

國立交通大學

光電工程研究所

碩士論文

以基因區域搜尋演算法  
設計繞射型濾波元件

Design of Diffractive Filter  
with Genetic Local Search Algorithm

研究生：張維中

指導教授：陸懋宏 教授

中華民國九十二年七月

以基因區域搜尋演算法設計繞設型濾波元件

Design of Diffractive Filter

with Genetic Local Search Algorithm

研究生：張維中

Student : Wei-Chung Chang

指導教授：陸懋宏

Advisor : Mao-Hong Lu

國立交通大學

電機資訊學院

光電工程研究所

碩士論文

A Thesis

Submitted to Institute of Electro-Optical Engineering  
College of Electronic Engineering and Computer Science  
National Chiao Tung University

In partial Fulfillment of the Requirements

For the Degree of

Master

In

Electro-Optical Engineering

July 2003

Hsinchu, Taiwan, Republic of China

中華民國 九十一年 七月

# 博碩士論文授權書

(國科會科學技術資料中心版本 92.6.17)

本授權書所授權之論文為本人在國立交通大學(學院)光電工程系所  
組 九十一 學年度第 二 學期取得 碩 士學位之論文。

論文名稱：以基因區域搜尋演算法設計繞射型濾波元件

同意     不同意 (政府機關重製上網)

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：\_\_\_\_\_，註明文號者請將全文資料延後半年再公開。

同意     不同意 (圖書館影印)

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鈎選，本人同意視同授權。

指導教授姓名：陸懋宏 教授

研究生簽名：張維中  
(親筆正楷)

學號：9024527

(務必填寫)

日期：民國 92 年 7 月 22 日

1. 本授權書(得自<http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html> 下載或至<http://www.stic.gov.tw> 首頁右下方下載)請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，請確認學校是否代收，若無者，請個別再寄論文一本至台北市(106-36)和平東路二段 106 號 1702 室 國科會科學技術資料中心 黃善平小姐。(本授權書諮詢電話：02-27377606 傳真：02-27377689)
3. 本授權書於民國 85 年 4 月 10 日送請內政部著作權委員會(現為經濟部智慧財產局)修正定稿，89.11.21 部份修正。
4. 本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

# 國立交通大學

## 論文口試委員會審定書

本校光電工程研究所碩士班\_\_\_\_\_張維中\_\_\_\_\_君

所提論文\_\_\_\_\_以基因區域搜尋演算法設計繞射型濾波元件\_\_\_\_\_

合於碩士資格標準、業經本委員會評審認可。

口試委員：\_\_\_\_\_張良知博士

\_\_\_\_\_葉茂勳博士

\_\_\_\_\_趙于飛教授

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

指導教授：\_\_\_\_\_陸懋宏教授

所長：\_\_\_\_\_謝文峰教授

中華民國 92 年 6 月 25 日

# 以基因區域搜尋演算法設計繞射型濾波元件

研究生：張維中

指導教授：陸懋宏

國立交通大學 光電工程研究所

## 摘要

以繞射理論為基礎為基礎發展出來的繞射光學元件，較傳統光學元件具有體積小、重量輕、適合量產等優點，且設計的自由度較大可以在單一元件上實現許多不同功能，使得繞射光學元件備受重視。

本論文結合基因演算法和共軛梯度法，發展出具有收斂快速且是全域最佳化的基因區域搜尋演算法。而繞射光學元件本身的特殊色散效果，利用此一演算法設計純相位型的繞射型濾波元件，使其達到窄波段濾波與聚焦的功能。

# **Design of Diffractive Filter with Genetic Local Search Algorithm**

Student: Wei-Chung Chang

Advisor: Prof. Mao-Hong Lu

Institute of Electro-Optical Engineering  
National Chiao Tung University

## **Abstract**

The diffractive optical elements (DOEs) based on diffraction theory have the advantages of small size, light weight, flexibility and suitable for mass production, which are hardly provided by traditional optical elements. Recently, DOEs become more and more attractive.

In this thesis, we combine the genetic algorithm (GAs) and conjugate gradient (CG) algorithm to develop a global and fast-convergence algorithm, called genetic local search algorithm (GLSA). The diffractive optical elements have uniquely dispersive property, so we use this algorithm to design a diffractive filtering device which has both narrow-band filtering and focusing functions.

## 誌謝

這本論文的完成，我必須要感謝人實在是很多，首先必須要感謝的是指導老師陸懋宏教授在這兩年的悉心指導，使我專業能力的學習有更深入的見解與廣闊的視野。感謝實驗室的學長姐們：林育正、林暉雄、姚柏宏、柯俊宏、施至柔、桑競存、高清芬、陳晏佐、徐得銘、鮑友南等學長姐在元件模擬、設計上提供我一些靈感；感謝同學們：王俊勛、王啟宗、李允民、朱淑君、吳銘鈞、洪文明、常皓源、曾郁潔、羅巧慧等在研究過程以及論文的製作上不斷的給予幫助以及經驗分享；實驗室的學弟妹們：呂怡瑱、陳建丞、陳慶勳、馮致華等的加入，帶給實驗室更多的歡樂與活力。

此外，必須要特別感謝父母親與家人一直給我打氣與支持，讓我能專心於研究無後顧之憂，能夠順利完成學業。最後，也向我身旁所有關心我的人，獻上由衷的感謝。

僅以此論文獻給所有曾經給予我幫助與鼓勵的人。

2003年7月於新竹交大

# 目錄

中文摘要	i
英文摘要	ii
致謝	iii
目錄	iv
圖目錄	vii
表目錄	x
第一章 緒論	1
1.1 引言	1
1.2 繞射光學元件的優點，以及與傳統光學元件的比較	2
1.3 繞射光學元件的設計方法	4
1.3.1 光線追跡法	4
1.3.2 最佳化演算法	5
1.4 關於本論文	6
第二章 基因區域搜尋法	7
2.1 最佳化演算法簡介	7
2.1.1 區域型最佳化演算法	7
2.1.2 全域型最佳化演算法	9



2.2 共軛梯度法	12
2.2.1 最速下降法	12
2.2.2 共軛梯度法	13
2.3 基因演算法	17
2.3.1 簡易基因演算法	19
2.3.2 輪廓理論	23
2.3.3 其他的運算方法	27
2.4 基因區域搜尋法	35
第三章 繞射光學元件的基本理論	38
3.1 基本繞射理論	39
3.2 菲涅耳波帶透鏡以及繞射光學匯聚元件的基本特性	43
3.2.1 基本公式的推導	43
3.2.2 繞射元件的規格與最小線寬之關係	47
3.3 濾波元件的設計概念	49
第四章 將基因區域搜尋演算法應用於繞射型濾波元件的設計	54
4.1 目標函數的設定與比較分析	54
4.2 改良型基因演算法	59
4.3 編碼以及兩種演算法交互迭代的方式	64
第五章 結論	69

5.1 模擬結果與討論	69
5.2 未來展望	78
附錄 A F-R 共軛梯度法	79
附錄 B 亂數產生器	85
參考文獻	89

## 圖目錄

圖 1-1 正透鏡對波光的相位進行調制，使得在成像面形成建設性干涉 .....	4
圖 2-1 模擬退火法流程圖.....	11
圖 2-2 最速下降法流程圖.....	13
圖 2-3 前進方向正交示意圖.....	14
圖 2-4 單變數函數最小化，在 $[X_a, X_b]$ 求一極小值.....	15
圖 2-5 F-R 共軛梯度法流程圖.....	17
圖 2-6 簡易基因演算法架構圖.....	19
圖 2-7 輪盤式選擇法示意圖.....	21
圖 2-8 單切點交配法示意圖.....	21
圖 2-9 反相法突變示意圖.....	22
圖 2-10 切點位置與定義長度關係示意圖.....	26
圖 2-11 基因演算法標準流程圖.....	29
圖 2-12 雙切點交配法示意圖.....	33
圖 2-13 交配遮罩與各種交配法的關係圖.....	33
圖 2-14 利用布林運算完成均勻交配法.....	34
圖 2-15 各種編碼方法所使用的突變運算.....	35
圖 2-16 GLSA 基本架構圖.....	37

圖 3-1	點光源照射到一平面屏	41
圖 3-2	菲涅耳波帶示意圖	44
圖 3-3	二進階繞射光學元件之相位函數與半徑平方的關係圖	46
圖 3-4	DOE 的最小線寬與光程示意圖	48
圖 3-5	濾波元件基本架構圖	49
圖 3-6	按照表 3-2 規格的 Fresnel 透鏡	52
圖 4-1	誤差函數示意圖	55
圖 4-2	分階前與分階後的 Fresnel 透鏡	59
圖 4-3	八種等效的結構	60
圖 4-4	整個族群中兩個較優秀的個體	62
圖 4-5	不同的交配方法隨時間收斂的曲線	64
圖 4-6	多個極大值集中在一起示意圖	67
圖 4-7	GLSA 流程圖	68
圖 5-1	使用目標函數 1，基因區域搜尋法計算的結果	70
圖 5-2	使用目標函數 1，所優化出來的表面輪廓圖	70
圖 5-3	使用目標函數 2，基因區域搜尋演算法的計算結果	71
圖 5-4	使用目標函數 2，所優化出來的表面輪廓圖	71
圖 5-5	光場譜線分佈圖	72
圖 5-6	使用目標函數 3，所優化出來的表面輪廓	76

圖 5-7 三種不同的最佳化的光場譜線分佈比較圖 ·····77

圖 B-1 以  $x=0.3$  代入  $2x^2-1$ ，不斷重複迭代的一組數列 ·····85

圖 B-2 以  $x=0.3001$  代入  $2x^2-1$ ，不斷重複迭代的一組數列 ·····86

圖 B-3 比較初始值  $x=0.3$  與  $0.3001$  的計算結果 ·····87

## 表目錄

表 1-1	傳統光學元件與繞射光學元件的比較	4
表 2-1	染色體與其所代表的值、適應度、挑選的機率	20
表 2-2	輪廓階數、長度與其字串的關係	24
表 2-3	7 位元之二元編碼與整數、浮點數的轉換	31
表 3-1	不同階數繞射光學元件的-1 級繞射效率	47
表 3-2	元件規格	51
表 3-3	元件對於各波長的繞射效率與訊噪比	52
表 4-1	各種不同的目標函數設定方式	58
表 5-1	高速個人電腦運算平台之規格	69
表 5-2	元件規格參數表	69
表 5-3	按照規格設計的 Fresnel 透鏡(優化前)以及經過基因區域搜尋演算法優化得到的濾波元件的比較表	72
表 5-4	目標函數 1，基因區域搜尋法與共軛梯度法、基因演算法的比較	73
表 5-5	目標函數 2，基因區域搜尋法與共軛梯度法、基因演算法的比較	73
表 5-6	以目標函數 3，針對 1517~1549nm 波段設計的元件，經基因區域搜尋演算法優化後與該規格下的 Fresnel 透鏡(優化前)之	

比較表 .....	76
表 B-1 本論文中 LCG 所使用的參數 .....	88

# 第 1 章 緒論

## 1.1 引言

近年來由於高科技的蓬勃發展、相關的製造技術的進步，帶動了 3C(Computer / Communications / Consumer electronics)產品不斷往輕薄短小發展，也使得微光學<sup>1</sup> (Micro-Optics)成為了一個熱門的新興研究領域。繞射光學元件 (Diffractive Optical Elements, DOEs)具有體積小、重量輕、結構緊密、成本低廉、可積體化等優點，並可以實現以往傳統折射式光學元件(Refractive Optical Elements, ROEs)無法實現的特殊功能，使得 DOE 成為微光學領域研究的重點之一。

DOE 的設計可由最佳化演算法計算而得，由於電腦科技的不斷進步，使得早年需要數星期甚至數個月的運算量，在今日僅需要數個小時到數天即可完成。因此，我們試著尋找更好的演算法，希望能得到更好的設計而不在于演算法會需要更大的計算量。最佳化演算法可視為是一種搜尋技術(搜尋最佳結果)。在早年，由於電腦運算速度的限制，所使用的演算法中每次的搜尋方向是由目前的搜尋位置所決定的，因此極有可能落入區域最佳解(Local Optimizations)。基因演算法<sup>2</sup> (Genetic Algorithm, GA)與模擬退火法(Simulated Annealing, SA)是一種全域的最佳化技術(Global Optimization Techniques)，



在此兩種演算法的計算的過程中，引入了隨機的概念，可以避免落入區域最佳解，但相較於搜索方向固定的演算法，此類的演算法需要更多更龐大的計算量。本文係以結合基因演算法全域最佳化的特性以及區域搜尋演算法快速搜尋的優點，用來設計 DOE 濾波元件，此種演算法稱為混合型基因演算法<sup>3</sup> (Hybrid Genetic Algorithm, Hybrid GA) 或稱為基因區域搜尋法<sup>4</sup> (Genetic Local Search Algorithm, GLSA)。

## 1.2 繞射光學元件的優點，以及與傳統光學元件的比較

由於繞射光學元件的可以靈活地對光波傳播波面進行調制，設計的自由度比傳統的光學元件大，因此可以實現許多傳統光學元件無法達成的特殊功能，以下是繞射光學元件的優點：

### 1. 具高繞射效率：

二進階(binary)光學元件是一種純相位型(Phase-only type)繞射光學元件，利用階梯狀相位分佈做近似連續的表面輪廓結構。一般利用  $N$  個光罩，可得到  $L=2^N$  個相位階數，其-1 級繞射效率  $\eta = \left[ \text{sinc}\left(\frac{1}{L}\right) \right]^2$ ，由此計算，當  $L=2、4、8、16$  時，分別有 40.5%、81%、94.9%和 98.6%的繞射效率。

### 2. 具獨特的色散性能：

因為二進階光學元件是一種色散元件，有別於傳統折射光學元件的色散特性，故可以在傳統光學系統中扮演消色差的角色。結合傳統

折射光學透鏡，構成複合式光學透鏡(Hybrid lens)以校正透鏡色差，達到同時校正球差與色差的功能。

### 3. 設計的自由度大：

在傳統光學系統的設計中，只能經由改變透鏡曲面曲率或使用不同光學材料來達到設計要求，而在繞射光學元件的設計中，我們則可以經由改變表面輪廓的結構與控制槽溝的深淺產生任意波面，大大地增加了設計的變量，也因此能設計出許多傳統光學元件所沒有之全新功能的光學元件。

### 4. 材料的選擇性大：

由於繞射光學元件的光學功能只與元件的表面設計輪廓有關，與材料的選擇無關，所以可用的材料範圍大。

### 5. 特殊的光學功能：

繞射光學元件可以達成一般傳統光學元件所不能實現的傳播波面，如環型面、錐形面、非球面等，或將光束在光軸上調制成具有長焦深、雙焦點、高橫向解析度等性能。

### 6. 配合塑膠射出成型技術，量產潛力大且成本低。

由於繞射光學元件所具有的獨特優點，使其在光纖通信、光學計算 (Optical computing)、大容量數據儲存、光學連接 (Optical interconnection) 等許多領域，已經有廣闊應用的前景。下表 1-1

是繞射光學元件與傳統光學元件的比較：

項目	傳統光學元件	繞射光學元件
特性	重量較重、體積大、量產較困難、製作成本高	體積小、重量輕、量產容易、製作成本低，可積體化、微型化、陣列化
原理	幾何光學、反射與折射原理	純量繞射理論、向量繞射理論
製作	機械研磨、精密加工、單件生產	光束干涉法、超大型積體電路製程技術、可量產複製
功能	單一功能的光學元件	多功能集一體的光學元件

表 1-1 傳統光學元件與繞射光學元件的比較

### 1.3 繞射光學元件的設計方法

繞射光學元件常用的設計方法大致有二：1. 光線追跡法(Ray tracing method)2. 最佳化演算法<sup>5</sup>。

#### 1.3.1 光線追跡法

純相位型繞射元件，是利用表面的輪廓結構來進行調制入射光波的相位，使得光波傳遞至成像面時，形成建設性或是破壞性的干涉。如下圖 1-1，是一個傳統折射光學元件，將一平面波整型成為一球面波，在焦點處形成建設性干涉。

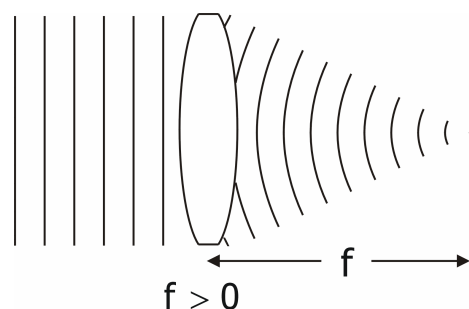


圖 1-1 正透鏡對波光的相位進行調制，使得在成像面形成建設性干涉

所謂的光線追跡法，即是利用幾何光學的方式分析光程，計算出元件每個位置所需要的相位補償(Phase compensation)，進而可計算出元件的輪廓結構。

對於功能簡單的元件，如單一焦點的繞射光學元件(Fresnel 透鏡)，光線追跡法是一種迅速、有效的設計方法。但對於功能較為特殊的元件，如本文所設計的濾波元件，就應該改採用最佳化演算法。

### 1.3.2 最佳化演算法

利用最佳化演算法設計繞射光學元件，大致可以用搜尋最符合要求的解或是誤差修正這兩種概念來解釋。設計時，先設定我們所要求的目標，然後透過評估函數或是誤差函數，利用最佳化演算法進行迭代的運算，使得最後的結果符合我們的要求。舉一個簡單的例子來說，假若我們需要利用最佳化演算法設計一個 Fresnel 透鏡，那可以將評估函數設定成繞射效率，搜尋一個解使得繞射效率有最大值，或者可以先在成像面，設定一個愛里盤(Airy disc)的光強度分佈函數(是一個理想聚焦的光強度分佈函數)，然後透過平方誤差和(Sum Square Error, SSE)函數來計算目前的解與目標函數相差的程度，透過最佳化演算法來使得誤差最小化，進而得到符合要求的解。

最佳化演算法大致可分成全域最佳化演算法與區域最佳化演算法。全域最佳化演算法可以得到全域最佳解，但卻需要更多、更龐大

的計算量，常用的全域最佳化演算法有：1. 模擬退火法 2. 隨機搜尋法 3. 基因演算法。而常用的區域最佳化演算法有：1. 梯度下降法 2. 實驗設計法(Design of Experiment，或稱為田口法)，而針對繞射光學元件設計發展出來的區域型最佳化演算法還有相位恢復迭代法以及 Y-G 演算法。本文將在稍後第二章的部分介紹這些演算法。

#### 1.4 關於本論文

本論文是結合基因演算法和共軛梯度法來設計繞射型濾波元件，針對繞射光學元件的特性修改基因演算法的一些運算機制，加快演算法收斂的速度。對於繞射型濾波元件的設計，本文也提出一些不同於以往的目標函數的設定方法，並討論這些設定方法的優缺點。

本文的第二章介紹了一般常用的演算法以及本論文所使用的基因區域搜尋演算法。第三章介紹 DOE 的特性以及基本的理論。第四、五章則將基因區域搜尋法套用在濾波元件的設計上，並討論模擬/設計的結果。

## 第 2 章 基因區域搜尋法

### 2.1 最佳化演算法簡介

一般而言，最佳化演算法可以被視為求某個函數的極大值(或極小值)，或者可將這些演算法當成搜尋技術，搜尋出較符合我們要求的答案。所以必須將光學元件上所希望達到的目的，適當轉換成函數的形式，再利用這些演算法求得此函數的極大值(或極小值)。舉例來說，如果將所求的函數定義為 SNR(Signal Noise Ratio，訊噪比)，則將是求此 SNR 的最大值；如果將函數定義為誤差函數，常用的函數形式為 SSE(Sum Square Error，平方誤差和)，則希望誤差越小越好，所以是求最小值。本章的第一節將介紹一些常使用的演算法，接著在第二與第三節將仔細說明本文所使用的共軛梯度法與基因演算法。

#### 2.1.1 區域型最佳化演算法

##### 1. 相位恢復迭代法(Iterative Fourier-Transform algorithms, IFT)：

此演算法在 1972 年由 Gerchberg 與 Saxton 首先提出，從輸出入平面測量兩個光場的照度數值為依據，用來恢復其輸出平面上光場相位分佈的一種演算法，通常稱為相位恢復迭代法，也稱為 G-S 演算法(G-S algorithms)。該演算法首先選擇初始相位分佈，然後進行富利葉轉換，再將輸出平面上測量得到的振幅分佈代替計算值，然後再進行

傅力葉轉換，最後於輸入平面上用實際測量值代替計算值，如此反覆迭代直到達到所預期的目標。在 G-S 演算法的計算中是利用快速傅力葉轉換(Fast Fourier Transform, FFT)，所花費的計算時間相當短，但也經常出現迭代停滯的現象。1989 年 Wyrowski 在 G-S 演算法上發展出一套稱為高效率量化演算法(efficient quantization algorithms)減少了迭代停滯的問題。

## 2. Y-G 演算法(Yan-Gu algorithms)：

此演算法是在 1980 年由中國科學院北京物理所的楊國楨(G. Z. Yang)和顧本源(B. Y. Gu)所提出，所以稱為 Y-G 演算法<sup>6</sup>。Y-G 法是改良自 G-S 演算法，所謂的 Y-G 法是在 G-S 法內引入了一非對角元素矩陣項，以致於整個系統的迭代有內外兩個循環，雖然費時較久，卻使其適合任意線性變換系統、能量有損耗系統、非么正變換系統，更具有抗噪訊、易推廣到多波長、多平面系統，以及對初值設定不敏感等優點。

## 3. 共軛梯度法(Conjugate Gradient Method)：

共軛梯度法是解決非線性規劃問題<sup>7</sup>常使用的一種演算法，也是目前處理無約束最佳化問題(unconstrained optimization problem)中最常被使用的方法之一。其基本原理是利用梯度下降的觀念，假設我們的問題為求  $F(\bar{X})$  的極小值，而  $\bar{X}^k$  為目前所在的位置， $\bar{\nabla}F(\bar{X}^k)$  為該位

置上的梯度(gradient)，則我們知道往梯度方向的反方向，是最快的下降方向，如此，令  $\bar{X}^{k+1} = \bar{X}^k - s\bar{\nabla}F(\bar{X}^k)$ ，取一個適當的  $s$  值(稱為步距)，經過多次迭代之後使得  $\bar{\nabla}F(\bar{X}^k)$  為 0，則滿足收斂條件。此演算法僅需要適當的定義出所希望最小化的函數，而此演算法與所求解的問題的形式無關，且具有快速收斂的特性，所以被大量運用在各種規劃問題上，包含合金材料的選擇、光學系統的最佳化、排程問題…等，本文將在本章的第二節詳細介紹此演算法。

#### 4. 實驗設計法(Design of Experiment)：

實驗設計法是 1960 年代初期由日本學者田口玄一(Taguchi)所提出，又稱為田口法。此一方法原本是應用於品質工程的領域中，但近年來已經陸續應用於許多非品質工程的領域上，例如利用田口法分析各個參數對於汽車操控性能的影響，藉由直交表(Orthogonal Array, OA)的分析，找出各個參數的影響性，減少實驗(分析)的次數，如此快速的求出各個參數的最佳化組合。

#### 2.1.2 全域型最佳化演算法

##### 1. 模擬退火法<sup>8</sup> (Simulated Annealing, SA)：

在冶金學上，如果將金屬緩慢冷卻(後續退火)，其金屬結構能自然的求出最小能量狀態；若將金屬急速冷卻，則無法獲得最小的狀態。而在長晶的過程也是如此，如果急速降溫，則無法獲得好的結晶，



但如果一系列的緩慢退火與淬火過程，則可獲得系統最低能量。科學家根據仿效、模擬此一物理現象，發展出模擬退火法。

此一演算法在剛開始時，先提升至某一個高溫狀態，此時溫度為  $T$ ， $\bar{X}$  表示目前的狀態， $E(\bar{X})$  是為能量函數，試著改變  $\bar{X}$ ，可以得到一個新的能量，如果新的狀態對應的能量比原先來得低，則接受此一狀態，反之，如果能量比原先來的高(此時能量差  $\Delta E$  為正值)，則以波茲曼(Boltzmann)分佈機率  $e^{-\frac{E^{k+1}-E^k}{kT}}$  來決定是否要接受改變成為此一狀態( $k$  為波茲曼常數， $T$  為溫度)，如此不斷迭代直到能量變化不大時，開始適當的降低一些溫度，並再作前述的迭代，最後當溫度降至很低時，會趨於收斂，當溫度小於所設定的結束溫度時，演算結束。

此一演算法與下坡搜尋法類似，但加入了隨機的概念，因此可以得到全域最佳解，而在溫度很低時  $T \sim 0$  時，改變成為能量較高狀態的機率非常低( $e^{-\frac{E^{k+1}-E^k}{kT}} \sim 0$ )，可確保其結果會收斂。此一演算法的流程圖如圖 2-1：

