

國立交通大學

多媒體工程研究所

碩士論文

粒子群演算法於圖形偵測與震測圖形識別之
應用

Particle Swarm Optimization for Pattern Detection
and Seismic Applications

研究生：董安晉

指導教授：黃國源 教授

中華民國九十六年六月

粒子群演算法於圖形偵測與震測圖形識別之應用
Particle Swarm Optimization for Pattern Detection and
Seismic Applications

研究生：董安晉

Student : An-Ching Tung

指導教授：黃國源 教授

Advisor : Dr. Kou-Yuan Huang

國立交通大學
多媒體工程研究所
碩士論文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2007

Hsinchu, Taiwan

中華民國九十六年六月

研究生：董安晉

指導教授：黃國源教授

國立交通大學多媒體工程研究所碩士班

摘要

我們採用一種以粒子群演算法 Particle Swarm Optimization (PSO) 為基礎的方法，用來偵測一張影像上的參數圖形（如：圓、橢圓、雙曲線與雙曲線的漸近線）的參數。我們定義了一個表示式來代表參數圖形，其中包含了平移及旋轉，並且定義了點到圖形的距離。我們用一個粒子來表示全部要偵測的參數圖形的參數，使用 PSO 在參數空間中尋找最佳的一個粒子，使影像空間上的所有影像點到此參數所代表的圖形的距離和為最小。我們先將 PSO 演算法應用於模擬的影像資料上，偵測模擬影像之後，再將之應用於偵測模擬的與真實的單炸點震測圖形上，偵測直接波圖形（直線）與反射波圖形（雙曲線），對於瞭解地層特性及後續的研究有很大的幫助。

關鍵字：粒子群演算法、霍夫演算法、震測圖形。

Particle Swarm Optimization for Pattern Detection and Seismic Applications

Student : An-Ching Tung Advisor : Dr. Kou-Yuan Huang

Institute of Multimedia Engineering

National Chiao Tung University



Abstract

Particle Swarm Optimization (PSO) is adopted to detect parameter pattern (e.g. circle, ellipse, hyperbola and asymptote.) Each particle is represented as parameters of patterns, then swarm of particles search the optimal solution in parameter space. We define mathematical formulas to represent various kinds of parameter patterns, and define the distance from points to patterns. Experiments on simulated image get good detection. The method is also applied to detect the parameters of direct wave (line) and reflected wave pattern (hyperbola) in simulated and real one-shot seismogram, the results can improve seismic interpretation and further seismic data processing.

Keywords: Particle Swarm Optimization, Hough transform, Seismic pattern.

致謝

首先誠摯的感謝指導教授黃國源博士，老師悉心的教導使我得以一窺類神經網路與圖形識別領域的深奧，不時的討論並指點我正確的方向，使我在這些年中獲益匪淺。老師對學問的嚴謹更是我輩學習的典範。本論文的完成得感謝老師的大力協助，因為有您的體諒及幫忙，使得本論文能夠更完整而嚴謹。

並感謝由 Colorado School of Mine.所建立的 Unix System，提供 Seismic data 作為實驗之用。

兩年裡的日子，實驗室裡共同的生活點滴，學術上的討論、言不及義的閒扯、讓人又愛又怕的宵夜、趕作業的革命情感，感謝眾位同學、學弟妹的共同砥礪，你們的陪伴讓兩年的研究生生活變得絢麗多彩。感謝學長不厭其煩的指出我研究中的缺失，且總能在我迷惘時為我解惑，也感謝楷儒同學的幫忙，恭喜我們順利走過這兩年。實驗室的映良、岳勳、俊宇學弟的幫忙及搞笑，讓我銘感在心。

最後，謹以此文獻給我摯愛的雙親。



目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
圖說明.....	v
表說明.....	vii
第一章 介紹.....	1
第二章 PSO 演算法.....	3
2.1 背景介紹.....	3
2.2 演算法參數與調整公式.....	3
2.3 演算法基本原理.....	4
2.4 其他參數設定.....	7
2.5 PSO 演算法流程.....	7
2.6 其他的 PSO 改進方法.....	8
第三章 PSO 偵測系統.....	10
3.1 偵測系統.....	10
3.2 輸入資料.....	10
3.3 定義圖形參數.....	10
3.4 點到圖形之距離.....	12
3.5 設定粒子參數.....	15
3.6 定義適應函數.....	15
第四章 實驗結果.....	20
4.1 模擬參數圖形偵測.....	20
4.2 模擬單炸點震測訊號圖形偵測.....	30
4.3 真實單炸點震測訊號圖形偵測.....	43
第五章 結論.....	52
第六章 討論.....	53
6.1 特殊圖形的參數偵測.....	53
6.2 圖形數目的設定.....	55
6.3 粒子數目的設定.....	57
6.3 學習參數的設定.....	58
參考資料.....	59

圖說明

圖 2.1	粒子調整方向與 $gBest$ 的關係：(a) 最佳解位置 m 與其它分佈粒子，(b)比較每個粒子的高度，找出最佳粒子 p ，(c) 根據 (2.1) 式及 (2.2) 式，其餘粒子往最佳粒子 p 的方向移動一段隨機距離，(d) 所有粒子移動到新位置.....4	4
圖 2.2	粒子調整方向與 $pBest$ 的關係：(a) 粒子 q 與粒子 p_1 ，因為 $f(q) < f(p_1)$ ， q 為 $gBest$ ，所以 p_1 會往 q 方向移動，(b) p_1 移動到達 p_2 ，發現 $f(p_1) < f(p_2)$ ，所以 p_1 為 $pBest$ ， p_2 的移動方向需考慮 q 及 p_1 ，(c) p_2 移動方向為兩個向量 $\alpha * rand() * (q - p_2)$ 與 $\beta * rand() * (p_1 - p_2)$ 之總和，來到 p_36	6
圖 3.1	PSO 圖形偵測系統.....10	10
圖 3.2	座標軸旋轉.....11	11
圖 3.3	座標軸平移.....12	12
圖 3.4	到圖形 $2x^2 + 2y^2 = 8$ 距離相同的點所形成的等高線圖.....14	14
圖 3.5	到圖形 $0.25x^2 + 0.25y^2 = 1$ 距離相同的點所形成的等高線圖.....14	14
圖 3.6	到圖形 $x^2 + y^2 = 1$ 距離的點所形成的等高線圖.....15	15
圖 3.7	一個點到全部圖形的距離計算.....16	16
圖 3.8	全部點到全部圖形的距離計算.....17	17
圖 3.9	PSO 偵測圖形參數演算法系統.....18	18
圖 4.1	(a) 圓的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....21	21
圖 4.2	(a) 橢圓的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....23	23
圖 4.3	(a) 一條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....24	24
圖 4.4	(a) 兩條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....26	26
圖 4.5	(a) 兩條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....28	28
圖 4.6	(a) 兩個雙曲線的偵測結果。(b) $gBest$ 的適應值 vs. iterations.....30	30
圖 4.7	PSO 單炸點震測訊號圖形偵測系統.....31	31
圖 4.8	模擬水平地層震測訊號.....31	31
圖 4.9	(a) 水平地層單炸點震測訊號圖。(b) 模擬單炸點震測訊號圖經過 Envelope 處理。(c) 經過 thresholding 後的模擬水平地層震測訊號輸入點。(d) 模擬水平地層單炸點震測訊號偵測結果。(e) 模擬水平地層單炸點震測訊號偵測結果與原始訊號。(f) $gBest$ 的適應值 vs. iterations。(g) 直接波圖形使用 4 個參數的偵測結果。(h) 直接波圖形使用 4 個參數的 $gBest$ 的適應值 vs.iterations.....32	32
圖 4.10	模擬傾斜地層震測訊號.....37	37
圖 4.11	(a) 傾斜地層單炸點震測訊號圖。(b) 模擬單炸點震測訊號圖經過	

Envelope 處理。(c) 經過 thresholding 後的傾斜水平地層震測訊號輸入點。(d) 模擬傾斜地層單炸點震測訊號偵測結果。(e) 模擬傾斜地層單炸點震測訊號偵測結果與原始訊號。(f) *gBest* 的適應值 vs. iterations。(g) 直接波圖形使用 4 個參數的偵測結果。(h) 直接波圖形使用 4 個參數的 *gBest* 的適應值 vs. iterations.....38

圖 4.12 (a) Canadian Artic 的單炸點震測訊號。(b) 震測訊號經過 Envelope 處理。(c) 經過 thresholding 後的偵測系統輸入點。(d) 震測訊號偵測結果。(e) 震測訊號偵測結果與原始訊號。(f) *gBest* 的適應值 vs. iterations.....44

圖 4.13 (a) Gulf of Cadiz 的單炸點震測訊號。(b) 震測訊號經過 Envelope 處理。(c) 經過 thresholding 後的偵測系統輸入點。(d) 震測訊號偵測結果。(e) 震測訊號偵測結果與原始訊號。(f) *gBest* 的適應值 vs. iterations..... 48

圖 6.1 以直線參數式來定義圖形的偵測結果.....54

圖 6.2 設定不同圖形個數的偵測結果與適應值.....56

圖 6.3 不同的圖形個數與適應值.....56



表說明

表 3.1	a 、 b 、 r 正負關係與圖形的形狀.....	11
表 4.1.1	圓的原始輸入參數.....	20
表 4.1.2	圓的參數偵測結果.....	21
表 4.2.1	橢圓的原始輸入參數.....	22
表 4.2.2	橢圓的參數偵測結果.....	22
表 4.3.1	一條直線的原始輸入參數.....	24
表 4.3.2	一條直線的參數偵測結果.....	24
表 4.4.1	兩條直線的原始輸入參數.....	25
表 4.4.2	兩條直線的參數偵測結果.....	25
表 4.5.1	兩條直線的原始輸入參數.....	27
表 4.5.2	兩條直線的參數偵測結果.....	27
表 4.6.1	兩個雙曲線的原始輸入參數.....	29
表 4.6.2	兩個雙曲線的參數偵測結果.....	29
表 4.7	模擬水平地層單炸點震測訊號圖參數偵測結果.....	35
表 4.8	直接波圖形使用 4 個參數的偵測結果.....	37
表 4.9	模擬傾斜地層單炸點震測訊號圖參數偵測結果.....	41
表 4.10	直接波圖形使用 4 個參數的偵測結果.....	43
表 4.11	震測訊號參數偵測結果.....	46
表 4.12	震測訊號參數偵測結果.....	50
表 6.1	直線的原始輸入參數.....	54
表 6.2	以直線參數式定義圖形的偵測結果.....	54

