過基因的編碼、複製、交配、突變來達到演化的目的，以下分別敘述這些機制的概念：

(1)編碼(Encoding)與解碼(Decoding)

使用基因演算法時，將問題的解以染色體的方式編碼是演算法是否運作的關鍵所在，主要關鍵有二[參考：Gen et al.]，一為如何將問題的顯性空間（phenotype space）的特性融入基因空間（genotype space）中，即如何將問題的解以基因方式表示；另一為當演算法在運作時基因的變化方式與型態。因此研究者須視本身問題的需要定義合適的染色體（chromosome），作為模式進行時之最基本個體單位（individual）。染色體的編碼方式在基因演算法的應用中是相當重要的部分，好的編碼方式會大幅提升運算之速度。一般常用的編碼機制有下列二種：

- (a) 二元編碼（Binary encoding）
- (b) 實數編碼或稱浮點編碼（Real-number encoding or floating point encoding）

解碼時，則將染色體上基因代表的數字或符號，轉換為一般顯示在外的結果，就如同人類基因X與Y顯示在真實世界即代表性別的不同，一般而言一個染色體解碼後的結果代表一個模式的可行解。

(2)母體大小(Population size)與初始值

母體（或稱群體）為每次演化時所存在的個體數目，母體較大表示在演算時同步搜尋的數目較多，因此有較大的機會可以找到最佳解，但因耗費的記憶體較大所以相對演算速度較慢；母體較小則較難搜尋到最佳解，且容易落入局部解陷阱中，因此研究者需視研究問題的需要定義其母體大小。基因演算法在運算之前，會先產生第一代的母體，其中所有染色體基因值是隨機產生的，並作為整個演算步驟的初始值。

(3)適合度(fitness)

為了評估母體內各個染色體的優劣，GA 必須導入一個評估指標作為演算機制運作的參考依據，此評估指標稱為適合度。依投入變數的不同，將適合度估算方式定義成適合度函數（fitness function），此函數是評估個體是否能適應模式所設定之環境的指標，亦是個體是否能繼續演化的標準。

(4)複製（Reproduction）

以「適者生存，不適者淘汰」的生物演化為主要的概念，即優良的基因有較大的機會參與遺傳演化。只要是具有高適存值的個體，就能複製較多的個體。一般的選擇種類以輪盤法（Roulette Wheel Selection）為研究中最常採用的方法，是一種啟發式的隨機選取方法。依照個體的適存函數及輪盤賭原理來做選擇的依據，將每一個個體被選擇機率按照比例依個體編號順序分佈成圓盤狀適存值大小的意義代表輪盤狹縫的寬，經過隨意旋轉輪盤選擇出下一世代的個體，可以發現的是具有較高適存值的個體，被選擇的機會較大，首先將母體中的所有個體（染色體）適合度予以加總，將總適合度視為一個輪盤全部面積。而每一個體會依其適合度大小在輪盤上佔有一定面積，適合度越高，所佔面積越大，在演算時則隨機挑選一編號，看其落於輪盤何處，便選取該處所代表的染色體出來進入下一步演算機制中。這種選擇的程序對於想要在短時間內搜尋到優良的個體，有著相當程度的影響。Goldberg於1989年提出輪盤法，依照各染色體的存活率。以隨機之方式每次抽出一個；就像賭博輪盤一樣，在輪盤上所佔的面積愈大則被抽重的機率則越大。詳細步驟如下：

步驟一：計算總是應函數值 S （母群體中所有染色體的適應值總合）；

步驟二：產生一隨機數 n （介於1與 S 之間）；

步驟三：選取適應函數數值累加至大於或等於 n 個的染色體，複製至子代；

步驟四：重複步驟二、三，直至複製個數等於族群個數。

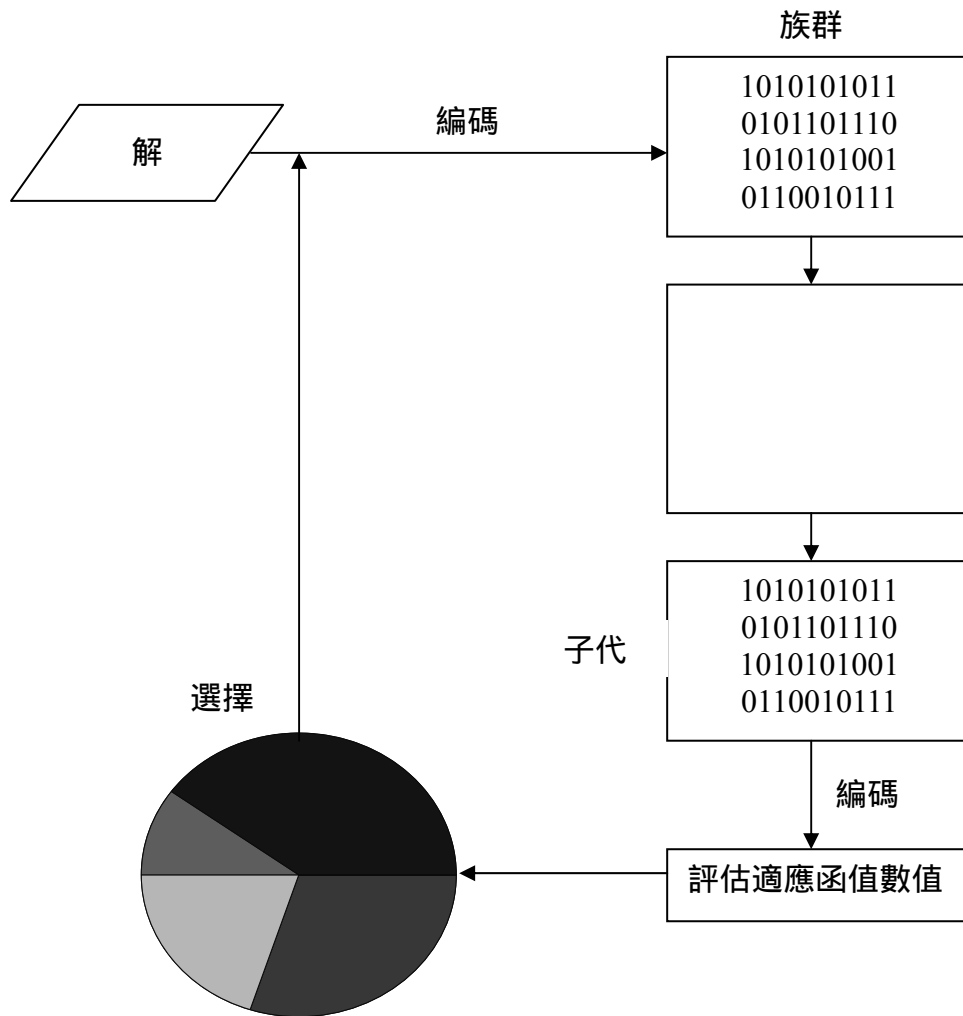


圖2-2 輪盤法

5. 交配 (Crossover)

交配之主要目的是讓被挑選出的兩個染色體交換彼此基因的訊息，以產生更優良的基因組合。在世代演化的過程中，若僅是藉由以上挑選的機制是無法使求解結果更進一步，一個染色體的進化，也許需藉助要其他染色體上的基因功能才能補全，因此必須藉由交配來達成以上目的。單點交配 (one-point crossover) 是最常採用交配方式之一，如圖2-3所示，在染色體中隨機挑選一個交配點，將兩染色體於交配點後的基因予以交換，以產生一對新的染色體，稱為子代 (offspring)，通常在GA 進行之前事先預設定一個交配機率crossover rate來控制染色體交配的頻率。

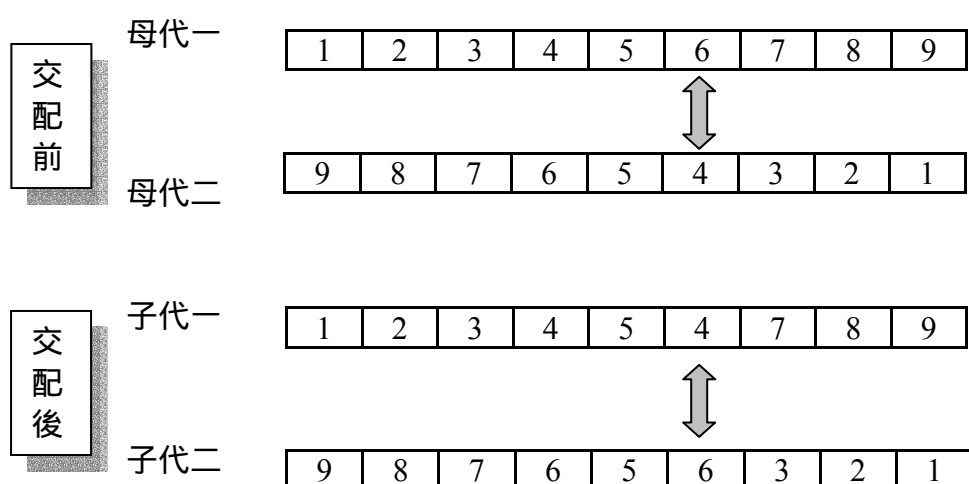


圖2-3 單點交配機制

6.突變(Mutation)

突變是基因演算法中相當特殊的機制，也是提供一個跳出目前搜尋空間避免落入局部最佳解的方式。突變機制一般來說目的有二；其一為開發新的搜尋區域，使求解方向產生無限可能；另一則為重新導入群體在演化過程中遺失的重要資訊。生物界也是藉由突變而產生出新物種，若此物種對於環境的適應更為優良，則越容易取代舊物種而形成演化的主體。與複製相同，在突變前系統會先預設一個突變機率 P_m ，作為基因是否突變的憑據，在突變機率的決定上，一般會設定在一個較低的值，以免過份破壞掉經由選擇、交配等機制所產生的優良染色體。一般文獻中所探討的突變機制有兩種，一為標準突變方式（standard mutation），設定一固定突變機率，按此機率然後逐步搜尋基因，當突變發生時，將基因值改變；一為累加突變機制（aggressive mutation），作法與標準突變相同，但突變機率隨演化代數增加，在演化初期，主要目的是收斂到最佳解附近，因此將突變率設定在較低的值，當染色體持續演化之後，由於擔心所得的解為局部最佳化，因此導入到高的突變機率，以尋找是否有逸失的最佳解。

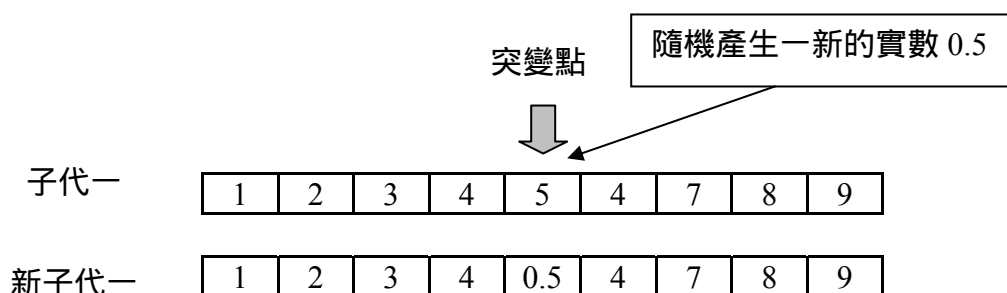


圖2-4 單點突變機制

總結，以上GA 由複製、交配、突變的運算機制構成了主要的演化流程，經過世代交替的演算過程後，直到符合適當條件下演算才會終止。終止的條件常使用的有兩類：

- (1) 達到指定的世代交替次數。
- (2) 適合度在演算過程中皆無法有效改進。

基因演算法依照運算特性的不同可分為二進位基因演算法(Binary Genetic Algorithms , BGA)與實數型基因演算法(Real-valued Genetic Algorithms , RGA)，此兩種方法之間的比較整理如下表2.2所示：

表2-1 BGA與RGA比較

演算法類別	二進位基因演算法(BGA)	實數型基因演算法(RGA)
運算過程	都包含有複製、交配、突變等三運算過程的運作	
編碼方式	以0 和1 的位元方式編碼來表示參數	以實數或稱浮點數編碼來表示參數
優缺點	<p>1. 當參數是自然量化時，BGA 有較好的配適。</p> <p>2. 當參數為連續時，藉由實數能更有邏輯的表達這些參數。</p> <p>4. RGA 比BGA 有較少記憶體儲存的優點。因為是由單一浮點數的參數來取代N 個bits 的整數，因此當N 個bits 增加時，使用BGA 則所需記憶體儲存的需求將變得更多，可能使運算速度減緩。</p> <p>5. RGA 不用作編碼及解碼的繁複運算。</p> <p>6. RGA 增加搜尋至系統整體最佳值的機會</p> <p>3. 因為BGA 是藉由二元的參數表達，因此有它的精確性限制。使用RGA 取代容易允許對機器精確性的表示，克服準確度不足的問題。因此對於在適應函數的計算上也會較為精確。</p>	

「參考資料：黃寶賢（民91）」

根據上述相關文獻尚未有以基因演算法的觀念來處理 AHP 的不完整矩陣問題，所以本研究將探討基因演算法結合成對比較矩陣要符合一致性的觀念，來求解遺漏之比較值，還原原本的完整矩陣，然後求出最終之優勢權重值。

三、不完整資訊成對比較矩陣遺漏值估計之方法

此章節的目的是要展現估算不完整資訊成對比較矩陣之遺漏值之各種方法，且決策者在填寫 AHP 問卷時，都被要求能盡可能的使一致性指標值(consistency index, 縮寫為 C.I.)越接近 0 越好，即表示此問卷是有邏輯的有效問卷；在此前提下，當成對比較之 AHP 問卷發生遺漏時（問卷為無效、不一致之問卷），過去相關之方法都朝改善一致性指標值的方向去找出遺漏值或權重值。因此，吾人所提出的基因演算法的適應函數，也依此設定為一致性值愈小越好；且本節包含說明基因演算法之操作流程與過去相關文獻方法的討論與比較。