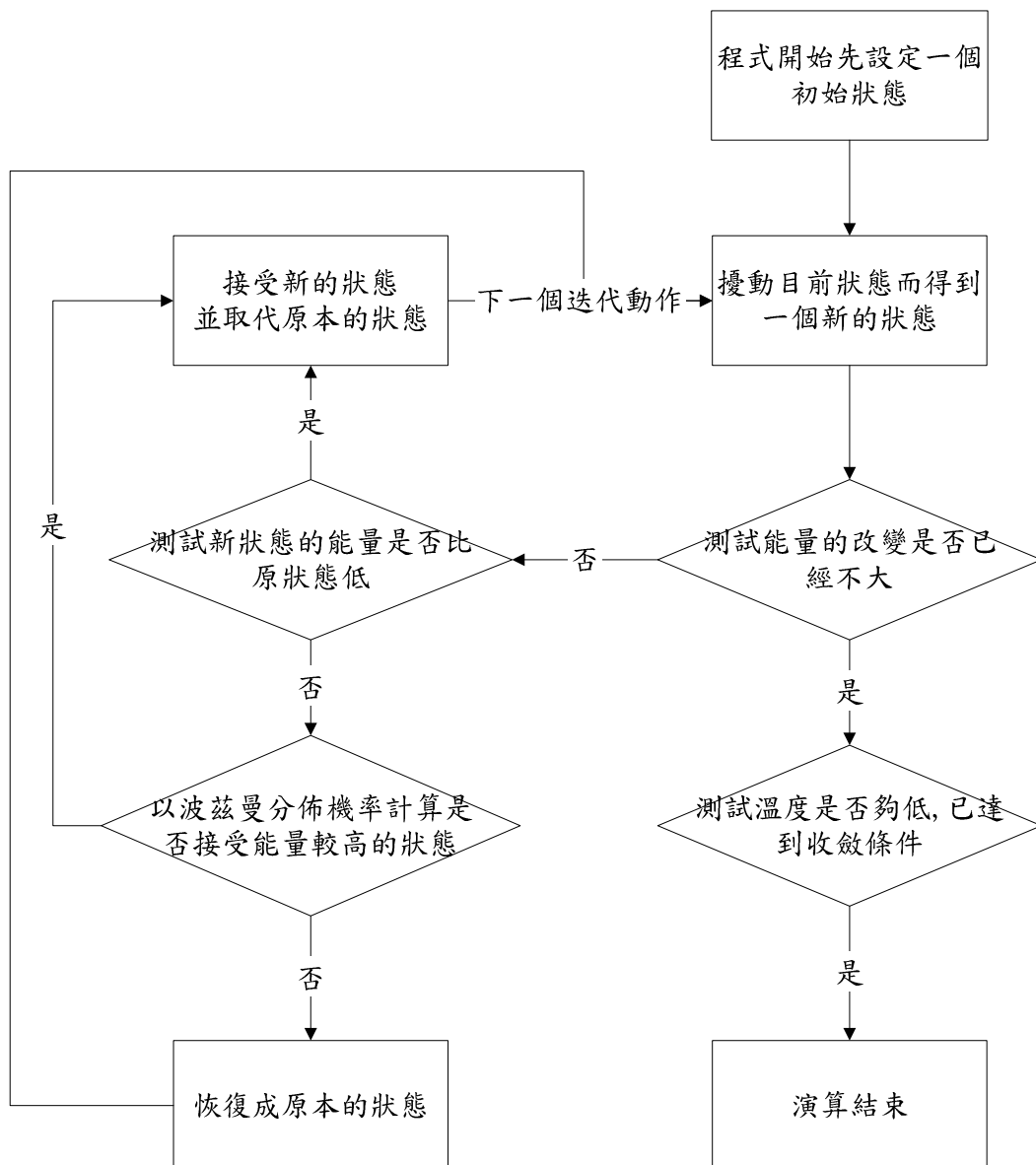


圖 2-1 模擬退火法流程圖

## 2. 基因演算法(Genetic Algorithm, GA)：

基因演算法<sup>2</sup>是在 1975 首先由密西根大學的 John Holland 教授所提出，是基於達爾文進化論所發展出的一套演算法，此演算法使用到了「挑選」(Selection)、「交配」(Crossover)、「突變」(Mutation)等機制，用來模擬自然界演化的現象，由於早年電腦速度較慢，使得基因演算法的發展受到了限制，近年來，由於電腦科技的進步，使得

基因演算法成為熱烈討論的主題，廣泛的應用於各種問題上，如人工智慧、決策問題、最佳化問題…等，在稍後的第三節裡頭，將有詳細的介紹。

## 2.2 共軛梯度法

### 2.2.1 最速下降法

最速下降法是無約束多變量函數最小化演算法，將所求的問題的解，以一個向量  $\bar{X} = (x_1, x_2, x_3, x_4, \dots, x_n)$  表示，而此演算法所要最小化的函數  $F = F(\bar{X})$  為一個純量函數，其原理如下：

$\bar{X}^k$  為目前所在的位置， $\bar{\nabla}F(\bar{X}^k)$  為該位置上的梯度 (gradient)，取  $\vec{d} = -\bar{\nabla}F(\bar{X})$  為移動的方向，由梯度的幾何上的概念可知，此移動方向是下降最快速的方向，因此令  $\bar{X}^{k+1} = \bar{X}^k - s\bar{\nabla}F(\bar{X}^k)$ ，取一個很小的  $s$  值 (稱為步距)，經過多次迭代之後使得  $\bar{\nabla}F(\bar{X}^k)$  為 0，則滿足收斂條件。此做法稱為最速下降法。其演算法的流程圖如圖 2-2：

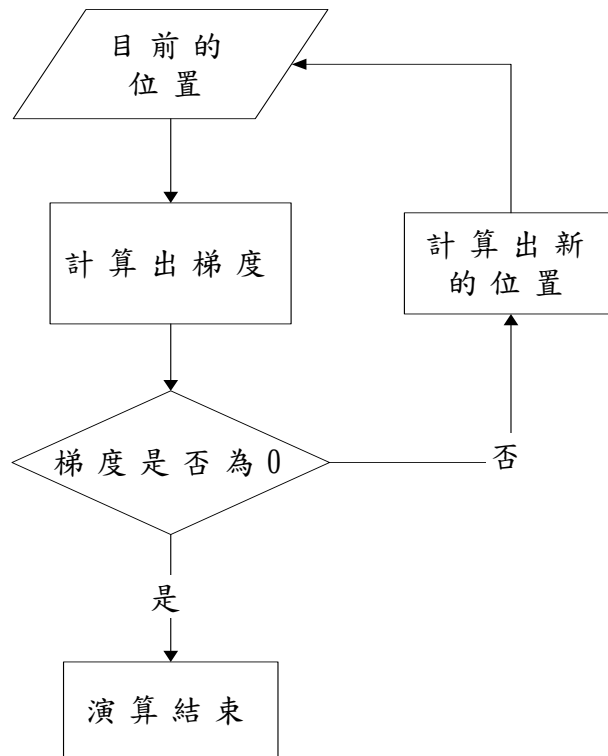


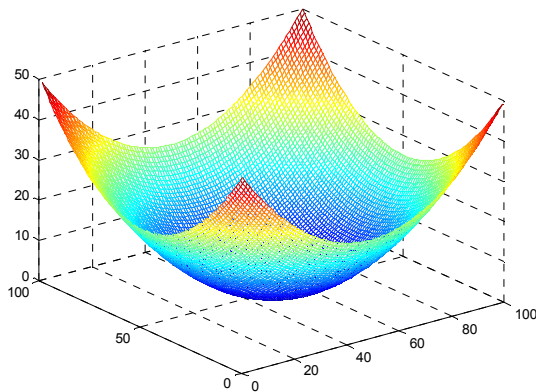
圖 2-2 最速下降法流程圖

### 2.2.2 共軛梯度法

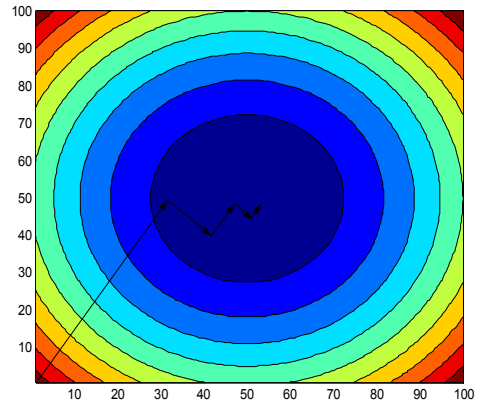
共軛梯度法<sup>9</sup>(Conjugate gradient method)與最速下降法類似，在最速下降法中，每次移動的步距  $s$  太小，導致收斂的速度過慢。因此的選擇步距時，希望使得  $F(\bar{X}^{k+1}) = F(\bar{X}^k - s\bar{\nabla}F(\bar{X}^k))$  最小。故考慮取一  $s$  值使得  $F$  有極小值，因此對  $s$  取導數為 0，可得

$$\frac{dF(\bar{X}^k - s\bar{\nabla}F(\bar{X}^k))}{ds} = -[\bar{\nabla}F(\bar{X}^{k+1})] \cdot \bar{\nabla}F(\bar{X}^k) = 0 \dots\dots\dots(2.1)$$

由此式可知， $-\bar{\nabla}F(\bar{X}^k)$  為第  $k$  次迭代的前進方向，而  $-\bar{\nabla}F(\bar{X}^{k+1})$  為下一次的前進方向，兩次前進方向內積為 0，代表著這逐次搜尋的方向正交，如圖 2-3(b)所示：



(a)



(b)

圖 2-3 前進方向正交示意圖

圖 2-3(a)表示一個具有兩個變量(2D)的純量函數圖形(方程式為  $z = (x-50)^2 + (y-50)^2$ )，其極值發生在  $x=50, y=50$  處，圖 2-3(b)是表示搜尋方向為正交的示意圖，由圖可知，此種搜尋方法並不是最好的。

共軛梯度法每次搜尋的方向取前次搜尋方向與目前梯度方向的組合得到逼近最佳解的新方向，共軛梯度法的基本公式為

$$\vec{d}^{k+1} = -\vec{g}^{k+1} + \beta \vec{d}^k \dots\dots\dots(2.2)$$

其中， $\vec{d}^{k+1}$ 表示第  $k+1$  次迭代所要搜尋的方向， $\vec{g}^{k+1}$ 表示第  $k$  次迭代過程中，該位置上的梯度， $\vec{d}^k$ 表示第  $k$  次迭代的搜尋方向，而  $\beta$ 表示一個比例因子。在共軛梯度法中，逐次搜尋的方向必須要是共軛的(見附錄 A)，因此對於對於一個指定的矩陣  $A$ ，必須滿足  $\vec{d}^{k+1} \cdot A \cdot \vec{d}^k = 0$ ，故可求得

$$\beta = \frac{|\bar{\nabla}F^{k+1}|^2}{|\bar{\nabla}F^k|^2} \dots\dots\dots(2.3)$$

而移動的方法就變成  $\bar{X}^{k+1} = \bar{X}^k + s\bar{d}^k$ ，當搜尋方向決定了之後，接著就必須要決定移動的步距  $s$ ，使得  $F(\bar{X} + s\bar{d})$  最小。此時所要求的問題就變成了單變數最小化的問題，在此使用黃金分割比例搜尋法，如下圖所示：

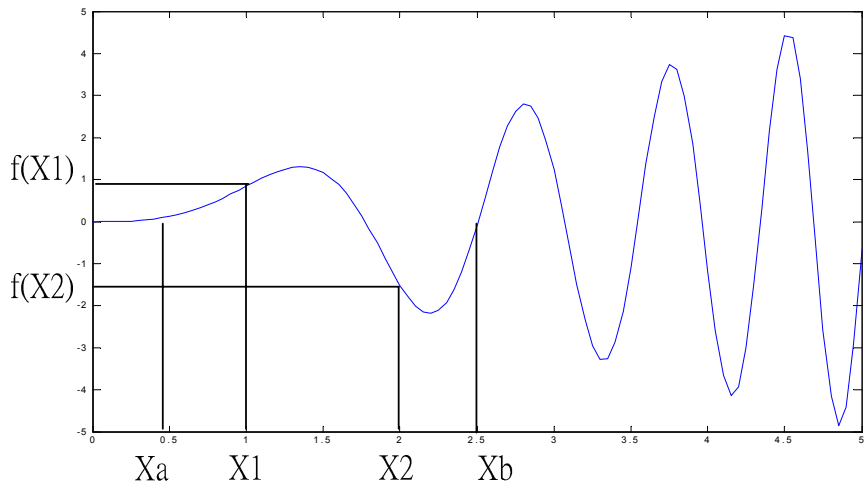


圖 2-4 單變數函數最小化，在  $[X_a, X_b]$  求一極小值

在圖 2-4 中， $[X_a, X_b]$  範圍內有一個極小值，任意選擇兩點， $x_1$  和  $x_2$ ，三等分此區間。假設  $x_a < x_1 < x_2 < x_b$ ，

若  $f(x_1) < f(x_2)$  則最小值需落於  $[X_a, X_2]$  內

若  $f(x_1) > f(x_2)$  則最小值需落於  $[X_1, X_b]$  內

任一最小值座落的範圍需小於區間  $[X_a, X_b]$ 。區間逐次縮減的過程可以連續重覆直到發現最小值座落之可接受的小區間。至於如何取  $x_1$  和  $x_2$ ，數學家提供一種黃金分割比例，如下：

$$g = \frac{1}{2}(-1 + \sqrt{5}) \approx 0.61803 \cdots \cdots (2.4)$$

$$x_1 = x_a + (x_b - x_a)(1 - g) \cdots \cdots (2.5)$$

$$x_2 = x_a + rg \cdots \cdots (2.6)$$

黃金分割比例與有名的 Fibonacci 數列有關，這個級數是 1, 1, 2, 3, 5, 8, 13, …，由下式產生

$$N_{k+1} = N_k + N_{k-1}, N_1 = N_2 = 1 \cdots \cdots (2.7)$$

$N_k$  是數列的第 k 項，當 k 趨近於無窮大時， $N_k / N_{k+1}$  趨近於黃金分割比例。

給定最大移動步距  $s_{\max}$ ，利用黃金分割比例搜尋法在  $[0, s_{\max}]$  範圍內求得移動步距 s 之後，可以利用 (2.2) 式求得新的位置，如此經過多次迭代之後使得為梯度的模  $norm(\bar{\nabla}F(\bar{X})) = |\bar{\nabla}F(\bar{X})|$  變為很小的值，滿足收斂條件。此做法是由 Fletcher 及 Reeves 所提出，稱為 F-R 共軛梯度法 (Fletcher-Reeves Conjugate Gradient Method)。其演算法流程如圖 2-5：

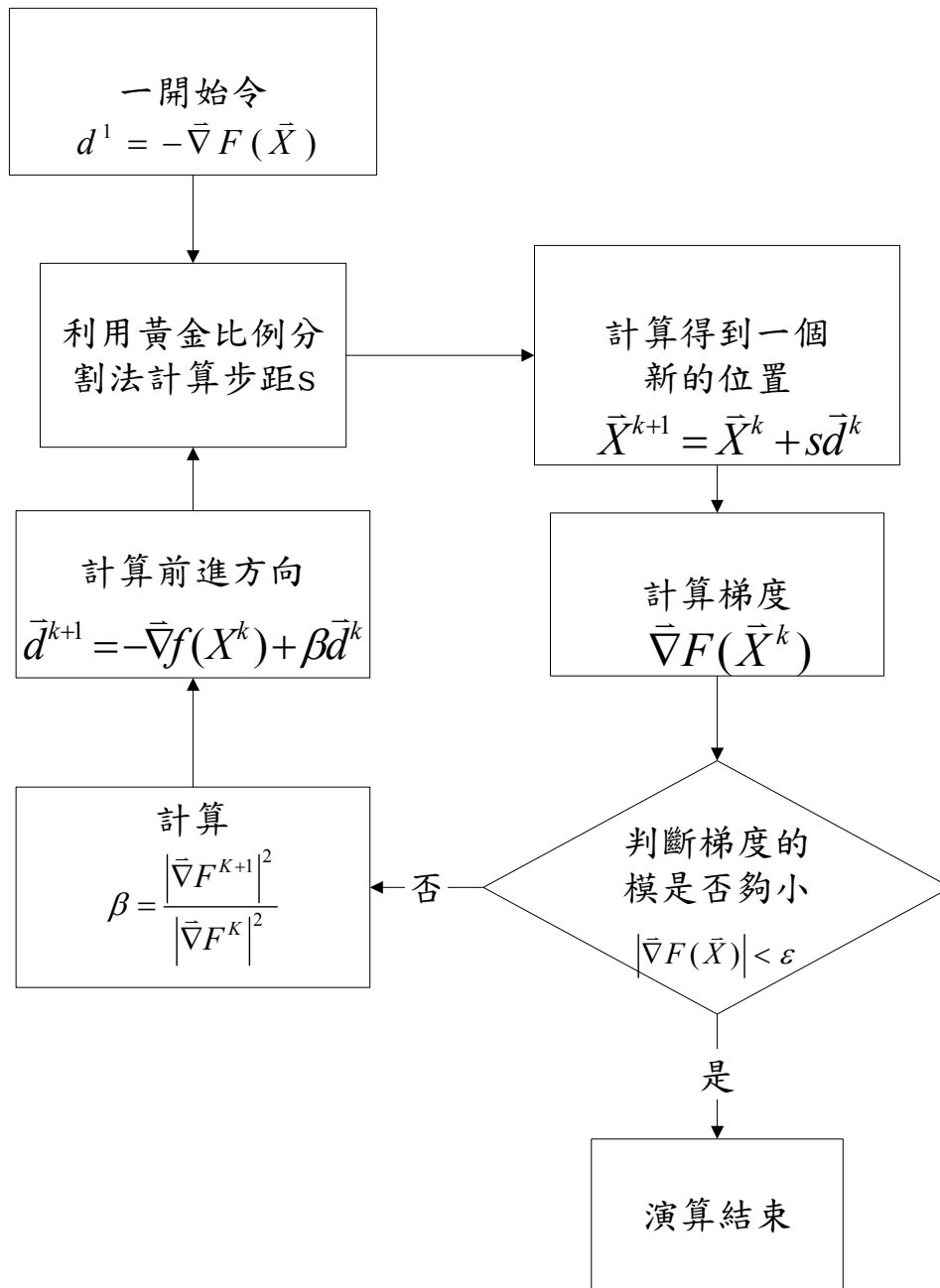


圖 2-5 F-R 共軛梯度法流程圖

### 2.3 基因演算法

基因演算法是近年來熱門研究的主題。將問題的解適當的編碼之後，根據達爾文的「進化論」(Evolutionism)，透過挑選、交配、突變等運算機制，搜尋出最佳解。本節的第一子小節首先介紹簡易基因



演算法(Simple Genetic Algorithm, SGA)，第二子小節介紹基因演算法的基本理論—輪廓理論，第三子小節詳細介紹基因演算法的運算機制。以下列出基因演算法所使用到的關鍵術語：

1. 族群 (Population)：一個族群中有許多獨立的個體 (Individual)，每一個獨立的個體就代表一個解。
2. 染色體 (Chromosome)：與個體同意，一個染色體是由一個或多個基因 (Gene) 構成，又稱為字串 (String)。
3. 基因：代表運算的基本單位，對於二元編碼法來說，一個位元 (bit) 就代表一個基因，染色體的長度 (Chromosome length) 就是指該染色體具有的基因數目。
4. 適應度函數 (Fitness function)：用來評估一個染色體的適應度，給定一個數值 (Fitness value) 大小代表該染色體的好壞，該值必須要是一個正數 (Positive value)。
5. 選擇運算子 (Selection operator)：挑選出適應度較大的染色體，以進行交配的動作。
6. 交配運算子 (Crossover operator)：將兩個染色體交換彼此的基因，產生新的染色體。
7. 突變運算子 (Mutation operator)：以隨機的方式，針對基因進行改變的動作，稱為突變。

### 2.3.1 簡易基因演算法

簡易基因演算法(Simple Genetic Algorithm, SGA)使用：1. 二元化編碼的染色體(Binary-coded chromosome)2. 單切點交配法(Single cut-point crossover) 3. 輪盤式挑選法(Roulette wheel selection)4. 使用標準二元突變運算子(Standard binary mutation operator, 意指「反相法」, bit inversion method)。以下是基本的架構圖：

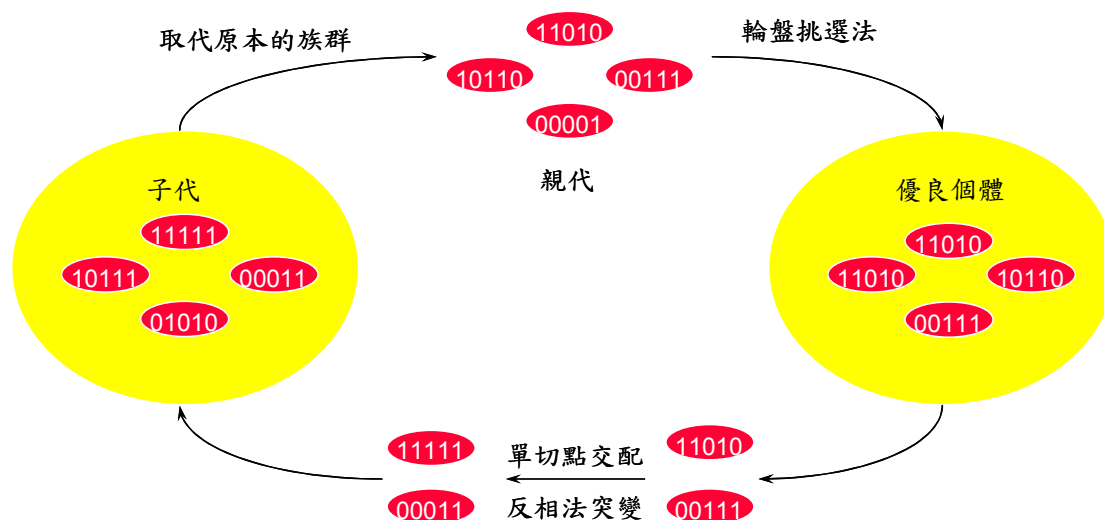


圖 2-6 簡易基因演算法架構圖

以下用一個簡單的例子說明簡易基因演算法迭代一次的過程，考慮有族群內有 4 個獨立的個體(染色體)，每個染色體具有 5 個基因，而每個染色體的適應度以及被挑選的機率如表 2-1 所示：

編號	染色體 $c_i$	所代表的值 $x_i$	適應度 $f(x_i)$	被挑選到的機率 $\frac{f(x_i)}{\sum_i f(x_i)} \times 100\%$
1	11010	26	676	55.8678%
2	00111	7	49	4.0496%
3	00001	1	1	0.0826%
4	10110	22	484	40%
總和			1210	100%

表 2-1 染色體與其所代表的值、適應度、挑選的機率

在簡易基因演算法裡，所使用的是輪盤挑選法，每個染色體被挑選出來交配的機率正比於該染色體的適應度值，而整個族群的適應度總和以  $F_{Total}$  表示，可以寫成以下的數學形式：

$$F_{total} = \sum_{i=1}^{N_{pop}} f(x_i) \dots\dots\dots (2.8)$$

$$P_i = \frac{f(x_i)}{F_{total}} \dots\dots\dots (2.9)$$

$P_i$  表示第  $i$  個染色體被挑選的機率， $N_{pop}$  表示族群中染色體的數目。圖 2-6 是輪盤式選擇法的示意圖，假想有一個輪盤，讓各個染色體所佔有的面積依適應度大小分配於盤面上，旋轉這個輪盤，然後再射出一個飛鏢，讓射中位置上的染色體為選中繁衍的進化者。

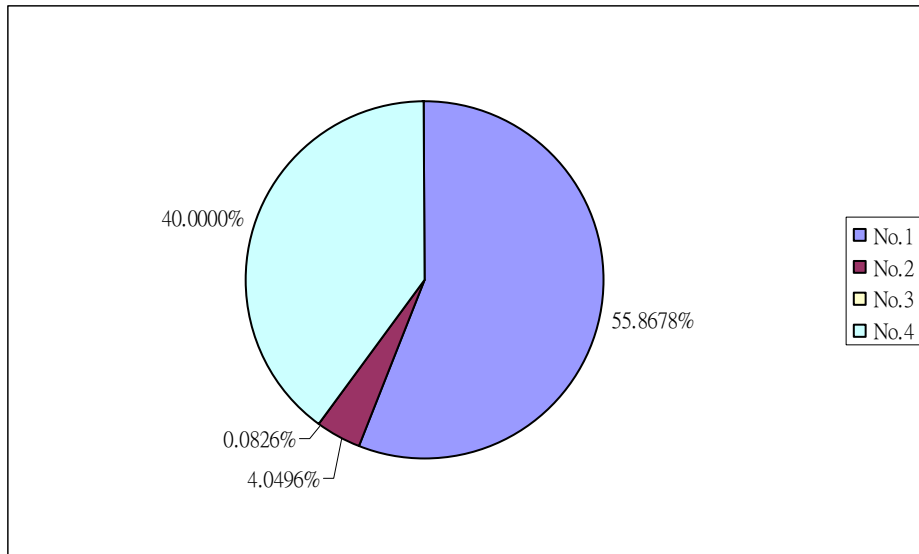


圖 2-7 輪盤式選擇法示意圖

在挑選出優良個體之後，接著進行交配的動作，假設挑選出來的染色體為 11010 與 00111，所謂的單切點交配法，便是將兩個長度為  $l$  的染色體，在  $1 \sim (l-1)$  位置的基因上，隨機找一個切割點，交換彼此的基因，圖 2-8 為單切點交配示意圖(假設切點為 2)。