Chapter 1 介紹

偵測一張影像裡的參數圖形在影像處理中是一項重要的研究，Hough [1] 在 1962 年提出了將影像空間上的點轉換到參數空間上，藉由累加器偵測出參數空間上的尖峰值，來找出所對應的圖形參數，此方法稱做霍夫轉換 Hough Transform (HT)。HT 在實作上有幾個缺點，如：1，累加器的偵測範圍與偵測視窗 (window) 大小不容易決定；2，同時偵測多個圖形，或圖形的影像有誤差或雜訊時，造成決定尖峰值高度，及選取尖峰值個數的困難；3，需耗費大量的記憶空間與計算量。

自 HT 發表以來，已有許多改良的 HT 發表 [2]-[6]，用來解決上述的問題。Basak 與 Das [7] 在 2002 年發表了 Hough Transform Neural Network (HTNN)，以類神經網路調整梯度的方法來找出圖形的參數，並用在偵測直線、圓、橢圓的圖形上。然而這些方法並沒有用於偵測雙曲線上。

震測訊號圖形識別對於地層訊號的處理與研究有很大的幫助。Huang et al. [8] 在 1985 提出以 HT 偵測單炸點震測圖的直接波 (直線) 與反射波 (雙曲線) 曲線。Huang et al. [9] 在 2006 年提出以 HTNN 偵測單炸點震測圖的直接波與反射波曲線。然而 HTNN 是以調整梯度的方法，在問題複雜時，有落入局部最佳解的問題。

粒子群演算法 Particle Swarm Optimization (PSO) [10]-[11] 在 1995 年由 Kennedy 與 Eberhart 所發表，是一種以演化計算為基礎的演算法。PSO 啟發自對於鳥群及魚群的覓食行為之觀察與模擬，PSO 以一個粒子來代表問題空間上的一個解，並在問題空間上分佈大量的粒子來達到廣域搜尋的能力，尋找最佳解的方法是藉由粒子間互相比較與學習的行為來實現。同時 PSO 也被證實能用於解決許多最佳化問題上，如旅行銷售員問題[12]，電路設計 [13]，動態系統 [14]等，同時也擁有尋找全域最佳解的能力。

假設我們要偵測的影像 \mathbf{X} 上有 n 個點， $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ，是由 k 個參數圖形 P_1, P_2, \dots, P_k 所組成，且 $P_1 \cup P_2 \cup \dots \cup P_k = \mathbf{X}$ 。我們要將影像分成 k 組，分別屬於圖形 P_1, P_2, \dots, P_k ，並且找出每個圖形的參數，使得屬於圖形 P_k 的所有點到 P_k 的距離和為最小。我們以一個粒子代表所有圖形的參數，使用 PSO 的方法在參數空間中尋找最佳的一個粒子，即是最佳的所有圖形的參數。

應用此方法，可以偵測出圓、橢圓、雙曲線的及雙曲線的漸近線圖形。

我們先將 PSO 演算法應用於模擬的影像資料上，偵測模擬影像之後，再將之應用於偵測模擬的與真實的單炸點震測圖形上，偵測直接波圖形（直線）與反射波圖形（雙曲線）圖形，對於瞭解地層特性及後續的研究有很大的幫助。



Chapter 2 粒子群演算法

2.1 背景介紹

粒子群演算法 Particle Swarm Optimization (PSO) [10]-[11] 在 1995 年由 Kennedy 與 Eberhart 所發表，是一種以演化計算 [15] 為基礎的演算法。PSO 的靈感來自於對鳥群及魚群的覓食行為之觀察與模擬。

假設一群鳥分佈在一個空間中，在這空間只有一點有食物，鳥群無法知道食物確切的方向與位置，但是鳥群能藉由別種方式，例如食物味道的濃度，來知道自己與食物之間的距離，而且鳥群們能互相溝通，因此牠們能知道目前距離食物最近的鳥，其餘的鳥便往這隻鳥的位置移動一段距離，如此便完成一次搜索。所有的鳥移動到新的位置之後，再重新比較一次，找出距離食物最近的鳥，再作一次移動。重複以上的動作，鳥群會集中在食物所在的位置，便可以找出食物的位置。

2.2 演算法參數與調整公式

在 PSO 中，我們以一個粒子 (particle) 來表示一隻鳥的位置，鳥群在空間中尋找食物的位置，即為最佳解。首先要定義一個適應函式 (fitness function)，輸入粒子到這個函式之中，可以得到一個適應值 (fitness value)。適應函式不一定是一個數學函式，它可能是一個多層的網路架構，或是一個反應系統，而輸出值便是粒子的適應值。依據問題的不同，我們去設計不同的適應函式，由適應函式得到粒子的適應值。藉由比較所有粒子適應值的大小，我們可以找出目前表現最好的粒子，我們稱它為 $gBest$ ，為 *global best* 的意思。此外我們也考慮單一粒子的表現，由於粒子並非每一次都會移動到表現更好的位置，也就是這次得到的適應值不一定會比前一次得到的適應值優秀，因此我們會記錄單一粒子從演化開始到目前時間所有的適應值，在其中找出表現最好的一次，將那一次的粒子稱為 $pBest$ ，也就是 *personal best* 的意思。也就是說，若是我們有 n 個粒子，在每一次的演化中，會產生 1 個 $gBest$ 和 n 個 $pBest$ 。找出 $gBest$ 和 $pBest$ 後，便可以依照 (2.1) 及 (2.2) 式來調整粒子的位置：

$$\Delta s_i(t) = \alpha * rand() * (gBest(t) - s_i(t)) + \beta * rand() * (pBest_i(t) - s_i(t)) \quad (2.1)$$

$$\mathbf{s}_i(t+1) = \mathbf{s}_i(t) + \Delta\mathbf{s}_i(t) \quad (2.2)$$

$\mathbf{s}_i(t)$ 代表粒子 i 在時間 t 時的位置， $\Delta\mathbf{s}_i(t)$ 為它此次的調整量。 α 、 β 為兩個常數，作為學習參數，通常我們設定 $\alpha = \beta = 2$ 。 $rand()$ 為介於 0~1 之間的均勻分佈隨機數 (uniform random variable)。

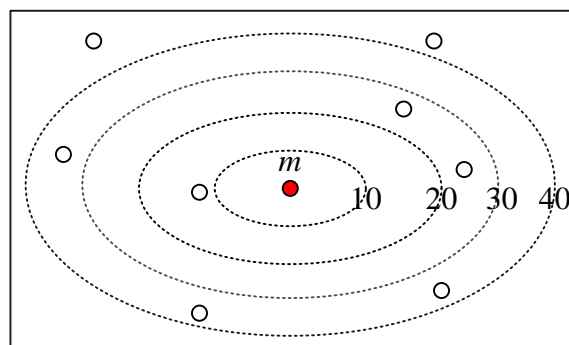
2.3 演算法基本原理

調整公式可分為兩部分，在以下分別討論。

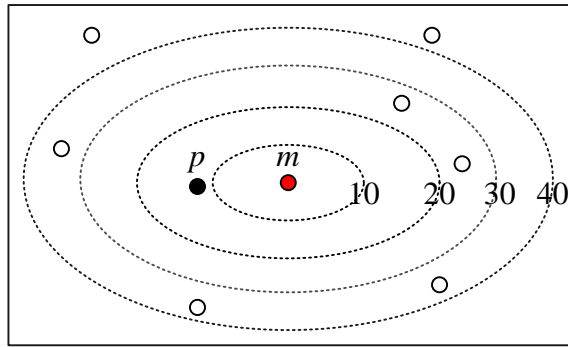
2.3.1 第一部分 $\alpha * rand() * (gBest(t) - \mathbf{s}_i(t))$

此部分代表了社會行爲 (social) [16]，也可稱之為模仿 (imitate)，粒子彼此之間能互相溝通、比較，並且學習表現較佳的粒子的行爲。我們設定學習參數 $\alpha = 2$ ，與 $rand()$ 介於 0~1 之間，使得這部分的改變量介於 $gBest(t) - \mathbf{s}_i(t)$ 的 0~2 倍之間，因此粒子不僅可能逼近 $gBest$ ，也可能超越 $gBest$ ，到達一個更佳的位置。

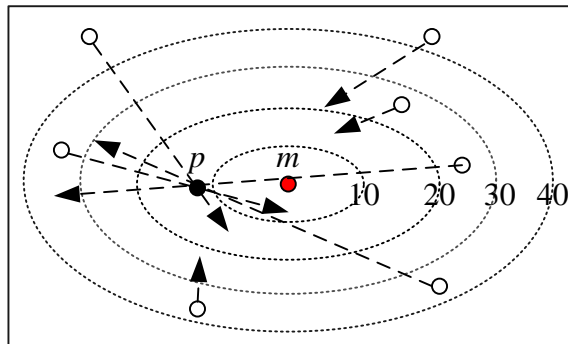
由一個簡單例子說明了粒子調整方向與 $gBest$ 的關係，圖 2.1 是一張 2D 等高線圖，虛線為等高度點的連線，越往中央高度越低， m 為圖上的最低點，我們想找出 m 的位置。因此我們在圖上隨機分佈一些粒子，接著由適應函式去計算出每一個粒子的適應值，在這裡我們定義適應值為粒子的高度。我們可以看出 p 的高度最低，是表現最好的一個粒子，我們便稱 p 為 $gBest$ ，而每個粒子一開始的 $pBset$ 都是自己。根據 (2.1) 式及 (2.2) 式，每個粒子會往 p 的方向移動一段隨機距離，所有粒子往 p 靠近，因此也有很大的機會靠近 m 。到達新位置後，所有粒子重新計算適應值，找出新的 $gBest$ ，以同樣的方法調整粒子位置。