3.1 不完整資訊成對比較矩陣基因演算法之操作

在不完整資訊之成對比較矩陣基因演算法，其產生及處理族群數（以實數表示）之操作，如圖 3-1 所示，且下述將詳細說明操作內容：

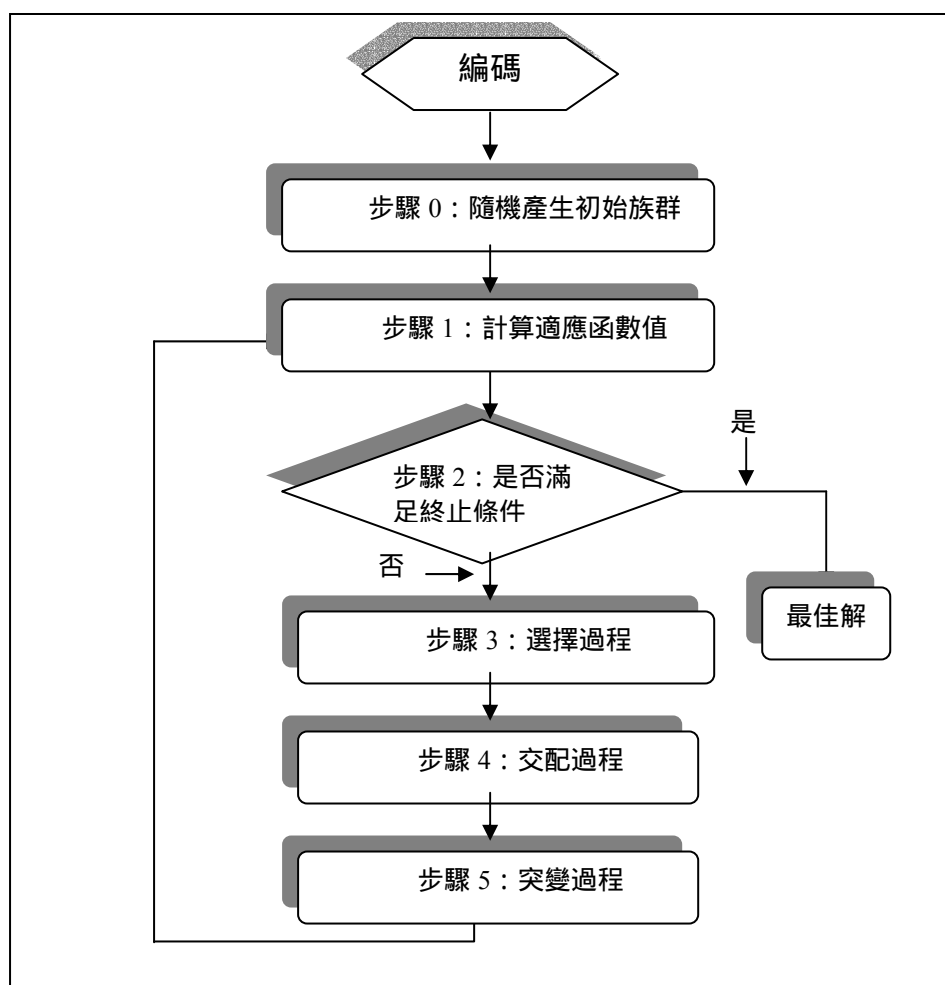


圖 3-1 基因演算法之演化流程圖

3.1.1 使用實數型基因演算法所需要考慮到的程序設定概念如下所示：

(1) 實數編碼與範圍

初始族群為第一代之染色體，一稱為母族群，一般基因演算法的初始族群的產生方式有兩種：最常見的為隨機產生；在考慮初始族群必須為可行解的條件下，且一個好的初始解產生，可加快搜尋的速度，故採有限制的隨機產生初始解。即實數編碼主要原則是在限定的範圍內 $[9 \sim 1/9]$ ，隨機產生的實數即可。

(2) 初始族群大小之設定

在基因演算法系統中，每一世代的群體大小是固定的，因此在產生初始解時，須先決定族群大小，族群大小(N_{pop})指的是染色體的數目。一般來說，族群大小是依照使用者個人主觀的判斷來決定，但假如問題之複雜性較高時，通常會設定較大的族群數量。

(3) 交配率

用來決定染色體被選擇交配的機率。主要目的是讓被挑選出的兩個染色體交換彼此基因的訊息，以產生更優良的基因組合。使用者可依照本身的問題自行設定之。

(4) 突變率

即染色體被選擇突變的機率，適度的突變機率可激發染色體內潛在的基因，以增加問題的搜尋空間，提供一個跳出目前搜尋空間避免落入局部最佳解的方式。

(5) 世代數

世代數是指基因演算過程中世代交替的次數，通常用來作為演算終止的標準。關於世代數的設定，通常依問題型態的不同，使用者自行決定疊代的次數。

3.1.2 基因演算法之操作步驟：

步驟 0：(隨機產生初始族群)：產生初始解是含有 N_{pop} 個實數之族群。在初始母體方面，初始母體是指系統內最初始的染色體狀態，也就是問題的起始解。首先定義一個 N_{pop} 染色體的初始母體。而一個 N_{pop} 染色體的初始母體之產生由下列公式所得出：

$$N_{pop}=a+ \text{Random} *(b-a)$$

其中Random：隨機產生一個介於0到1之間的數值； a ：參數範圍裡的最大數(上限)； b ：參數範圍裡的最小數(下限)。

步驟 1：(計算適應函數值)：計算出各實數之適應函數值。在實數型基因演算法中，需定義出一個符合演化目標的標準，即所謂的適應函數，它是基因演化過程中一項重要的績效指標(performance index)。將每一段染色體，即族群中之每一實數值代入適應函數中，經由運算可得到一個值，稱之為染色體的適應值。適應值代表每個染色體在演化過程的重要性與否，適應值越高，表示該染色體具有較優的特質，相對地存活或被複製的機率也較高。而在本研究裡，則是採用最小化一致性指標值：

$$C.I. = \frac{\lambda_{\max} - n}{n - 1}$$

其中 n ：倒置矩陣(reciprocal matrix)之階數(order)； λ_{\max} ：最大之特徵值。

步驟2：(是否滿足終止條件)：當已達到規定的條件，即終止運算，否則繼續運算；終止條件是指基因演化停止的基準，為了結束基因演算法的演化循環，必須事先設定終止規則，使得當基因演化過程中滿足所設定的終止規則時，能夠停止演化循環。本研究則是採用設定最大演化世代數的方式來作為測試收斂性的依據。此為最常被使用的方法，透過多次試驗程式，合理的最大世代數可被觀察出來。

步驟 3：(選擇過程)：利用輪盤法(roulette wheel selection)選擇出較佳之基因，視為交配池中的母代。複製是依據每一條染色體的適應值來決定其在下一子代中應該被淘汰或保留的一種運算過程。通常適合值越高的染色體，其存活機率也越大，也代表其擁有的基因越優秀。由於本研究的適應函數目標為最小化，所以必須將適應值低的染色體轉換為較高的適應值，以便能符合適應值越高、存活率越大的原則。當每條染色體在輪盤上所佔有的面積(也就是機率)計算完成之後，則被保留複製的機會會越大。

步驟 4：(交配)：透過母代隨機交配，產生新的子代。在挑選較優的母代染色體之後再做交配的操作，隨機產生每一段母代染色體的相對應的交配率，以作為決定哪些染色體被選擇來配對，為避免交配過度的氾濫，而造成亂倫，因此設定交配條件為，且採兩點交配的方式進行交配，當每一段染色體的交配率小於所設定的交配

率則被選擇出來交配，每次挑選兩段染色體才進行交配，未被挑選出來的染色體片段則保留原本的基因。

例如：設定之交配率為0.25，機率小於0.25之染色體將被挑出來進行交配。

交配前	母代一	1	2	3	4	5	4	7	8	9
	隨機機率	0.1	0.5	0.3	0.5	0.3	0.6	0.9	0.8	0.9
	母代二	1	2	3	4	5	4	7	8	9
	隨機機率	0.3	0.5	0.2	0.6	0.3	0.6	0.4	0.5	0.6
交配後	子代一	3	2	3	4	5	4	7	8	9
	子代二	1	2	3	1	5	4	7	8	9

步驟 5 : (突變)：在新的子代中選擇一些進行突變，跳回步驟 1。在突變方面，突變率通常設定為 0.01 左右，不過仍需視問題的特性作調整，可透過多次試驗程式後，較好的突變率可被觀察出來。突變率的方式為將交配後的子代串聯起來，然後隨機產生相對應的突變機率，若某片段的染色體其突變的機率小於所設定的突變機率，則將被選出來進行突變；突變為採單點突變，即將選出的染色體以隨機產生一新的染色體取代掉。而隨機產生之新染色體(X_{new})的方程式如下所示：

$$X_{new} = a + \text{Random} * (b - a)$$

其中Random：隨機產生一個介於0到1之間的數值； a ：參數範圍裡的最大數(上限)； b ：參數範圍裡的最小數(下限)。

例如：設定之突變率為 0.01，機率小於 0.01 之染色體將被挑出來進行突變。



3.2 方法討論

本研究所要探討解決不完整成對比較矩陣之遺漏值的方法共有五種,以下將說明各方法之基本思考及理論基礎,方法的特色,最後再以表來表示。

表 3-1：方法符號對照表

符號	作者
Har	Harker, (1987)
Non-LP	Triantaphyllou, (1995)
LP	Triantaphyllou, (1995)
Polynomial	Shiraishi etal. (1998)
GA	John Holland 教授於 1960 與 1970 年代所發展出的理論

方法一：Har 法

Harker (1987)最早提出解決不完整矩陣方法，其提出一方法來處理在不完整矩陣下之優勢權重求取。此方法是乘冪法(power method 理論基礎如附錄)之一延伸，其做法為若 (i,j) 成份是遺漏的，為 a_{ij} ，則以人工變數 w_i/w_j 來代替，完成一完整倒值矩陣 $A(w)$ ，接著為計算其特徵值。而能以 w_i/w_j 來代替原本決策者所填的成對比較數值 a_{ij} 是以線性代數和圖形理論為基礎，因此 Harker 方法條件限制：因為 AHP 成對比較內容所考慮成對的準則或方案間一定存在一間接或直接的比較，基於線性代數與圖形理論下，若一不完整之成對比較矩陣，為不可縮減矩陣，則可以保證不完整成對比較中遺漏的比較與有填的比較間是有較強的關聯，所以可有效的從有填的比較中推估出遺漏的比較，故任一不完整成對比較矩陣（未知比較以 0 表示）必須為不能縮減矩陣(irreducible)。

方法二：Non-LP 法

Triantaphyllou (1995)年提出兩個方法來估計未知的比較：(1)為簡單易懂法；(2)為使誤差最小之線性規劃法。方法基本想法為在完全一致性之成對比較倒值矩陣滿足 $a_{ij} = a_{ik}a_{kj}$ for $i, j, k = 1, 2, \dots, n$ 的關係，透過此關係任一不完整 $n \times n$ 矩陣，都可分割成兩完整之方正矩陣，此兩完整之子矩陣，分別為 $n_1 \times n_1$ 矩陣與 $n_2 \times n_2$ 矩陣，並且必須符合 $n_1 + n_2 > n$ 才可求解。

因此，在完全一致性的情況下，未知比較 x_{ij} 可以 $a_{ij} = a_{ik}a_{kj}$ for $i, j, k = 1, 2, \dots, n$ 求得，若在不一致性情況下 $a_{ij} = a_{ik}a_{kj}$ for $i, j, k = 1, 2, \dots, n$ 之關係不一定為真，但是未知之 x_{ij} 我們仍可期望去求出一很接近的值，因此以綜合所有可能的值的平均值，來作為合理衡量出未知 x_{ij} 。

方法三：LP 法

Triantaphyllou，認為簡單易懂法未考量到以下關係：

$$x_{ij} \approx (a_{ii'} \times a_{jj'}) \times x_{i'j'} \quad \text{for } i, i' = 1, 2, \dots, n - n_2; j, j' = (n_1 + 1), \dots, n$$

即沒考量到簡單易懂法所估計出的值不完全為實際值，因此提出一線性規劃法，以估計誤差最小為目標的線性規劃法，來解除 $x_{ij} = (a_{ii'} \times a_{jj'}) \times x_{i'j'}$ 的條件。

方法四：Polynomial 法

Shiraishi et al. (1998)年發現正倒值矩陣之特徵多項式(characteristic polynomial)之某些性質與 AHP 成對比較矩陣之一致性有很大的關聯存在。基於此結果，作者提出一啟發式之方法來估計不完整成對比較矩陣之遺漏要素。

因為期望求得最佳 $\lambda_{\max}(x)$ ，令 $c_3(x)$ 為 $A(x)$ 特徵多項式中 λ^{n-3} 之係數。從理論 1，可知 $c_3 \leq 0$ 且假設我們能夠使 $c_3(x) = 0$ 則 $A(x)$ 能變成一致性。因此，期望 $A(x)$ 能有較好的一致性，則 c_3 越接近 0 越好，(理論 1)若 A 為一 $n \times n$ 的正倒值矩陣，則 A 為一致是只有在 $P_A(\lambda) = \lambda^n - n\lambda^{n-1}$ ，最後提出一公式求算遺漏值，但此法只能求算一個遺漏值。

方法五：GA 法

本研究嘗試以最佳化工具之一基因演算法來求算遺漏值，因基因演算法為一演算法，不需任何的數學理論證明，且操做簡單，而在設定適應函數，即基因在搜尋最佳遺漏值的準則為，以一致性最小為適應函數之目標，因為 AHP 問卷是絕對要求決策者盡可能的符合一致性的填寫此份問卷，也因此決策者在填寫問卷時，是朝一致性高的思考方向來填寫 AHP 問卷，所以一致性指標值與漏填的比較值有很大的關係存在，且前述的各方法也是以此想法而發展出各種不同的方法。最後基因演算法是透過交配與突變的生物生存法則來選擇出最佳的解，因此對於遺漏值的各數或位置完全沒有任何限制。

各方法之前提假設：決策者填寫 AHP 問卷時是盡可能的一致性。

表 3-2 方法之比較

方法名稱	方法基本想法	使用之限制條件
Har (1987)	在線性代數法與圖形理論為基礎下，以乘冪法為延伸，在不完整矩陣下，求取近似之權重。	不完整矩陣不可成為可縮減矩陣。
Non-LP (1995)	將不完整矩陣分割成兩完整之子矩陣，依據層級分析法中的遞移性，以簡單平均法，求出遺漏值。	分割之兩子矩陣階數總和必須大於不完矩陣之階數。
LP (1995)	修正 Non-LP，以估計值與實際值誤差最小之線性規劃法來求出遺漏值。	分割之兩子矩陣階數總和必須大於不完矩陣之階數。
Polynomial (1998)	以特徵多項式為主，發現其中 C_3 係數與一致性有很密切之關係，然後以 C_3 係數最小化，提出一求算遺漏值之公式。	只能求算一個遺漏值。
GA	因一致性指標值與漏填的比較值有很大的關係	無任何限制

存在，故以一致性指標值最小化為適應函數，來求算遺漏值。

3.3 方法之求解範圍限制

在討論 4-6 之矩陣大小的研究範圍下，各方法求解遺漏值的位置與個數限制，將以表格的方式詳細說明，且層級分析法之問卷設計只問決策者上三角矩陣的部分，與其相對應之下三角矩陣的值為倒數關係，因此下三角矩陣的值可由上三角矩陣的值求出，故當問卷發生遺漏狀況時，即表示上三角矩陣的某一成對比較值未知，當然其相對應之下三角矩陣之成對比較值也未知；最後之整理結果如下所示（ n ，矩陣大小），以下數表所整理的遺漏值位置與個數，因為求簡潔，所以只表示上三角矩陣的遺漏個數與位置，其相對應的下三角矩陣的部分則省略：