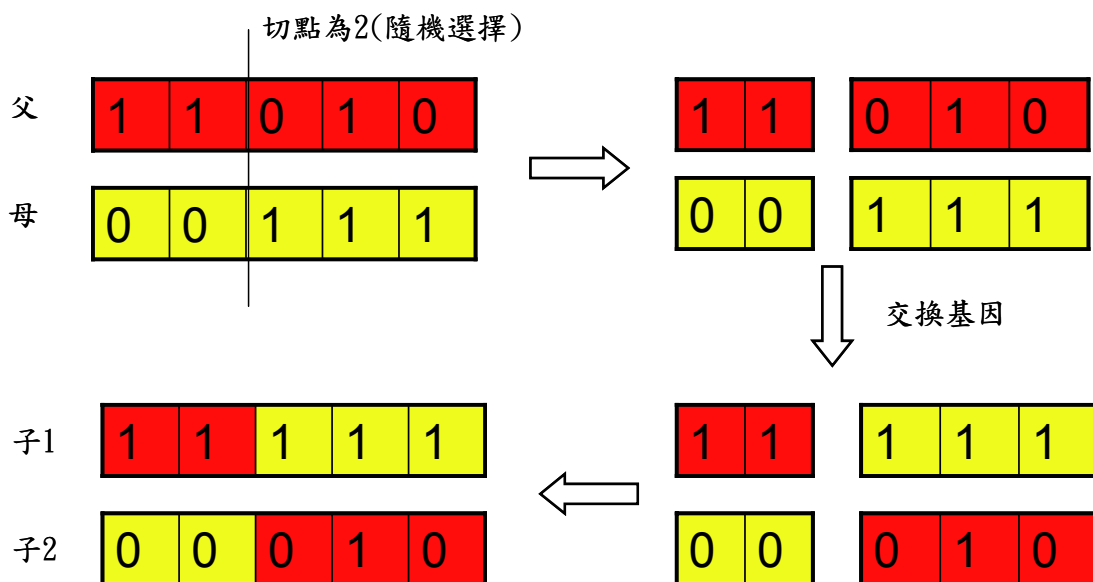


圖 2-8 單切點交配法示意圖

交配完成之後，下一個步驟就是突變，在簡易基因演算法中，每

個基因(位元)，突變的機率是相等的(突變的機率必須是一個很小的值，一般而言將會設定在 0.01~0.1 之間)，假設突變的機率是  $p_m$ ，計算的方式就是檢測每一個位元，利用隨機抽取一個 0~1 之間的隨機亂數(見附錄 B)，比較這個亂數是否小於  $p_m$ ，如果小於  $p_m$ ，將該位元反轉(bit inversion)，意即是將 1 變為 0，而將 0 變為 1，也就是布林(Boolean)代數中的 Not operator，如圖 2-9 所示，此作法稱為反相法。

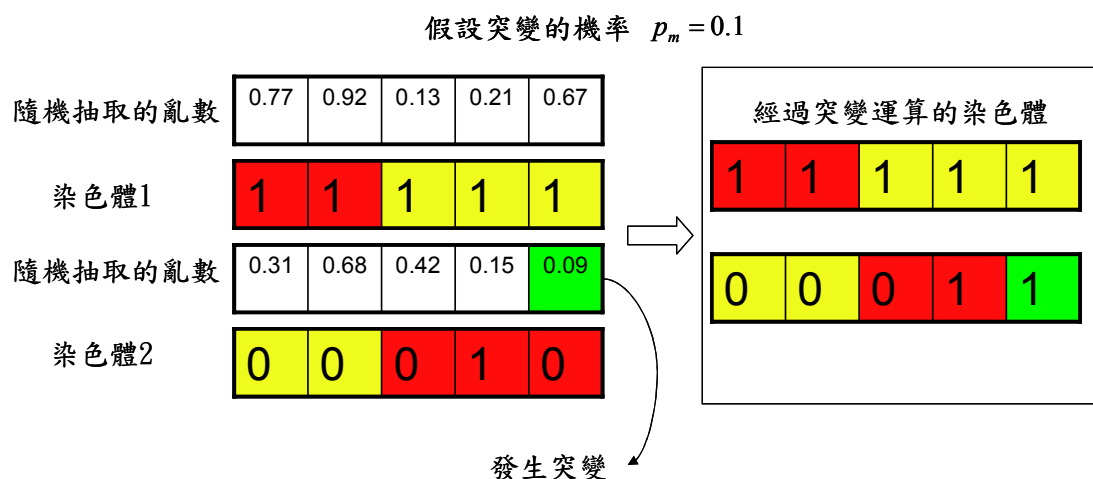


圖 2-9 反相法突變示意圖

完成突變運算之後，將可以得到一群新的個體，稱為子代(Child generation)，接著用子代取代原本的親代，如此便完成了基因演算法迭代一次的動作。稍後在 2.3.2 子小節裡，將會說明基因演算法的基本數學理論，討論其收斂性，而在 2.3.3 子小節裡，將會介紹別種編碼、挑選、交配、突變的方法，並討論其特徵。

### 2.3.2 輪廓理論

基因演算法創始者 John Holland 於 1992 年所提出輪廓理論 (Schemata theorem)，此一理論成功的描述了基因演算法之所以會收斂的原因，奠定了數學上的基礎。為了降低討論基因演算法理論的複雜性，在此僅考慮簡易基因演算法。在此，首先介紹輪廓(Schemata)的概念：如果研究基因演算法所產生的字串，會發現族群中常常出現某些特定的基因字串。高適應度的染色體，通常會有許多共同的基因字串結構。例如族群中高適應度的個體時常出現以 10 為首，以 010 為尾的字串，或是時常出現中間兩位為 10 的字串，將這些字串以  $10*****010$  和  $****10****$  表示，這些星號表示「外卡」(wild card，美國職棒大聯盟的術語)元素，外卡的元素的值是不固定的。如此，便可以利用輪廓(或稱為「相似樣本」，Similarity template，符號記為  $H$ )描述族群中染色體的共同特徵，以下先介紹輪廓理論中的一些術語及一些符號：1.  $P(t)$ 、 $P'(t)$ 、 $P''(t)$  以及  $P'''(t)$  分別代表第  $t$  代的族群、第  $t$  代完成挑選運算的族群、第  $t$  代完成交配運算的族群以及第  $t$  代完成突變運算的族群 2.  $o(H)$  代表輪廓的階數(Schemata order) 3.  $\delta(H)$  代表輪廓的定義長度(Schemata defining length)。

相似模版	字串	階數 $o(H)$	長度 $\delta(H)$
$H_1$	10*****010	5	9
$H_2$	****10****	2	1
$H_3$	11*1*****	3	3
$H_4$	10**1*****	3	4
$H_5$	*0*****	1	0
$H_6$	1*****0	2	9

表 2-2 輪廓階數、長度與其字串的關係

輪廓的「定義長度」的定義為：字串中出現第一個固定值和最後一個固定值的位置的差。表 2-2 中，第一個輪廓  $H_1$ ，出現第一個固定值的位置為 1，而最後一個固定值的位置為 10，所以定義長度為  $10-1=9$ 。同理，第二個輪廓  $H_2$ ，其定義長度為  $6-5=1$ 。

輪廓階數的定義為：字串中出現固定值的數目。表 2-2 中，第一個輪廓  $H_1$ ，總共有 5 個固定值，第二個輪廓  $H_2$  有 2 個固定值。

在介紹完輪廓的長度與階數之後，為了描述輪廓的優劣，接著必須定義族群中輪廓的適應度  $f(H)$ ，輪廓的適應度定義為族群中擁有輪廓  $H$  的適應度的平均值。數學式如下：

$$f(H) = \frac{\sum_{i, c_i \in P \cap H} f(c_i)}{m_p(H, t)} \dots \dots \dots (2.10)$$

式中的  $m_p(H, t)$  表示第  $t$  代族群內擁有輪廓  $H$  的染色體數目。

回顧簡易基因演算法中的輪盤式挑選法，某個染色體  $c^*$  被挑選到

的機率  $p_s$  為：

$$p_s(c^*) = \frac{f(c^*)}{\sum_{i=1}^{N_{pop}} f(c_i)} \dots\dots\dots (2.11)$$

如此，第  $t$  代被挑選出來擁有輪廓  $H$  的數目為

$$\begin{aligned} m_{p'}(H, t) &= N_{pop} \sum_{i, c_i \in H \cap P(t)} p_s(c_i) \\ &= N_{pop} \frac{\sum_{i, c_i \in H \cap P(t)} f(c_i)}{\sum_{i=1}^{N_{pop}} f(c_i)} \dots\dots\dots (2.12) \end{aligned}$$

將第(2.10)式代入(2.12)式可得

$$\begin{aligned} m_{p'}(H, t) &= N_{pop} \frac{\sum_{i, c_i \in H \cap P(t)} f(c_i)}{\sum_{i=1}^{N_{pop}} f(c_i)} \\ &= \frac{N_{pop}}{\sum_{i=1}^{N_{pop}} f(c_i)} \times \frac{\sum_{i, c_i \in H \cap P(t)} f(c_i)}{m_p(H, t)} \times m_p(H, t) \dots\dots\dots (2.13) \\ &= \frac{f(H)}{\bar{f}} m_p(H, t) \end{aligned}$$

式中的  $\bar{f}$  表示族群中染色體適應度的平均值，接下來，考慮特徵的定義長度以及單切點交配法的特色，便可以建構出經過交配運算後仍然保有輪廓  $H$  的數目，假設有一個輪廓字串定義長度為 3，如圖 2-10 所示：



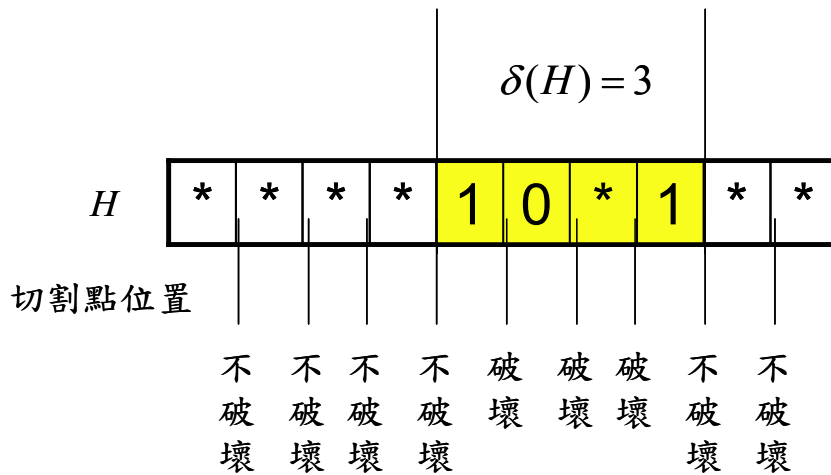


圖 2-10 切點位置與定義長度關係示意圖

由圖 2-10 可以清楚的瞭解，只有當切割點發生在第一個固定值與最後一個固定值的位置之內時，才會破壞原本的特徵，因此被破壞的機率正比於定義長度  $\delta(H)$ 。定義被破壞的機率如下：

$$p_{dest} = \frac{\delta(H)}{l-1} p_c \dots\dots\dots (2.14)$$

式中的  $l$  表示染色體長度，而長度為  $l$  的染色體，其具有  $l-1$  個切割點，而式中的  $p_c$  是交配的機率，也就是兩組染色體具有不同的輪廓的機率。接著可以得到經過交配運算後，族群中擁有輪廓  $H$  的數目

$$m_{p^*}(H, t) \geq m_{p^*}(H, t)(1 - p_{dest}) = \frac{f(H)}{\bar{f}} m_p(H, t) [1 - \frac{\delta(H)}{l-1} p_c] \dots\dots\dots (2.15)$$

式中的大於等於符號表示某種輪廓被破壞之後，可能會產生另外一種輪廓，在最極端的情況之下，就是每次被破壞之後並沒有產生其他種輪廓。接下來，最後一道運算就是突變，突變運算之後的族群就是下一代的族群，意即  $P(t+1) = P''(t)$ ，而定義某種輪廓不被突變運算給破壞，生存下來的機率為  $p_{surv}$ ，其機率跟輪廓階數有關，輪廓中每個固

定值不突變的機率為  $p_{nm} = 1 - p_m$ ， $p_m$  為突變的機率，因此可得

$$p_{surv} = (1 - p_m)^{o(H)} \dots \dots \dots (2.16)$$

其中，由於突變的機率  $p_m$  是很小的值，因此使用泰勒級數(Taylor's series)展開，忽略高階項可得

$$p_{surv} \approx 1 - p_m o(H) \dots \dots \dots (2.17)$$

將(2.17)式結合(2.15)式可以得到

$$\begin{aligned} m_p(H, t+1) &= m_{p^m}(H, t) \\ &\geq m_p(H, t) \frac{f(H)}{\bar{f}} \left[ 1 - \frac{\delta(H)}{l-1} p_c - p_m o(H) \right] \dots \dots \dots (2.18) \end{aligned}$$

藉由觀察(2.18)式可以得知，階數小、定義長度短且適應度大於平均的輪廓會不斷的增加，而適應度低於平均的輪廓會以指數的方式迅速被淘汰。

### 2.3.3 其他的運算方法

在 2.3.1 子小節裡介紹了簡易基因法，在簡易基因演算法中，使用的是二元編碼法、輪盤式挑選法、單切點交配法、以及反相突變法，而在本節將介紹基因演算法其他經常使用到的方法，包含整數編碼法、浮點數編碼法、多切點交配法、均勻交配法…等不同的運算方法，本節將針對流程圖中每一個步驟做詳細的介紹，次頁圖 2-11 是基因演算法的標準流程圖。

#### 1. 產生一個族群：

執行基因演算法之前，必須產生一個族群，最常使用的產生族群

的方法是充分利用隨機亂數來產生族群中每一個染色體的每一個基因。而基因區域搜尋法是結合基因演算法以及區域型演算法，而區域型演算法如共軛梯度法只有一個解(也就是一個染色體)，如何將一個染色體適當的產生一整個族群，便成為結合兩種演算法所必須克服的重要關鍵技術。在此，提出「播種」(Seeding)的概念，由其他演算法所得到的解，就是一個種子，將種子播灑填滿整個族群，如此便可產生一個族群以進行基因演化的程序。

但現在又產生了另一個問題，就是如果丟入族群中的種子僅僅只是原始種子的複製品(Clone)，由於每個染色體都相同，那將會造成演化的停滯不前，如果僅是光靠突變機制產生新的優良基因，那將需要漫長的等待時間。這個解決的方法其實在農業改良上，已經行之有年了，就是利用人工外加的因素，例如照射微量的 X 光，讓種子發生突變。而我們模仿此一機制，在種子被放入族群之前，將每個種子照射不同強度的 X-ray(發生突變的機率正比於 X-ray 的強度)。

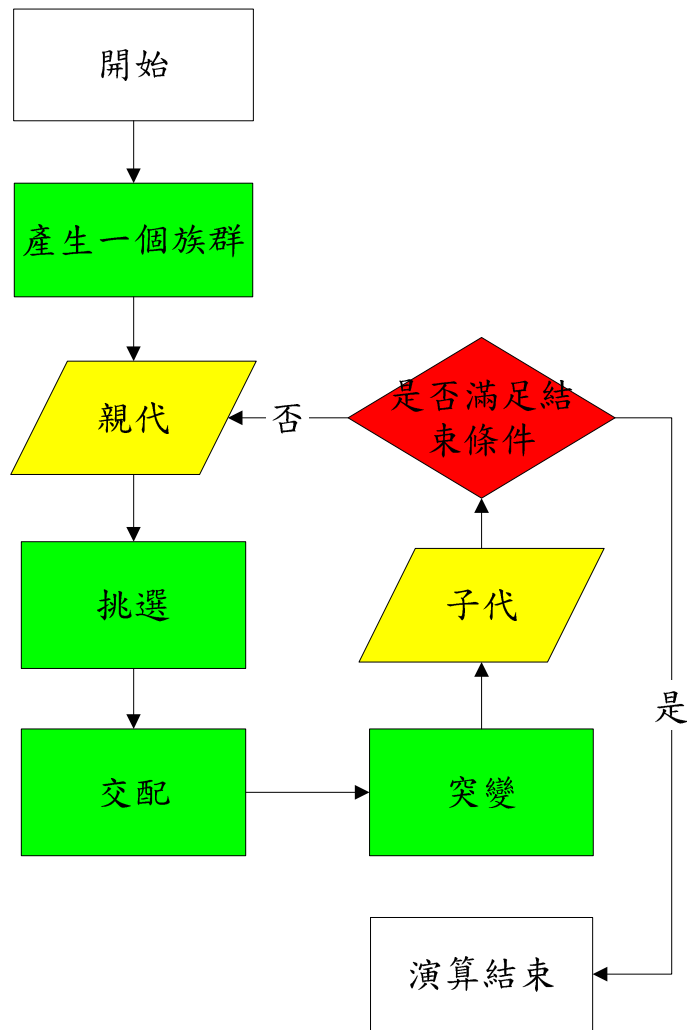


圖 2-11 基因演算法標準流程圖

在此過程之中而第一個種子不照射 X-ray，以保存原有的基因，並且讓族群具有一定的差異性。在程式的實做上，使用的方法是考慮「保留的機率」 $p_{reserve}$ （不突變的機率， $p_{reserve} = 1 - p_{mutate}$ ），而第  $i$  個種子填入族群，被保留的機率考慮使用指數下降的方式

$$p_{reserve} = p_{rd}^{i-1} \dots \dots \dots (2.19)$$

其中  $p_{rd}$  是保留機率衰減的因子，其值可以簡單的被控制。假若我們希望族群中有一半以上的染色體保留的機率在一半以上，可以得到

$$0.5 = p_{rd}^{\frac{N_{pop}-1}{2}} \dots\dots\dots(2.20)$$

計算可得

$$p_{rd} = \frac{N_{pop}-1}{2} \sqrt{0.5} \dots\dots\dots(2.21)$$

此一作法確保了族群中每個個體的差異性，同時又保留了原本優良的基因，成功的解決了基因演算法與其他演算法交互迭代所會碰到的問題。

## 2. 編碼(encode)/解碼(decode)：

常見的編碼方法有：1. 二元編碼法(Binary coding)2. 整數編碼法(Integer coding)2. 浮點數編碼法(Float-point coding，又稱為 Real-number coding)4. 混合型編碼法(Mixed)。事實上，任何編碼的方式最後都可以轉換成為二元編碼法，只是表達形式的不同而已，可以視實際的需要，挑選一種最直接的編碼方式用來表示解的形式，在本論文裡頭，所採用的編碼方式是整數編碼法，一個基因代表的數值為 0~7 的整數值，與二元化編碼法不同的地方在於二元化編碼法使用 3 個基因來表示某個位置的相位補償，而整數編碼法使用一個基因來表示，這兩種編碼的差別在交配以及突變時都會稍有差異，稍後會在挑選及突變的部分有詳細的討論。而一個整數  $0 \sim 2^{l-1}$  的整數與二元編碼法的轉換公式為

$$V_{int} = \sum_{i=0}^{l-1} B_i \times 2^i \dots\dots\dots(2.22)$$

式中的  $B_i$  代表二元編碼法中，位置為  $i$  的基因， $V_{int}$  表示轉換成整數後所代表的數值。對於一個介於  $[a, b]$  範圍內的浮點數，可以用下列公式轉換：

$$V_{float} = a + (b - a) \frac{\sum_{i=0}^{l-1} B_i \times 2^i}{2^l - 1} \dots\dots\dots (2.23)$$

$$precision = (b - a) \frac{1}{2^l - 1} \dots\dots\dots (2.24)$$

(2.24) 式表示資料的精確度。有了(2.22)及(2.23)式，便可以將不同的編碼方法轉換成二元化編碼。

二元編碼	整數值	a	b	精確度	浮點值
0010110	22	3.0	4.5	0.0118	3.2598
1011001	89	-5.0	2.1	0.0559	-0.0244

表 2-3 7 位元之二元編碼與整數、浮點數的轉換

### 3. 挑選：

在基因演算法裡，其挑選的過程是使用「物競天擇」、「適者生存」(fittest-of-the survival)、「合適再製」(reproduction based on fitness)的概念，藉由物種間彼此不斷的競爭，快速地淘汰不良的基因。而普遍的挑選方法有三種：1. 輪盤式挑選法 2. 競爭式挑選法 (Tournament selection) 3. 隨機式挑選法(Random selection)。其中，第一種輪盤式挑選法是一種按照適應度比例挑選(proportionate selection scheme)的方法，具有運算簡單、快速等特性，是最常使

用的方法，本論文也採用此一方法，由於在第 2.3.1 子小節已經介紹過了，在此便不重複說明。第二種挑選方法是競爭式挑選法，其運算的方法是先將染色體依照適應度的高低，進行排序(Sorting)，排名超過「及格排名」(一個參數，經常使用的值為  $N_{pop}/2$ )的染色體直接淘汰，僅有優良的個體能夠進入交配池(Mating pool)產生子代。此一演算法必須適當的控制及格排名這個參數，如果及格排名取的太小，將會導致收斂過早的現象。反之，如果及格排名取的太大，又會發生收斂過慢的現象。此外，由於此種挑選方法還需要一個額外的排序動作，因此較不常被使用。至於第三種挑選的方法由於其過程過於隨機，導致收斂速度過慢，因此僅有在較特殊問題時，才會採用此種挑選方法，以確保演算不至於收斂過早。

#### 4. 交配：

先前介紹過單切點交配法，而實際上經常使用到的方法還有：1. 雙切點交配法 2. 均勻交配法(可視為多切點交配法的一種)。所謂的雙切點交配法是在  $1 \sim (l-1)$  位置的基因上，隨機找兩個切割點，交換彼此的基因，如圖 2-12 所示。基本上，雙切點交配法可以利用兩次的單切點交配法運算來完成。

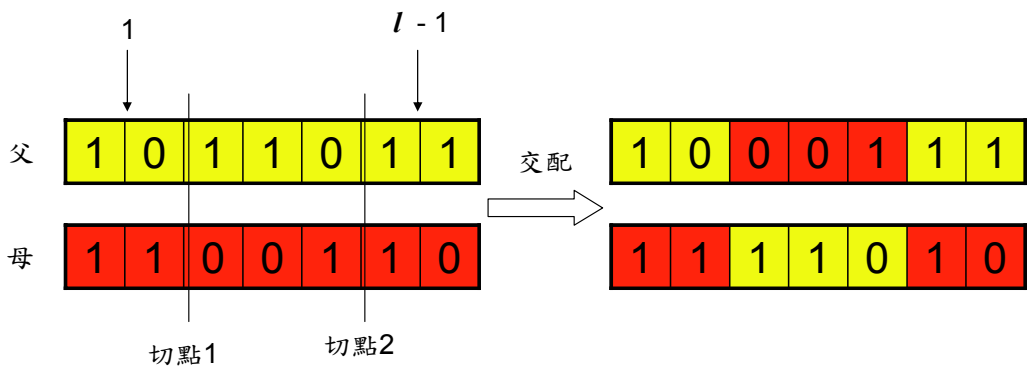


圖 2-12 雙切點交配法示意圖

所謂的均勻交配法(Uniform crossover)是將父母的基因，以隨機的方式均勻的混合成子代染色體。事實上，這三種不同的交配方法可以利用交配遮罩(Crossover mask)來描述這些交配動作，圖 2-13 為示意圖。均勻交配法可以先用隨機亂數產生一個交配遮罩，然後可以利用幾個簡單的布林運算，完成均勻交配法的動作，如圖 2-14：

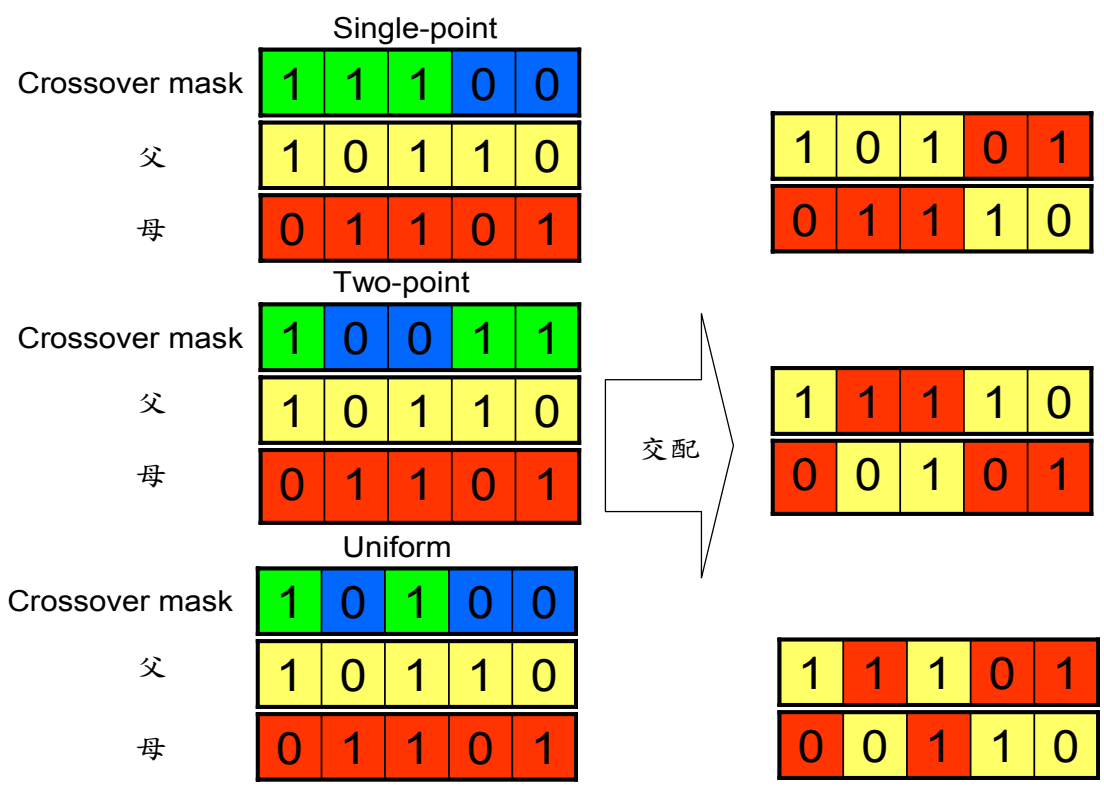


圖 2-13 交配遮罩與各種交配法的關係圖



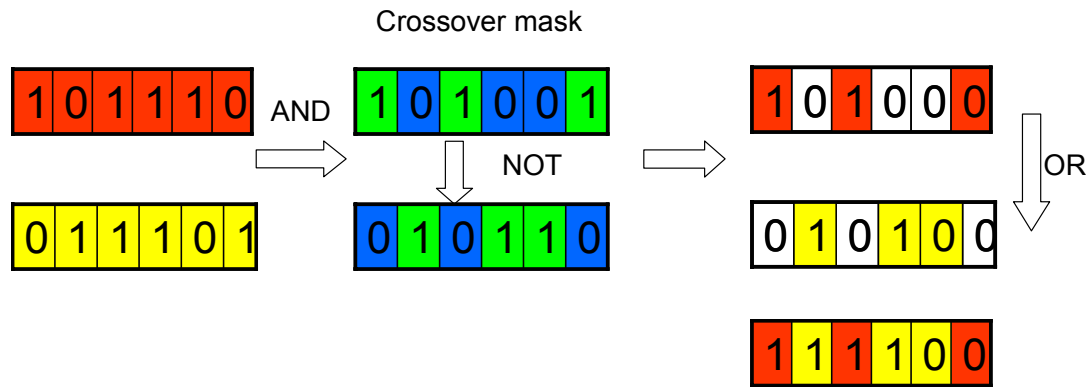


圖 2-14 利用布林運算完成均勻交配法

由圖 2-14 可以看出，均勻交配法特別適合二元化的編碼(切割的最小單位為位元)，而整數編碼法中，一個基因可能需要使用多個位元，因此較不適用於均勻交配法(事實上，仍可以利用位元擴充的概念解決此一問題，不過計算較複雜)。