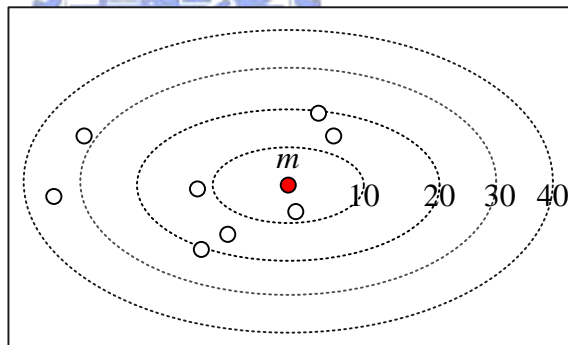
(a)



(b)



(c)



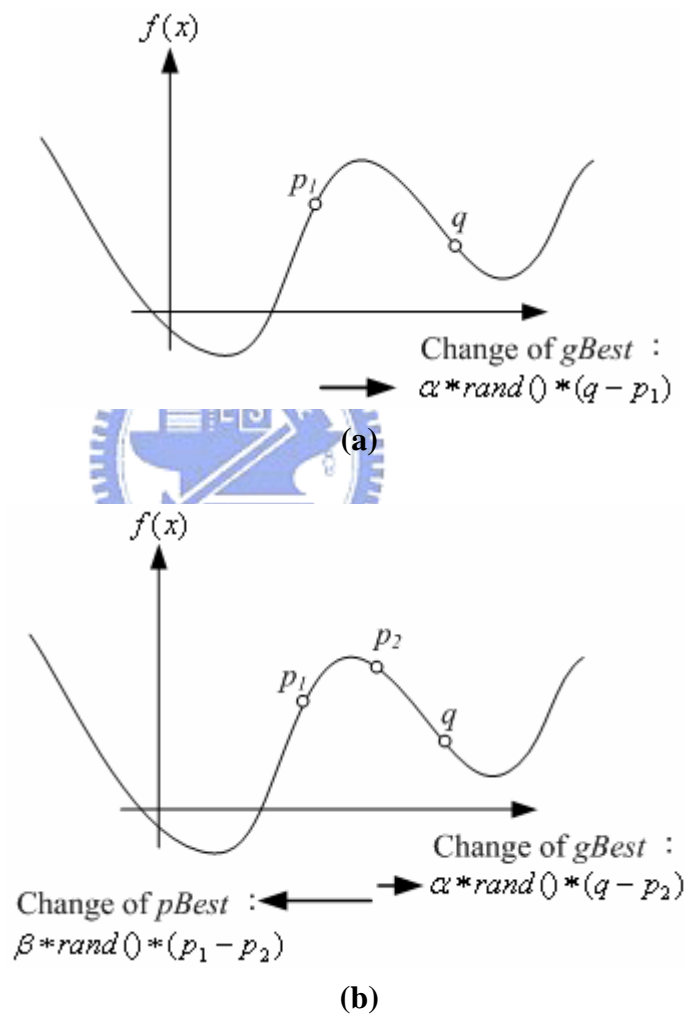
(d)

圖 2.1. 粒子調整方向與 $gBest$ 的關係：(a) 最佳解位置 m 與其它分佈粒子，(b) 比較每個粒子的高度，找出最佳粒子 p ，(c) 根據 (2.1) 式及 (2.2) 式，其餘粒子往最佳粒子 p 的方向移動一段隨機距離，(d) 所有粒子移動到新位置。

2.3.2 第二部分 $\beta * rand() * (pBest_i(t) - s_i(t))$ 。

此部分代表了自我認知 (cognition) [16]，亦可稱之為經驗法則 (experience)，粒子會記錄自己從演化開始到目前時間的所有適應值，並在其中找出表現最好的一次。有了此項，可以避免粒子落入局部最佳解之中。

圖 2.2 為一個簡單例子，一個函數曲線 $f(x)$ ，如圖所示，我們想找到 $f(x)$ 的最小值。一開始在 $f(x)$ 上有兩個粒子，粒子 q 與粒子 p_1 。由圖可知 $f(q) < f(p_1)$ ，因此我們設定 q 為 $gBest$ ， p_1 為自己的 $pBest$ 。根據 PSO 公式， p_1 會往 q 移動一段隨機距離到達 p_2 ，但是 $f(p_1) < f(p_2)$ ，所以 p_1 依然為 $pBest$ 。所以下一次移動時， p_2 除了會往 q 的方向移動，也會往 p_1 的方向移動，由於移動的向量會乘上一個 0~2 之間的隨機值，因此 p_2 有可能會往正確的方向，也就是 p_1 前進，到達 p_3 。我們可以看出，在調整時如果都不考慮 $pBest$ ， p_1 會一直往 q 的方向前進，而落入局部最佳解之中。



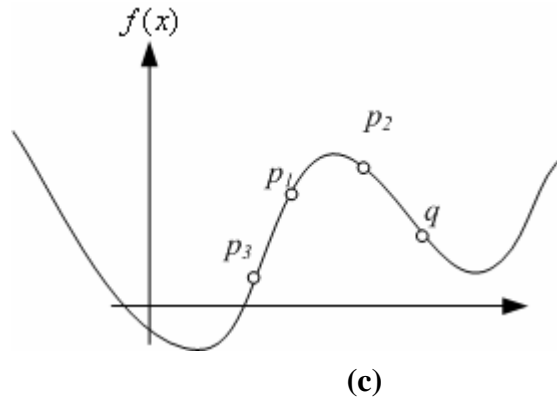


圖 2.2. 粒子調整方向與 $pBest$ 的關係：(a) 粒子 q 與粒子 p_1 ，因為 $f(q) < f(p_1)$ ， q 為 $gBest$ ，所以 p_1 會往 q 方向移動，(b) p_1 移動到達 p_2 ，發現 $f(p_1) < f(p_2)$ ，所以 p_1 為 $pBest$ ， p_2 的移動方向需考慮 q 及 p_1 ，(c) p_2 移動方向為兩個向量 $\alpha * rand() * (q - p_2)$ 與 $\beta * rand() * (p_1 - p_2)$ 之總和，來到 p_3 。

2.4 其他參數設定

除此之外，我們也可以設定參數 Δs_{max} ，代表粒子在一個維度上最大的調整量，若 $\Delta s_i(t)$ 的某一維度調整量大於 Δs_{max} ，則會被限制調整量最大為 Δs_{max} 。 Δs_{max} 太高，可能使粒子移動量過大，造成震盪產生， Δs_{max} 太小，使得每一次的改變量過小，可能會收斂太慢，或使粒子落入局部最佳解中。

PSO 演算法的終止條件，可設定為最大的演化次數 N ，若演化次數大於 N 則終止演化。或是設定最小的調整量 Δs_{min} ，當 $\Delta s_i(t)$ 每一維度的改變量皆小於 Δs_{min} ，代表全部的粒子都已經集中於一點，此時也可終止演化。

從上述的說明，我們可以得知 PSO 演算法，在問題空間上分佈大量的粒子，利用粒子的移動來搜尋最佳解的位置，因此 PSO 具有廣域搜尋的能力。而且 PSO 不像梯度下降 (Gradient descent) 演算法，總是往梯度下降的方向前進，PSO 尋找最佳解的方式是由粒子間互相比較、模仿，因此粒子移動的方向能夠往梯度上升的方向移動，也能越過局部最佳解的位置，當有粒子移動到全域最佳解附近時，其他的粒子也能很快的到達此位置，所以 PSO 也擁有尋找全域最佳解的能力。

2.5 PSO 演算法流程

我們將 PSO 演算法 [10]-[11] 流程整理如下：

Algorithm : Particle Swarm Optimization。

Input : 輸入資料、粒子數目與分佈位置、終止條件。

Output : 一個最佳的粒子，代表最佳解在問題空間中的位置。

Step 1. 設定適應函式。

Step 2. 輸入資料到適應函式，計算出每一個粒子的適應值。

Step 3. 對於單一粒子，記錄從演化開始到目前時間的所有適應值，找出最好的一次，將該次的粒子設定為 $pBest$ 。

Step 4. 對於全部粒子，比較全部粒子的適應值，找出目前最好的粒子，將該粒子設定為 $gBest$ 。

Step 5. 由 (2.1) 式及 (2.2) 式調整粒子位置。

$$\Delta \mathbf{s}_i(t) = \alpha * rand() * (gBest(t) - \mathbf{s}_i(t)) + \beta * rand() * (pBest_i(t) - \mathbf{s}_i(t))$$

$$\mathbf{s}_i(t+1) = \mathbf{s}_i(t) + \Delta \mathbf{s}_i(t)$$

Step 6. 重複 Step 2~5，直到滿足終止條件。

2.6 其他的 PSO 改進方法

PSO 發表以來，已被證實用在解決最佳化問題上有良好的效率及可靠性，並已經成功被應用於許多領域，如：類神經網路 [17]-[21]、模糊系統控制 [22] 等。PSO 演算法最常與同為演化計算之下的基因演算法 [23] 做比較，因為兩者有許多相同的特性，如：