3.3.1 Har 法

Har 法之方法限制為在多大之矩陣大小，其遺漏值的個數或位置都必須為不可縮減矩陣，即 $n=4$ ，必須回答至少 4 個上三角矩陣內的成對比較元素，因此矩陣大小為 4 之成對比較矩陣，只能遺漏 2 個遺漏值，其他矩陣大小，同理類推，而針對遺漏一個與兩個遺漏值的詳細內容如表 3-3 所示。

表 3-3：Har 法之一個與兩個遺漏值之求解範圍

n	圖形說明	一個遺漏值之位置	二個遺漏值的位置
4	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} \\ x_{21} & 1 & x_{23} & x_{24} \\ x_{31} & x_{32} & 1 & x_{34} \\ x_{41} & x_{42} & x_{43} & 1 \end{bmatrix}$	<p>共有 6 種遺漏位置：</p> <p>$x_{12} ; x_{13} ; x_{14} ;$</p> <p>$x_{23} ; x_{24} ; x_{34}$</p>	<p>共有 15 種遺漏位置：</p> <p>將 $x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34}$ 做兩兩組合。</p>
5	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 \end{bmatrix}$	同理類推	同理類推

6	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} & x_{36} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} & x_{46} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 & x_{56} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 1 \end{bmatrix}$	同理類推	同理類推
---	--	------	------

3.3.2 Non-LP 與 LP 兩方法

Non-LP 與 LP 是將不完整資訊之成對比較矩陣分割成兩個完整的子矩陣，左上方的子矩陣大小以 n_1 表示，右下方之子矩陣大小以 n_2 表示，且此兩子矩陣的階數(order)加總要大於原本成對比較矩陣之階數，如表 3-4 所示。

表 3-4：Non-LP 與 LP 兩方法之求解範圍

n	n_1	n_2	圖形說明	遺漏位置
4	3	3	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} \\ x_{21} & 1 & x_{23} & x_{24} \\ x_{31} & x_{32} & 1 & x_{34} \\ x_{41} & x_{42} & x_{43} & 1 \end{bmatrix}$	x_{14}
5	3	3	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 \end{bmatrix}$	$x_{14}, x_{15}, x_{24}, x_{25}$
5	3	4	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 \end{bmatrix}$	x_{14}, x_{15}
5	4	3	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 \end{bmatrix}$	x_{15}, x_{25}

5 4 4

1	x_{12}	x_{13}	x_{14}	x_{15}
x_{21}	1	x_{23}	x_{24}	x_{25}
x_{31}	x_{32}	1	x_{34}	x_{35}
x_{41}	x_{42}	x_{43}	1	x_{45}
x_{51}	x_{52}	x_{53}	x_{54}	1

x_{15}

6 3 4

1	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	1	x_{23}	x_{24}	x_{25}	x_{26}
x_{31}	x_{32}	1	x_{34}	x_{35}	x_{36}
x_{41}	x_{42}	x_{43}	1	x_{45}	x_{46}
x_{51}	x_{52}	x_{53}	x_{54}	1	x_{56}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	1

$x_{14}, x_{15}, x_{16}, x_{24}, x_{25}, x_{26}$

6 4 3

1	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	1	x_{23}	x_{24}	x_{25}	x_{26}
x_{31}	x_{32}	1	x_{34}	x_{35}	x_{36}
x_{41}	x_{42}	x_{43}	1	x_{45}	x_{46}
x_{51}	x_{52}	x_{53}	x_{54}	1	x_{56}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	1

$x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36}$

6 3 5

1	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	1	x_{23}	x_{24}	x_{25}	x_{26}
x_{31}	x_{32}	1	x_{34}	x_{35}	x_{36}
x_{41}	x_{42}	x_{43}	1	x_{45}	x_{46}
x_{51}	x_{52}	x_{53}	x_{54}	1	x_{56}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	1

x_{41}, x_{51}, x_{61}

6 4 4

1	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	1	x_{23}	x_{24}	x_{25}	x_{26}
x_{31}	x_{32}	1	x_{34}	x_{35}	x_{36}
x_{41}	x_{42}	x_{43}	1	x_{45}	x_{46}
x_{51}	x_{52}	x_{53}	x_{54}	1	x_{56}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	1

$x_{15}, x_{16}, x_{25}, x_{26}$

6	4	5	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} & x_{36} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} & x_{46} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 & x_{56} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 1 \end{bmatrix}$	x_{15}, x_{16}
---	---	---	--	------------------

6	5	4	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} & x_{36} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} & x_{46} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 & x_{56} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 1 \end{bmatrix}$	x_{16}, x_{26}
---	---	---	--	------------------

6	5	5	$\begin{bmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{21} & 1 & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{31} & x_{32} & 1 & x_{34} & x_{35} & x_{36} \\ x_{41} & x_{42} & x_{43} & 1 & x_{45} & x_{46} \\ x_{51} & x_{52} & x_{53} & x_{54} & 1 & x_{56} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & 1 \end{bmatrix}$	x_{16}
---	---	---	--	----------

3.3.3 Polynomial 法

Polynomial 法因方法的限制，針對上三角矩陣而言只能求算一個遺漏值，如表 3-5 所示。

表 3-5:Polynomial 法之求解範圍

n	遺漏位置
4	$x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34}$
5	$x_{12}, x_{13}, x_{14}, x_{15}, x_{23}, x_{24}, x_{25}, x_{34}, x_{35}, x_{45}$
6	$x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{23}, x_{24}, x_{25}, x_{26}, x_{34}, x_{35}, x_{36}, x_{45}, x_{46}, x_{56}$

3.3.4 基因演算法

基因演算法只需設定適應函數，來選擇出最佳的解，因此沒有任何的限制式，故任何的遺漏位置或個數，都可使用。

3.4 方法之績效衡量

不同的方法在預測遺漏值的效果上一定會有不同的結果，而如何衡量出各方法在此問題上的績效？就必須採用重要的衡量指標來進行衡量，在探討相關預測的問題時，預測模式的好壞，都是以預測之誤差來作為最重要的衡量指標，因此估計遺漏之比較值時，當然也是估計值要盡可能的與實際比較值越接近越好，故以估計誤差為衡量指標。另一方面，各方法是否能夠在估計出遺漏值後的同時，能使問卷為有效之問卷（一致性指標值小於 0.1），故以命中率為第二個衡量指標，來衡量各方法救回問卷的績效。而指標三為各方法的程式執行時間，此為表示求解過程的速度績效指標，提供要使用各方法所需花費時間的參考。

此三指標以一、二指標最重要，因為已足夠代表方法在執行上的績效，指標三則代表在實際執行時所需花費的時間，即代表實作上是否方便使用（註：本研究以 Matlab 程式撰寫）。本研究會將此三指標的在不同實驗下的模擬結果，作一整理。

指標一：估計誤差 = $\sum (\text{估計值} - \text{實際值})^2 / \text{總矩陣樣本數}$

指標二：命中率 = $\frac{\text{估計遺漏值後成對比較矩陣之一致性指標值小於 0.1 之樣本數}}{\text{總成對比較矩陣樣本數}}$

指標三：方法之程式執行時間（500 個樣本的總執行時間）

四、數值例模擬與模擬結果之討論

我們應用數值模擬來評估比較各方法之估算誤差（即平均(估算出的遺漏值與實際比較值差的平方)），與各方法之命中率。然當 AHP 之成對比較矩陣在回收問卷時，極可能發生部分之成對比較有缺填的情況，即有遺漏之比較值，希望能透過方法還原原本的比較值，以減少時間或成本的損失；吾人可將此之觀念轉移以 3.1 節所述之基因演算法來處理此不完整資訊成對比較矩陣問題，對於不完整資訊成對比較矩陣問題，基因演算法頗有效率。

而實驗的範圍與結果將在下面討論，且實驗結果也顯示出在估算誤差部分，整體而言，基因演算法比過去相關方法的預測能力好。為了使實驗模擬具代表性，針對如下所示之矩陣以電腦隨機產生一致性指標值小於等於 0.1 之隨機矩陣，共 500 個，然後進行模擬實驗；共有三個模擬實驗，模擬範圍為 4-6 之成對比較矩陣，其中實驗一，隨機遺漏一個成對比較元素；實驗二，隨機遺漏二個成對比較元素；實驗三，由於 Non-LP 與 LP 法之方法限制，因此僅有 12 種遺漏狀況，整體之實驗模擬內容如下圖所示：

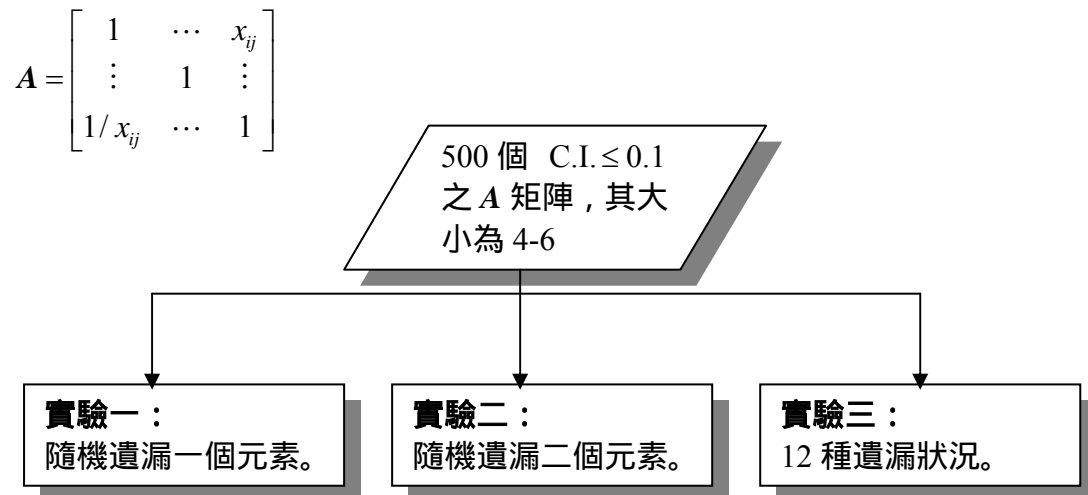


圖 4-1 實驗模擬內容

4.1 實驗步驟與模擬實驗一

實驗一主要模擬的狀況為，在4-6大小的成對比較矩陣中，發生隨機遺漏一個比較值的問題模擬；模擬的方法共有三種。在參數設定方面，主要著重於使用實數型基因演

算法所需決定之參數，包括族群大小、交配率、突變率以及世代數。世代數的取決需依照搜尋過程中適應值收斂的程度來決定，經由多次模擬試驗結果，在矩陣大小為4之情況下大致在60代後就趨於收斂；矩陣大小為5之情況下大致在40代後就趨於收斂；矩陣大小為6之情況下也在40代後就趨於收斂，最後各個參數的設定值如下表4-1所示：

◆ 表4-1實驗一之GA運算子設定

	● 矩陣大小：4	● 矩陣大小：5	● 矩陣大小：6
群體大小	40	60	60
演化代數	60	40	40
選擇型態	輪盤法	輪盤法	輪盤法
交配型態	兩點交配	兩點交配	兩點交配
交配機率	0.25	0.35	0.35
突變機率	0.01	0.01	0.02