## 5. 突變：

突變運算是根據突變機率(mutation probability)對某些染色體進行突變動作，與交配機制不同的地方在於交配機制不會產生新的基因物質，而突變能在在母體種產生新的資訊。基本上，突變運算在基因演算法中所負責的任務就是修補遺失(lost)或是未探訪到(unexplored)的基因到母體中，以避免演算法收斂過早，這種獨特的運算機制能夠使得基因演算法避免陷入區域最佳解。

常使用的突變機制有二元型(binary)，整數型(integer)，以及實數型(real value)。二元型編碼常利用反相法來達到突變，使用布林運算中的 NOT 運算來達到位元反轉的功能，將 0 改為 1，或將 1 改為

0。另外由於某些特殊的編碼,例如染色體中的 0 與 1 的數目是固定的,此時所使用的方法為位移法(shifting),將某段基因與另一個位置的某段基因交換。整數型編碼法時常使用的方法為亂相法(random phase),當突變發生時,將該基因以一個亂數值取代。實數型編碼法經常使用的方法為漸進法,將原本的基因值加減某個量的數,這個數稱為步進值(step value),如圖 2-15 所示。

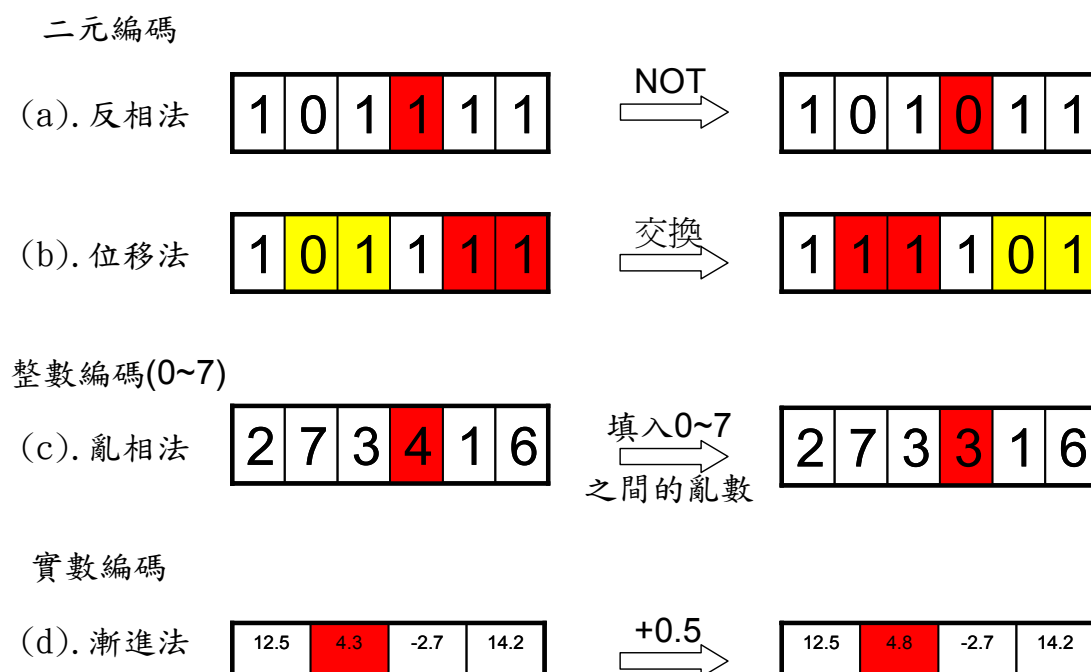


圖 2-15 各種編碼方法所使用的突變運算

## 2.4 基因區域搜尋法

大部分傳統的最佳化法(搜尋法),都是只能考慮一個點的決策法則,這些方法對於多峰值模式(multi-peak model)的問題而言,將有可能陷入區域最佳解的陷阱裡。基因演算法可同時評估多點且具有交配、突變等機制,可以防止落入區域最佳解的陷阱。另外,基因演算

法僅需要瞭解所有適應函數所包含的參數即可運算，與問題的形式無關，不像其他種演算法必須有諸多限制(例如相位迭代法需要進行傅力葉轉換)。一般而言，基因演算法與其他傳統的最佳化演算法互相比較，基因演算法具有以下優點：

1. 基因演算法同時考慮搜尋空間中的多個峰值，而不是單個點，因此可以迅速獲得整體最佳解。同時也可以避免陷入局部最佳解，此乃基因演算法最大的優點。
2. 基因演算法直接以參數集合(parameter set)的字串進行編碼運算，而非參數本身，因此可以跳脫搜尋空間分析上的限制。
3. 基因演算法使用機率轉換法則(probabilistic transition rules)去引導搜尋方向，而非決定性法則(deterministic rules)，因此較能符合各種不同類型的最佳化問題。
4. 基因演算法採用適應度函數資訊，而非導數或其他輔助知識，對於連續或不連續之系統皆可適用。

但基因演算法也並不是毫無缺點，最令人詬病的一個重大缺點就是運算速度過慢，為了解決此一問題，我們利用區域型搜尋法輔助基因演算法，以加速收斂的速度。要交互使用兩種不同的演算法必須要注意幾點：1. 兩種演算法目標函數的設定必須要是完全等效的，2. 編碼上必須要容易轉換。在此所選擇搭配的演算法為共軛梯度法，使用共軛

梯度法的原因是因為共軛梯度法是一種極快速且通用(General)的演算法，完全符合我們的要求，故在此選擇共軛梯度法來輔助基因演算法。其基本的演算流程圖如下：

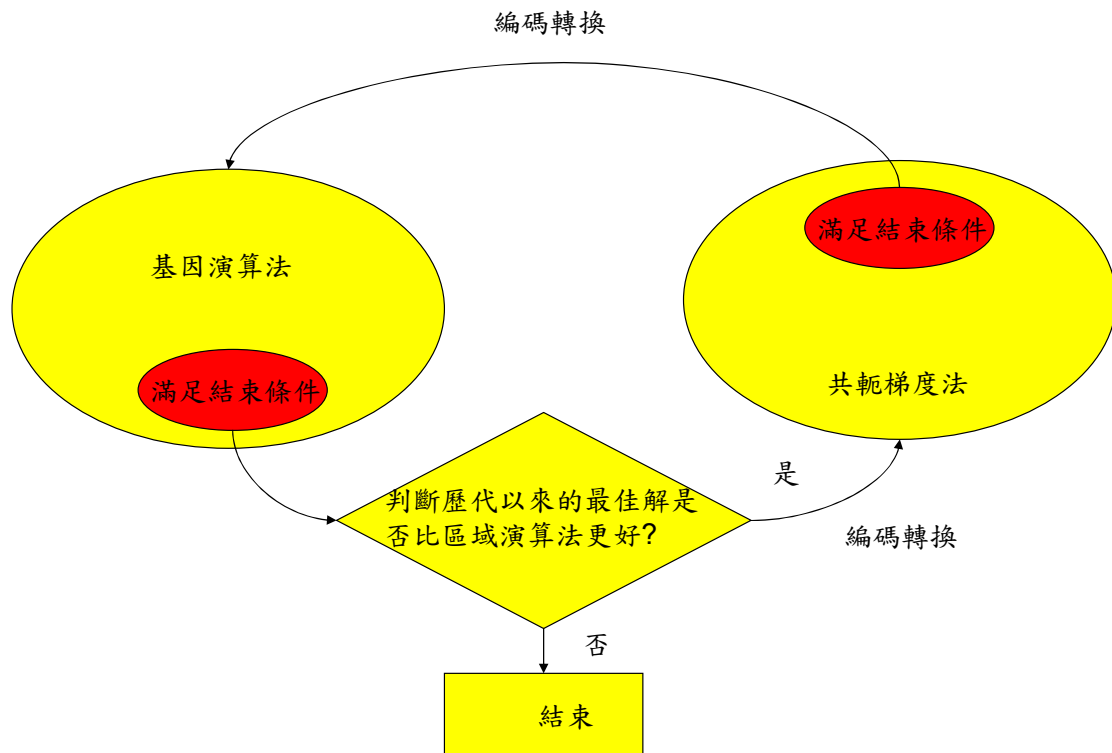


圖 2-16 GLSA 基本架構圖

編碼轉換的過程取決於編碼的形式，本文將於第四章再做說明。在基因演算法開始運算之前，使用前面所提的播種方式，將共軛梯度法所得的最佳解進行複製並做適當的改變(以不同的突變機率進行突變運算)，以便產生一整個族群開始進行演算。本文將於第五章實際展示此演算法的優點，並討論此種混合型演算法的一些調整方法。

### 第 3 章 繞射光學元件的基本理論

當我們用一光束照射一個尺寸較小的不透明物體，並將它的影像投影到一個屏幕上，如果仔細觀察陰影的輪廓，會發現不但光線會進入物體後方的陰影裡，同時在光線直接照射的邊緣區域也會有暗影存在，造成物體陰影輪廓模糊的現象，與光線經過一個針孔或一條狹縫後的擴散現象有密切的關係。索莫非(Sommerfeld)將此種無法以光線以直線傳播的現象稱為「繞射」(Diffraction)。

一般設計及分析繞射光學元件可依不同的需求及適合性採用幾何光學、向量繞射(Vector diffraction)、純量繞射(Scalar diffraction)等理論。其中幾何光學方面可以利用光柵方程式及光線追跡來預測光波經由繞射光學元件作用後的行進方向，較適合用來配合傳統的光學設計，但他不能用來計算繞射效率，同時其計算的結果比較不精確。向量繞射理論對於繞射元件的模擬分析是以馬士威爾波動方程式(Maxwell's wave equation)為基礎，利用數值方法解出光波傳遞的行為，常使用的方法有光束傳播法(Beam Propagation Method, BPM)以及時域有限差分法(Finite Difference Time Domain, FDTD)。而純量繞射理論則是向量繞射理論的一個近似結果，當元件的最小線寬大於波長的數倍時，可以忽略元件不連續處以及光波的偏振態所造成的效應。在採用近似之後，光波的繞射現象

可以用一個積分公式來表示，計算量比利用數值方法實際去解馬士威爾波動方程式小許多，本論文採用此一分析方式。

### 3.1 基本繞射理論

在一定限定條件下，為了簡化數學運算的過程，在模擬重建波場時我們採用數學模型較簡單的純量繞射理論<sup>10</sup>，捨棄在計算分析上較為複雜的向量繞射理論 (vector diffraction theory)。另外，實際製程中由於製程技術的限制，我們並無法實現由電腦計算求解出的最佳元件表面輪廓之連續分佈，而是利用二進階光學 (binary optics) 理論以分階方式做連續表面輪廓結構的近似。

接著我們從自由空間 (free space) 中的馬克斯威爾方程式 (Maxwell's equations) 出發：

$$\begin{cases} \vec{\nabla} \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t} \\ \vec{\nabla} \times \vec{H} = \varepsilon \frac{\partial \vec{E}}{\partial t} \dots\dots\dots(3.1) \\ \vec{\nabla} \cdot (\varepsilon \vec{E}) = 0 \\ \vec{\nabla} \cdot (\mu \vec{H}) = 0 \end{cases}$$

其中  $\vec{E}$  是空間中的電場，在直角座標系上以座標  $(E_x, E_y, E_z)$  表示； $\vec{H}$  是空間中的磁場，在直角座標系上以座標  $(H_x, H_y, H_z)$  表示； $\mu$  和  $\varepsilon$  分別是空間中的磁導率 (permeability) 與電容率 (permittivity)。

假設電磁波是在介電質中傳播，為了簡化整個數學計算，我們必須進一步對介電質做出一些假設：

1. 線性的 (linear), 介電質滿足線性條件 (linearity property)。
2. 各向同性的 (isotropic), 即電場、磁場、極化與方向無關。
3. 均勻性的 (homogeneous), 在整各傳播空間裡介電係數是一個常數。
4. 無磁性的 (nonmagnetic), 假設磁導率永遠等於真空中的磁導率。

當傳播的介質滿足上面四個條件, 經由數學代換後, 我們可以將馬克斯威爾方程式簡化為:

$$\begin{cases} \nabla^2 \bar{E} - \frac{n^2}{c^2} \frac{\partial^2 \bar{E}}{\partial t^2} = 0 \\ \nabla^2 \bar{H} - \frac{n^2}{c^2} \frac{\partial^2 \bar{H}}{\partial t^2} = 0 \end{cases} \dots\dots\dots (3.2)$$

其中  $n$  為折射率, 定義為  $n = (\epsilon/\epsilon_0)^{1/2}$ ,  $\epsilon_0$  是真空中的電容率;  $c = 1/\sqrt{\mu_0\epsilon_0}$  是真空中的光速。綜合以上討論, 在直角座標系裡的電場與磁場各有三個分量, 我們可以得到一個純量的傳播方程式, 如下所表示:

$$\nabla^2 U(P,t) - \frac{n^2}{c^2} \frac{\partial^2 U(P,t)}{\partial t^2} = 0 \dots\dots\dots (3.3)$$

其中  $U(P,t)$  可以是向量場的任何一個分量, 我們可以發現此函數只與位置  $P$  和時間  $t$  有關。接著我們考慮在時間簡諧場 (time-harmonic field) 成立時, 因此可以得到:

$$(\nabla^2 + k^2)U = 0 \dots\dots\dots (3.4)$$

這裡的  $k$  是波數 (wave number), 即  $k = 2n\pi\omega/c = 2\pi/\lambda$ ,  $\lambda$  是在介質中的光波波長, (3.4) 式就是著名的赫姆霍茲方程式 (Helmholtz' s

equation)。

為了求解複數型式的電場分佈  $U$ ，我們可以藉由格林定理 (Green' s theorem) 進行數學運算

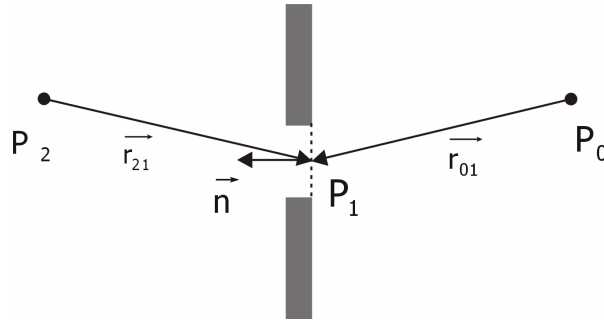


圖 3-1 點光源照射到一平面屏

經由一連串的數學假設、推導與運算，最後我們可以得到菲涅耳-克希荷夫繞射公式 (Fresnel-Kirchhoff diffraction formula)，即

$$U(P_0) = \frac{A}{j\lambda} \iint_{\Sigma} \frac{e^{jk(r_{21}+r_{01})}}{r_{21}r_{01}} \left[ \frac{\cos(\vec{n}, \vec{r}_{21}) - \cos(\vec{n}, \vec{r}_{01})}{2} \right] ds \dots\dots\dots (3.5)$$

如圖 3-1，其中  $P_2$  是點光源的位置， $P_1$  孔徑上的一點， $P_0$  則是任意輸出面之點的位置。接著再將公式重寫如下：

$$U(P_0) = \iint_{\Sigma} U(P_1) \frac{e^{jkr_{01}}}{r_{01}} ds \dots\dots\dots (3.6)$$

其中

$$U(P_1) = \frac{1}{j\lambda} \left[ \frac{Ae^{jkr_{21}}}{r_{21}} \right] \left[ \frac{\cos(\vec{n}, \vec{r}_{21}) - \cos(\vec{n}, \vec{r}_{01})}{2} \right] \dots\dots\dots (3.7)$$

上式即惠更斯原理 (Huygens' principle) 所描述的次級子波



(secondary wavelet) 的複數振幅，該複數振幅包含了菲涅耳與克希荷夫對惠更斯原理的修正因子與訊息：

- 1)  $1/j$  項隱含著次級子波的波前超越入射波波前  $\pi/2$ 。
- 2)  $\frac{\cos(\vec{n}, \vec{r}_{21}) - \cos(\vec{n}, \vec{r}_{01})}{2}$  為傾斜因子 (obliquity factor)，表示次級子波對  $P_0$  的貢獻度。

在式(3.7)中，考慮純相位型元件(振幅 A 為 1)以及平行光入射，則可以得到：

$$U(P_0) = \frac{1}{j\lambda} \iint_{\Sigma} e^{j\phi(P_1)} \frac{e^{jk r_{01}}}{r_{01}} \frac{\cos(\vec{n}, \vec{r}_{21}) + 1}{2} ds \dots\dots\dots(3.8)$$

式中  $\phi(P_1)$  表示由於元件表面浮雕的輪廓所造成相位變化，由於所設計元件是二進階式，在最小線寬內相位的補償都是固定的，因此將(3.8)式改寫成：

$$U(P_0) = \frac{1}{j\lambda} \sum_n \left[ U_{P_1}(S_n) \cdot \iint_{S_n} \frac{e^{jk r_{01}}}{r_{01}} \frac{\cos(\vec{n}, \vec{r}_{01}) + 1}{2} ds \right] \dots\dots\dots(3.9)$$

其中  $U_{P_1}(S_n)$  表示在  $S_n$  區域內的複數振幅，其與相位補償的對應關係為

$$U_{P_1}(S_n) = e^{j\phi(S_n)} \dots\dots\dots(3.10)$$

此一運算即為向量內積運算，因此令

$$\begin{aligned} \vec{U}_R &= (U_1, U_2, \dots, U_n) \\ \vec{G}_{P_1 \rightarrow P_0} &= (G_1, G_2, \dots, G_n) \dots\dots\dots(3.11) \end{aligned}$$

$$\begin{aligned} G_i &= \iint_{S_n} \frac{e^{jk r_{01}}}{r_{01}} \frac{\cos(\vec{n}, \vec{r}_{01}) + 1}{2} ds \\ U(P_0) &= \vec{G}_{P_1 \rightarrow P_0} \cdot \vec{U}_R \dots\dots\dots(3.12) \end{aligned}$$

如此在程式中，僅需在一開始計算好  $G$  向量，然後利用向量內積的方式可以迅速的重建出像平面的光場分佈，而不需要每次都進行積分計算。而對於二進階光學來說，例如分成八階，僅有八種不同相位補償，因此在(3.12)式中的計算，可以藉由提出公因數的方式，進一步減少運算量，對於利用基因演算法設計元件的運算速度有很大幫助。

### 3.2 菲涅耳波帶透鏡以及繞射光學匯聚元件的基本特性

具有浮雕表面結構的透鏡稱為菲涅耳波帶透鏡(Fresnel lens)，很久以來就為光學科學家所關注。其起源應追溯到近一個世紀前 L. Rayleigh, Soret 和 Wood 等。人們很快發現，作為實用的成像透鏡，必須克服波帶片因存在多級像而損失光能量的缺點。其解決途徑是將波帶片製作成具有浮雕表面的相位結構，浮雕的厚度在波長量級，且圖形應盡可能接近設計值以實現預期的高繞射效率。這些設想直到近年來，由於加工技術的進步才得以實現。

#### 3.2.1 基本公式的推導

我們由菲涅耳波帶 (Fresnel zones) 談起，如圖 3-2 所示， $S$  表示點光源之位置，假定  $P$  為點光源經過繞射光學元件後的成像位置， $O$  是元件中心並且在光軸  $\overline{SP}$  直線上，元件上一點  $Q$  到  $S$ 、 $P$  的距離分別為  $R$ 、 $R'$ 。

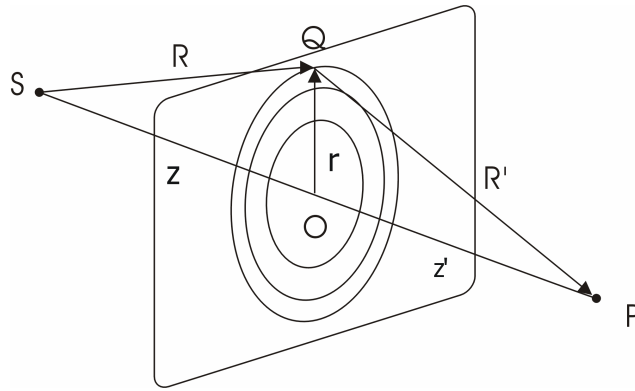


圖 3-2 菲涅耳波帶示意圖

由於菲涅耳波帶每一個圓環代表的意義即是相鄰兩圓環間光程差為設計波長的一半，我們可以利用數學式表示如下：

$$\begin{aligned}
 R + R' &= z\left[1 + \frac{1}{2}\left(\frac{r}{z}\right)^2 + \dots\right] + z'\left[1 + \frac{1}{2}\left(\frac{r}{z'}\right)^2 + \dots\right] \dots\dots\dots(3.13) \\
 &\approx z + z' + \frac{r^2}{2}\left(\frac{1}{z} + \frac{1}{z'}\right)
 \end{aligned}$$

再經由簡單的數學推導可以得到： $r_1 = \sqrt{\lambda_0 f}$ ， $r_2 = \sqrt{2\lambda_0 f}$ ， $\dots$ ， $r_n = \sqrt{n\lambda_0 f}$ ，其中 $\lambda_0$ 是繞射透鏡的設計波長， $f = (1/z + 1/z')^{-1}$ 為繞射透鏡的主焦點焦距。我們可以發現菲涅耳波帶上每一環的面積都是相等的，即 $\pi r_n^2 - \pi r_{n-1}^2 = \pi r_1^2$ 。接著我們假設 $g(x, y)$ 為繞射透鏡的相位分佈函數，因為它具有圓對稱徑向平方的週期性，我們可以將其表示為

$$g(x, y) = g(x^2 + y^2) = g(r^2) = g(r^2 + Nr_p^2) \dots\dots\dots(3.14)$$

其中 $r_p = \sqrt{2\lambda_0 f}$ ，N 為一正整數。由於 $g(x, y)$ 是一個週期性函數，我們利用富利葉級數 (Fourier series) 展開如下

$$g(x, y) = g(x^2 + y^2) = g(r^2) = \sum_{m=-\infty}^{\infty} A_m \exp\left(j2m\pi \frac{x^2 + y^2}{r_p^2}\right) \dots\dots\dots(3.15)$$

$$A_m = \frac{1}{r_p^2} \int g(r^2) \exp\left(-j2m\pi \frac{r^2}{r_p^2}\right) dr^2$$

將(3.15)式代入近場繞射積分公式 (near field diffraction integral formula)

$$U(x', y', z') = \frac{e^{jkz'}}{j\lambda z'} \iint_{\infty} g(x, y) \exp\left(jk \frac{(x-x')^2 + (y-y')^2}{2z'}\right) dx dy \dots\dots(3.16)$$

整理後可得到繞射場分佈

$$U(x', y', z') = \sum_{m=-\infty}^{\infty} A_m \exp\left[\frac{j\pi}{\lambda z'}(x'^2 + y'^2)\right]$$

$$\times \left\{ \iint \exp\left[j2\pi \left(\frac{m}{r_p^2} + \frac{1}{2\lambda z'}\right)(x^2 + y^2)\right] \exp\left[j2\pi \left(\frac{xx' + yy'}{\lambda z'}\right)\right] dx dy \right\} \dots\dots$$

..... (3.17)

其中  $\lambda$  為入射光波長 (與設計波長  $\lambda_0$  不同)。令 (3.17) 式中的積分

項中指數  $\left(\frac{m}{r_p^2} + \frac{1}{2\lambda z'}\right)$  在焦點處為零，其中  $r_p = \sqrt{2\lambda_0 f_0}$ ，則我們可以定義：

$$z' \equiv f = -\frac{f_0 \lambda_0}{m\lambda} \dots\dots\dots(3.18)$$

從 (3.18) 我們可以看出入射波長與焦距成反比，即它的色散性質與傳統折射式透鏡相反，這也就是為什麼我們可以利用繞射透鏡製作消色差複合透鏡 (hybrid lens) 的原因，稍後我們針對繞射透鏡此一物理特性，設計濾波元件。接著，我們考慮如圖 3-3 繞射光學元件的