1. 在一開始設定許多的粒子 (染色體)。
2. 計算粒子 (染色體) 的適應值。
3. 調整粒子位置 (基因交配、重組、突變)。
4. 重複同樣的步驟，直到滿足終止條件。

兩者僅在調整公式部分有所不同。

PSO 發表至今，有許多學者致力於改良 PSO 之效率與正確率。Shi 與 Eberhart [24] 提出在調整公式中加入 *inertia weight*，使目前的調整保有前一次的趨勢。Liang 與 Qin [25] 提出將粒子每個維度分開作調整，每一次的調整不是調整粒子的全部維度，而是依照機率改變其中某部分的維度，如此可避免粒子陷入局部最佳解。Angeline [26] 提出混合演化計算與 PSO 的方法，在每一次演化中，依照適應值高低將粒子作排序，並留下表現較好的粒子做調整，並以之取代表現較差的粒子。Kennedy [27] 提出將粒子先以拓撲 (topology) 分佈，

將粒子分群，找出每一群中最好的粒子，設定為 *lBest*，即 *local best* 之意，並在調整公式中加入 *lBest* 項。Hu 與 Eberhart [28] 將 PSO 應用於動態系統之中，讓粒子能自動適應系統變化，更快找出最佳解。以上研究都被證實能在某些問題上加速 PSO 的收斂，並提高 PSO 的正確率。



Chapter 3 PSO 偵測系統

3.1 偵測系統

我們設計出一套以 PSO 為基礎之偵測系統，如圖 3.1 表示。我們以原始影像上點的座標做為輸入資料，以 PSO 系統偵測出圖形參數，接著再以參數重建圖形。

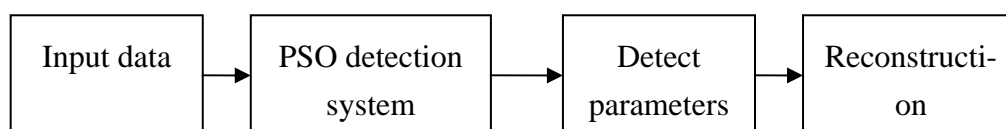


圖 3.1. PSO 圖形偵測系統。

3.2 輸入資料

輸入資料為欲偵測圖形 (object) 上每一點的座標 (x_i, y_i) ， $i = 1, \dots, n$ 。

3.3 定義圖形參數

在 PSO 偵測系統中，首先要知道的便是適應函數 (Fitness Function) 及適應值 (Fitness Value) 的定義。在說明之前，我們要先解釋圖形的參數定義。

我們所要偵測參數圖形皆為二次曲線，包括了圓、橢圓、雙曲線及雙曲線的漸近線。在經過平移與旋轉後的座標軸 $x'' - y''$ 中，圓、橢圓、雙曲線及雙曲線的漸近線皆可以下式表示：

$$ax''^2 + by''^2 = r \quad (3.1)$$

我們用 (3.1) 式來表示二次曲線的基本型態，由 a 、 b 、 r 的正負關係可得圖形的形狀，見表 3.1。當 a 、 b 、 r 同號，圖形為圓或橢圓， a 、 b 異號時則為雙曲線， a 、 b 異號且 $r = 0$ 時為雙曲線的漸近線。

表 3.1. a 、 b 、 r 正負關係與圖形的形狀。

a	b	r	圖形
+	+	+	圓或橢圓
-	-	-	
+	-	+	雙曲線
+	-	-	
-	+	+	
-	+	-	
+	-	0	雙曲線的漸近線
-	+	0	
+	+	-	無圖形
-	-	+	

因為 $x'' - y''$ 座標軸是由 $x' - y'$ 軸旋轉 θ 角度而來，兩者之間的關係可由 (3.2.1) 及 (3.2.2) 式來表示，圖 3.2 說明了座標軸旋轉的關係。

$$x'' = x' \cos \theta + y' \sin \theta \quad (3.2.1)$$

$$y'' = -x' \sin \theta + y' \cos \theta \quad (3.2.2)$$

將座標軸旋轉回 $x' - y'$ 後，得到新的表示式如下：

$$a(x' \cos \theta + y' \sin \theta)^2 + b(-x' \sin \theta + y' \cos \theta)^2 = r \quad (3.3)$$

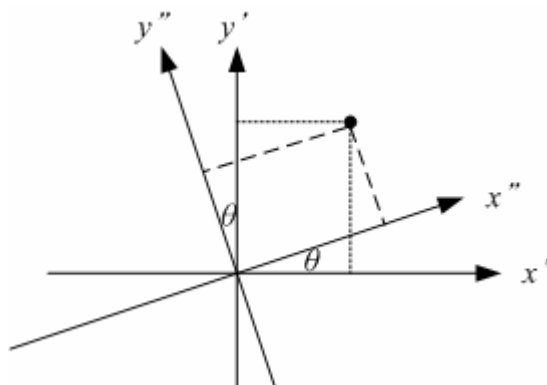


圖 3.2. 座標軸旋轉。

同樣的， $x' - y'$ 座標軸是由 $x - y$ 座標軸由原點平移至 (c_x, c_y) 而來，兩者之

間的關係可由 (3.4.1) 及 (3.4.2) 式來表示，圖 3.3 說明了座標軸平移的關係。

$$x' = x - c_x \quad (3.4.1)$$

$$y' = y - c_y \quad (3.4.2)$$

將 $x' - y'$ 座標軸平移回 $x - y$ 座標軸後，得到新的表示式如下：

$$a((x - c_x) \cos \theta + (y - c_y) \sin \theta)^2 + b(-(x - c_x) \sin \theta + (y - c_y) \cos \theta)^2 = r \quad (3.5)$$

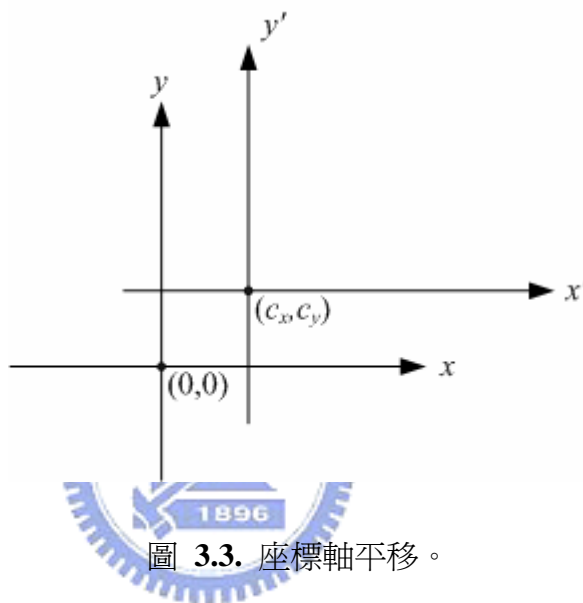


圖 3.3. 座標軸平移。

我們以 (3.5) 式來表示一個二次曲線參數圖形，包括圓、橢圓、雙曲線及雙曲線的漸近線，其中 (c_x, c_y) 為圖形的中心， θ 為旋轉的角度， r 為圖形大小，由 a 、 b 、 r 的正負關係可得圖形的形狀，見表 3.1。要注意的是，在這樣的表示式下，一條直線會被視為雙曲線漸近線的其中一條。

3.4 點到圖形之距離

對於輸入影像中一點 (x_i, y_i) 到一個二次曲線參數圖形的距離 d_i 則可以定義為 (3.6) 式：

$$d_i = |a((x_i - c_x) \cos \theta + (y_i - c_y) \sin \theta)^2 + b(-(x_i - c_x) \sin \theta + (y_i - c_y) \cos \theta)^2 - r| \quad (3.6)$$

我們要做的是去找到一組係數 c_x 、 c_y 、 a 、 b 、 r 、 θ ，使得 d_i 為最小。但是這樣的距離表示法會有兩個問題產生，第一，除了真正的解，還會得到一組係數

a 、 b 、 r 全為0的解，使得 $d_i=0$ ，然而這不是我們要的解；第二，相同的一個圖形，因為係數的不同，計算出來的距離也不同，例如 (3.7.1) 與 (3.7.2) 式。

$$2x^2 + 2y^2 = 8 \quad (3.7.1)$$

$$0.25x^2 + 0.25y^2 = 1 \quad (3.7.2)$$

這兩個方程式在幾何意義上是相同的圖形，但是同一個點到這兩個方程式計算出的距離卻不同，如圖 3.4 與圖 3.5，可看出兩者差異。因此我們必須做正規化的處理，我們將 (3.5) 式的係數全部除以 $\sqrt{|ab|}$ ，如 (3.8) 式。

$$\frac{a}{\sqrt{|ab|}}((x-c_x)\cos\theta+(y-c_y)\sin\theta)^2+\frac{b}{\sqrt{|ab|}}(-(x-c_x)\sin\theta+(y-c_y)\cos\theta)^2=\frac{r}{\sqrt{|ab|}} \quad (3.8)$$

正規化後得到新的表示式如下：

$$a'((x-c_x)\cos\theta+(y-c_y)\sin\theta)^2+b'(-(x-c_x)\sin\theta+(y-c_y)\cos\theta)^2=r' \quad (3.9)$$

新的距離定義如下：

$$d_i=|a'((x_i-c_x)\cos\theta+(y_i-c_y)\sin\theta)^2+b'(-(x_i-c_x)\sin\theta+(y_i-c_y)\cos\theta)^2-r'| \quad (3.10)$$