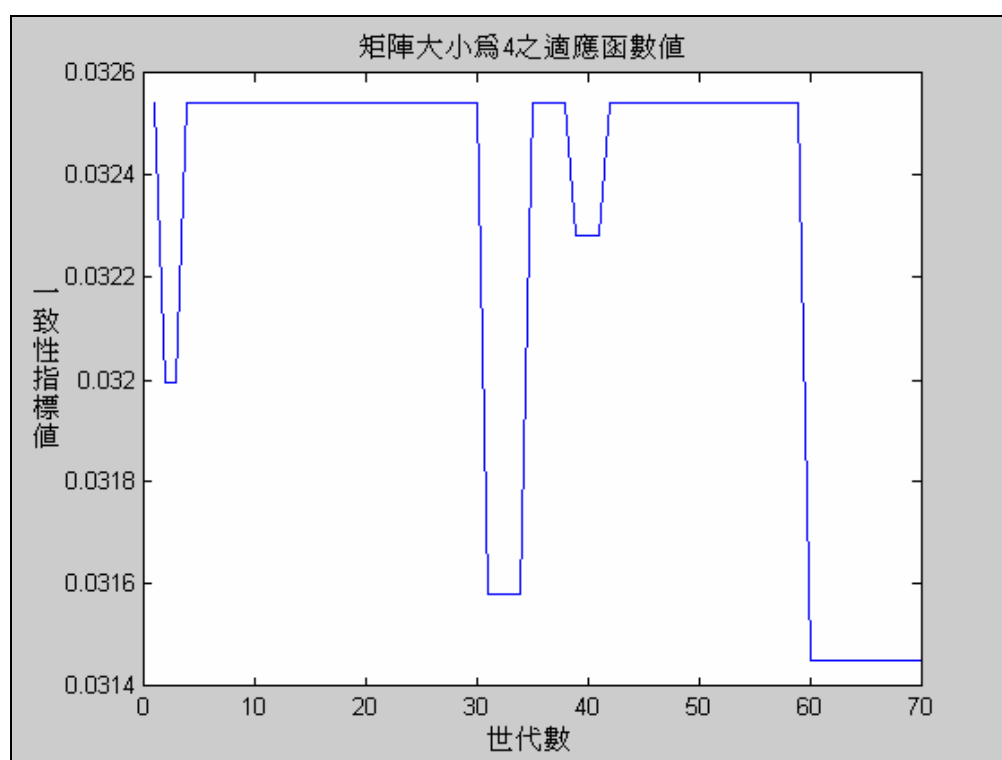


圖 4-2 適應值收斂圖

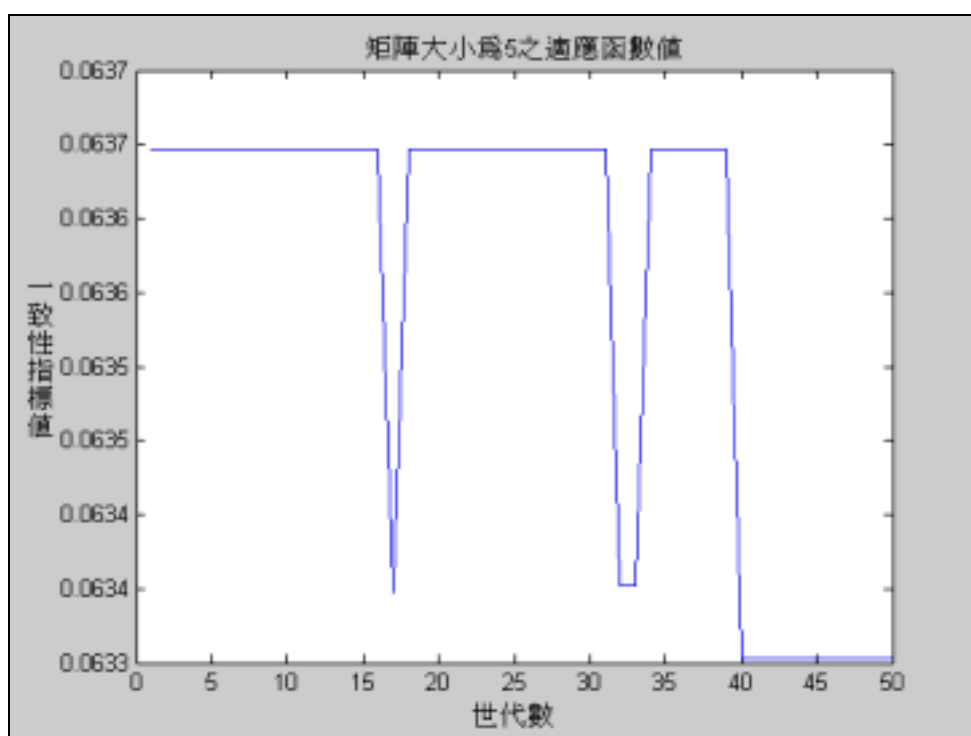


圖 4-3 適應值收斂圖

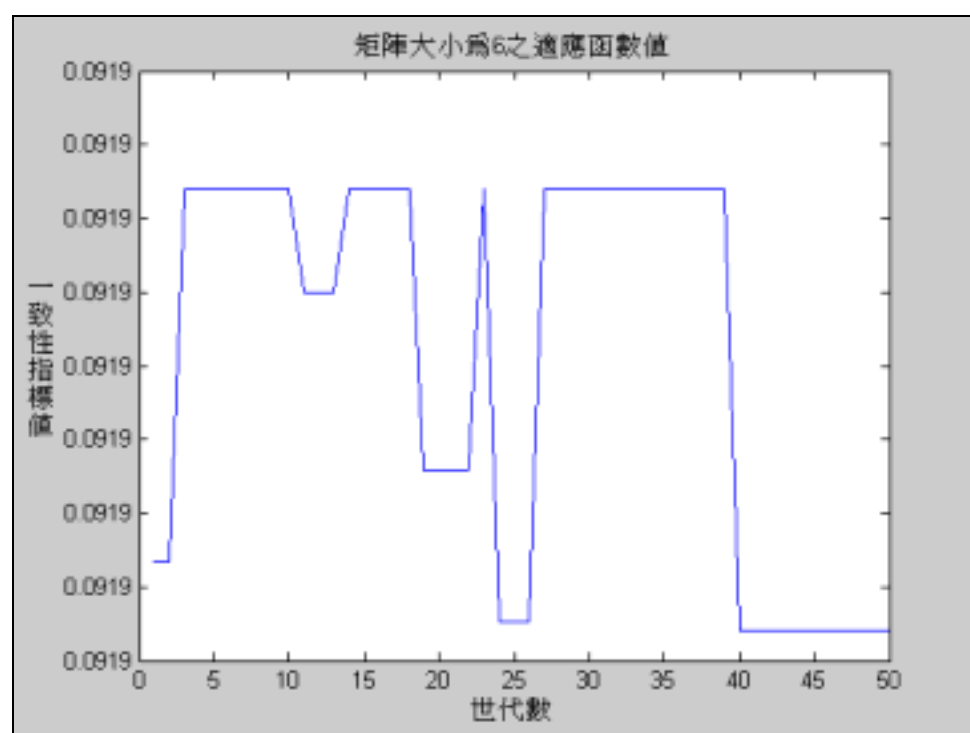


圖 4-4 適應值收斂圖

◆ 實驗一之 GA 執行步驟：

步驟 0：利用電腦，隨機產生 x_{ij} ，其值的範圍為 $\{9 \sim 1/9\}$ 的正倒值矩陣(reciprocal matrix)，各產生 500 個符合一致性($C.I. \leq 0.1$)之矩陣，大小為 4-6 之成對比較矩陣，以此為各實驗模擬的資料庫。

步驟 1：針對矩陣大小為 4-6 的完整矩陣，分別隨機遺漏一個成對比較矩陣上三角之元素，和其相對應之下三角的元素。

步驟 2：使用三方法（Har、polynomial、GA）分別估算出遺漏值。

步驟 3：使用 Matlab 程式來執行模擬實驗，且我們以平均估算誤差平方與命中率兩指標，來探討方法對遺漏值的估計能力。分別計算出 4-6 矩陣之整體遺漏值之平均誤差與救回問卷數之百分比，若救回問卷未達百分之百時，扣除失敗救回的問卷（一致性值大於 0.1），然後再重新計算估計誤差。

步驟 4：基因演算法之 Npop、世代數、交配率、突變率等，參照上表所整理的實驗一之 GA 運算子設定來執行基因演算法。

步驟 5：其結果應用 Excel 來繪圖展示結果。

表 4-2 至 4-4 之估計誤差，表示平均原本值減估計值之平方；命中率，表示估算出遺漏值後，能使問卷還原為有一致性之成對比較矩陣，佔全部隨機矩陣樣本的百分比。修正之誤差，表示扣除掉不一致之矩陣樣本（估計出遺漏值後，此成對比較矩陣仍為不一致），再重新平均估計誤差平方。圖 4-5 與圖 4-6 之 X 座標（矩陣大小）：表示 AHP 矩陣之階數(order)。Y 座標（估計誤差）：表示對應不同大小矩陣之全部遺漏位置的平均誤差平方，故當平均誤差越小即表示此方法的估算績效較好。

表 4-2：矩陣大小為 4 之各方法結果

	Har	Polynomial	GA
估計誤差	9.8676	146.6641	2.8235
命中率	100 %	42.8 %	93.4 %
修正之估計誤差	9.8676	54.5915	2.7782

表 4-3：矩陣大小為 5 之各方法結果

	Har	Polynomial	GA
估計誤差	6.0078	155.0002	2.5397
命中率	100 %	33.8 %	92.8 %
修正之估計誤差	6.0078	79.1150	2.4252

表 4-4：矩陣大小為 6 之各方法結果

	Har	Polynomial	GA
估計誤差	5.7005	139.6792	2.2821
命中率	100 %	38.8 %	93.8 %
修正之估計誤差	5.7005	34.7033	2.3737

表 4-5：矩陣大小為 4-6 之各方法程式執行時間（秒）

	Har	Polynomial	GA
矩陣大小為 4	2	1.719	1227
矩陣大小為 5	2.063	1.812	1336
矩陣大小為 6	2.188	1.921	1890

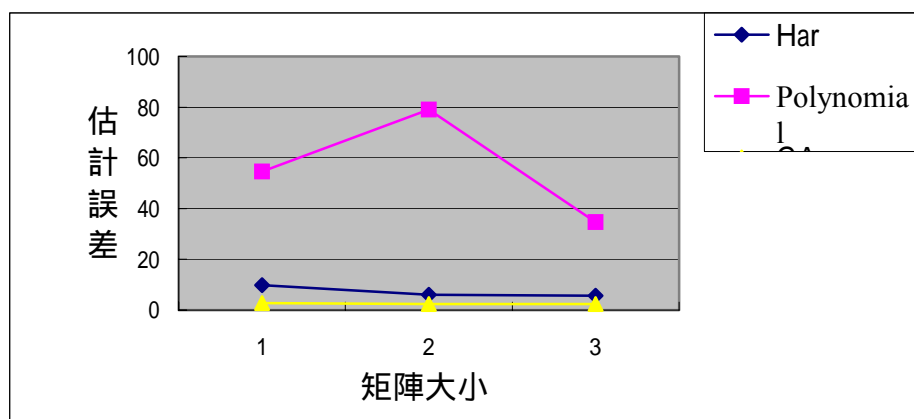


圖 4-5：三方法之估計誤差

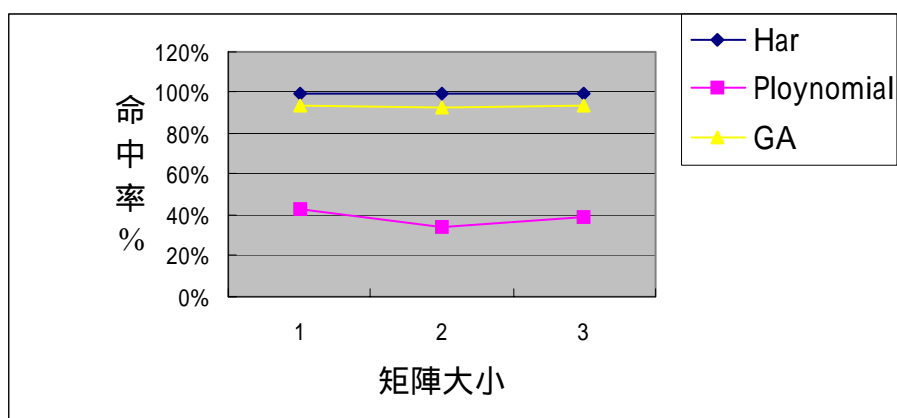


圖 4-6：三方法之救回問卷百分比

由上述實驗一之模擬結果，從表 4-1 至 4-3 和圖 4-1 至 4-2 都可明顯看出 GA 法在隨機遺漏一個遺漏值的估計誤差是最小的，但在於命中率以 Har 法最好，其可百分之百救回問卷，其次為 GA 法，也有九十百分比的命中率。而 Polynomial 法不論在估計誤差或命中率都是最差的。

4.2 實驗步驟與模擬實驗二

實驗二主要模擬的狀況為，在 4-6 大小的成對比較矩陣中，發生隨機遺漏兩個比較值的問題模擬；模擬的方法共有兩種。在參數設定方面，主要著重於使用實數型基因演算法所需決定之參數，包括族群大小、交配率、突變率以及世代數。世代數的取決需依照搜尋過程中適應值收斂的程度來決定，經由多次模擬試驗結果，在矩陣大小為 4 之情況下大致在 48 代後就趨於收斂；矩陣大小為 5 之情況下大致在 55 代後就趨於收斂；矩陣大小為 6 之情況下也大致在 55 代後就趨於收斂，最後各個參數的設定值如表 4-6 所示：

◆ 表4-6 實驗二之GA運算子設定

	● 矩陣大小：4	● 矩陣大小：5	● 矩陣大小：6
群體大小	40	40	40
演化代數	60	60	60
選擇型態	輪盤法	輪盤法	輪盤法
交配型態	兩點交配	兩點交配	兩點交配
交配機率	0.25	0.35	0.35
突變機率	0.01	0.01	0.02

◆ 實驗二之 GA 執行步驟：

步驟 1：針對矩陣大小為 4-6 的完整矩陣，分別隨機遺漏二個成對比較矩陣上三角之元素，和其相對應之下三角的元素。

步驟 2：使用二方法（Har、GA）分別估算出遺漏值。

步驟 3：基因演算法之 Npop、世代數、交配率、突變率等，參照上表所整理的實驗二之 GA 運算子設定來執行基因演算法。

步驟 4：使用 Matlab 程式來執行模擬實驗，且我們以平均估算誤差平方與救回問卷之能力兩指標，來探討方法對遺漏值的估計能力。分別計算出 4-6 矩陣之整體遺漏值之平均誤差與救回問卷數之百分比，若救回問卷未達百分之百時，扣除失敗救回的問卷（一致性值大於 0.1），然後再重新計算估計誤差。