表面輪廓（已經根據二進階光學分階）。

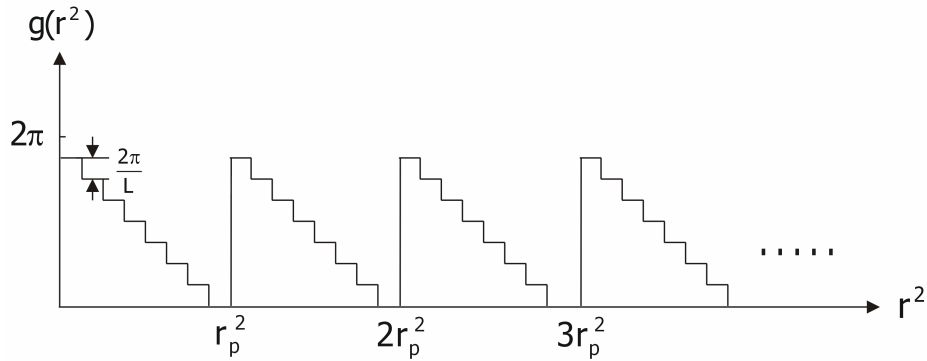


圖 3-3 二進階繞射光學元件之相位函數與半徑平方的關係圖

假設光在通過元件後的相位分佈函數為

$$g(r^2) = \sum_{n=0}^{N(L-1)} \exp\left(\frac{j2\pi(L-1)}{L}\right) \exp\left(\frac{-j2n\pi}{L}\right) \text{rect}\left(\frac{r^2 - nr_p^2/L - r_p^2/2L}{r_p^2/L}\right) \dots\dots\dots$$

.....(3.19)

其中  $L$  為元件分階的階數。我們令  $r_p^2 = 1$ ，並省略  $\exp(j2\pi(L-1)/L)$  項，

可以得到第  $m$  級繞射的複數振幅為

$$A_m = \frac{1}{r_p^2} \int_0^{r_p^2} \sum_{n=0}^{N(L-1)} \exp\left(\frac{-j2n\pi}{L}\right) \text{rect}\left(\frac{r^2 - nr_p^2/L - r_p^2/2L}{r_p^2/L}\right) \exp(j2m\pi r^2) dr^2 \dots\dots\dots$$

$$= \frac{1}{L} \exp\left(\frac{-jm\pi}{L}\right) \text{sinc}\left(\frac{m}{L}\right) \sum_{n=0}^{(L-1)} \exp\left(-j2\pi \frac{n(m+1)}{L}\right)$$

.....(3.20)

其中，

$$\sum_{n=0}^{(L-1)} \exp\left(\frac{-j2n(m+1)\pi}{L}\right) = \begin{cases} L, & m = aL - 1 \quad (a \in Z) \dots\dots\dots(3.21) \\ 0, & \text{otherwise} \end{cases}$$

我們定義第  $m$  級繞射效率為：

$$\eta_m = \frac{I_m}{\sum_{m=-\infty}^{\infty} I_m} \dots\dots\dots(3.22)$$

(3.22) 式若滿足  $m = aL - 1$ ，則 (3.20) 式可化簡為

$$A_m = \exp\left(\frac{-jm\pi}{L}\right) \text{sinc}\left(\frac{m}{L}\right) \dots\dots\dots(3.23)$$

$$I_m = |A_m|^2 = \text{sinc}^2\left(\frac{m}{L}\right)$$

接著，我們將各階繞射光強度歸一化，藉由 (3.22)、(3.23) 可以得到第-1級繞射效率

$$\eta_{-1} = \text{sinc}^2\left(\frac{1}{L}\right) \dots\dots\dots(3.24)$$

根據 (3.24) 式，我們可以估算繞射光學元件在不同階數的第-1級繞射效率，如表 3-1。

階數	繞射效率
2	40.53%
4	81.06%
8	94.96%
16	98.72%

表 3-1 不同階數繞射光學元件的-1 級繞射效率

### 3.2.2 繞射元件的規格與最小線寬之關係

菲涅耳波帶透鏡可由光線追跡法求得其表面輪廓，由於元件是二進階式，因此存在著最小線寬的問題，在最小線寬之內，元件的浮雕高度是固定的，也就代表著元件給予光波的相位補償是固定的。如圖

3-4 所示：

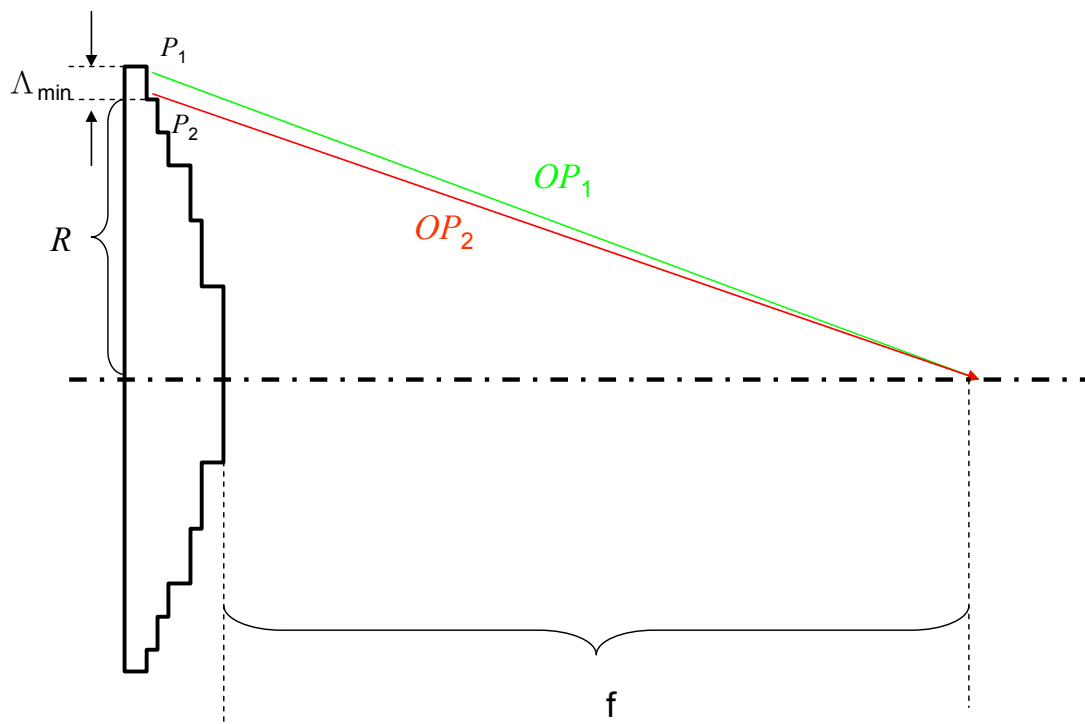


圖 3-4 DOE 的最小線寬與光程示意圖

在圖 3-4 中， $R$  是元件的半徑，元件在  $P_1$  位置與  $P_2$  位置處所給予相位補償是相同的，但是實際上，由  $P_1$  位置走到焦點與  $P_2$  位置走到焦點位置的光程是不同的，假若光程的變化不大，則可以相同的相位補償近似原本應該連續變化的相位補償，但假若光程變化過大，通過  $P_1$  位置與  $P_2$  位置的光波勢必不能形成建設性干涉，因此考慮雷利的  $\frac{\lambda}{4}$  判據可得

$$OPD = OP_1 - OP_2 \leq \frac{\lambda}{4} \dots\dots\dots (3.25)$$

$$OP_1 - OP_2 = \sqrt{(R + \Lambda_{\min})^2 + f^2} - \sqrt{R^2 + f^2} \approx \frac{1}{2} \frac{2R\Lambda_{\min} + \Lambda_{\min}^2}{f} \dots\dots\dots (3.26)$$

考慮  $R \gg \Lambda_{\min}$ ，整理以上兩式可得

$$f \geq 4R \frac{\Lambda_{\min}}{\lambda} \dots\dots\dots(3.27)$$

其中， $\frac{\Lambda_{\min}}{\lambda}$  是線寬與波長的比例，考慮使用純量繞射模擬的限制，此值必須恆大於 1，如此便可以決定出繞射型聚焦元件的數值孔徑 (Numerical Aperture, NA)，也就是焦距與元件尺寸的關係。

### 3.3 濾波元件的設計概念

之前已經介紹過繞射光學元件較一般傳統光學元件更具色散特性，故利用此特殊性質來設計有濾波功能的光學元件<sup>11</sup>

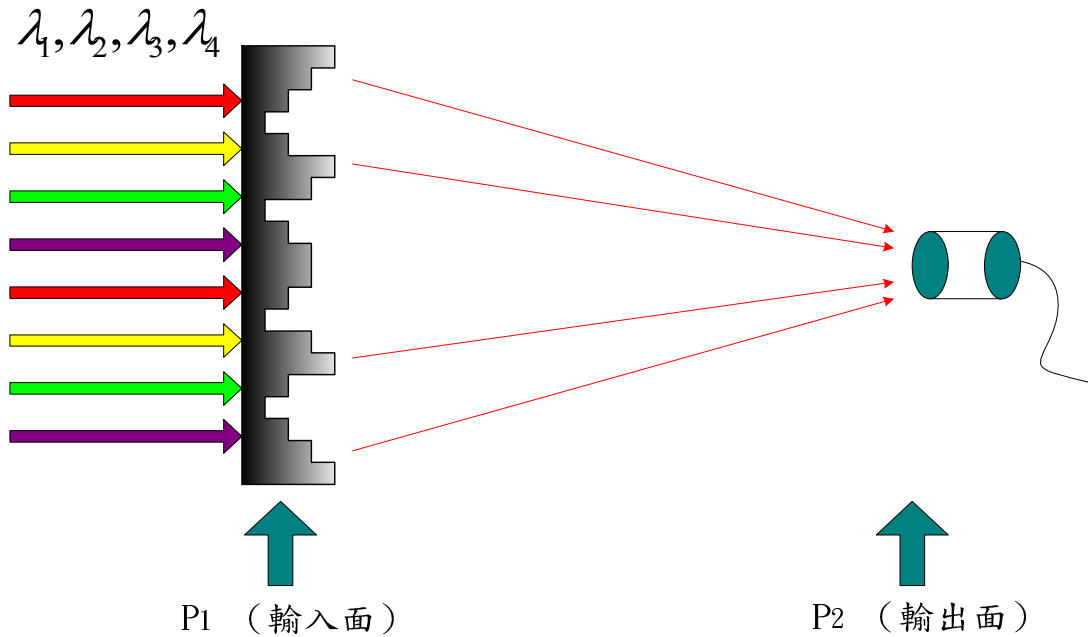


圖 3-5 濾波元件基本架構圖

上圖為所設計之濾波繞射光學元件，此元件不但具有濾波功能，亦具有聚焦功能，可將所需要的光波指定在設定的訊號區域內，而其餘波長均不會在我們的訊號區域內有太大的能量分佈。當然一般的聚焦透鏡也可以達到這個功用，不過其色散的效果不彰，又由於設計的



四個輸入光的波長分別為：1533nm、1541nm、1549nm 和 1557nm，各波長之間差 8nm，這可不是一般的透鏡所能辦到的。故利用繞射元件的色散特性來達到此一目的。

基本上，此濾波元件針對主波長 $\lambda_1$ 而言，是屬於一種聚焦元件，而根據(3.18)式可知，其他的波長仍然會聚焦在 Z 軸上，而焦點的位置沿 Z 軸發生偏移，其偏移量為 $\Delta f_{\lambda_1\lambda_2}$

$$|\Delta f_{\lambda_1\lambda_2}| = |f_2 - f_1| = \left| \frac{\lambda_1}{\lambda_2} f_1 - f_1 \right| = \frac{|\lambda_1 - \lambda_2|}{\lambda_2} f_1 \dots\dots\dots (3.28)$$

現在考慮一個透鏡的焦深(Depth of focus) $\delta$ ，如果焦距的偏移量比焦深還大，那其他的波長在主波長的焦點處將會發生散焦的現象，這就是繞射型濾波元件的基本光學性質。將以上條件寫成下式：

$$|\Delta f_{\lambda_1\lambda_2}| > \delta \dots\dots\dots (3.29)$$

而 $\delta = \frac{2\pi r_1^2}{\lambda_1}$ ，式中的 $r_1$ 為光斑半徑， $r_1 = 1.22 \frac{\lambda_1 f_1}{D}$ ，考慮 $\lambda_2 > \lambda_1$ ，則將(3.29)

是化成

$$f_1 < \frac{(\lambda_2 - \lambda_1) D^2}{2\pi(1.22)^2 \lambda_1 \lambda_2} \dots\dots\dots (3.30)$$

結合(3.27)式，則可以得到

$$\frac{4R\Lambda_{\min}}{\lambda_1} \leq f_1 < \frac{(\lambda_2 - \lambda_1) D^2}{2\pi(1.22)^2 \lambda_1 \lambda_2} \dots\dots\dots (3.31)$$

考慮四個輸入波長為 1533nm、1541nm、1549nm 和 1557nm，其中主波長為 1533nm，各波長之間差 8nm，由(3.31)可以決定出元件的規格，我們訂出以下規格：

元件半徑(μm)	10800
主波長(nm)	1533
其他波長(nm)	1541、1549、1557
分階數(階)	8
焦距(μm)	144000
訊號區半徑(μm)	12.4684
最小線寬(μm)	6
元件分割點數	1800

表 3-2 元件規格

當我們制訂主波長、元件的半徑以及焦聚等規格之後，可以知道該元件焦點處的光斑大小(Spot size)，此大小就是愛里盤(Airy Disc)的大小，其半徑大小可由  $r=1.22\frac{\lambda f}{D}$  求得，並將該大小定義為訊號區。

由表 3-2 的規格，利用光線追跡的方法，可以得到一個 Fresnel 透鏡，如圖 3-6 所示。圖形中的 y 軸所代表的是相對的相位補償，其與輪廓表面浮雕高度的轉換關係為：

$$h = \frac{RPC}{L} \frac{1}{(n_{\lambda} - n_{air})} \lambda_1 = \frac{RPC}{L} \frac{1}{(n_{\lambda} - 1)} \lambda_1 \dots\dots\dots(3.32)$$

式中的 RPC 是相對相位補償，L 代表分階數。而繞射效率以及訊噪比 (Signal Noise Ratio, SNR) 的定義如下：

$$\text{繞射效率 } \eta = \frac{\text{輸出光場中訊號區內的光通量}}{\text{輸入光場的光通量}} \times 100\% \dots\dots\dots(3.33)$$

$$SNR = \frac{\eta_1}{\sum_{i \neq 1} \eta_i} \dots \dots \dots (3.34)$$

而該元件各個波長的繞射效率以及 SNR 如表 3-3 所示：

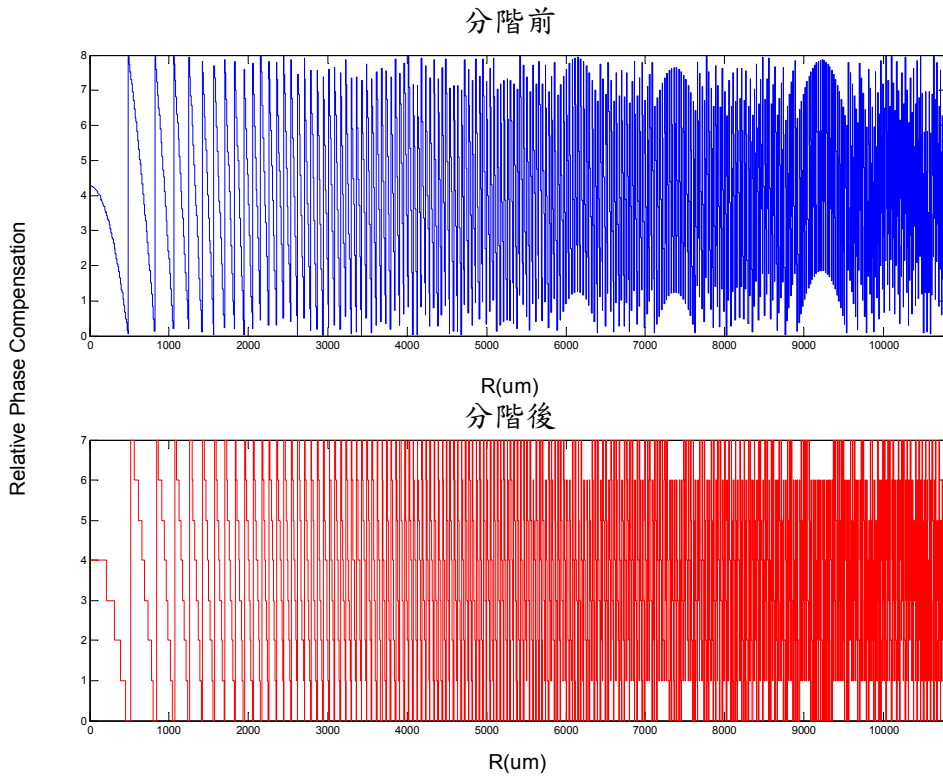


圖 3-6 按照表 3-2 規格的 Fresnel 透鏡

	分階前	分階後
波長(nm)	效率(%)	
1533	71.0821	68.2965
1541	4.73951	4.4929
1549	1.39251	1.31486
1557	0.552307	0.532997
SNR	10.6341	10.7710

表 3-3 元件對於各波長的繞射效率與訊噪比

此一元件並未經過優化，其 SNR 僅約為 10，顯示對於功能較為

複雜的元件，使用光線追跡法並不理想，在稍後的章節裡，將利用基因區域搜尋演算法來設計此一元件，以期能夠得到 SNR 較高的元件。

## 第 4 章 將基因區域搜尋法應用於繞射型濾波器的設計

利用演算法設計繞射型光學元件，首先最重要的就是定義出目標函數。目標函數的設定方式的好壞將會直接影響運算的結果，一個不恰當的設定，將會導致結果無法符合預期，本章將在第一小節裡討論幾種不同的設定方式，並比較其優缺點。在第二小節裡將介紹針對繞射光學元件所修改的基因演算法，而第三小節將介紹本論文所採用的編碼方式以及演算法交互迭代的方法。

### 4.1 目標函數(Object function)的設定與比較分析

由於許多最佳化演算法都是求函數的極小值，因此，在繞射光學元件的設計上，一般常見的目標函數設定都是採用誤差函數(Error function)，如平方誤差和(Sum Square Error, SSE)：

$$SSE(P) = \sum_i [I_i(P) - \tilde{I}_i]^2 \dots\dots\dots (4.1)$$

式中的  $P$  表示是演算法目前搜尋的位置(解)，而  $I_i(P)$  是訊號區內的光強度， $\tilde{I}_i$  是光強度的目標值。當  $SSE=0$  時，也就代表已經達到我們所要求的目標了，但這只是一種理想的狀況，實際上我們只能找到一個極小值而已，圖 4-1 為誤差函數示意圖。而基因演算法是一種求極大值的演算法，其目標函數設定方式跟一般求最小值的演算法可以利用簡單的取倒數或是取相反數的方式來完成：

$$f_{\min} = \frac{1}{F} \dots\dots\dots (4.2)$$

或

$$f_{\min} = -F \dots\dots\dots(4.3)$$

式中大寫的 F 表示基因演算法的適應度函數(求最大值)，而  $f_{\min}$  是共軛梯度法以及其他種求最小值最佳化演算法的目標函數。

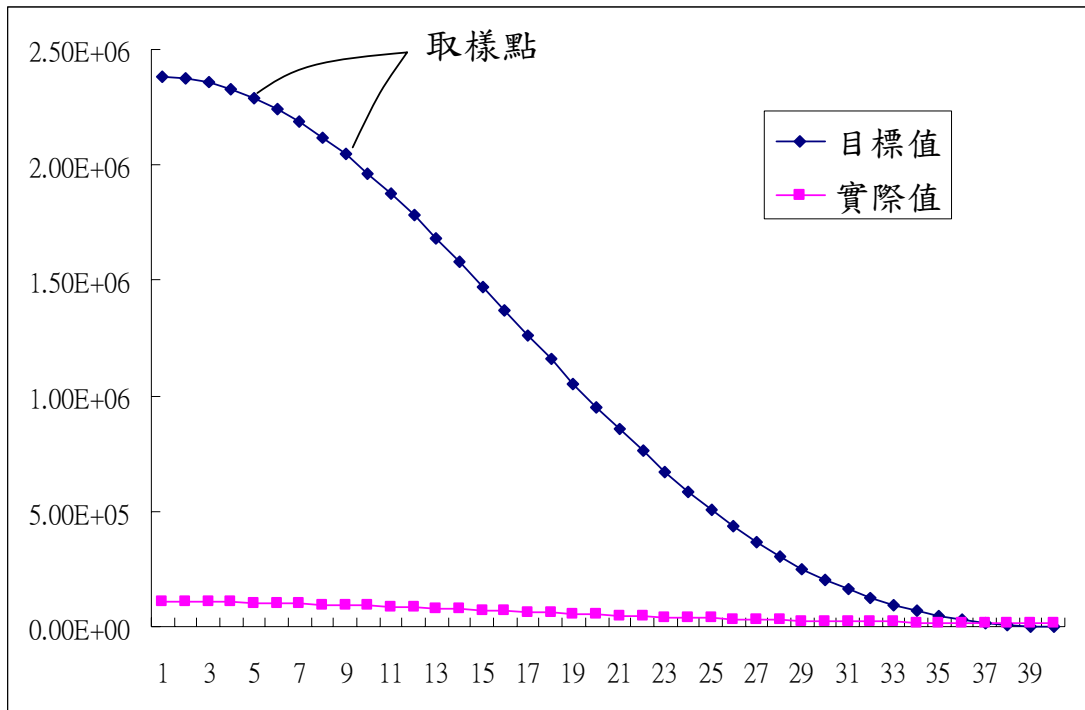


圖 4-1 誤差函數示意圖

濾波型元件有幾點要求：1. 主要波長具有高繞射效率 2. 具有高 SNR 值。因此希望主波長在訊號區的光強度分佈近似於理想的愛里盤 (理想聚焦元件的分佈)，而其他波長則希望在訊號區內強度為 0，因此可以寫成下列的式子：

$$SSE = \sum_i [I_{\lambda_i} - \tilde{I}_{\lambda_i}]^2 + \sum_i [I_{\lambda_{2i}} - 0]^2 + \sum_i [I_{\lambda_{3i}} - 0]^2 + \sum_i [I_{\lambda_{4i}} - 0]^2 \dots\dots(4.4)$$

式中的  $\tilde{I}_{\lambda_i}$  是愛里盤的強度分佈，而由於主波長的誤差跟其他波長的

誤差的大小會有所不同，所以習慣上會針對第  $n$  個波長加上一個誤差的加權(weight)。

$$SSE = \sum_i [I_{\lambda_i} - \tilde{I}_{\lambda_i}]^2 + \sum_i W_{\lambda_{2i}} I_{\lambda_{2i}}^2 + \sum_i W_{\lambda_{3i}} I_{\lambda_{3i}}^2 + \sum_i W_{\lambda_{4i}} I_{\lambda_{4i}}^2 \dots \dots \dots (4.5)$$

上式是一般普遍所使用的設定方法，然而由於主波長的光強度誤差跟其他光強度的誤差大小，往往相差數個數量級，往往需要經過不斷的測試，才能找出合適的加權值。當取樣點改變或是元件規格改變時，都必須要再調整加權值的大小。顯然這不是一種非常好的設定方式，因此本論文考慮基因演算法求最大值的特性，使用幾種不同的目標函數設定方式：

1. 求  $SNR$  最大值：

這個概念是來自於我們希望主波長具有在訊號區內有極高的光通量，而其他波長的光通量趨近於 0，也就是  $SNR$  極大的概念。考慮四個波長的計算，使用基因區域搜尋法，其計算結果為  $\eta_1 = 0.32051\%$ 、 $\eta_2 = 0.000012\%$ 、 $\eta_3 = 0.0000026\%$ 、 $\eta_4 = 0.00000072\%$ ， $SNR = 20921$ ，此一結果雖然有極高的  $SNR$  值，但主波長的繞射效率過低，並不符合我們希望得到的結果。

2. 求  $\eta_1 \times 100 + W \cdot SNR'$  或  $\eta_1 \times SNR^w$  極大：

單純考慮  $SNR$  並不是一個很好的方式，因此加入主波長繞射效率的因素進來，同時根據(3.34)式  $SNR$  的定義，為了避免其值會在分母

接近 0 時會發生趨近無窮大的現象，考慮在分母加入一個非 0 值，使其 SNR 是有限值

$$SNR' = \frac{\eta_1}{\eta_{nontrivial} + \sum_{i \neq 1} \eta_i} \dots\dots\dots(4.6)$$

$\eta_1 \times 100 + W \cdot SNR'$  與  $\eta_1 \times SNR'^W$  的差別在於一個是取加權平均數的最大值，另一個是取幾何平均數的最大值。 $\eta_1 \times 100 + W \cdot SNR'$  式中  $\eta_1$  乘上 100 的目的是為了讓目標函數有適當大小的值(well-scaled)，如此考慮設定  $\eta_{nontrivial} = 0.5\%$  與  $W = 0.5$ ，根據此種設定的方式使用基因區域搜尋法則可以得到  $\eta_1 = 57.1086\%$ 、 $SNR = 711.7223$ ，由此結果可以看出，此種設定方式相較於先前所介紹的 SSE 設定方式，是一種比較好的方法。

3. 求  $\eta_1 - W \sum_{i \neq 1} \eta_i$  極大：

這個設計方式的概念在於假若已知某個規格的繞射聚焦元件的各個波長的繞射效率，那我們可以用主要波長的聚焦效率換取其他波長聚焦效率的降低，以提升 SNR 值。如表 3-3，已知  $\eta_1 = 68.2965\%$ 、 $\eta_2 = 4.4929\%$ 、 $\eta_3 = 1.31486\%$ 、 $\eta_4 = 0.532997\%$ ， $\sum_{i \neq 1} \eta_i = 6.3408\%$ ，則在最糟糕的情況下，主波長的聚焦效率能夠以  $\frac{1}{W}$  的倍率交換其他波長的效率降低量，這便是此加權參數  $W$  所隱含物理意義。如此我們可以很容易制定出加權函數的大小，考慮主波長聚焦效率至少具有 60%，則可以