經過正規化處理後， $|a'b|$ 的值固定為1，點到圖形的距離也會一致，且不會改變原本圖形的形狀。經過處理後 (3.7.1) 式及 (3.7.2) 式都會變成 (3.11) 式。

$$x^2 + y^2 = 4 \quad (3.11)$$

這樣的處理，除了避免得到係數 a 、 b 、 r 全為0的方程式，點到圖形的距離也會一致。所以每演化出一組新的參數時，都要做 (3.8) 式的正規化處理。

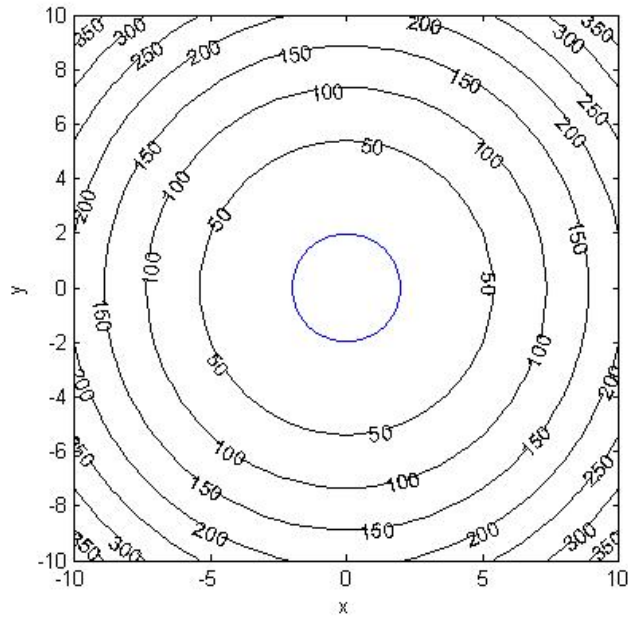


圖 3.4. 到圖形 $2x^2 + 2y^2 = 8$ 距離相同的點所形成的等高線圖。

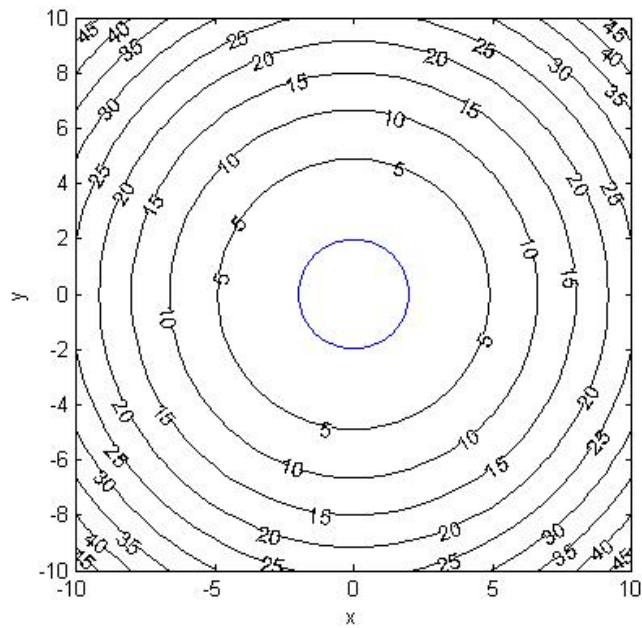


圖 3.5. 到圖形 $0.25x^2 + 0.25y^2 = 1$ 距離相同的點所形成的等高線圖。

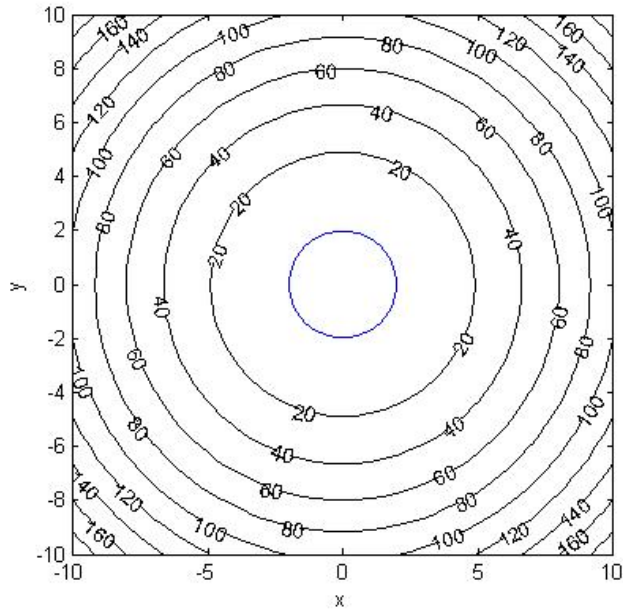


圖 3.6. 到圖形 $x^2 + y^2 = 1$ 距離相同的點所形成的等高線圖。

3.5 設定粒子參數

我們可以用上面提到的方式來定義一個圖形，並用一個粒子 s 來表示。我們可以將粒子 s 的係數設定為 $[c_x \ c_y \ a \ b \ r \ \theta]^T$ ，來代表一個圖形，其中係數 $[a \ b \ r]$ 要經過 (3.8) 式的正規化處理，使得 $|ab| = 1$ 。當偵測的圖形有兩個，則設定一個粒子為 $[c_{1x} \ c_{1y} \ a_1 \ b_1 \ r_1 \ \theta_1 \ c_{2x} \ c_{2y} \ a_2 \ b_2 \ r_2 \ \theta_2]^T$ ，以此推類。

$[c_x \ c_y]$ 的範圍可以由輸入的資料的分佈來觀察，猜測大概的位置， $[a \ b \ r]$ 與圖形的形狀有關，在實驗中，我們將這些值的初始範圍分佈於 $[-20 \sim 20]$ 之間。如果我們已經知道圖形的形狀為何，由表 3.1 可知道 $[a \ b \ r]$ 這三個維度的範圍，例如我們已經知道偵測的圖形是圓或橢圓，則限制 $[a \ b \ r]$ 要同號，如果是雙曲線，則限制 $ab < 0$ 。

3.6 定義適應函數

影像上有 n 個點，而偵測的圖形數目為一個時，我們依據 (3.10) 式計算影像上每一個點到粒子所代表的一個參數圖形的距離 d_1, d_2, \dots, d_n ，然後將全部距離相加後，除以影像上點的數目 n ， $f = (d_1 + d_2 + \dots + d_n)/n$ ，即為 n 個偵測點與一個參數圖形之間的平均距離。

當偵測的圖形有兩個以上時，我們先定義一個點到全部圖形的距離為『點到個別圖形距離的乘積』。假設圖形上的點可分為 k 個圖形 P_1, P_2, \dots, P_k ，

我們一樣依照 (3.10) 式分別來計算點到每個圖形的距離，因此可得到輸入影像中一點 (x_i, y_i) 到這些圖形的距離分別為 $d_{i,1}, d_{i,2}, \dots, d_{i,k}$ ，最後我們定義一個點到全部圖形的距離為個別距離的乘積，如同 (3.12) 式。計算一個點到全部圖形的距離如圖 3.7 所示。

$$D_i = d_{i,1} * d_{i,2} * \dots * d_{i,k} \quad (3.12)$$

我們如此定義的理由是，當點落在其中一個圖形上時，它到此圖形的距離為 0，因此到全部的圖形距離乘積也會是 0。

接著我們將影像上全部 n 個點依照同樣方法，計算出個別點到全部圖形的距離 D_1, D_2, \dots, D_n ，接著將全部相加後，除以影像上點的數目 n ，得到 (3.13) 式

$$f_j = \frac{(D_1 + D_2 + \dots + D_n)}{n} \quad (3.13)$$

f_j 就代表著全部的輸入點到粒子 j 所表示的全部參數圖形的平均距離，也就是粒子 j 所得到的適應值 (Fitness Value)。計算粒子適應值的過程如圖 3.8 所示。