步驟 5：其結果應用 Excel 來繪圖展示結果。

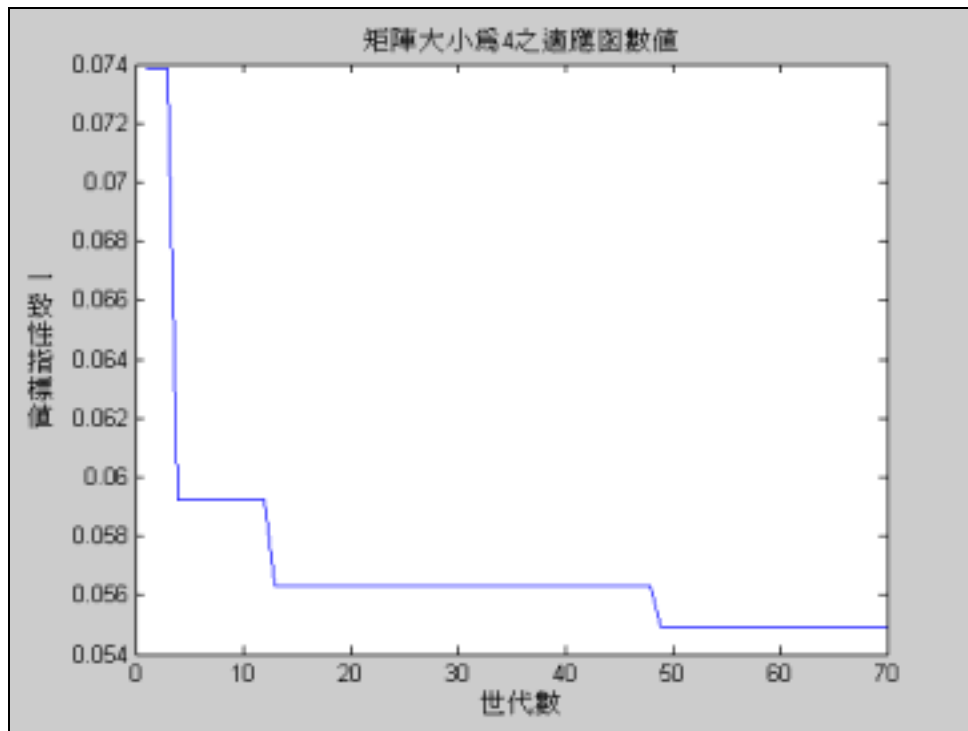


圖 4-7 適應值收斂圖

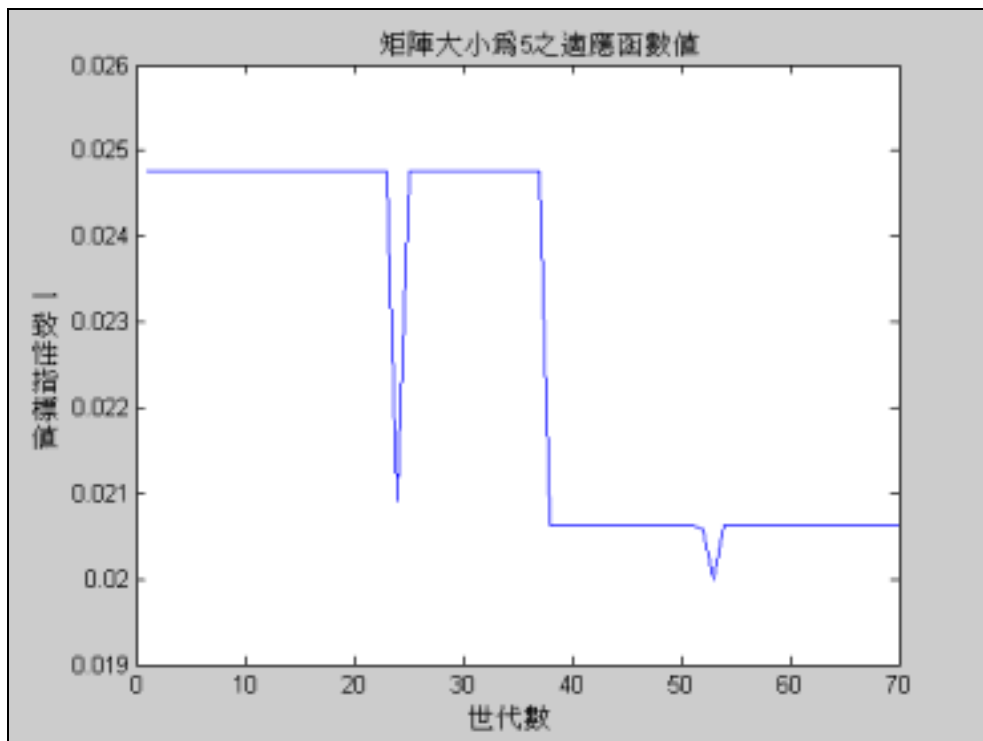


圖 4-8 適應值收斂圖

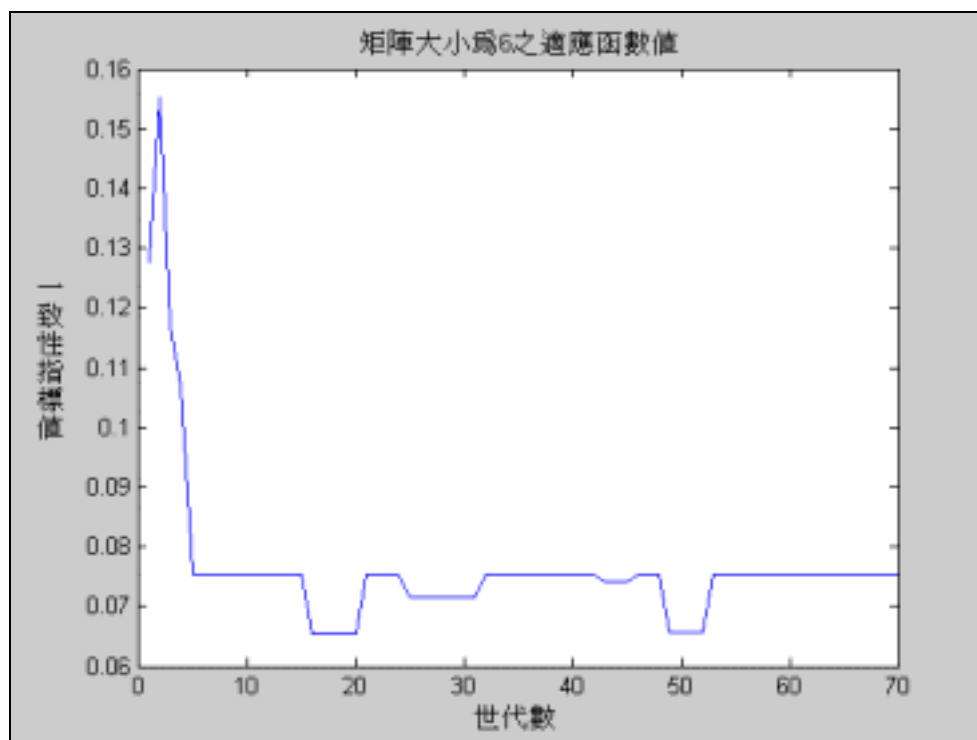


圖 4-9 適應值收斂圖

表 4-7：矩陣大小為 4 之各方法結果

	Har	GA
估計誤差	16.8291	3.2125
命中率	100 %	92 %
修正之估計誤差	16.8291	3.1455

表 4-8：矩陣大小為 5 之各方法結果

	Har	GA
估計誤差	9.6352	2.6541
命中率	100 %	91.4 %
修正之估計誤差	9.6352	2.6249

表 4-9：矩陣大小為 6 之各方法結果

	Har	GA
估計誤差	5.1197	2.5323
命中率	100 %	100 %
修正之估計誤差	5.1197	2.5323

表 4-10：矩陣大小為 4-6 之各方法程式執行時間（秒）

	Har	GA
矩陣大小為 4	1.906	929.5
矩陣大小為 5	1.656	1234.5
矩陣大小為 6	1.719	1140

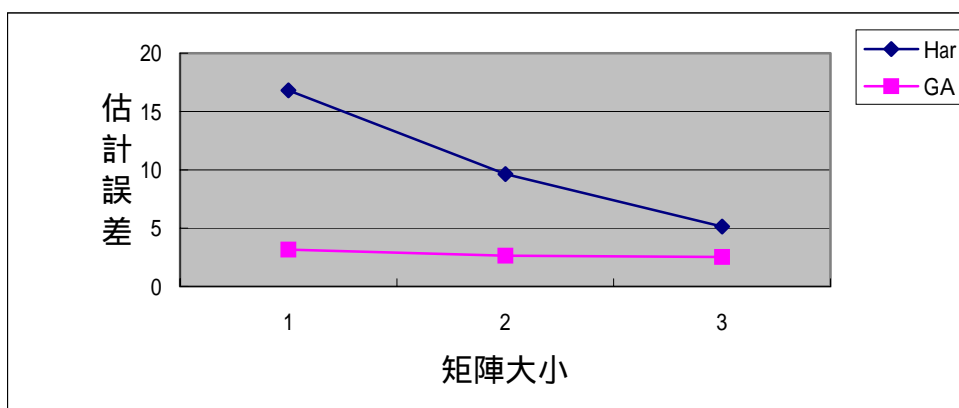


圖 4-10：二方法之估計誤差

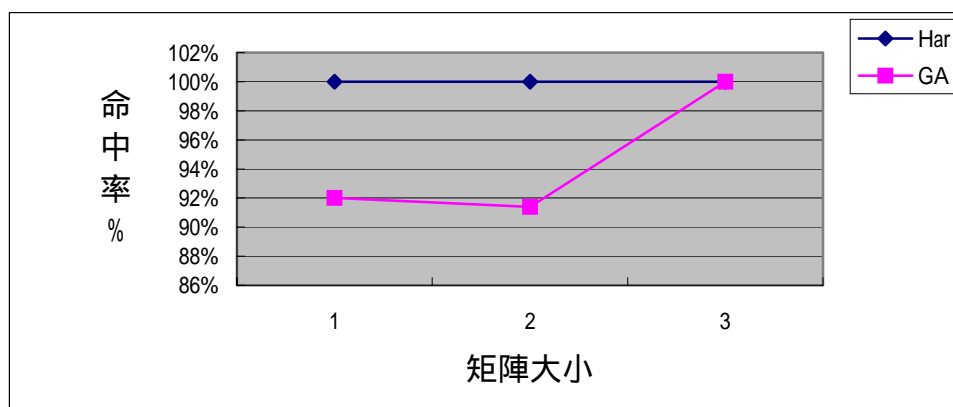


圖 4-11：二方法之命中率

由實驗模擬之結果可看出，隨機遺漏兩個元素，GA 法的估計誤差仍比 Har 好，而在命中率方面 Har 法還是能夠百分之百救回問卷，而在矩陣大小為 6 時，GA 之命中率與 Har 相同。

4.3 實驗步驟與模擬實驗三

此部分之實驗為以 Non-LP 和 LP 方法的為主，然後整理出十二種狀況，然後其他方法包括 Har 法、Polynominal 法、GA 法再對照十二種狀況是否可求解，所以模擬的方法共有五種，整理結果如下述表 4-12 所示。在參數設定方面，主要著重於使用實數型基因演算法所需決定之參數，包括族群大小、交配率、突變率以及世代數。世代數的取決需依照搜尋過程中適應值收斂的程度來決定，經由多次模擬試驗結果，最後整理各個參數的設定值如表 4-11 所示：

◆ 表4-11 實驗三之GA運算子設定：

	群體大小	演化代數	選擇型態	交配型態	交配機率	突變機率
狀況一	50	20	輪盤法	兩點交配	0.25	0.02
狀況二	30	60	輪盤法	兩點交配	0.35	0.02
狀況三	60	40	輪盤法	兩點交配	0.25	0.02
狀況四	50	30	輪盤法	兩點交配	0.25	0.02
狀況五	30	40	輪盤法	兩點交配	0.25	0.01
狀況六	40	60	輪盤法	兩點交配	0.35	0.02
狀況七	60	40	輪盤法	兩點交配	0.35	0.02
狀況八	60	40	輪盤法	兩點交配	0.35	0.02
狀況九	60	60	輪盤法	兩點交配	0.35	0.02
狀況十	60	40	輪盤法	兩點交配	0.35	0.02
狀況十一	60	40	輪盤法	兩點交配	0.35	0.02
狀況十二	50	20	輪盤法	兩點交配	0.25	0.02

◆ 實驗三之 GA 執行步驟：

步驟 1：分析 Non-LP 和 LP 方法的限制，故共只能模擬 12 種狀況。

步驟 2：整理詳細模擬狀況與可解決此 12 種狀況的其他方法，結果如下表 4-7 所示；遺漏狀況的表示，只列出上三角矩陣的部分，詳細內容請參看上述之表 3-3。

步驟 3：在基因演算法之部分，各狀況之運算子之設定，即對照上述表 4-7 之設定來執行各狀況的實驗模擬。

步驟 4：使用 Matlab 程式來執行模擬實驗，分別計算出 12 種遺漏狀況之各方法的估計誤差與命中率，若救回問卷未達百分之百時，扣除失敗救回的問卷（一致性值大於 0.1），然後再重新計算估計誤差。

步驟 5：其結果應用 Excel 來繪圖展示結果。

表 4-12：十二種遺漏狀況與解決方法（*，表示此方法可求解）

矩陣大小	遺漏狀況(上三角矩陣)	方法名稱				
		Non-LP	LP	Har	Polynomial	GA
n=4	x_{14}	*	*	*	*	*
	$x_{14}, x_{15}, x_{24}, x_{25}$	*	*	*	-	*
n=5	x_{14}, x_{15}	*	*	*	-	*
	x_{15}, x_{25}	*	*	*	-	*
	x_{15}	*	*	*	*	*
	$x_{14}, x_{15}, x_{16}, x_{24}, x_{25}, x_{26}$	*	*	*	-	*
n=6	$x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36}$	*	*	*	-	*
	x_{14}, x_{15}, x_{16}	*	*	*	-	*
	$x_{15}, x_{16}, x_{25}, x_{26}$	*	*	*	-	*
	x_{15}, x_{16}	*	*	*	-	*
	x_{16}, x_{26}	*	*	*	-	*
	x_{16}	*	*	*	*	*

表 4-13：十二種遺漏狀況各方法之估計誤差

矩陣大小	遺漏狀況	Non-LP	LP	Har	Polynomial	GA
n=4	x_{14}	14.2544	5.3635	11.2969	26.9798	3.6788
	$x_{14}, x_{15}, x_{24}, x_{25}$	15.6369	11.7858	9.1927	-	3.0387
n=5	x_{14}, x_{15}	13.34805	12.8	10.4183	-	3.6213
	x_{15}, x_{25}	18.5176	17.5029	11.9366	-	3.2219
	x_{15}	12.8675	5.0773	9.2006	27.2469	3.0317
	$x_{14}, x_{15}, x_{16}, x_{24}, x_{25}, x_{26}$	13.3445	7.365	8.8490	-	3.2689
n=6	$x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36}$	13.6263	13.4407	12.1791	-	3.0074
	x_{14}, x_{15}, x_{16}	15.2463	17.3204	8.6111	-	2.8103
	$x_{15}, x_{16}, x_{25}, x_{26}$	14.7152	13.1897	9.9683	-	3.2469
	x_{15}, x_{16}	19.6272	18.9682	11.0933	-	3.0942
	x_{16}, x_{26}	22.7527	21.3401	16.1999	-	2.6968
	x_{16}	12.2026	4.8843	8.5290	27.3789	3.1263

表 4-14：十二種遺漏狀況各方法之命中率

矩陣大小	遺漏狀況	Non-LP	LP	Har	Polynomial	GA
$n=4$	x_{14}	99.2%	98.2%	100%	90.8 %	95.8%
$n=5$	$x_{14}, x_{15}, x_{24}, x_{25}$	96.8%	24.8%	100%	-	92.4%
	x_{14}, x_{15}	97.6%	93.8%	100%	-	91.6%
	x_{15}, x_{25}	90.8%	91%	100%	-	91.4%
	x_{15}	99%	92.2%	100%	83.8 %	95%
	$x_{14}, x_{15}, x_{16}, x_{24}, x_{25}, x_{26}$	97.6%	78.4%	100%	-	90.4 %
$n=6$	$x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36}$	98%	92.6%	100%	-	90.4 %
	x_{14}, x_{15}, x_{16}	96.4%	19%	100%	-	86.8 %
	$x_{15}, x_{16}, x_{25}, x_{26}$	96.6%	93%	100%	-	89.2 %
	x_{15}, x_{16}	88.2%	88.4%	100%	-	90.6%
	x_{16}, x_{26}	88.4%	91%	100%	-	92.4 %
	x_{16}	98.8%	93.4%	100%	81.8 %	96.6%

表 4-15：十二種遺漏狀況各方法之程式執行時間（秒）

矩陣大小	遺漏狀況	Non-LP	LP	Har	Polynomial	GA
$n=4$	x_{14}	1.687	8.984	1.968	1.625	609.5
$n=5$	$x_{14}, x_{15}, x_{24}, x_{25}$	1.182	11.36	1.187	-	750
	x_{14}, x_{15}	1.687	10.296	1.703	-	880
	x_{15}, x_{25}	2.422	12.766	1.469	-	890
	x_{15}	1.734	8.61	2.047	1.672	890
	$x_{14}, x_{15}, x_{16}, x_{24}, x_{25}, x_{26}$	2.171	14.61	1.015	-	1220
$n=6$	$x_{15}, x_{16}, x_{25}, x_{26}, x_{35}, x_{36}$	2.265	15.391	1.141	-	1500
	x_{14}, x_{15}, x_{16}	0.703	12.734	1.141	-	1300
	$x_{15}, x_{16}, x_{25}, x_{26}$	0.75	14.7	1.109	-	1515.5
	x_{15}, x_{16}	1.938	13.11	1.766	-	1250
	x_{16}, x_{26}	1.859	13.437	1.884	-	1258
	x_{16}	2.031	90.093	2.141	1.797	453