簡單估計  $W \rightarrow \frac{68.2965\% - 60\%}{6.3408\%} \approx 1$ 。然而實際上若將  $W$  設定為 1，並不能真正的把主波長的效率完全交換成其他波長效率和的降低，我們必須要增加  $W$  參數的值以放寬交換成立的條件，以順利的壓低其他波長的效率和， $W$  所隱含的另外一個物理概念便是寬鬆交換條件。因為參數  $W$  具有明確的物理概念，所以能夠用簡單的方式估計出範圍，具有容易設定的優點。考慮使用  $W=5$ ，經過基因區域搜尋法優化，則可以得到  $\eta_1 = 63.2527\%$ 、 $SNR = 85.7276$ ，此一結果主波長繞射效率仍然相當的高，而  $SNR$  值也高達 80 以上，符合我們的需求。

根據(4.2)與(4.3)式的轉換方法，將以上幾種基因演算法的目標函數轉換成共軛梯度法的目標函數，使得兩種不同的演算法具有相同的目標，如表 4-1：

設定方式		
概念	基因演算法(極大值)	共軛梯度法(極小值)
SSE	$F = \frac{1}{SSE}$	$f_{\min} = SSE$
SNR	$F = SNR$	$f_{\min} = -SNR$
$\eta_1 + W \cdot SNR'$	$F = \eta_1 + W \cdot SNR'$	$f_{\min} = \eta_1 + W \cdot SNR'$
$\eta_1 + SNR'^W$	$F = \eta_1 + SNR'^W$	$f_{\min} = -(\eta_1 + SNR'^W)$
$\eta_1 - W \sum_{i \neq 1} \eta_i$	$F = \eta_1 - W \sum_{i \neq 1} \eta_i$	$f_{\min} = W \sum_{i \neq 1} \eta_i - \eta_1$

表 4-1 各種不同的目標函數設定方式

## 4.2 改良型基因演算法<sup>12</sup>

在本章節裡，主要是針對繞射元件的特性，修改「交配」(Crossover)動作，以達到更好的收斂效果。純相位型繞射光學元件是利用相位補償調制入射光波的相位，使得光波通過元件之後在成像面處達到建設性或破壞性的干涉。首先，用一個簡單的 Fresnel 透鏡的例子來解釋為何要將基因演算法做如此的修改。如圖 4-2，左圖是一個標準的 Fresnel 透鏡結構，右邊是為了要使用半導體製程，而將此結構進行分階化(分成 8 階)後的結構。

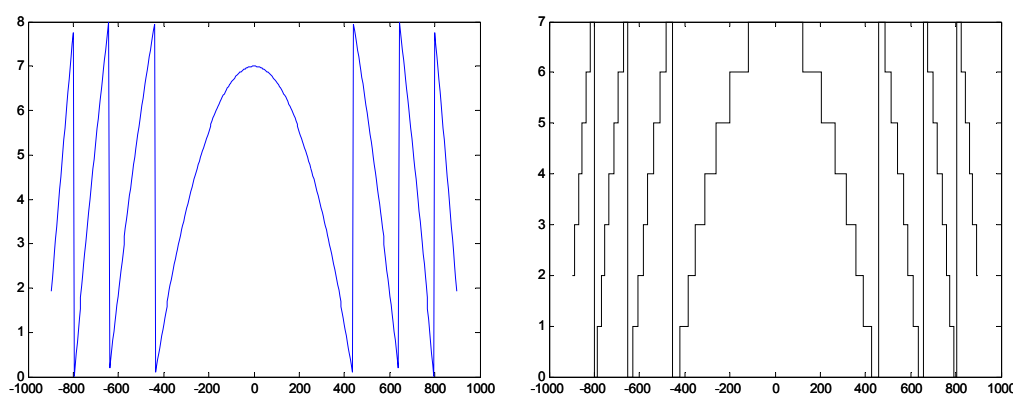


圖 4-2 分階前與分階後的 Fresnel 透鏡

在圖中，橫軸是繞射元件的徑向的部分，而縱軸代表著元件所必須進行的相位補償(也就是光程的補償)，其補償的範圍應該是 $[0, 2\pi)$ 。而在此，為了閱讀的方便以及使得與分階的結果一致，這裡將值域的範圍歸一化，使其範圍落於 $[0, 8)$ 之間。對於一個單一波長的元件來說，所有的相位補償都應該被限制在 $[0, 2\pi)$ 之間，因為在光學上，相位補

償超過 $2\pi$ 的部分都是可以忽略的，數學形式可以寫成 $\phi' = \phi \bmod 2\pi$ ，其中 mod 是取餘數的意思。在繞射型光學元件中，我們將相位的補償約束至 $[0, 2\pi)$ 的範圍內，其結構記錄的是相位補償的相對值(差值)，而不是一個絕對值。如下圖 4-3 所示，這八種結構在光學上是完全等效的，其中的差距僅是一個相位的常數。

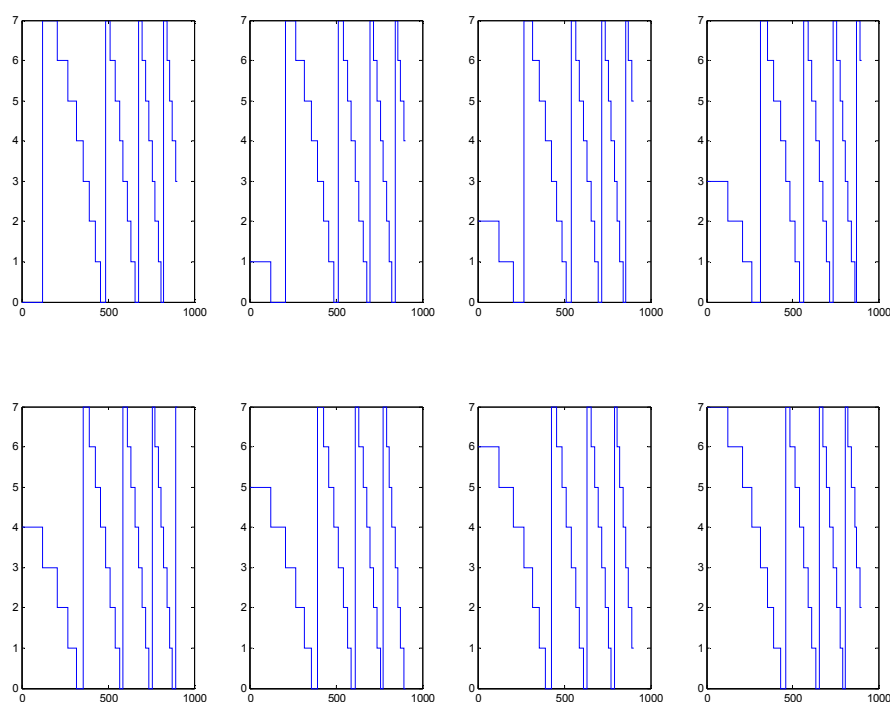


圖 4-3 八種等效的結構

如果把左上第一個圖全部加上 1，然後對 8 取餘數，則可以得到第二個圖形，而我們可以使用波的疊加的概念來的解釋繞射光學元件的光學特性與行為。在圖 4-3 中，左上方的第二個結構，可以當成是相位常數為 1 的結構圖(實際的相位常數應該是 $\frac{1}{8} \times 2\pi$ )，而右邊一個

結構的相位常數是 2，再右邊一個的相位常數是 3，依此類推。當光線通過這八個不同的結構時，全部將會聚焦到同一點上，強度也會完全相同，但是，相位卻不會相同，也就是第一個結構的相位是跟他的相位常數 0 有關，而第二個結構在聚焦點上，光的相位是跟他的相位常數 1 有關，同時我們也知道，這兩個元件聚焦點上光的相位會相差  $1$  (也就是  $\frac{1}{8} \times 2\pi$ )，如果將這兩個結構各取一半(結構一取左邊一半，結構二取右邊一半)之後拼在一起，所得到的新結構，其聚焦效果會不如原本的結構，這是因為這兩個結構的相位常數不相同，最後進行波的疊加時，反而不能達到良好的建設性疊加。在瞭解了光學上特殊的性質之後，我們開始進入基因演算法的正題，考慮圖 4-2 是一個 Fresnel 透鏡結構，是繞射行純聚焦元件的最佳解，也就是我們使用基因演算法，想要得到的解。

而基因演算法中，每一個染色體個體代表一個結構，而會有多個染色體個體形成一個族群，最後再由此一族群進行交配、突變等演化的行為，最後演化出一個最好的解。在演化的過程中，優秀的個體容易被挑選出來(被挑選的機率正比於該染色體的是否優秀的評估值)，因此我們考慮在一個廣大的族群中，就在演化的過程中出現兩組較優秀的個體如圖 4-4 所示：

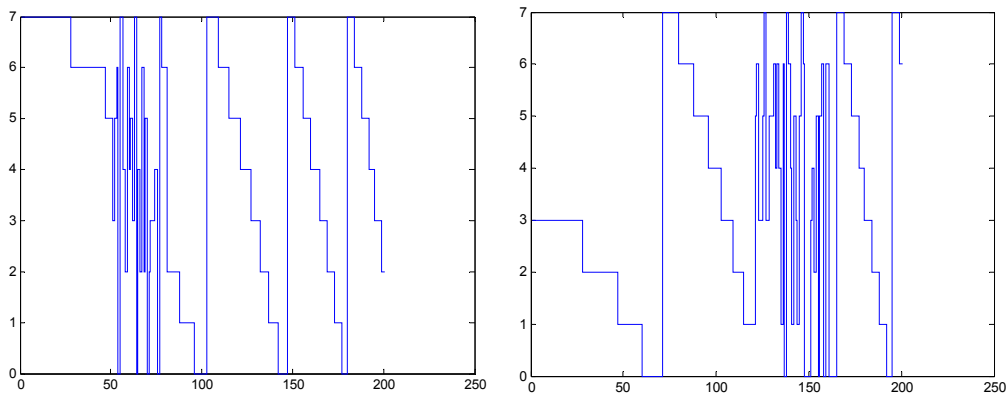


圖 4-4 整個族群中兩個較優秀的個體

因為此兩組染色體都十分優秀(由圖可知，已經蠻接近 Fresnel 透鏡了)，因此極有可能被挑選出來互相交配。考慮其交配的切割點(單切割點交配法)是落在橫軸 100 的位置，左邊那個結構，取的是 100 以後的基因，又邊那個結構，取的是 100 以前的基因，如此我們得到一個新的染色體個體，照理說，此組染色體遺傳自兩個相當優秀的個體，這組染色體應該也會有相當好的表現才是，但事實卻完全相反，此組染色體所代表的結構，其聚焦效果將會十分的差，原不如親代的結構的聚焦效果。其原因可以簡單的看出，圖 4-4 左邊圖形，其結構的相位常數大致可以看成是 7(對於 Fresnel 透鏡來說)，而右邊圖形的相位常數卻是 3，兩者之間差了 4(也就是  $\frac{4}{8} \times 2\pi = \pi$ )，因此反而造成波的破壞性的疊加了。當我們瞭解問題就出在相位常數上時，為了解決這個問題，我們必須將兩個結構的相位常數差消去。而我們知道只有當發生破壞性干涉時(相位常數相差  $\frac{\pi}{2} \sim \frac{3\pi}{2}$  時)，才會造

成子代比親代還差，因此提出一個簡單的方法，也就是當發生子代比兩個親代還要差的時候，考慮補償  $\frac{\pi}{2} \sim \frac{3\pi}{2}$  的相位常數，就是將右邊那個結構，考慮相位常數，因此可以得到 4 個結構(分 8 階，故  $\frac{\pi}{2} \sim \frac{3\pi}{2}$  具有 4 種結構)，此 4 個結構(或稱光學性質等效的 4 組染色體)再與左邊那個結構(染色體)交配，形成 4 個子代，最後由此 4 個子代挑選出最好的，當成新一代。在之前所提到的這個例子中，如果改採這種修改型的交配方法，我們馬上就可以得到當相位常數改變 4(就是相位常數相差變成 0)，將會與左邊那組交配出最好的結果，而此結果就會馬上與 Fresnel 透鏡完全相同。當然，或許這僅僅是一個方便解釋的例子，但是，這也是實際演化的時候，極有可能碰到的問題，當我們改採此種修改型交配法時，可以很輕鬆的克服此一嚴重的問題。

假若整個族群中不存在相位常數的問題，而剛才所介紹的方法需要額外測試相位的常數，勢必浪費許多運算，因此考慮使用統計的方式，假若整個族群大小為  $N_{pop}$ ，由第  $t$  代產生第  $t+1$  個子代時，如果有  $rN_{pop}$  個子代需要使用修改型交配的方法，則認定此一族群存在相位常數的問題，在  $t+1$  代裡仍然使用修改型的交配方法，反之若需要修改型交配法的子代數目小於  $rN_{pop}$ ，則認定此族群不存在相位常數的， $t+1$  代之後將只使用一般的交配方法，其中  $r$  是一個比例常數，本論文的設定是 0.5，以下圖 4-5 是利用修改型演算法與一般演算法設計

一個使用波長為  $1\ \mu\text{m}$ ，元件大小  $1500\ \mu\text{m}$ ，300 個分割點，焦距為  $40000\ \mu\text{m}$ ，光斑半徑為  $16.26670\ \mu\text{m}$ ，其目標函數為愛里盤的繞射效率，其收斂速度的比較曲線圖。

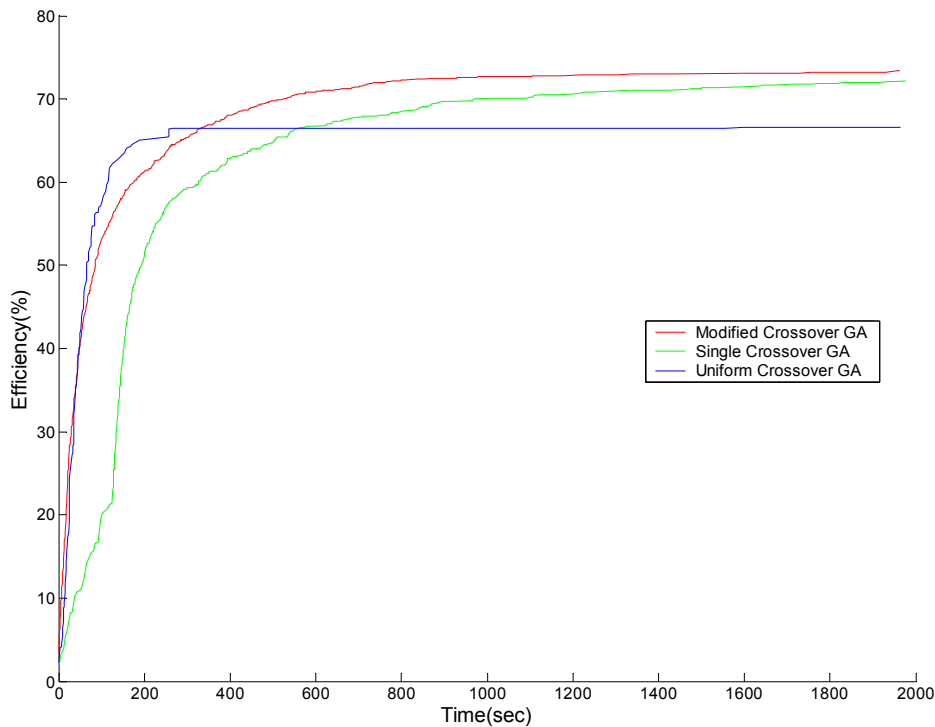


圖 4-5 不同的交配方法隨時間收斂的曲線

由圖 4-5 可以看出，均勻交配法(藍色曲線)的收斂速度雖然快，但卻容易發生收斂過早的現象，而單切點交配(綠色曲線)的收斂速度不如針對繞射元件改良的交配方法(紅色曲線)。本論文中均採用此種改良型的交配方法。

#### 4.3 編碼以及兩種演算法交互迭代的方式

本論文所設計的元件規格如表 3-2 所示，採取二進階制，共分 8

階，所以相位的補償總共有  $\left\{ \frac{0}{4}\pi, \frac{1}{4}\pi, \frac{2}{4}\pi, \frac{3}{4}\pi, \frac{4}{4}\pi, \frac{5}{4}\pi, \frac{6}{4}\pi, \frac{7}{4}\pi \right\}$ ，在基因演算法中使用整數編碼法，以 0~7 表示  $0 \sim \frac{7}{4}\pi$  相位的補償。

在共軛梯度法中，為了配合基因演算法，限定相位的補償必須要在  $[0, 8)$  之內，然而共軛梯度法是一種無約束最佳化演算法，所以必須要引入一個約束條件，使的相位的補償最後能夠被限制在  $[0, 8)$  之內。

假若限制範圍是  $[0, L]$ ，經常使用以下轉換方法：

$$\begin{aligned} \bar{X}_c &= \frac{L}{2} \sin(\bar{X}) + \frac{L}{2} \\ &= \left( \frac{L}{2} \sin(X_1) + \frac{L}{2}, \frac{L}{2} \sin(X_2) + \frac{L}{2}, \dots, \frac{L}{2} \sin(X_n) + \frac{L}{2} \right) \end{aligned} \quad \dots\dots\dots(4.7)$$

如此便能夠使得轉換出來的值被約束在  $[0, L]$  的範圍內了，不過對於以導數資訊為基礎的演算法而言，這種轉換方式在某些情況下可能會出問題，假設最小化的函數為  $f_{\min}(\bar{X}_c)$ ，則

$$\begin{aligned} \bar{\nabla} f(\bar{X}_c) &= \left( \frac{\partial f}{\partial X_{c1}} \frac{dX_{c1}}{dX_1}, \frac{\partial f}{\partial X_{c2}} \frac{dX_{c2}}{dX_2}, \dots, \frac{\partial f}{\partial X_{cn}} \frac{dX_{cn}}{dX_n} \right) \\ &= \left( \frac{L}{2} \frac{\partial f}{\partial X_{c1}} \cos(X_1), \frac{L}{2} \frac{\partial f}{\partial X_{c2}} \cos(X_2), \dots, \frac{L}{2} \frac{\partial f}{\partial X_{cn}} \cos(X_n) \right) \end{aligned} \quad \dots\dots\dots(4.8)$$

如此，當  $X$  向量的第  $k$  個元素，滿足  $X_k = \frac{\pi}{2} + n\pi$  條件時(其中  $n$  為整數)，由(4.8)式可以清楚的知道，梯度的第  $k$  個元素會為 0，而停止該方向的移動。且此種轉換方式需要使用到計算較為複雜的正弦函數，因此本論文提出另外一種轉換方法，如下所示：

$$\bar{X}_c = MOD(\bar{X}, L) = \bar{X} - \left\lfloor \frac{\bar{X}}{L} \right\rfloor \cdot L \quad \dots\dots\dots(4.9)$$



式中的[]符號是高斯算符。雖然(4.9)的轉換看起來是不連續的，但 $X$ 是表示相位的補償，以相位的角度而言，超過一個週期便開始重複，因此這個轉換對於相位而言，仍然是連續的。與(4.7)的轉換方法相較，此一轉換方法僅需幾個簡單的四則運算便可完成，在計算速度有著明顯的優勢，本論文採用此種約束的方法。

基因演算法計算的直接就是分階後的結構，而共軛梯度法所計算是分階前的結構。所以當共軛梯度法收斂要把計算結果交給基因演算法之前必須做分階量化的動作，其轉換的方法如下：

$$\vec{X}_L = [\text{MOD}(\vec{X}_c + 0.5, L)] \cdots \cdots \cdots (4.10)$$

如此便能夠將共軛梯度法所求得的區域最佳解傳遞給基因演算法。接著討論基因演算法與共軛梯度法相互迭代的方法：基因演算法是一種收斂緩慢的全域型最佳化法，而共軛梯度法是一種收斂快速但卻可能落入區域最佳解陷阱的演算法，在結合此兩種演算法時，究竟應該先跑哪一種演算法呢？如果目標函數本身存在多個區域最佳解，那先跑共軛梯度法將會落入區域最佳解的陷阱裡，在交給基因演算法時，必須要依靠良好的突變機制才能跳脫此一區域最佳解，然而突變機制可視為一種隨機搜尋的方式，是必須要反覆大量的測試(try & error)才可以跳脫出此一區域最佳解，顯然的，這會需要龐大的計算量，此時，建議先從基因演算法開始，一開始就是全域搜索最佳解，當族群

開始收斂時，再挑出最佳的染色體，然後交給共軛梯度法進行快速的計算而求得全域最佳解。反之，如果目標函數本身僅有一個最佳解(具有單一峰值)，則先跑基因演算法就不是一個明智的作法，其原因是基因演算法的收斂速度跟共軛梯度法相比，實在是太慢了。但我們如何知道目標函數究竟是屬於哪一種的函數呢？解決的方法就是可以先用共軛梯度法測試，因為共軛梯度法收斂的速度遠快於基因演算法，所以多跑幾次也沒有關係，每次跑共軛梯度法時設定不同的初值(建議用隨機亂數選取)，如果每次計算出來的結果都十分相近(如同4.2節所述，必須扣除相位常數再進行比較)，那表示即使這個目標函數具有多個區域最佳解，其全域最佳解也與共軛梯度法所求得的解相差不遠了(也就是多個極值都集中在某處)，如圖4-6所示，

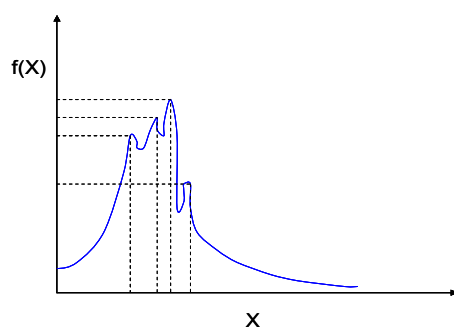


圖 4-6 多個極大值集中在一起示意圖

此時應該先跑共軛梯度法，然後再交由基因演算法搜尋全域最佳解。反之，如果每次使用不同的初值所求得的結果相差甚遠，此時就需要先執行基因演算法，搜尋全域最佳解，等族群發生集中於某處的現象時，再利用共軛梯度法快速收斂至最佳解。圖4-7是基因區域搜尋演

算法的流程圖。

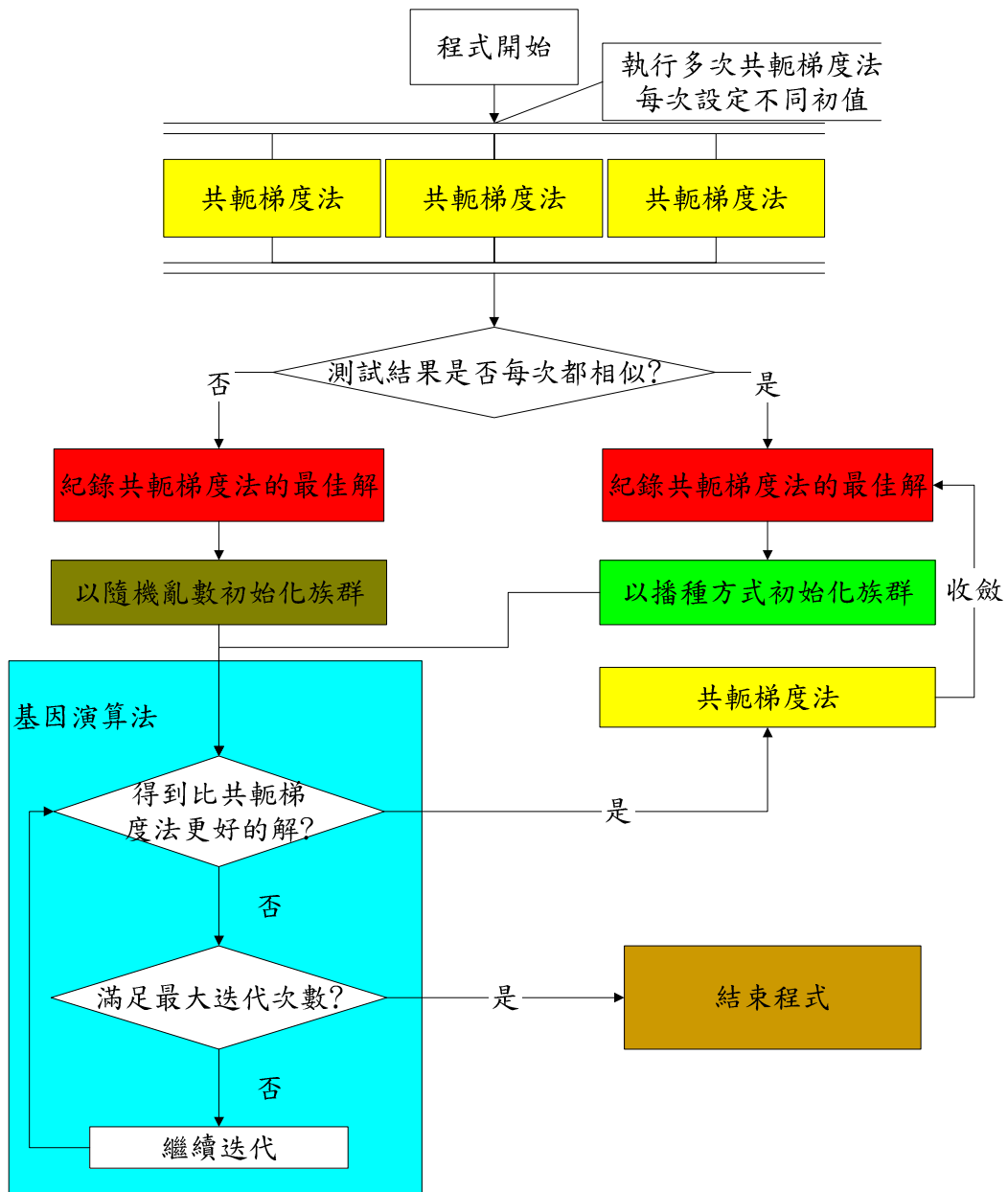


圖 4-7 GLSA 流程圖

## 第 5 章 結論

### 5.1 模擬結果與討論

以下列出此次模擬的電腦運算平台：

中央處理器	Intel Pentium 4 1600MHz
第二階快取記憶體容量	512KB
主記憶體	1024MB DDR SDRAM
作業系統	Microsoft Windows XP SP1