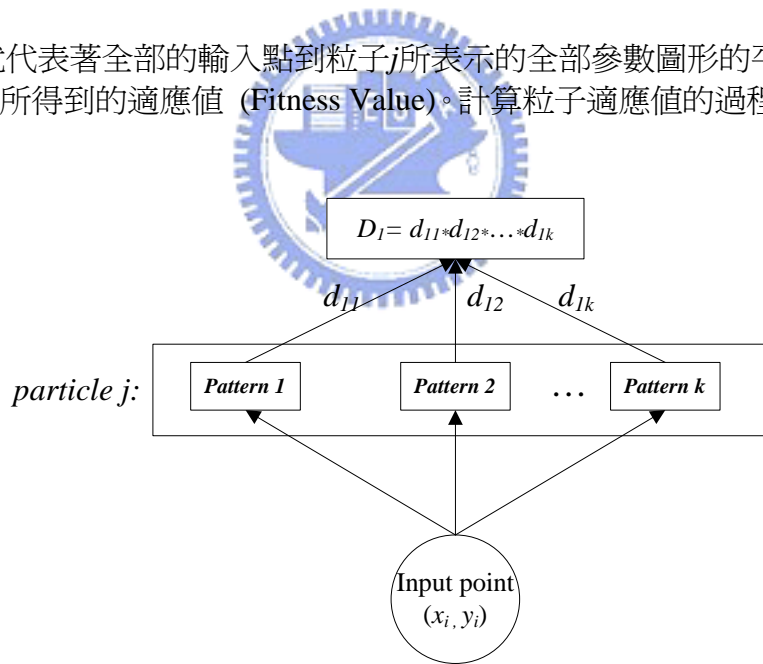


圖 3.7. 一個點到全部圖形的距離計算。

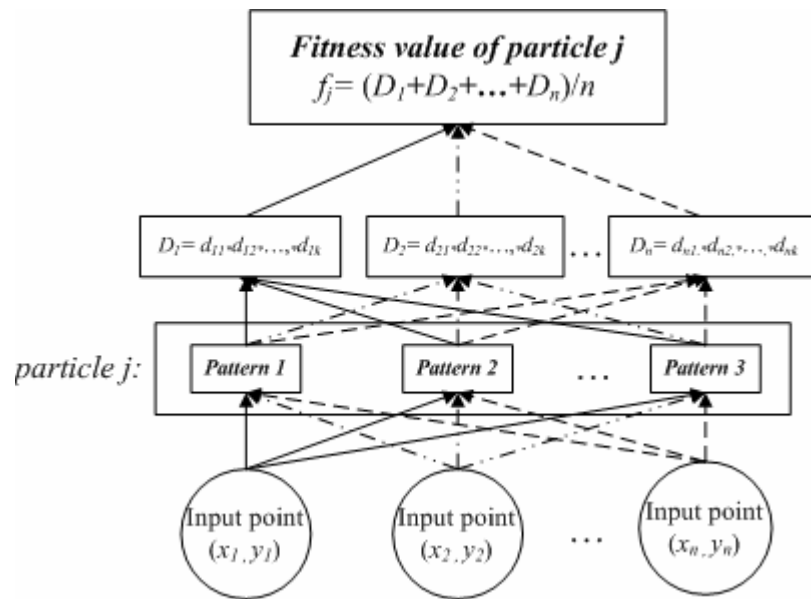


圖 3.8. 全部點到全部圖形的距離計算。

我們在參數空間上隨機分佈 m 個粒子，每個粒子依照欲偵測影像上參數圖形的數量設定維度。接著依上述的方法去計算每個粒子的適應值，適應值越低，代表粒子表示的參數圖形與欲偵測影像的圖形越相似。比較適應值，找出每一次演化的 $pBest$ 與 $gBest$ ，並以 PSO 演算法去調整粒子位置，直到演化次數到達停止條件，也就是最大的演化次數 N 後，得到一組最好的粒子，並以粒子的參數去還原圖形。整個系統如圖 3.9 所示。

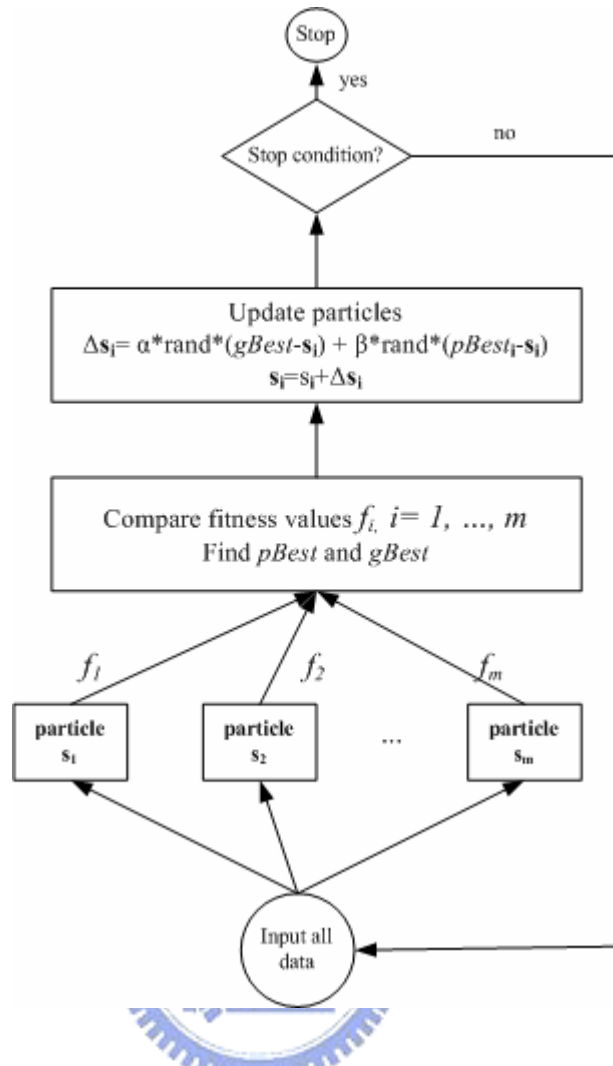


圖 3.9. PSO 偵測圖形參數演算法系統。

我們將 PSO 演算法偵測圖形參數系統整理如下：

Algorithm：PSO 偵測圖形參數演算法。

Input：欲偵測影像中點的座標 (x_i, y_i) ， $i = 1, \dots, n$ ，設定要偵測的圖形數目 k ，粒子數目 m ，學習參數 α 、 β ，終止條件：最大演化次數 N 。

Output：適應值最佳的一個粒子，代表最佳的一組圖形參數，使偵測影像中的點到此圖形參數所代表的圖形距離最小。

Step 1. 初始設定：依照要偵測的圖形數目 k 設定一個粒子 \mathbf{s}_i ，

$$\mathbf{s}_i = [c_{i,1x} \ c_{i,1y} \ a_{i,1} \ b_{i,1} \ r_{i,1} \ \theta_{i,1}, \dots, c_{i,kx} \ c_{i,ky} \ a_{i,k} \ b_{i,k} \ r_{i,k} \ \theta_{i,k}]^T$$

同理，設定 $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m$ 。

Step 2. 圖形參數經過正規化處理後，輸入點座標，計算一個點到一個

圖形的距離，如 (3.10) 式：

$$d_i = \left| a'((x_i - c_x) \cos \theta + (y_i - c_y) \sin \theta)^2 + b'(-(x_i - c_x) \sin \theta + (y_i - c_y) \cos \theta)^2 - r \right|$$

計算一個點到全部圖形的距離，如 (3.12) 式：

$$D_i = d_{i,1} * d_{i,2} * \dots * d_{i,k}$$

計算所有點到全部圖形的平均距離，如 (3.13) 式：

$$f_j = \frac{D_1 + D_2 + \dots + D_n}{n}$$

計算出每個粒子的適應值 f_1, f_2, \dots, f_m 。

Step 3. 比較適應值，找出每個粒子的 $pBest$ 及全部粒子的 $gBest$ 。

Step 4. 調整粒子的位置，如 (2.1) 式及 (2.2) 式：

$$\Delta \mathbf{s}_i(t) = \alpha * rand() * (gBest(t) - \mathbf{s}_i(t)) + \beta * rand() * (pBest_i(t) - \mathbf{s}_i(t))$$

$$\mathbf{s}_i(t+1) = \mathbf{s}_i(t) + \Delta \mathbf{s}_i(t)$$

Step 5. 重複 Step 2~4，直到符合終止條件（最大演化次數 N ）。

Chapter 4 實驗與結果

PSO 參數圖形偵測系統的實驗分為三個部分，第一部分為模擬的參數圖形，包括混合的直線、圓、橢圓以及雙曲線。第二部分為模擬的震測訊號，我們用已知的地層深度、震波速度、傾斜角度等，去模擬出單炸點震測圖，其中有水平地層及傾斜地層。第三部分偵測為真實的單炸點震測圖。

實驗的程式以 Matlab R2006a 執行，作業系統為 Windows XP，Intel Pentium(R) 4，3.00GHz CPU，1.5GB RAM。

4.1 模擬參數圖形偵測

訓練的影像樣本範圍皆為 50×50 ，每一個圖形有 50 個點，每個點的座標加上 $[-1 \sim 1]$ 的 uniform random noise。

我們設定粒子個數為 300 個，根據圖形數目 k 去設定粒子維度，每個維度的初始分佈皆是隨機分佈於 $[-20 \sim 20]$ ，實驗中沒有限制粒子移動的大小或範圍。演化次數設定為 300 次，學習參數 $\alpha = \beta = 2$ 。

實驗 1. 偵測圓的實驗

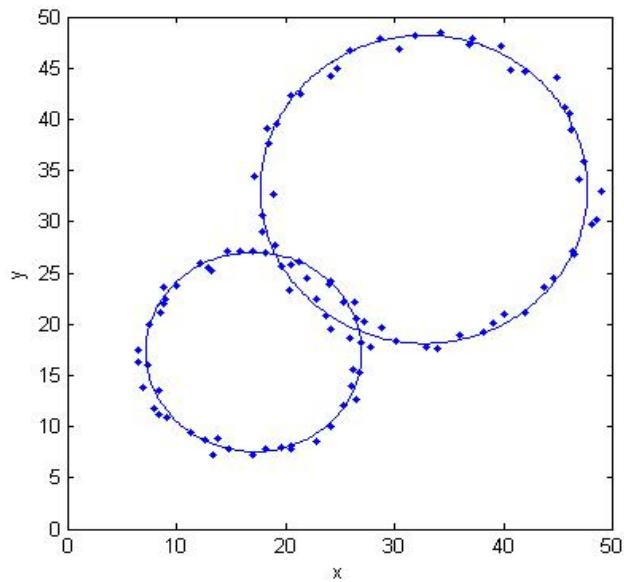
輸入點為兩個圓形，範圍為 50×50 ，共有 100 個輸入點，每個點的座標加上 $(-1 \sim 1)$ 的 uniform random noise。設定偵測的圖形數目 $k = 2$ 。表 4.1.1 為原始輸入參數，表 4.1.2 為參數偵測結果，圖 4.1(a) 為兩個圓的偵測結果，4.1(b) 為 $gBest$ 的適應值 vs. iterations。

表 4.1.1. 圓的原始輸入參數。

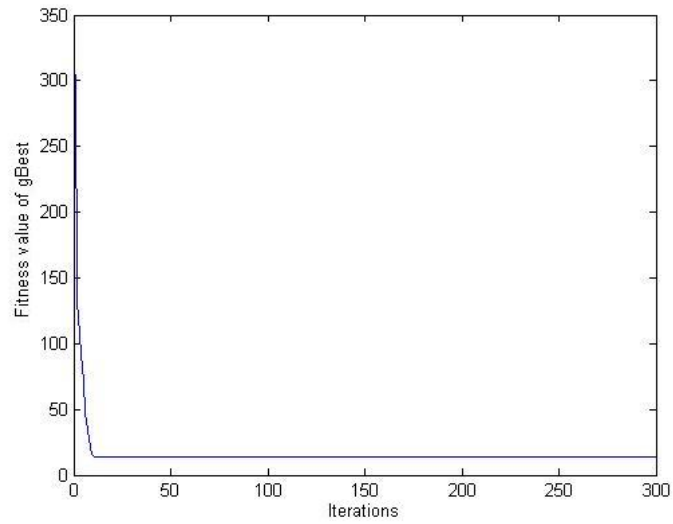
	c_x	c_y	r
Pattern 1	17	17	10
Pattern 2	33	33	15