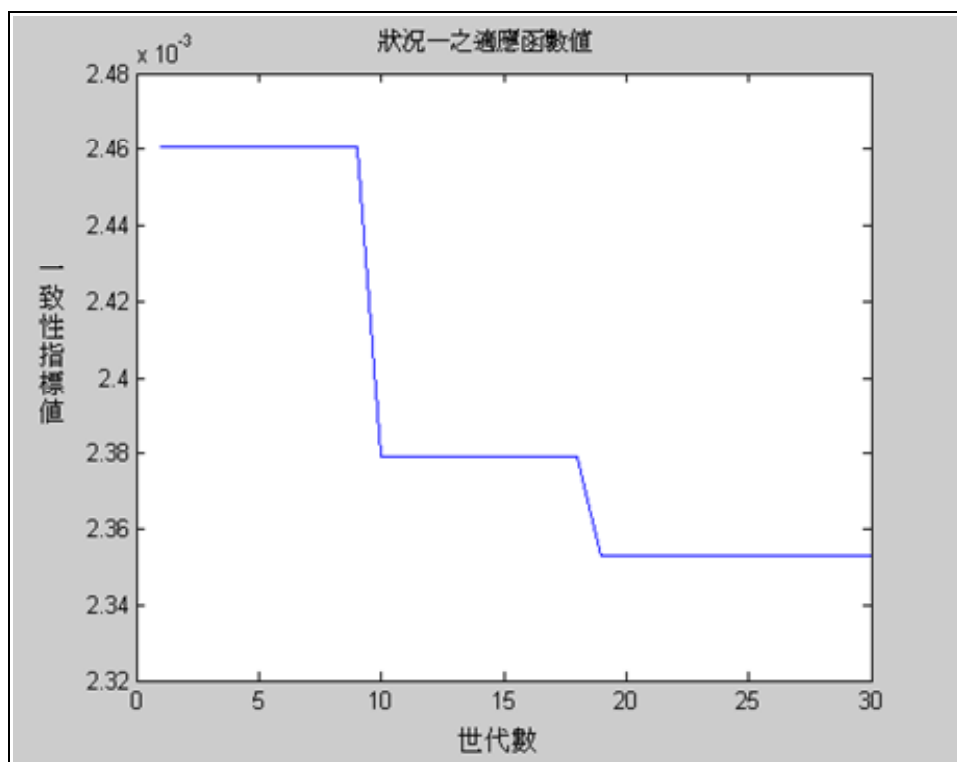


圖 4-12 狀況一適應值收斂圖

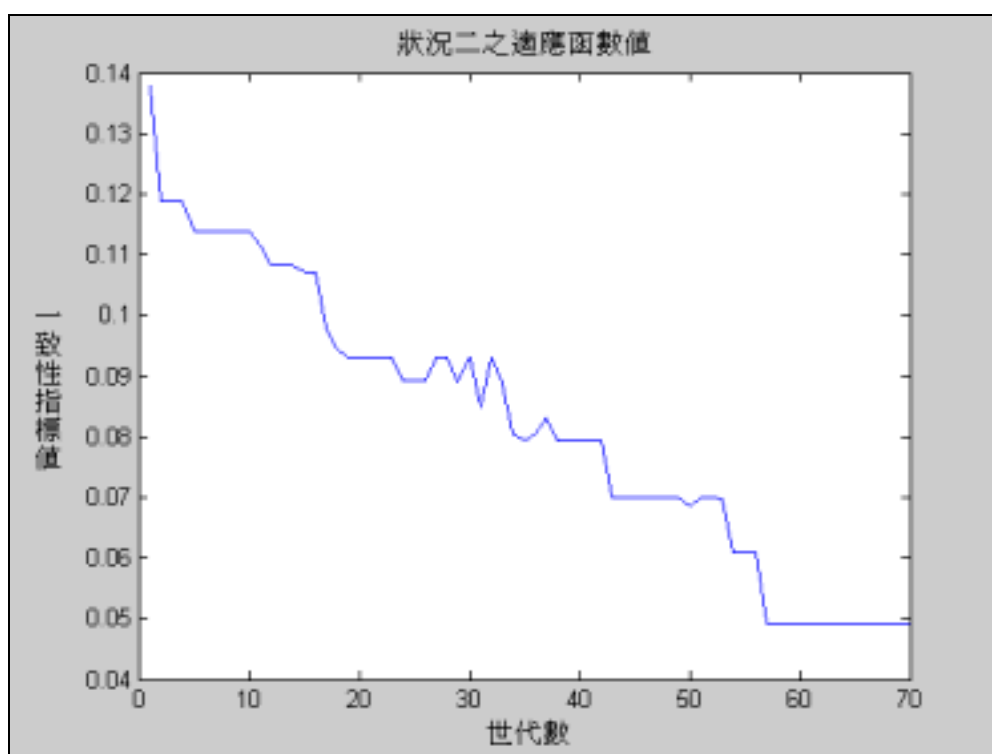


圖 4-13 狀況二適應值收斂圖

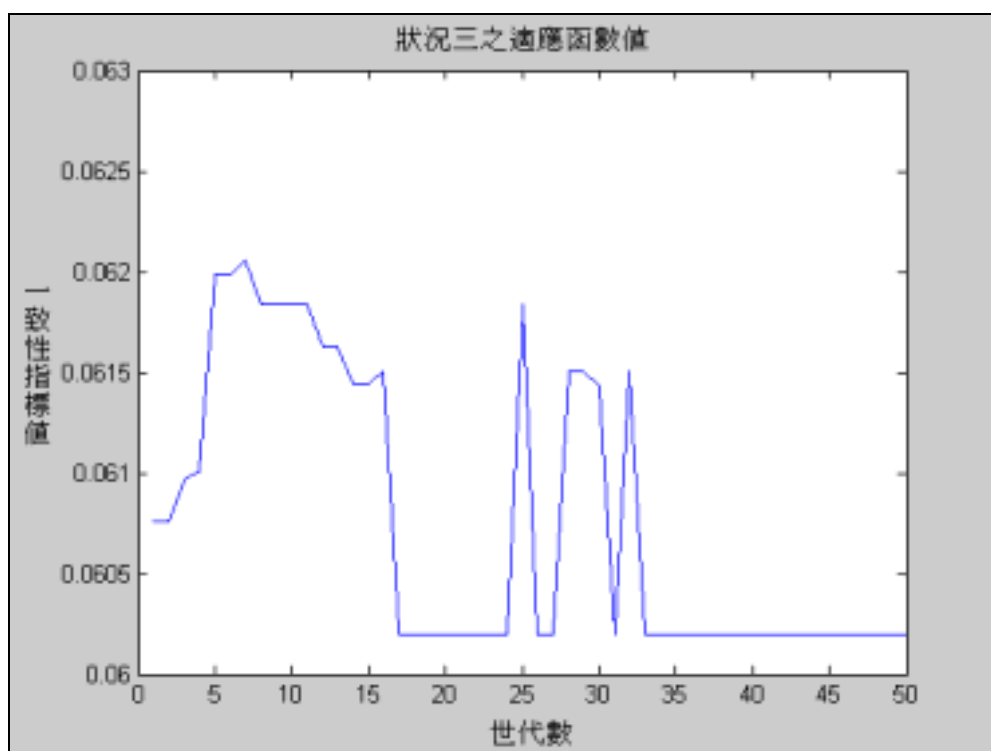


圖 4-14 狀況三適應值收斂圖

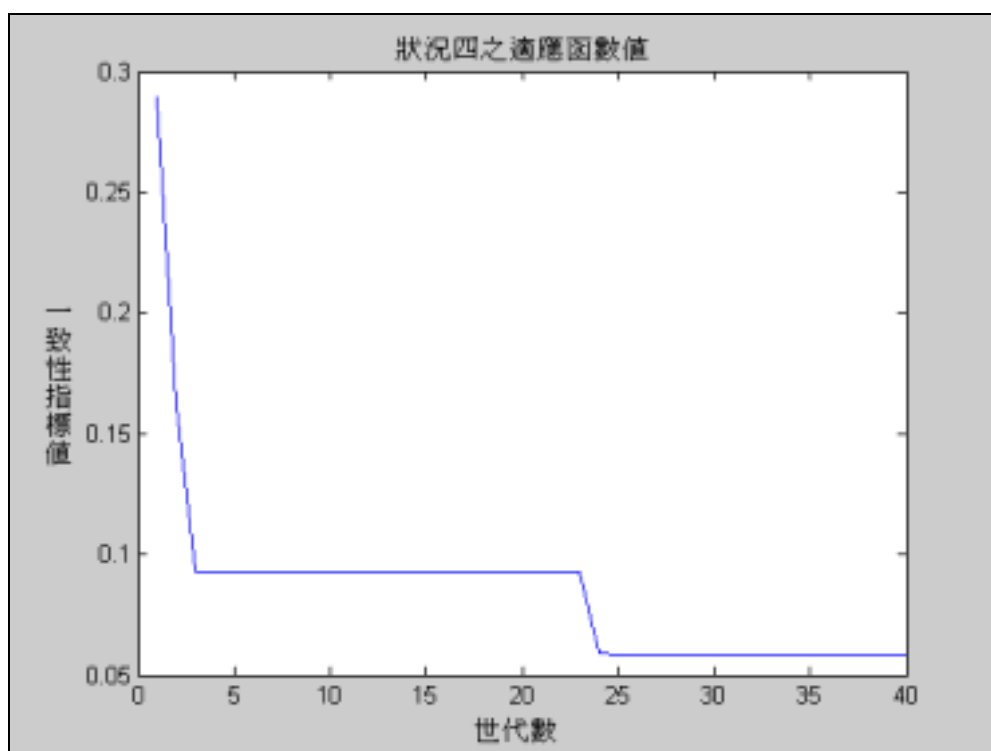


圖 4-15 狀況四適應值收斂圖

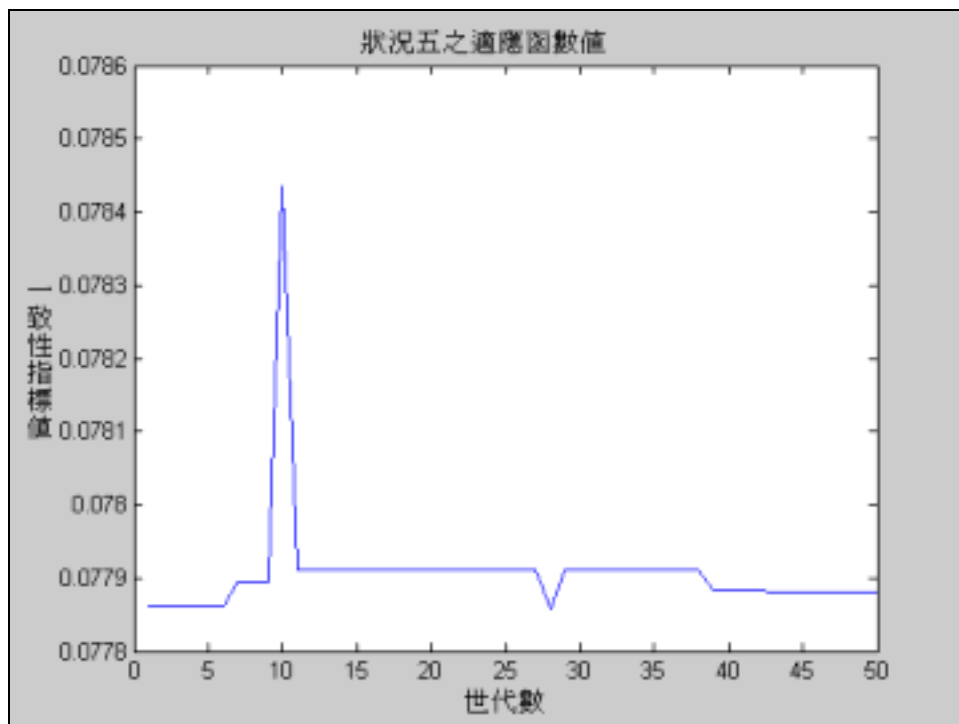


圖 4-16 狀況五適應值收斂圖

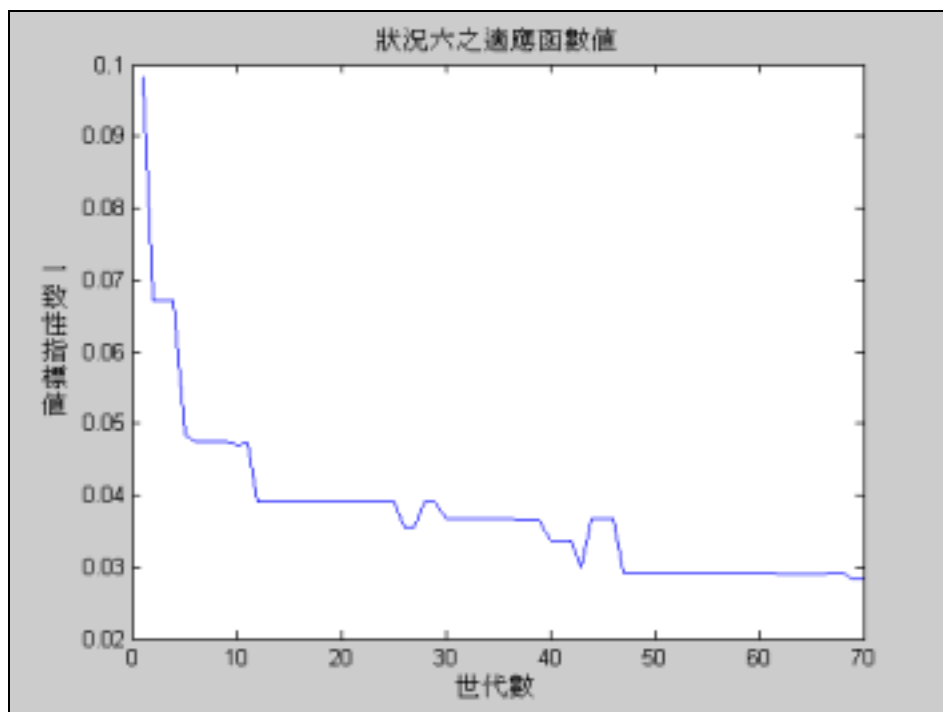


圖 4-17 狀況六適應值收斂圖

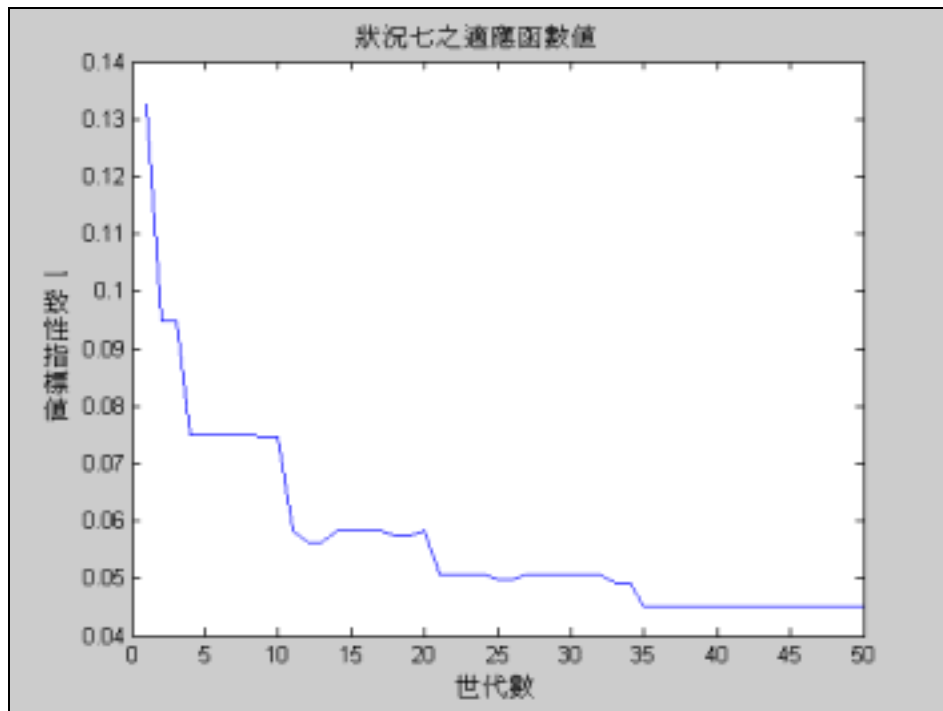


圖 4-18 狀況七適應值收斂圖

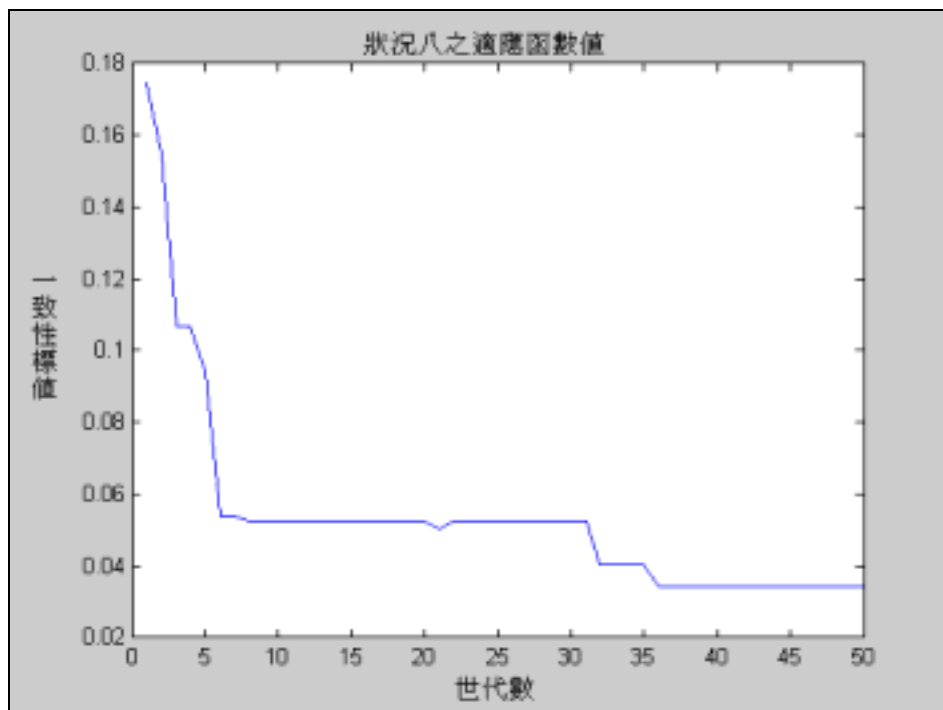


圖 4-19 狀況八適應值收斂圖

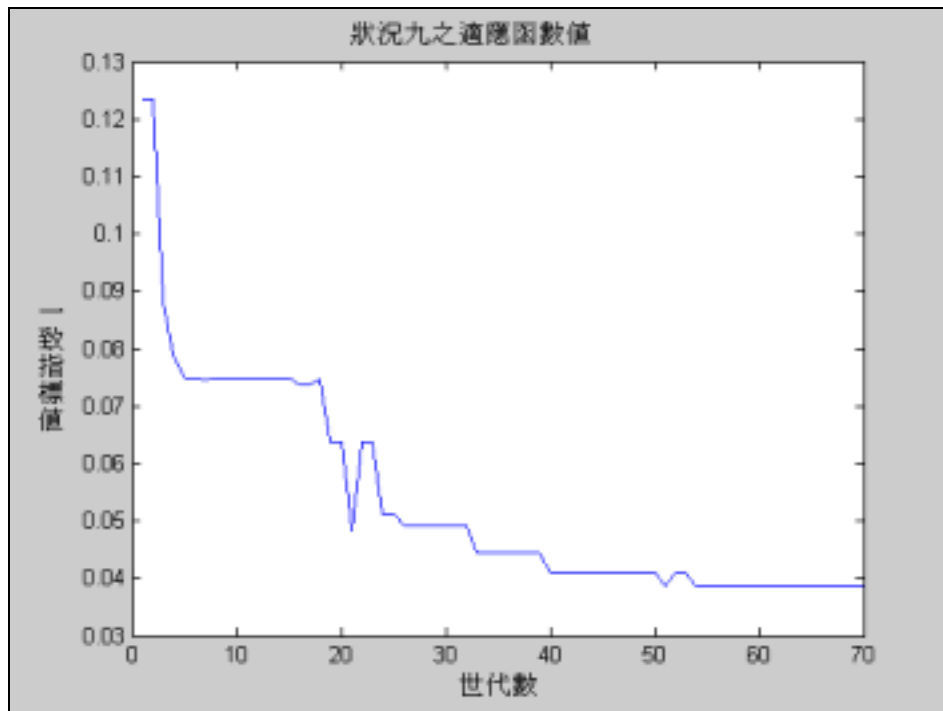


圖 4-20 狀況九適應值收斂圖

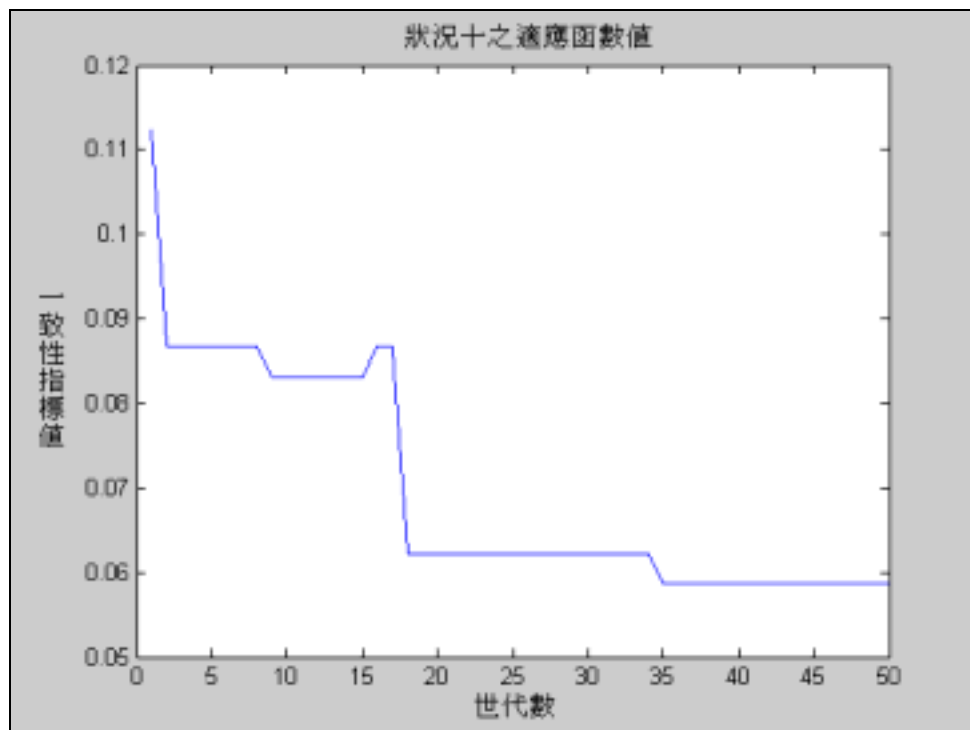


圖 4-21 狀況十適應值收斂圖

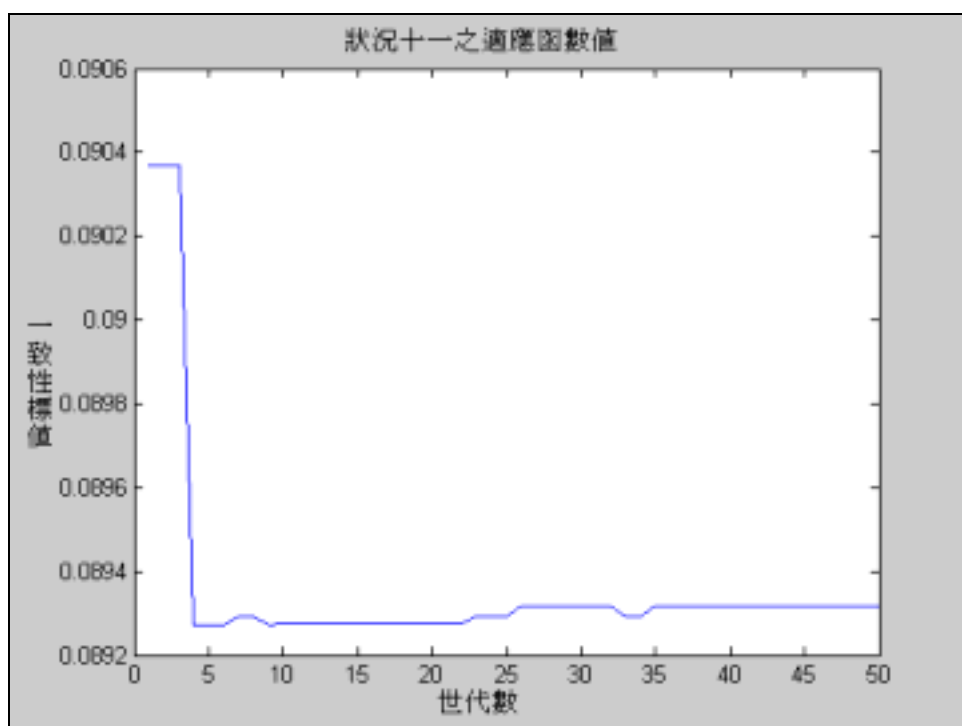


圖 4-22 狀況十一適應值收斂圖

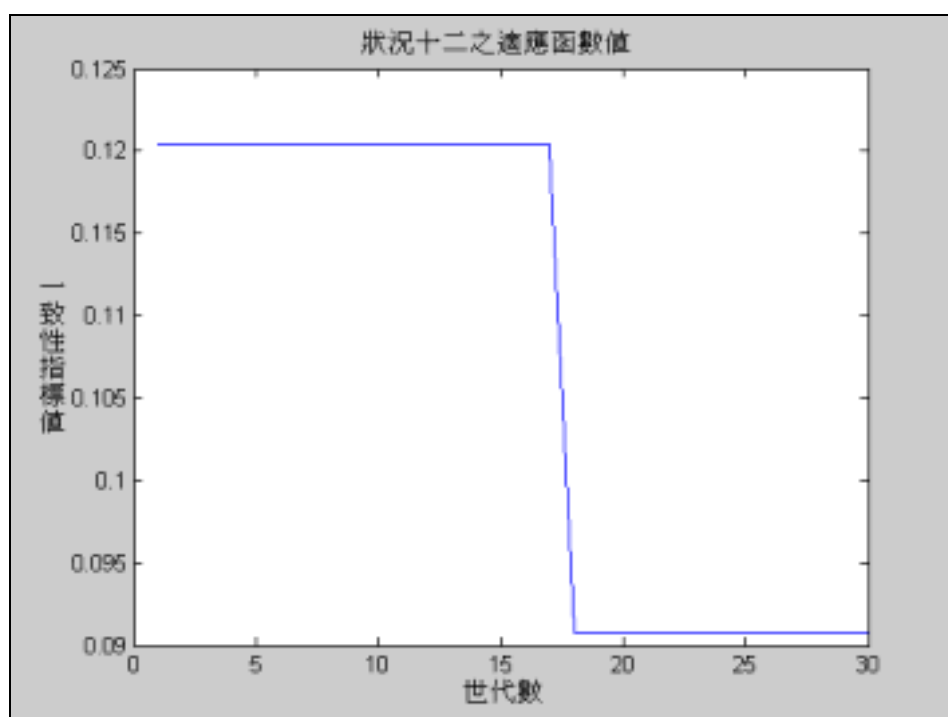


圖 4-23 狀況十二之適應值收斂圖

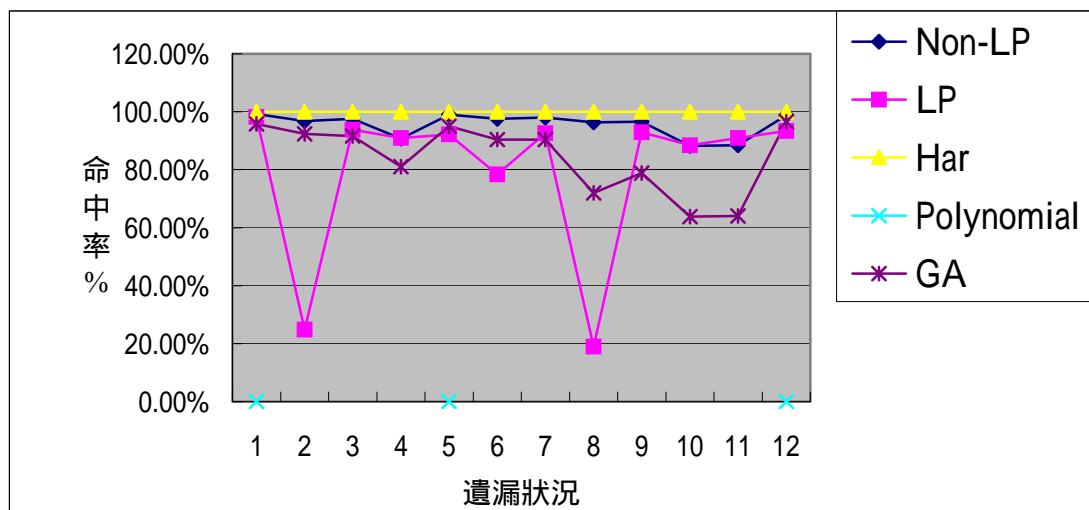


圖 4-24：十二種遺漏狀況各方法之估計誤差

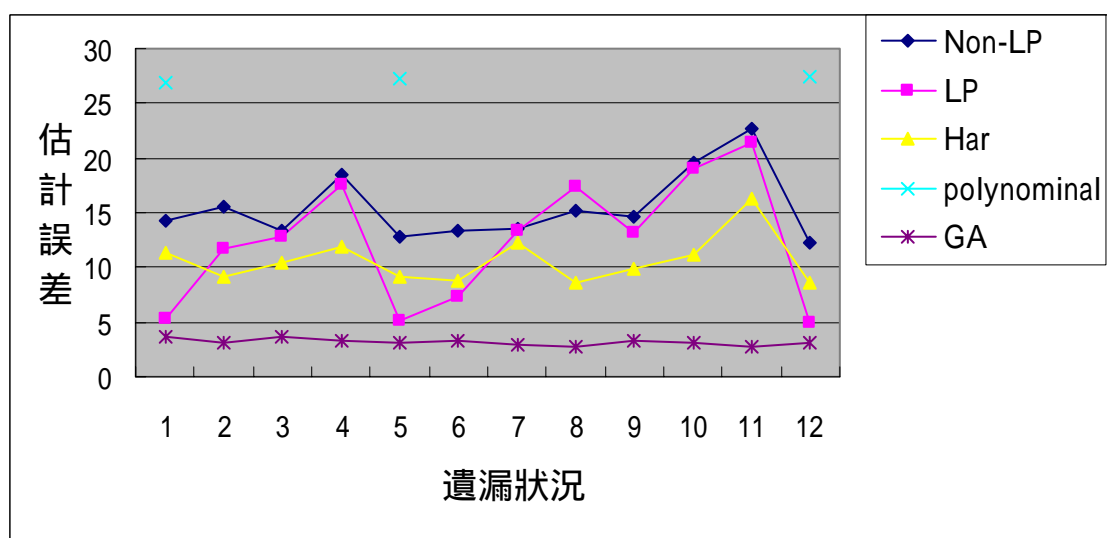


圖 4-25：十二種遺漏狀況各方法之命中率

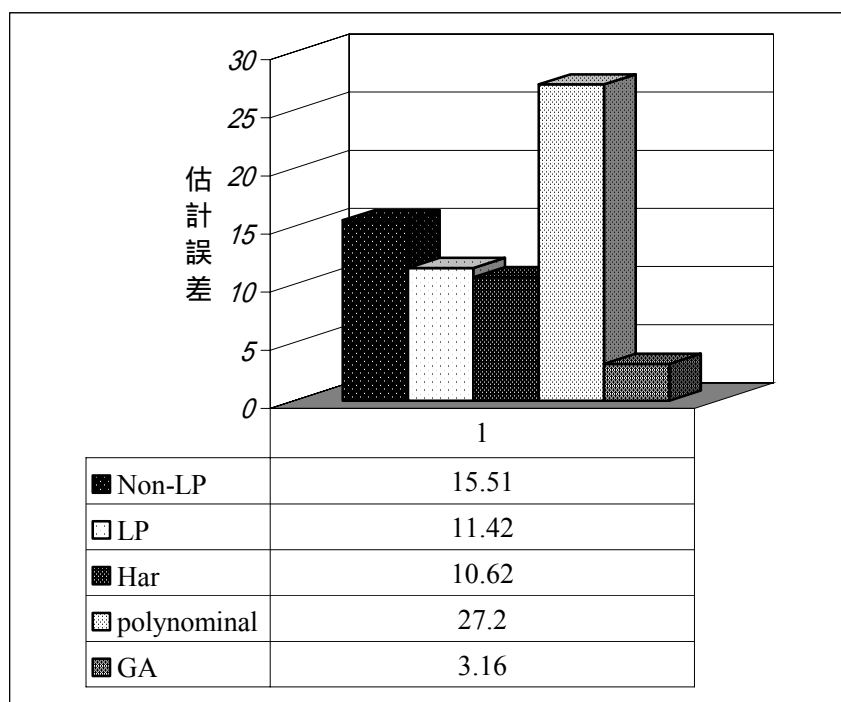


圖 4-26：十二種狀況之整體平均估計誤差

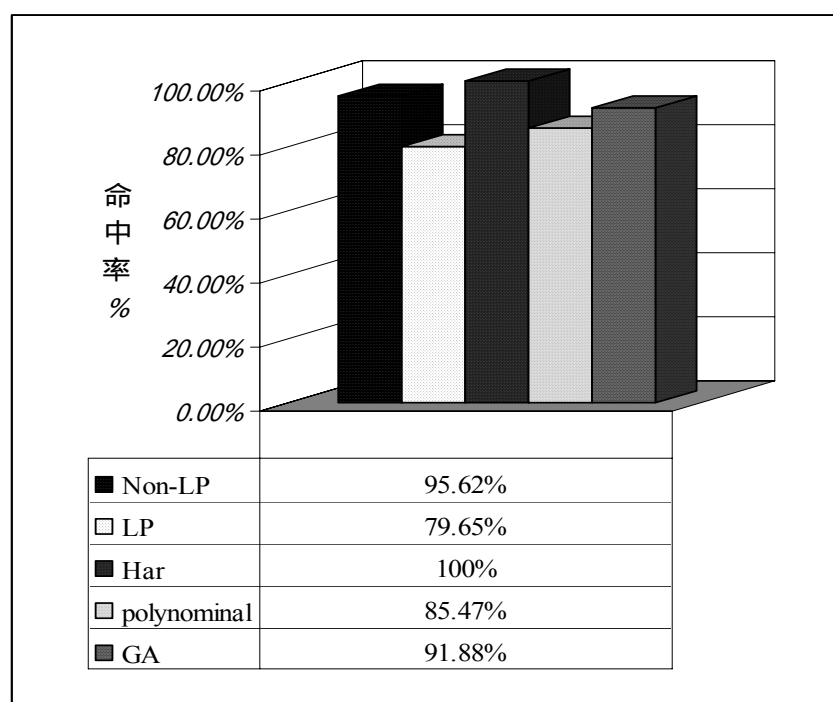


圖 4-27：十二種狀況之整體命中率

上述之 12 種遺漏狀況中，遺漏個數有一、二、三、四、六等五種遺漏個數，就整體估計誤差而言由圖 4-7 可清楚看出 GA 法的估計誤差是最小，其次為 Har 法，第三為

LP 法；命中率方面，以 Har 法最好，其次為 Non-LP 法，第三為 GA 法。而從圖 4-5 中，發現遺漏一個狀況的實驗模擬，GA 法估計誤差最小，LP 法為第二小的，但 LP 法在其他的遺漏狀況估計力較差。

4.4 結果與討論

指標一與二的討論：針對各方法對求解遺漏值的限制下，因此我們可從實驗模擬的結果看出，並不是每種方法都可做所有的實驗，討論在實驗一至三的遺漏狀況下，從上述實驗可看出 Polynomial 法的求解範圍最小，其次為 Non-LP 與 LP 法，而 Har 與 GA 都可模擬上述的所有實驗。綜合討論估計誤差方面，整體而言 Polynomial 法的估計誤差最大，其次為 Non-LP 與 LP 法，Har 法為第二小，GA 之估計誤差是明顯比其他各方法小。但在實驗模擬三中，求算單一之 x_{14} , x_{15} , x_{16} 等三狀況，LP 法是明顯比 Har 法好。在命中率方面，以 Har 法最好，三實驗所模擬的各種狀況都能夠達到百分之百的命中率，而 Non-LP 法平均都有九成五以上，而 LP 法的命中率變化很大，最差之命中率只有 19%，而 GA 法也是有小幅度的變化，變化較 LP 法小。

若針對隨機遺漏一個的狀況討論命中率，實驗一，Har 法最好，其次為 GA 法，Polynomial 法最低。若針對固定遺漏一個遺漏值的位置，實驗三，Har 法>Non-LP 法>Non-LP>Polynomial 法。

指標三的討論：三實驗結果，我們可明顯看出基因演算法的確需要較長的執行時間，且雖然都明顯比其他方法需花更多的時間，但最長時間都在三十分鐘左右（500 個樣本單位），所亦在實作上並不會造成太大的時間等後。

最後，綜合討論估計誤差、命中率與程式執行時間，基因演算法所需花費的時間雖較長，且 GA 法雖在命中率無法像 Har 法好，但在一般填回遺漏值的主要觀念下，仍著重在估計誤差越小越好，結果顯示在估計誤差與求解遺漏範圍，GA 法都是最好的，故此本研究以實驗模擬結果證實基因演算法在求解不完整成對比較矩陣之 AHP 問題上是有較好的績效。

五、結論與建議

5.1 結論

本研究蒐集整理分析相對於不完整資訊織成對比較矩陣的方法，作一完整性的比較與模擬，而我們嘗試提出一基因演算法來求出遺漏之比較值，且綜合數值模擬實驗一至三中可發現，GA 法的估計誤差都為最小，且在一般填回遺漏值的主要觀念為估計誤差越小越好，即修正的資訊偏誤都比其他方法小，因此 GA 法雖在命率無法像 Har 法好，但在求解遺漏值的範圍上，GA 法沒有任何的條件限制方法，故實際施行上，採用 GA 法在求解不完整成對比較矩陣之 AHP 問題上是較其他方法有較好的績效。總結為基因演算法在還原遺漏值之估計誤差方面是優於其他的方法，並且三實驗之模擬結果能提供解決層級分析法之問卷回收時所面臨的資料遺漏問題的參考。