表 5-1 高速個人電腦運算平台之規格

而元件規格以及所使用的參數列表如下：

元件半徑 $R(\mu\text{m})$	10800
分割點數	1800
最小線寬 $(\mu\text{m})$	6
輸入光主要波長 $\lambda_1(\text{nm})$	1533
輸入光次要波長 $\lambda_2$ 、 $\lambda_3$ 、 $\lambda_4(\text{nm})$	1541、1549、1557
材料以及折射率	Si, 3.479(針對波長 1533nm 而言)
焦距 $f(\mu\text{m})$	144000
輸出面訊號區半徑 $r(\mu\text{m})$	12.4684
目標函數的設定方法 1	$F_{\max} = (\eta_1 - W \sum_{i=2,3,4} \eta_i) \times 100$ $f_{\min} = (W \sum_{i=2,3,4} \eta_i - \eta_1) \times 100, W = 5$
目標函數的設定方法 2	$F_{\max} = \eta_1 \times 100 + W \frac{\eta_1}{\eta_{\text{nontrival}} + \sum_{i=2,3,4} \eta_i}$ $f_{\min} = -\eta_1 \times 100 - W \frac{\eta_1}{\eta_{\text{nontrival}} + \sum_{i=2,3,4} \eta_i},$ $W = 0.5, \eta_{\text{nontrival}} = 0.005$
基因演算法中，族群的大小(個)	8000

表 5-2 元件規格參數表

執行基因區域搜尋法最佳化的結果如下所示：

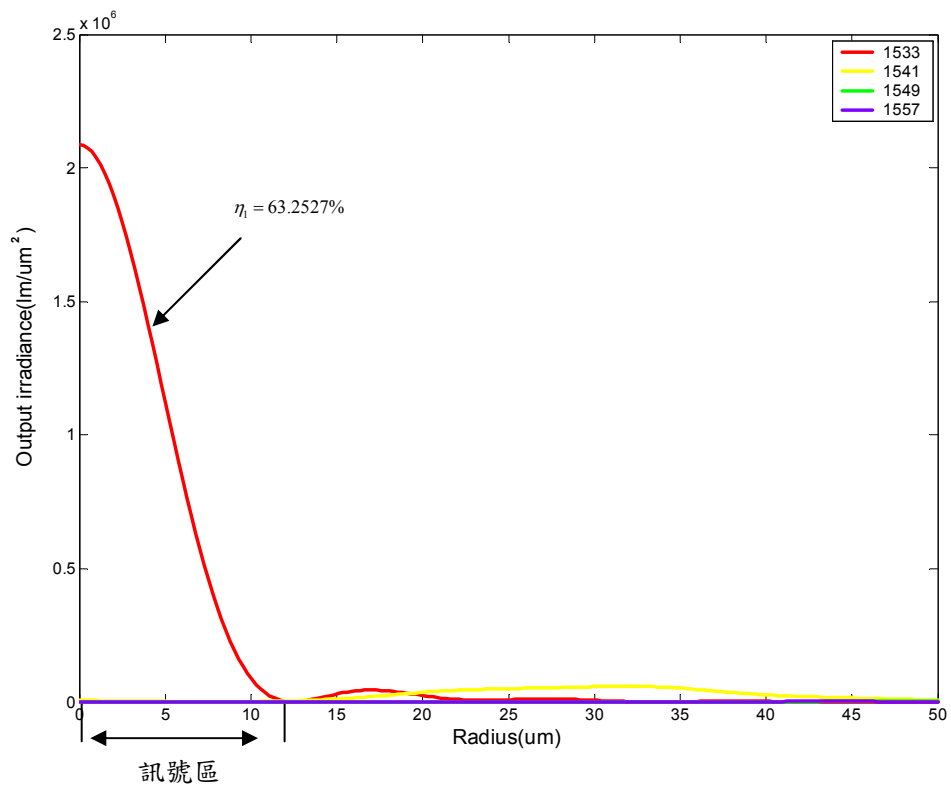


圖 5-1 使用目標函數 1，基因區域搜尋法計算的結果

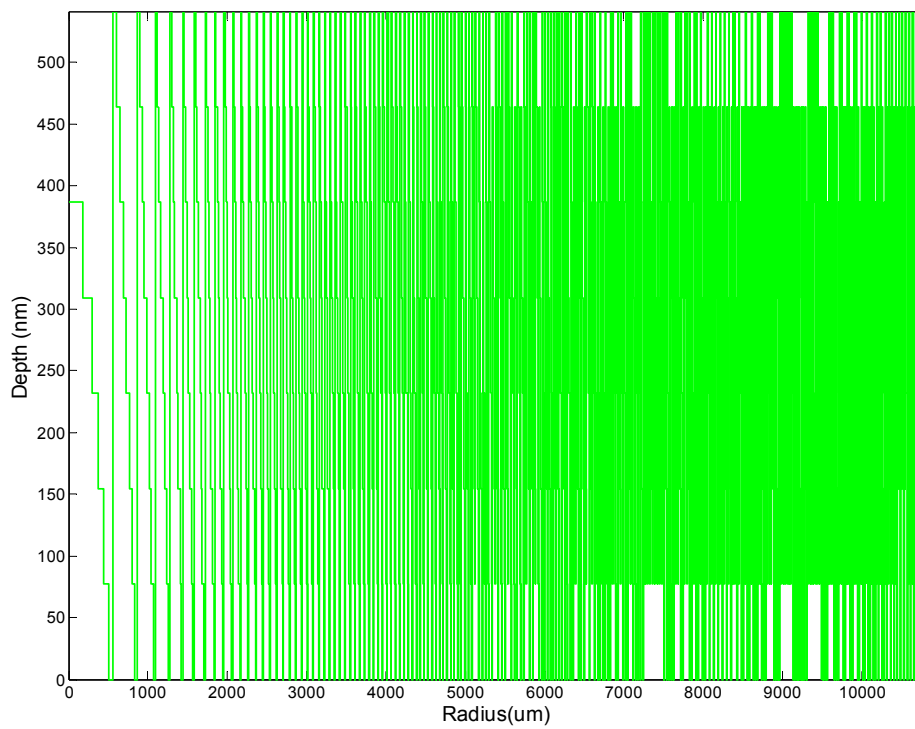


圖 5-2 使用目標函數 1，所優化出來的表面輪廓圖

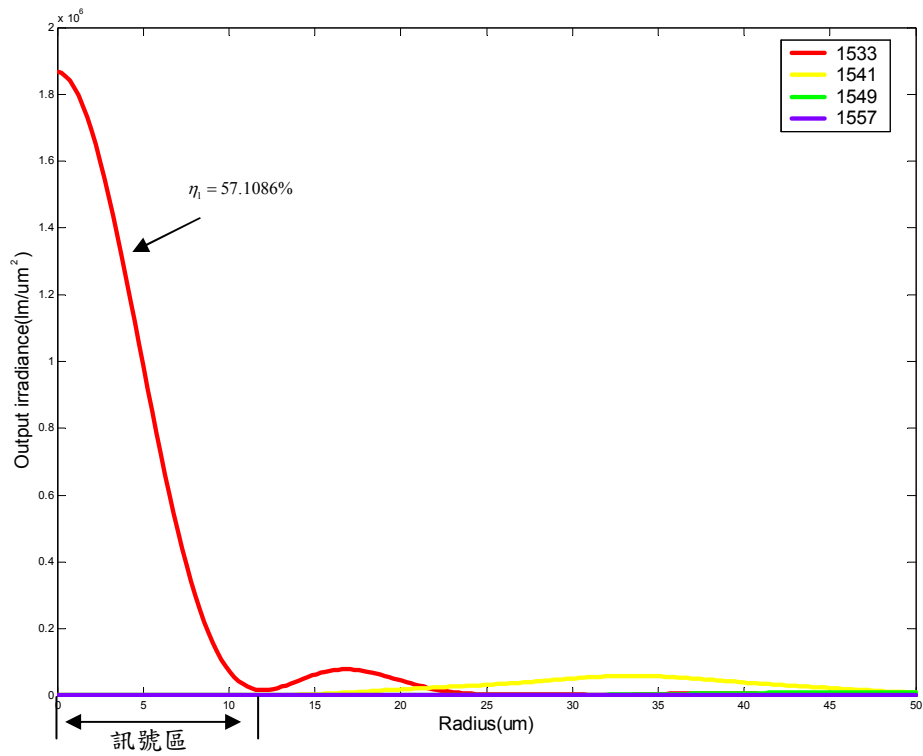


圖 5-3 使用目標函數 2，基因區域搜尋演算法的計算結果

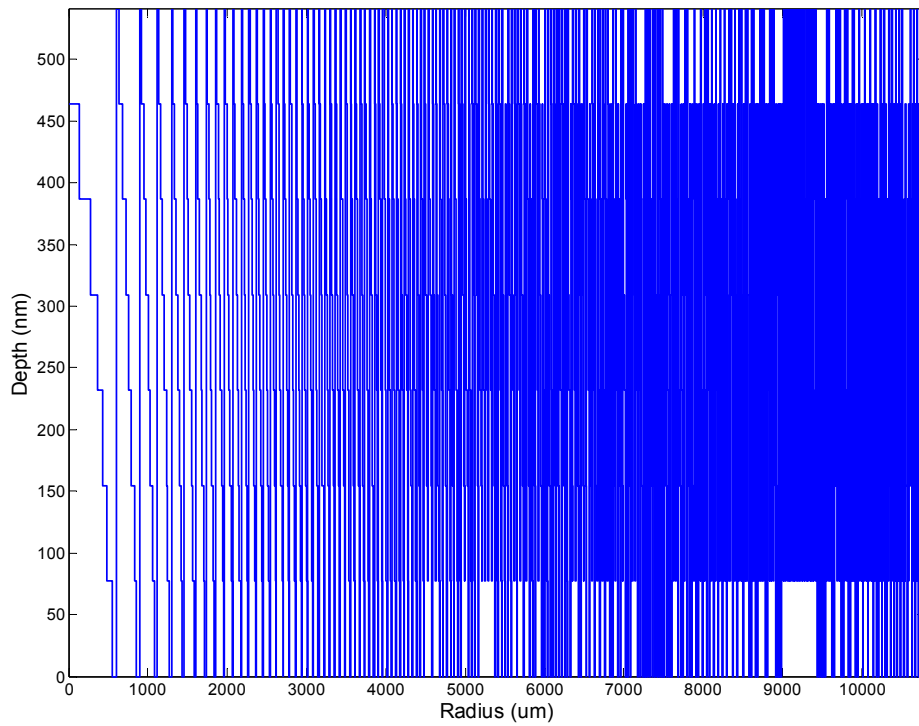


圖 5-4 使用目標函數 2，所優化出來的表面輪廓圖

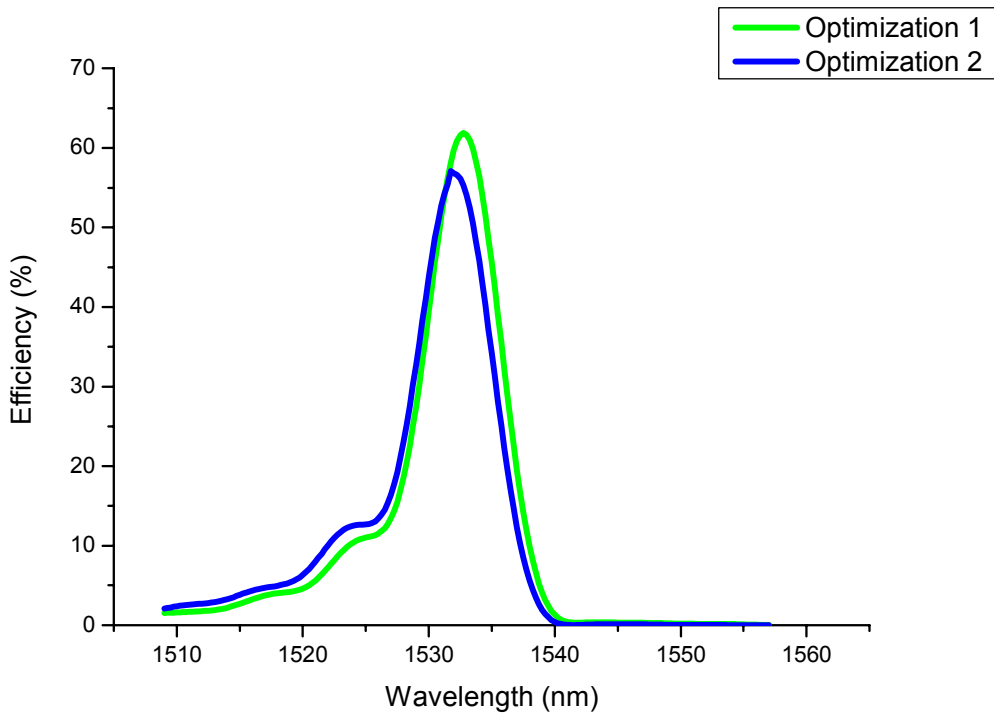


圖 5-5 光場譜線分佈圖

波長(nm)	優化後訊號區內的效率(%)		優化前訊號區內的效率(%)
	目標函數 1	目標函數 2	
1533	63.2527	57.1086	68.2965
1541	0.4424	0.03732	4.4929
1549	0.253919	0.02413	1.31486
1557	0.0414383	0.01879	0.532997
訊雜比			
	85.7276	711.7223	10.7710

表 5-3 按照規格設計的 Fresnel 透鏡(優化前)以及經過基因區域搜尋演算法優化得到的濾波元件的比較表

由表 5-3 可知，經過基因區域搜尋演算法優化之後，主波長 1533nm 的效率降低約 5%~10%，但 SNR 卻可以大幅的提升，符合濾波元件設計的要求。

接著比較基因區域搜尋法、共軛梯度法、以及基因演算法，如下

所示：

	基因區域搜尋演算法	共軛梯度法		基因演算法
最大迭代次數*	2000	**		8000
主波長效率 $\eta_1$ (%)	63.2527	分階前	分階後	31.2815
		65.1115	61.6841	
SNR	85.7276	84.5724	84.8351	57.4135
$(\eta_1 - 5 \sum_{i=2,3,4} \eta_i) \times 100$	59.5639	61.2620	58.0486	28.5573
所花費的時間	366217 秒 (約 4 天 6 小時)	15091 秒 (約 4 小時)		514823 秒 (約 7 天) (尚未收斂)

表 5-4 目標函數 1，基因區域搜尋法與共軛梯度法、基因演算法的比較

\*基因演算法必須設定最大的迭代次數(最多世代交替幾次)

\*\*共軛梯度法收斂的判據是「梯度的模(norm) <  $\epsilon$ 」，其中  $\epsilon$  是由我們給定的值

	基因區域搜尋演算法	共軛梯度法		基因演算法
最大迭代次數*	2000	**		8000
主波長效率 $\eta_1$ (%)	57.1086	分階前	分階後	34.237
		30.427	27.501	
SNR	711.7223	210.8633	219.8697	235.9893
$\eta_1 \times 100 + 0.5 \times \frac{\eta_1}{0.005 + \sum_{i=2,3,4} \eta_i}$	106.3198	54.0395	52.6631	60.7741
所花費的時間	305026 秒 (約 3.5 天)	11151 秒 (約 3 小時)		653948 秒 (約 7.5 天) (尚未收斂)

表 5-5 目標函數 2，基因區域搜尋法與共軛梯度法、基因演算法的比較

由表 5-4 與 5-5 可以知道，基因區域搜尋演算法的速度遠較傳統一般的基因演算法來得快速，在本次的模擬中，使用基因區域搜尋法在 5 天內已經求得全域最佳解，而傳統的基因演算法執行 7 天後卻仍然還未收斂至最佳解。



當我們採用第一種目標函數的設定，使用共軛梯度法所求得未分階之前具有比基因區域搜尋法更好的解，但實際製造上時必須進行分階的動作，而分階後共軛梯度法的解比基因區域搜尋法來得差，這是因為我們採用(4.10)式進行分階的運算，該運算是最簡單的四捨五入的方法，然而四捨五入之後的結構卻不一定是分階後最佳解，此時使用基因演算法仍然可以得到一些改進( $GLSA = CG + GA$ )。從這點來看，表示此一目標函數，是個單一極值的函數，如此，全域型演算法跟區域型演算法將會得到同一個解。

而當我們採用第二種目標函數的設定，使用共軛梯度法所求的解遠不如使用基因演算法以及基因區域演算法的結果好，這顯示共軛梯度法已經落入了區域最佳解的陷阱裡了。事實上，並不是所有問題都必須交給基因演算法，如果確定共軛梯度法已經能夠求得全域最佳解時，使用基因演算法無疑是浪費時間。然而由於我們並不清楚目標函數究竟是屬於僅有單一峰值還是具有多個峰值，我們在跑全域型最佳化演算法之前，我們永遠無法確定所求得的解究竟是全域的最佳解還是區域的最佳解，因此全域型最佳化演算法是有其存在的必要的。使用區域型演算法時可以適當的選擇不同的目標函數，以避免發生落於區域最佳解的陷阱，但是如果過於遷就挑選單一最佳解的目標函數，有時反而無法求得符合我們真實要求的條件。使用基因區域搜尋演算

法則可以不必理會目標函數會有區域最佳解陷阱的這種問題，可以將注意力完全的集中在符合我們需求的條件上。綜合以上幾點可以知道基因區域搜尋演算法是一種相當優良的演算法，所耗費的時間雖然比區域型最佳化演算法多，但卻遠少於一般全域型的最佳化演算法，同時能夠求得全域最佳解。

在圖 5-5 中，可以清楚的看到優化後得到的濾波元件，對於 1541~1557nm 的波長，有極佳濾波效果，但對於 1509~1527nm 的波段，其濾波的效果並不好。會造成這種結果的原因是來自於目標函數的設定，也就是我們一開始並未考慮到 1509~1527nm 的部分，所以在優化的過程中，並未能降低 1509~1527nm 波段的聚焦效率。如果今天希望設計一個能夠濾掉 1509~1527nm 以及 1541~1557nm 兩個波段的元件，則僅需要再把 1509~1527nm 的波段的影響加入原始的目標函數即可。

因此現在考慮一個 5 個波長的最佳化計算，其波長分別為  $\lambda_1 = 1517nm$ 、 $\lambda_2 = 1525nm$ 、 $\lambda_3 = 1533nm$ （主波長）、 $\lambda_4 = 1541nm$ 、 $\lambda_5 = 1549nm$ ，其使用的目標函數為  $F_{Max} = \eta_3 \times 100 + 2 \times \frac{\eta_3}{0.3\% + \sum_{i=1,2,4,5} \eta_i}$ （以下稱為目標函數 3），其他參數與表 5-2 相同，進行最佳化運算，得到的結果列於表 5-6。

目標函數 3		
波長(nm)	優化後訊號區內的效率(%)	優化前訊號區內的效率(%)
1517	0.0206637	1.37243
1525	0.0819142	4.64941
1533	25.7481	68.2965
1541	0.0675251	4.4929
1549	0.0422205	1.31486
訊雜比		
	121.2683	5.7734

表 5-6 以目標函數 3，針對 1517~1549nm 波段設計的元件，經基因區域搜尋演算法優化後與該規格下的 Fresnel 透鏡(優化前)之比較表

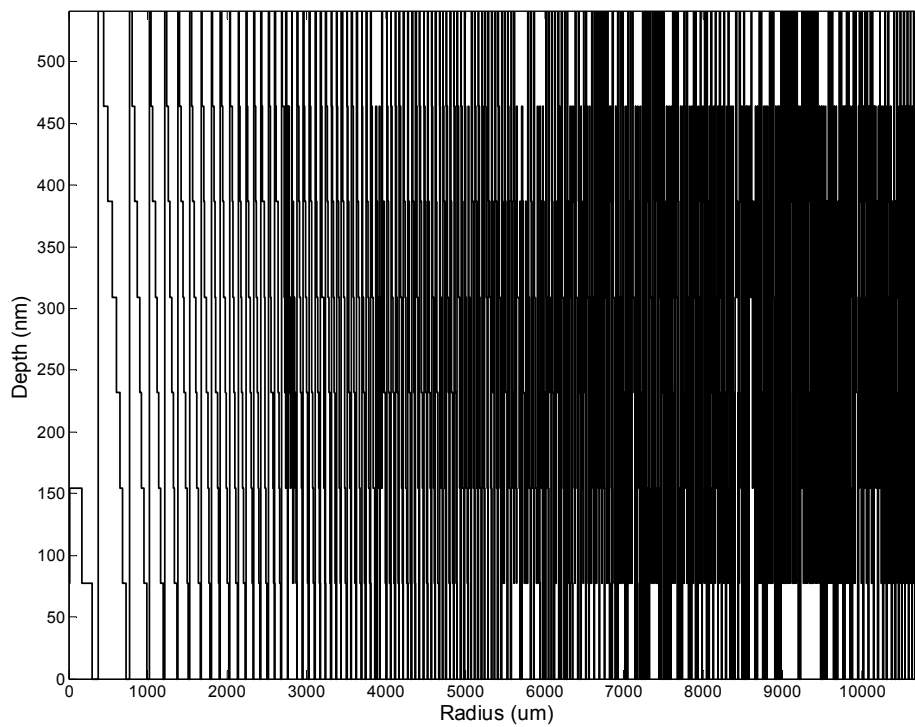


圖 5-6 使用目標函數 3，所優化出來的表面輪廓

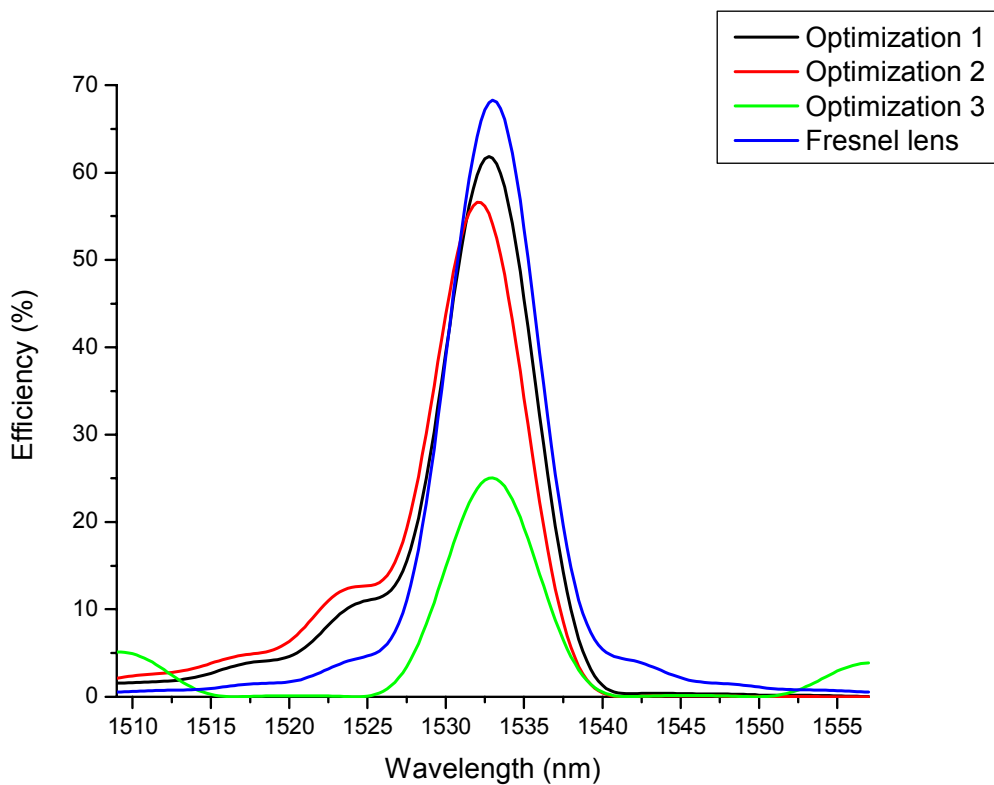


圖 5-7 三種不同的最佳化的光場譜線分佈比較圖

由圖 5-7 可知，本論文所發展的最佳化演算法依然會找出該目標函數的最佳解，適當的把 1517~1525nm、1541~1549nm 等波段給濾掉了，比起前述的兩種最佳解，具有更窄的頻譜半高寬，符合我們目標函數的設定要求，但是在新的最佳解中，主波長 1533nm 的聚焦效率降低了，其原因是來自於我們加入了 1517~1525nm 波段的考慮，使得限制的條件變多，所以主波長效率的降低是種必然的結果。因此，實際上我們在設計元件時，必須謹慎適當地考慮所需使用到的波長及條件，不需要用到的波長儘量不要放入目標函數中，如此便可以提升元件的性能。

## 5.2 未來展望

本論文所發展的基因區域搜尋法，是與精密儀器中心合作的計畫，主旨是在發展一種通用的快速型全域最佳化演算法平台，未來希望能夠應用在包含凹面光柵的參數最佳化、長焦深的 CD/DVD 讀寫頭透鏡設計、多色光學系統的參數最佳化、波導參數的最佳化設計以及各種參數的最佳化問題上。