表 4.1.2. 圓的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間 (sec)
Pattern 1	17.04	17.28	-0.98	-1.01	-96.53	9.31°	77.14
Pattern 2	32.67	33.15	-1.00	-0.99	-225.86	4.12°	



(a)



(b)

圖 4.1. (a) 圓的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

實驗 2. 偵測橢圓的實驗

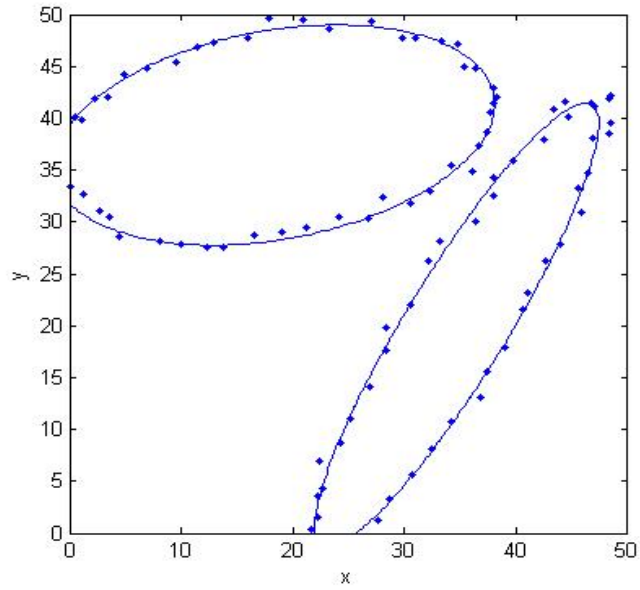
輸入點為兩個橢圓形，範圍為 50x50，共有 100 個輸入點，每個點的座標加上(-1~1)的 uniform random noise。設定偵測的圖形數目 $k = 2$ 。表 4.2.1 為原始輸入參數，表 4.2.2 為參數偵測結果，圖 4.2(a) 為兩個橢圓的偵測結果，4.2(b) 為 $gBest$ 的適應值 vs. iterations。

表 4.2.1. 橢圓的原始輸入參數。

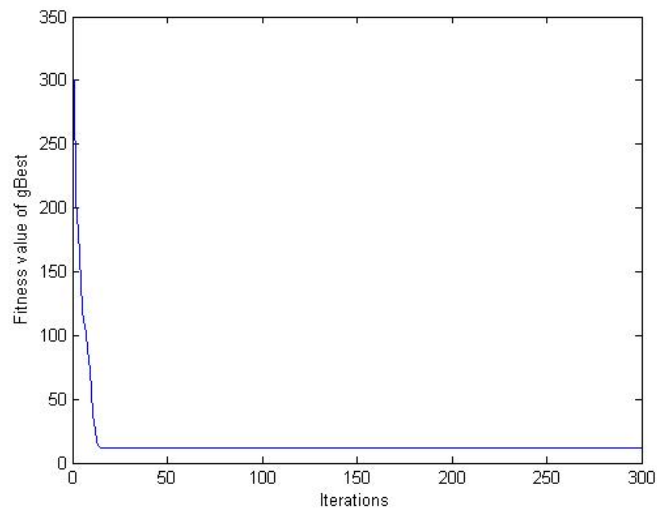
	c_x	c_y	a	b	r	θ
Pattern 1	20	35	5	25	10	-30°
Pattern 2	25	35	20	10	15	10

表 4.2.2. 橢圓的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間 (sec)
Pattern 1	34.70	20.08	5.30	0.18	113.28	29.61	69.69
Pattern 2	18.31	38.37	0.49	2.00	201.67	-11.51	



(a)



(b)

圖 4.2. (a) 橢圓的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

實驗 3. 偵測一條直線的實驗

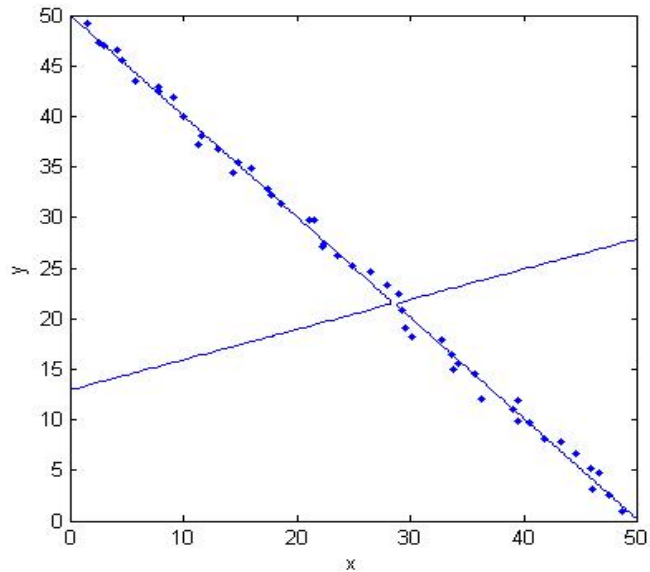
輸入點為一條直線，範圍為 50×50 ，共有 50 個輸入點，每個點的座標加上 $(-1 \sim 1)$ 的 uniform random noise。設定偵測的圖形數目 $k = 1$ 。表 4.3.1 為原始輸入參數，表 4.3.2 為參數偵測結果，圖 4.3(a) 為一條直線的偵測結果，4.3(b) 為 $gBest$ 的適應值 vs. iterations。

表 4.3.1. 一條直線的原始輸入參數。

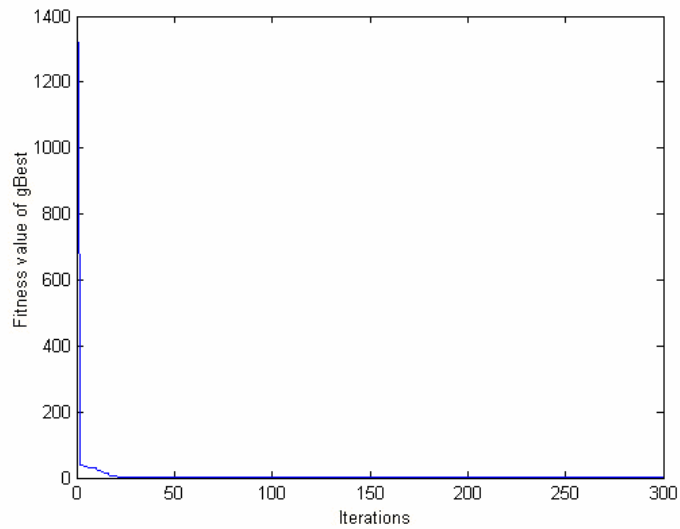
Pattern	$x + y = 50$
---------	--------------

表 4.3.2. 一條直線的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
Pattern	28.57	21.49	0.59	-1.68	0.04	14.13	49.69



189 (a)



(b)

圖 4.3. (a) 一條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

實驗 4. 偵測兩條直線的實驗

輸入點為兩條直線，範圍為 50x50，共有 100 個輸入點，每個點的座標加上(-1~1)的 uniform random noise。設定偵測的圖形數目 $k=1$ 。表 4.4.1 為原始輸入參數，表 4.4.2 為參數偵測結果，圖 4.4(a) 為兩條直線的偵測結果，4.4(b) 為 $gBest$ 的適應值 vs. iterations。

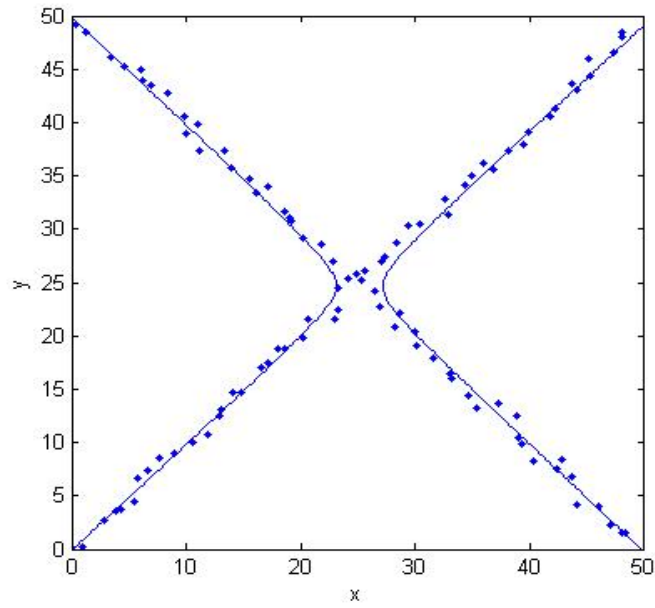
表 4.4.1. 兩條直線的原始輸入參數。

Pattern 1	$x + y = 50$
Pattern 2	$x - y = 0$

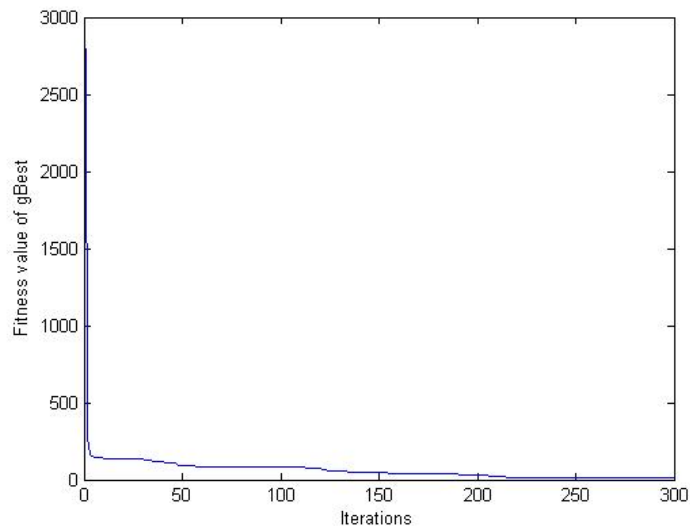
表 4.4.2. 兩條直線的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
Pattern	25.12	24.66	0.99	-1.00	4.16	0.26	47.39