5.2 建議

1. 基因在命中率方面無法達百分之百，也許是因為適應函數的設定考慮不夠，未來可再做調整。
2. 未來可採用神經網路來求算遺漏值。
3. 因本次研究是假設不完整成對比較矩陣原本是一致之矩陣，在未來可探討若不完整成對比較矩陣原本為不一致之矩陣，可去探討各方法錯誤還原的機率。

5.3 未來研究課題

以基因演算法來找出導致不一致（不符合遞移性）的成對比較元素。

參考文獻：

- Goldberg, D. E. (1989), "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison - Wesley, MA.
- Harker, P. L. (1987), "Alternative modes of Questioning In the Analytic Hierarchy Process ", Mathl. Modelling, Vol. 9, No. 3-5, pp.353-360.
- Holland, J. H. (1992), "Genetic Algorithms ", Sci. Am., pp.66-72.
- Jong, D. (1975), "Analysis of the Behavior of A Class of A Genetic Adaptive System ", Ph. D. Dissertation, The University of Michigan, Ann Arbor.
- Levary, R. R, and Wan, K. (1998), "A Simulation Approach for Handling Uncertainty in the Analysis Hierarchy Process", European Journal of Operational Research, Vol. 106, No. 1, pp. 116-122.
- Man, K. F., Tang, K. S. and Kwong, S. (1999), " Genetic Algorithms: Concepts and Designs ", Springer, London.
- M. Gen, and R. Cheng (2000), "Genetic algorithms and engineering optimization ", John Wiley & Sons.
- Michalewicz (1999), "Genetic Algorithms + Data Structures = Evolution Programs", Springs.
- Nishizawa, K. (1997), "A Method to Estimate Results of Unknown Comparisons In Binary AHP", Journal of the Operations Research Society of Japan, Vol. 40, No. 1, pp. 404-414.
- Obata, T. , Shiraishi, S., Daigo, M. and Nakajima, N. (1999), " Assessment for An Incomplete Comparison Matrix and Improvement Comparison: Computational Experiments", ISAHP.
- Rooij, A. J. F., Jain, L. C., and Johnson, R. P. (1996), "Neural Network Training Using Genetic Algorithms", Singapore: World Scientific.
- Saaty, T. L. (1980), " The Analytic Hierarchy Process ", McGraw-Hill, New York.
- Saaty, T. L. and Vargas L. G. (1987), "Uncertainty & Rank Order in the Analytic Hierarchy Process ", European Journal of Operational Research, Vol. 32, No.2, pp.107-117.
- Saaty, T. L. (1989), "Group Decision Making and the AHP", in: Golden, B. L. and Wasil, E.

- (eds.), *The Analytic Hierarchy Process: Applications and Studies*, Springer-verlag, New York, pp.59-67.
- Saaty, T. L. (1994), “Fundamentals of Decision Making and Priority Theory ”, RWS publication.