在本論文裡，已經成功的利用基因區域搜尋法設計出具有高效率、高訊雜比的繞射型濾波元件。而此一窄波段濾波元件是以繞射的方式來達成，這是一種比較新式的方法，若我們的設計能夠進一步縮小其譜線帶寬，未來便可期望設計成波長解多工器，運用在光纖通訊上。

## 附錄 A F-R 共軛梯度法

在多維空間 (multi-dimension) 中我們使用一種稱作共軛方向法 (conjugate direction method) 來處理無約束最佳化問題，此演算法可以決定出最快速的搜尋方向  $Z$ ，共軛方向法的定理說明如下：  
 $f$  為二次式函數，若搜尋方向  $Z^{(k)}$  都互相個別與二次式 (quadratic,  $f(x) = a + b^T X + \frac{1}{2} X^T Q X$ ) 函數中的正定矩陣  $Q$  共軛正交，其中  $k = 1, 2, 3, \dots, n$ ，則到達之位置  $X^{(n)}$  將是二次式函數  $f$  的精確最小值。接下來我們將介紹如何利用共軛方向法來尋找  $Z$  之方向。

$$Z^{(1)} = -\nabla f(X^{(0)}) \dots \dots \dots (A. 1)$$

所以下一步之位置應該為

$$X^{(1)} = X^{(0)} + \alpha_1 Z^{(1)} \dots \dots \dots (A. 2)$$

接著求  $\nabla f(X^{(1)})$  的數值，並取

$$Z^{(2)} = -\nabla f(X^{(1)}) + \beta_{11} Z^{(1)} \dots \dots \dots (A. 3)$$

其中  $\beta_{11}$  是一個被選定的數值，使得  $Z^{(1)}$  與  $Z^{(2)}$  會分別對矩陣  $Q$  共軛正交，即  $[Z^{(1)}]^T Q Z^{(2)} = 0$ 。可以得到：

$$\begin{aligned} [Z^{(1)}]^T Q Z^{(2)} &= [Z^{(1)}]^T Q [-\nabla f(X^{(1)}) + \beta_{11} Z^{(1)}] \\ &= -[Z^{(1)}]^T Q \nabla f(X^{(1)}) + \beta_{11} [Z^{(1)}]^T Q Z^{(1)} = 0 \\ \Rightarrow \beta_{11} &= \frac{[Z^{(1)}]^T Q \nabla f(X^{(1)})}{[Z^{(1)}]^T Q Z^{(1)}} \dots \dots \dots (A. 4) \end{aligned}$$

現在再由  $X^{(1)}$  開始沿著  $Z^{(2)}$  移動到新的位置  $X^{(2)}$ ，且計算  $\nabla f(X^{(2)})$ 。

同樣地，新的方向  $Z^{(3)}$  必須與  $Z^{(1)}$ 、 $Z^{(2)}$  對矩陣  $Q$  共軛正交， $Z^{(3)}$  可以表示為：

$$Z^{(3)} = -\nabla f(X^{(2)}) + \beta_{21}Z^{(1)} + \beta_{22}Z^{(2)} \dots\dots\dots(A. 5)$$

其中  $\beta_{21}$  與  $\beta_{22}$  的選取必須使得  $[Z^{(1)}]^T QZ^{(3)} = [Z^{(2)}]^T QZ^{(3)} = 0$ 。所以，我們可以得到一般的表示式：

$$Z^{(k+1)} = -\nabla f(X^k) + \sum_{j=1}^k \beta_{kj} Z^{(j)}, k=0,1,2,\dots,n-1 \dots\dots\dots(A. 6)$$

由於上式 (A. 6) 中之係數  $\beta_{kj}$  是矩陣  $Q$  的函數，要求解這種二次式 Hessian 矩陣是一件困難的事。因此，接下來我們要介紹的是 1964 年 Fletcher、Reeves 二人利用 Hestenes、Stiefel 於 1952 年所提出求解線性系統之共軛梯度法為基礎，所發展出特別適用於二次式函數的無約束最小化演算法 (unconstrained minimization algorithm)，透過此方法我們在尋找這些方向時並無需求解矩陣  $Q$ 。敘述如下：假設函數  $f$  是一個二次式 (quadratic) 函數，並假設有  $m$  個非零向量  $Z^{(1)}, Z^{(2)}, Z^{(3)}, \dots, Z^{(m)}$  且  $Z^{(j)} \in R^{(n)}$ ，其中  $m \leq n$ ，這些向量彼此互相各別與矩陣  $Q$  共軛正交。我們從  $X^{(0)} \in R^{(n)}$  開始，沿著方向  $Z^{(1)}, Z^{(2)}, Z^{(3)}, \dots, Z^{(m)}$  分別移動到  $X^{(1)}, X^{(2)}, X^{(3)}, \dots, X^{(m)}$ ，使得：

$$[Z^{(j)}]^T \nabla f(X^j) = 0, j=1,2,3,\dots,m \dots\dots\dots(A. 7)$$

則

$$[Z^{(j)}]^T \nabla f(X^m) = 0, j=1,2,3,\dots,m \dots\dots\dots(A. 8)$$

證明如下：

$$X^{(j)} = X^{(j-1)} + \alpha_j' Z^{(j)}, j = 1, 2, 3, \dots, m$$

$$\nabla f(X^{(m)}) = \nabla f(X^{(j)}) + Q[X^{(m)} - X^{(j)}], j = 0, 1, 2, \dots, m-1 \dots\dots\dots (A. 9)$$

其中  $\alpha_j'$  的選取使其滿足

$$\begin{aligned} [Z^{(k)}]^T \nabla f(X^{(k)}) &= [Z^{(k)}]^T \nabla f(X^{(k-1)}) + [Z^{(k)}]^T Q[\alpha_k' Z^{(k)}] = 0 \\ \Rightarrow \alpha_k' &= -\frac{[Z^{(k)}]^T \nabla f(X^{(k-1)})}{[Z^{(k)}]^T QZ^{(k)}} \dots\dots\dots (A. 10) \end{aligned}$$

應用遞迴公式可得到：

$$\begin{aligned} X^{(m)} - X^{(j)} &= \sum_{i=j+1}^m \alpha_i' Z^{(i)}, j = 0, 1, 2, \dots, m-1 \\ \nabla f(X^{(m)}) &= \nabla f(X^{(j)}) + Q[X^{(m)} - X^{(j)}], j = 0, 1, 2, \dots, m-1 \dots\dots\dots (A. 11) \end{aligned}$$

將上式 (A. 11) 左右兩方各乘上  $[Z^{(j)}]^T$ ，即

$$[Z^{(j)}]^T \nabla f(X^{(m)}) = [Z^{(j)}]^T \nabla f(X^{(j)}) + \sum_{i=j+1}^m \alpha_i' [Z^{(j)}]^T QZ^{(i)}, j = 1, 2, 3, \dots, m-1 \quad (A. 12)$$

上式 (A. 12) 右式第一項由 (A. 7) 式知道會等於零，第二項則因為共軛關係所以也會等於零。經由上述之推導過程可獲得重要的推論：

若  $m=n$ ，則

$$\nabla f(X^n) = 0 \dots\dots\dots (A. 13)$$

於是我們得到  $X^{(n)}$  就是函數  $f$  在無約束條件下之最小值。在  $n$  維空間，函數  $f$  在  $X^{(n)}$  位置的梯度向量與  $Z^{(j)}$  向量都正交 ( $Z^{(j)}$  為線性獨立向量)，則函數  $f$  在  $X^{(n)}$  的梯度為零。

以符號  $\gamma^{(i)}$  來代表函數  $f$  在連續兩個點  $X^{(i)}$ 、 $X^{(i-1)}$  的梯度差，即



$$\begin{aligned} \gamma^{(i)} &= \nabla f(X^{(i)}) - \nabla f(X^{(i-1)}), \\ \Rightarrow \gamma^{(i)} &= Q[X^{(i)} - X^{(i-1)}] = \alpha_i' QZ^{(i)}, i=1,2,3,\dots \dots \dots (A. 14) \end{aligned}$$

位置  $\alpha_i'$  與方向  $Z^{(i)}$  的選取皆要使得(A. 7)式與(A. 10)式有效。因此

$$[\gamma^{(i)}]^T Z^{(j)} = \alpha_i' [Z^{(i)}]^T QZ^{(j)}, i, j=1,2,3,\dots, n \dots \dots \dots (A. 15)$$

假設  $Z^{(1)}, Z^{(2)}, Z^{(3)}, \dots, Z^{(k)}, k \leq n$  為所選之方向，且彼此各別與矩陣  $Q$  共軛正交，當  $i \neq j$  時，可以得到：

$$[\gamma^{(i)}]^T Z^{(j)} = 0, i \neq j, i=1,2,3,\dots, k; j=1,2,3,\dots, n \dots \dots \dots (A. 16)$$

根據上述的結果可以知道：

$$\begin{aligned} [\gamma^{(1)}]^T Z^{(2)} &= [\nabla f(X^{(1)}) - \nabla f(X^{(0)})]^T [-\nabla f(X^{(1)}) + \beta_{11} Z^{(1)}] \\ &= [\nabla f(X^{(1)}) - \nabla f(X^{(0)})]^T [-\nabla f(X^{(1)}) - \beta_{11} \nabla f(X^{(0)})] = 0 \end{aligned} \quad (A. 17)$$

其中  $Z^{(1)} = -\nabla f(X^{(0)})$ ，我們得到：

$$\beta_{11} = \frac{-[\gamma^{(1)}]^T \nabla f(X^{(1)})}{[\gamma^{(1)}]^T \nabla f(X^{(0)})} = \frac{[\nabla f(X^{(1)}) - \nabla f(X^{(0)})]^T \nabla f(X^{(1)})}{[\nabla f(X^{(1)}) - \nabla f(X^{(0)})]^T [-\nabla f(X^{(0)})]} \dots \dots \dots (A. 18)$$

由 ( A. 17 ) 式 與 ( A. 18 ) 式，可推得到

$$[Z^{(1)}]^T \nabla f(X^{(1)}) = -[\nabla f(X^{(0)})]^T \nabla f(X^{(1)}) = 0, \text{再假設 } \nabla f(X^{(0)}) \neq 0, \text{所以得到 } \beta_{11}$$

另一種表示式

$$\beta_{11} = \frac{[\nabla f(X^{(1)})]^T \nabla f(X^{(1)})}{[\nabla f(X^{(0)})]^T [\nabla f(X^{(0)})]} \dots \dots \dots (A. 19)$$

因此，由位置  $X^{(2)}$  沿著共軛方向  $Z^{(2)}$  可以到達最小值。同樣地，由(A. 7)

式與 (A. 8) 式兩式得知：

$$[Z^{(1)}]^T \nabla f(X^{(2)}) = [Z^{(2)}]^T \nabla f(X^{(2)}) = 0 \dots \dots \dots (A. 20)$$

因為  $Z^{(1)} = -\nabla f(X^{(0)})$ ，代入 (A. 20) 式左方，可以得到：

$$-\left[\nabla f(X^{(0)})\right]^T \nabla f(X^{(2)}) = 0 \dots\dots\dots (A. 21)$$

又  $Z^{(2)} = -\nabla f(X^{(1)}) + \beta_{11}Z^{(1)}$ ，將  $Z^{(2)}$  代入 (A. 20) 式右方得到：

$$\begin{aligned} & \left[(-\nabla f(X^{(1)})) + \beta_{11}\nabla f(X^{(0)})\right]^T \nabla f(X^{(2)}) = 0 \\ & -\left[\nabla f(X^{(1)})\right]^T \nabla f(X^{(2)}) + \beta_{11}\left[\nabla f(X^{(0)})\right]^T \nabla f(X^{(2)}) = 0 \dots\dots\dots (A. 22) \end{aligned}$$

由於 (A. 22) 式左式之第二項為零，所以第一項也會為零，即

$$-\left[\nabla f(X^{(1)})\right]^T \nabla f(X^{(2)}) = 0 \dots\dots\dots (A. 23)$$

繼續我們要找出 (A. 5) 式裡面之係數  $\beta_{21}$  與  $\beta_{22}$  所代表之的公式，由

(A. 5)、(A. 16)、(A. 20) 及 (A. 23) 式進一步推導：

$$\begin{cases} Z^{(3)} = -\nabla f(X^{(2)}) + \beta_{21}Z^{(1)} + \beta_{22}Z^{(2)} \\ \left[\gamma^{(1)}\right]^T Z^{(3)} = 0 \\ \left[\gamma^{(2)}\right]^T Z^{(3)} = 0 \end{cases}$$

我們可以得到

$$\beta_{21} = 0, \beta_{22} = \frac{\left[\nabla f(X^{(2)})\right]^T \nabla f(X^{(2)})}{\left[\nabla f(X^{(1)})\right]^T \nabla f(X^{(1)})} \dots\dots\dots (A. 24)$$

同理，以相同的方法我們也可以得到下面結果：

$$\left[\nabla f(X^{(i)})\right]^T \nabla f(X^{(j)}) = 0, 0 \leq i < j \leq n, \beta_{kj} = 0, k \neq j \dots\dots\dots (A. 25)$$

其中

$$\beta_k = \beta_{kk} = \frac{\left[\nabla f(X^{(k)})\right]^T \nabla f(X^{(k)})}{\left[\nabla f(X^{(k-1)})\right]^T \nabla f(X^{(k-1)})}, k = 1, 2, 3, \dots, n \dots\dots\dots (A. 26)$$

則

$$\begin{aligned}
Z^{(k+1)} &= -\nabla f(X^{(k)}) + \beta_k Z^{(k)} \\
&= -\nabla f(X^{(k)}) + \frac{[\nabla f(X^{(k)})]^T \nabla f(X^{(k)})}{[\nabla f(X^{(k-1)})]^T \nabla f(X^{(k-1)})} Z^{(k)} \dots\dots\dots (A. 27)
\end{aligned}$$

將以上的推導整理如下：

$$\left\{ \begin{array}{l}
X^{(k+1)} = X^{(k)} + \alpha_k Z^{(k)}, \alpha_k = -\frac{[Z^{(k)}]^T \nabla f(X^{(k)})}{[Z^{(k)}]^T QZ^{(k)}}, \\
\qquad \qquad \qquad k = 0, 1, 2, \dots, n \\
Z^{(k)} = -\nabla f(X^{(k-1)}) + \beta_k Z^{(k-1)}, \beta_k = \frac{[\nabla f(X^{(k)})]^T \nabla f(X^{(k)})}{[\nabla f(X^{(k-1)})]^T \nabla f(X^{(k-1)})}, \\
\qquad \qquad \qquad k = 1, 2, 3, \dots, n
\end{array} \right. \dots\dots\dots (A. 28)$$

## 附錄 B 亂數產生器

### B.1 混沌系統

混沌在物理上的真正解釋其實是指某些物理系統的看似無序運動模式。在一個世紀之前，認為物理系統只有兩種現象：1. 週期性的運動，例如月亮繞地球公轉，週期為一個月，而地球會自轉，週期為一天，如此週而復始。2. 演化到最後會趨於穩定的運動，例如一個由高處掉落的物體，他會在地面上反彈許多次之後，由於受到空氣摩擦的阻力，運動過程中會導致能量的損失，最後他便會停下來。直到二十世紀，才有人提出混沌這種完全不同的運動模式。可以用一個簡單的數學式子  $x_{n+1} = 2x_n^2 - 1$  來瞭解混沌系統，如果一開始在  $n=1$  你把當中  $x_1$  的代入  $(-1,1)$  之間任何一個數字，可得  $x_2$ ，如此反覆再代入可得  $x_3$ 、 $x_4 \cdots x_n$ ，便會得出一大堆看似雜亂無章的數字，如圖 B-1 所示。

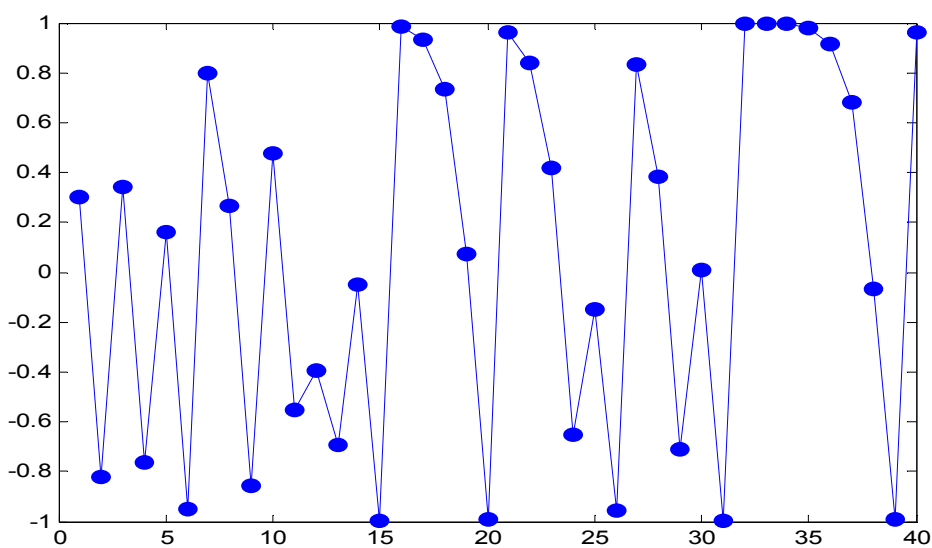


圖 B-1 以  $x=0.3$  代入  $2x^2-1$ ，不斷重複迭代的一組數列

由  $x_1 = 0.3$  當成初始值所形成的一組數列，而橫軸代表計算的次數，縱軸代表數值，這堆數字看起來好像毫無規則，無法預測，它既不是週期性的，亦不會趨向一個特定的數值，但事實上它是根據一條數式計算出來的，絕對不是一些隨機的數字，更有趣的是如果把初始值由 0.3 改成 0.3001，這些數字便會有截然不同的變化，如圖 B-2 所示。而把圖 B-1 與 B-2 重疊可得圖 B-3，如此可以清楚的看出兩組數列的差別。

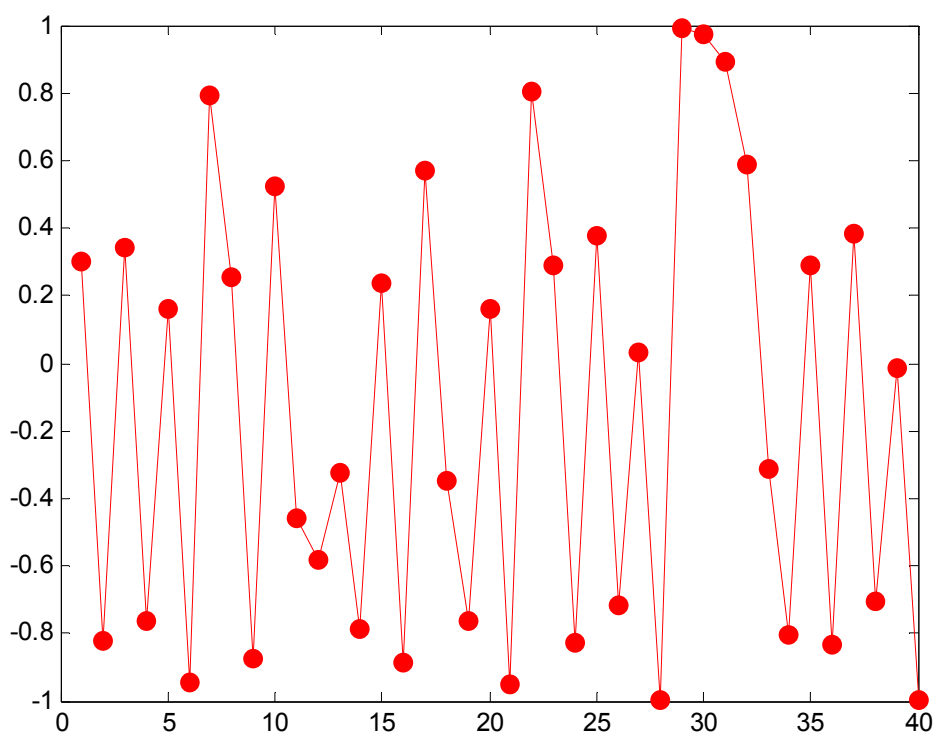


圖 B-2 以  $x=0.3001$  代入  $2x^2-1$ ，不斷重複迭代的一組數列

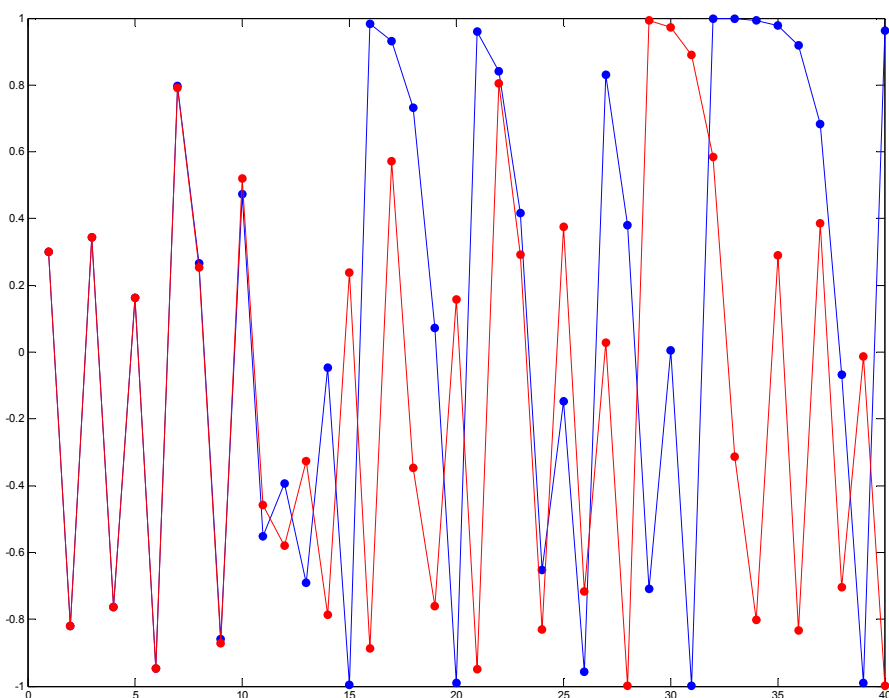


圖 B-3 比較初始值  $x=0.3$  與  $0.3001$  的計算結果

上述例子指出混沌系統兩個十分重要的特性：1. 系統的變化看似是毫無規則，但實際上是由物理定律所決定的。2. 系統的演化對初始條件的選取非常敏感，初始條件極微小的分別（就例如  $0.3$  和  $0.3001$  只相差三千分之一），在一定的迭代後會產生截然不同的結果。

## B.2 線性同餘產生器

經過上一小節的介紹，可以知道混沌系統的特性就是看似無規則、無週期的運動，但實際上卻是遵照一定的物理定律(或數學式子)。在電腦中的亂數也是如此，在此介紹最常使用的線性同餘產生

器(Linear Congruential Generator, LCG)，其演算規則如下：

$$x_{n+1} = (ax_n + b) \bmod m \cdots \cdots \cdots (B.1)$$

其中  $m$  稱為同餘因子(Congruential factor)，決定了亂數序列的最大週期，適當的挑選參數  $a$  和  $b$  則可以得到一組均勻分佈於  $[0, m)$  之間的亂數，在本論文中，分別使用到兩組週期不同的亂數，所使用的參數為

當 $m=2^{32}$	$a=25214903917$	$b=11$
當 $m=2^{48}$	$a= 214013$	$b= 2531011$

表 B-1 本論文中 LCG 所使用的參數

此兩組參數經過數學及統計學家證實，具有均勻分佈、滿週期等優良特性，本論文中所使用到的亂數皆由此產生。

## 參考文獻

1. Stefan Sinzinger, Jürgen Jahns, “Micro Optics” , Wiley-VCH, (1999)
2. J.H. Holland, “Genetic algorithm,” Scientific American, (1992)
3. G.Y. Zhou, X.C. Yuan, P. Dowd, Y.L. Lam, and Y.C Chan, "Diffractive optical elements designed by hybrid genetic algorithm for the generation of nondiffrating beams," SPIE's Photonics West ' 2001, January 2001, San Jose CA, USA.
4. Guangya Zhou, Xiao-Lin Zhao, Zongguang Wang, Yixin Chen, Mingsheng Zhang, “Optimization design of diffractive optical elements by genetic local search algorithms,” SPIE Proceedings Vol. 3740, pp.222-226(1999)
5. Eero Noponen, Jari Turunen, antti Vasara, “Parametric optimization of multilevel diffractive optical elements by electromagnetic theory,” Applied Optics, Vol.31, No. 28, P5910-5912 (1992)
6. Guo-zhen Yang et al.,” Gerchberg-Saxton and Yang-Gu algorithms for phase retrieval in a nonunitary transform



- system: a comparison,” Applied Optics, Vol.33  
No. 2 ,P209(1986)
7. Mordecai Avriel, “Nonlinear Programming: analysis and methods,” Prentice-Hall, Englewood Cliffs, New Jersey, P299-307 (1976)
  8. S.Kirk Patrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, Vol. 220, P671-680 (1983)
  9. R. Fletcher, “Practical Methods of Optimization” , second edition, John Wiley & Sons, P6-10, 19-23, 80-92(1999)
  10. Joseph W. Goodman, “Introduction to Fourier Optics,” second edition, McGraw-Hill, (1996)
  11. Ben-Yuan Gu, et al., “Diffractive-phase-elements design that implements several optical functions,” Applied Optics, Vol. 34, No.14, P2564 (May 1995)
  12. Yahya Rahmat-samii, Eric Michielssen, “Electromagnetic Optimization by Genetic Algorithms” , John Wiley & Sons, (1999)