- Shiraishi, S. Obata, T. and Daigo, M. (1998), “Property of a Positive Reciprocal Matrix and Their Application to AHP”, *Journal of the Operations Research Society of Japan*, Vol. 41, No. 3, pp.404-414.
- Steven, J. L. (1998), “Linear Algebra with Application”, Prentice Hall, Inc. Simon & Schuster/A Viacom Company Upper Saddle River, New Jersey.
- Triantaphyllou, E., and Mann, S. H., (1994), “A Computational Evaluation of the AHP and the REvised AHP When the Eigenvalue Method Is Used under A Continuity Assumption”, *Computers and Industrial Engineering*, Vol. 26, pp. 609-618.
- Triantaphyllou, E. (1995), “Linear Programming Based Decomposition Approach In Evaluating Priorities from Pairwise Comparisons and Error Analysis”, *Journal of Optimization Theory and Applications*, Vol. 84, No. 1, pp. 207-234.
- Zahedi, F. (1986), “A Simulation Study of Estimation Methods in the Analytic Hierarchy Process ”, *Socio-Econ. Plann. Sci.*, Vol. 2, No. 6, pp.347-356.
- Haupt, L. Randy, and S. E. Haupt (1998), “Practical genetic algorithms ”, Wiley, New York,
- 蘇木春、張孝德 ,(民國 86 年 12 月), 「機器學習：類神經網路、模糊系統、以及基因演算法則」, 全華科技股份有限公司印行。
- 周鵬程 ,(民國 90 年 11 月), 「遺傳演算法原理與應用-活用 Matlab」, 全華科技股份有限公司印行。
- 鄧振源 曾國雄 ,(民國 78 年 6 月), 「層級分析法(AHP)的內涵特性與研究(上)(下)」, *中國統計學報* ,(上) 27 卷 6 期 , 頁次 13707-13724、(下) 27 卷 7 期 , 頁次 13767-13786。
- 詹德光 ,(民國 82 年 6 月), 「資訊不完整下層級分析問題之解析-以 0-1 分數規劃法求解」, 資訊管理研究所碩士論文 , 國立交通大學。
- 林傳生 ,(民國 90 年 10 月), 「Matlab 之使用與應用」, 儒林圖書公司印行。

吳嘉祥，(民國 84 年 9 月)，「初等線性代數」，台灣東華書局股份有限公司印行。

黃寶賢，(民國91年6月)，「應用物件導向技術及智慧型計算建構半導體製程之最佳化品質控制系統--以化學機械研磨為例」，工業工程學系碩士論文，私立大葉大學。

謝欣宏，(民國91年6月)，「台鐵司機員排班與輪班問題之研究 - 以基因演算法求解」，交通管理科學研究所碩士論文，國立成功大學。

附錄一：

(一) Harker 方法之理論基礎：

舉例說明：給定一不完整 3×3 矩陣 C ：

$$C = \begin{bmatrix} 1 & 2 & w_1 / w_3 \\ 1/2 & 1 & 2 \\ w_3 / w_1 & 1/2 & 1 \end{bmatrix}$$

定義一新矩陣 A ，為 $Aw = Cw$ ：

$$A = \begin{bmatrix} 2 & 2 & 0 \\ 1/2 & 1 & 2 \\ 0 & 1/2 & 1 \end{bmatrix}$$

理論 1：若 B 是一 n 階非負不可縮減矩陣，所以 $(I + B)^{n-1} > 0$ ；即 $A = (I + B)$ 是一原始之矩陣(primitive matrix)。

理論 2：對於原始的矩陣 A

$$\lim_{k \rightarrow \infty} \frac{A^k e}{e^T A^k e} = c w$$

e 為單位向量， c 為一常數， w 為矩陣 A 之主要向量 (principle eigenvector)。

簡 歷

姓名：羅英嫻

生日：68 年 5 月 9 日

求學過程：田中國小、田中國中、彰化女中、逢甲大學（交通工程與管理系）、交通大學（交通運輸研究所）

住址：彰化縣田中鎮中路里福安路 32 巷 69 號