(a)



(b)

圖 4.4. (a) 兩條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

實驗 5. 偵測兩條直線的實驗

輸入點為兩條直線，範圍為 50×50 ，共有 100 個輸入點，每個點的座標加上 $(-1 \sim 1)$ 的 uniform random noise。設定偵測的圖形數目 $k=2$ 。表 4.5.1 為原始輸入參數，表 4.5.2 為參數偵測結果，圖 4.5(a) 為兩條直線的偵測結果，4.5(b) 為 $gBest$ 的適應值 vs. iterations。

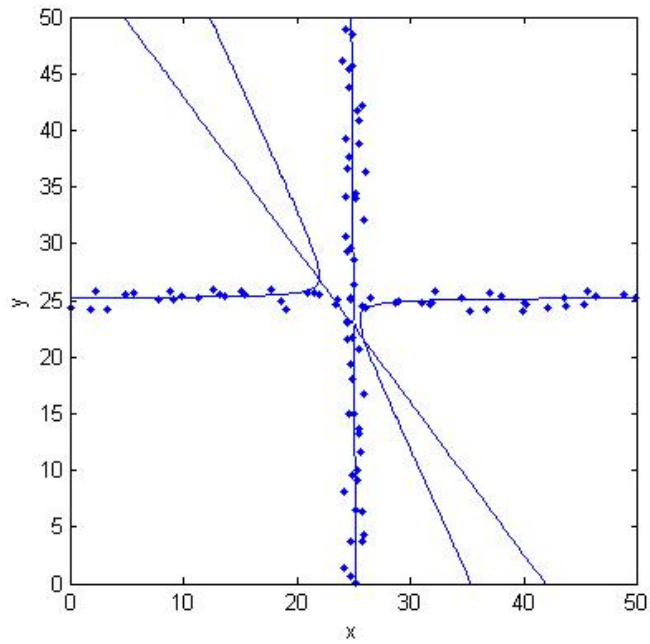
表 4.5.1. 兩條直線的原始輸入參數。

Pattern 1	$x = 25$
Pattern 2	$y = 25$

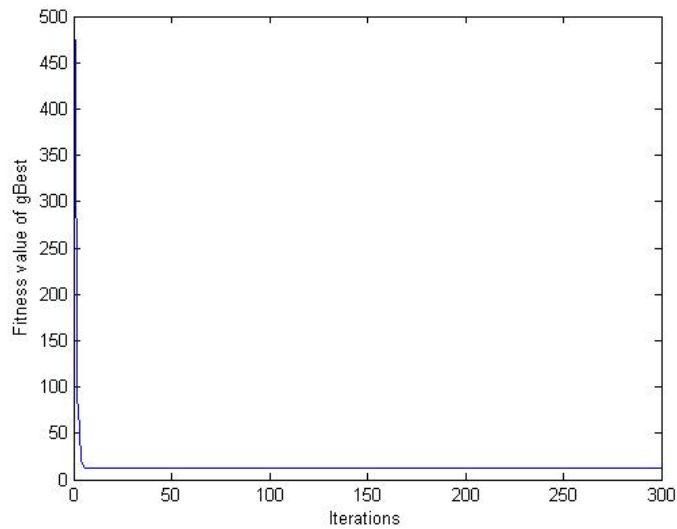
表 4.5.2. 兩條直線的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
Pattern 1	24.98	22.74	3.06	-0.32	0.06	-18.55°	76.35
Pattern 2	23.73	25.19	1.55	-0.64	-3.42	-57.43°	





(a)



(b)

圖 4.5. (a) 兩條直線的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

實驗 6. 偵測兩個雙曲線的實驗

輸入點為兩個雙曲線圖形，範圍皆為 50×50 ，共有 100 個輸入點，每個點的座標加上 $(-1 \sim 1)$ 的 uniform random noise。設定偵測的圖形數目 $k = 2$ 。表 4.6.1 為原始輸入參數，表 4.6.2 為參數偵測結果，圖 4.6(a) 為兩條直線的偵測結果，4.6(b) 為 $gBest$ 的適應值 vs. iterations。

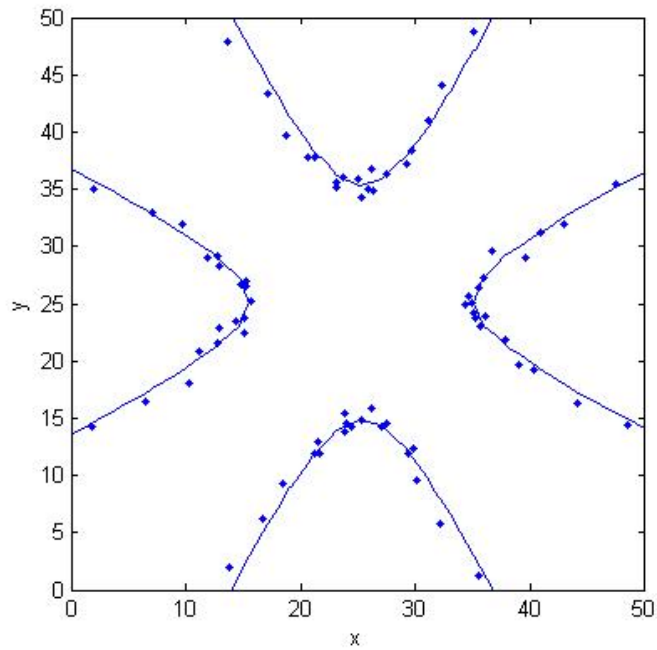
表 4.6.1. 兩個雙曲線的原始輸入參數。

	c_x	c_y	a	b	r	θ
Pattern 1	25	25	1	-4	10	0°
Pattern 2	25	25	-4	1	10	0°

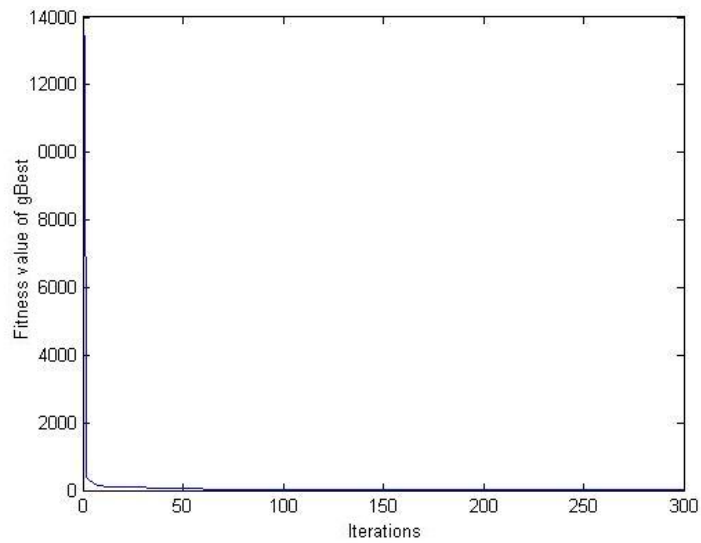
表 4.6.2. 兩個雙曲線的參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
Pattern 1	25.07	25.01	1.99	-0.50	-2.09	0.06°	71.38
Pattern 2	25.05	25.04	0.49	-2.01	1.93	-0.12°	





(a)



(b)

圖 4.6. (a) 兩個雙曲線的偵測結果。(b) $gBest$ 的適應值 vs. iterations。

4.2 模擬單炸點震測訊號圖形偵測

我們將 PSO 偵測系統應用在震測訊號圖形偵測上，首先模擬出單炸點震測訊號圖，接著將訊號作 Envelope [29]處理，接著作 thresholding，取出震幅大於 threshold 的點，取出來的點便作為 PSO 偵測系統的輸入點，偵測出參數後，依據參數重建圖形，整個流程如圖 4.7。

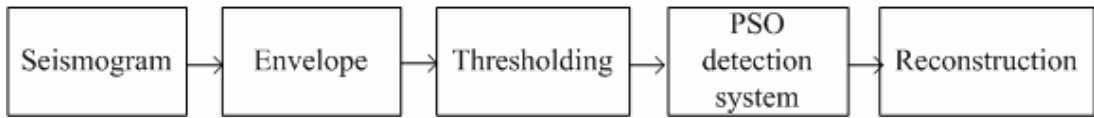


圖 4.7. PSO 單炸點震測訊號圖形偵測系統。

4.2.1 偵測水平地層單炸點震測訊號

水平地層單炸點震測訊號如圖 4.8 所示 [30]，共有兩層地層，地面上 Source 為爆炸點，在爆炸點左右各有 64 個接收站，加上爆炸點處 1 個，共有 129 個接收站，每個接收站之間距離 Δx 為 50 meters，接收站與爆炸點之間距離 $x = k * \Delta x$ ， k 為接收站編號。接收站之 sampling interval 為 0.004 sec，總共有 512 個 samples。地表與反射層的垂直深度 h 為 2,500 meters，為爆炸點到地下地層的垂直深度。直接波與反射波速度 v 皆為 2,500 meters/sec，反射係數為 0.2。由圖 4.8 可知直接波及反射波的行進方向，由計算可知直接波訊號為直線，(4.1) 式為直接波訊號公式，反射波訊號為雙曲線，(4.2) 式為反射波訊號公式。

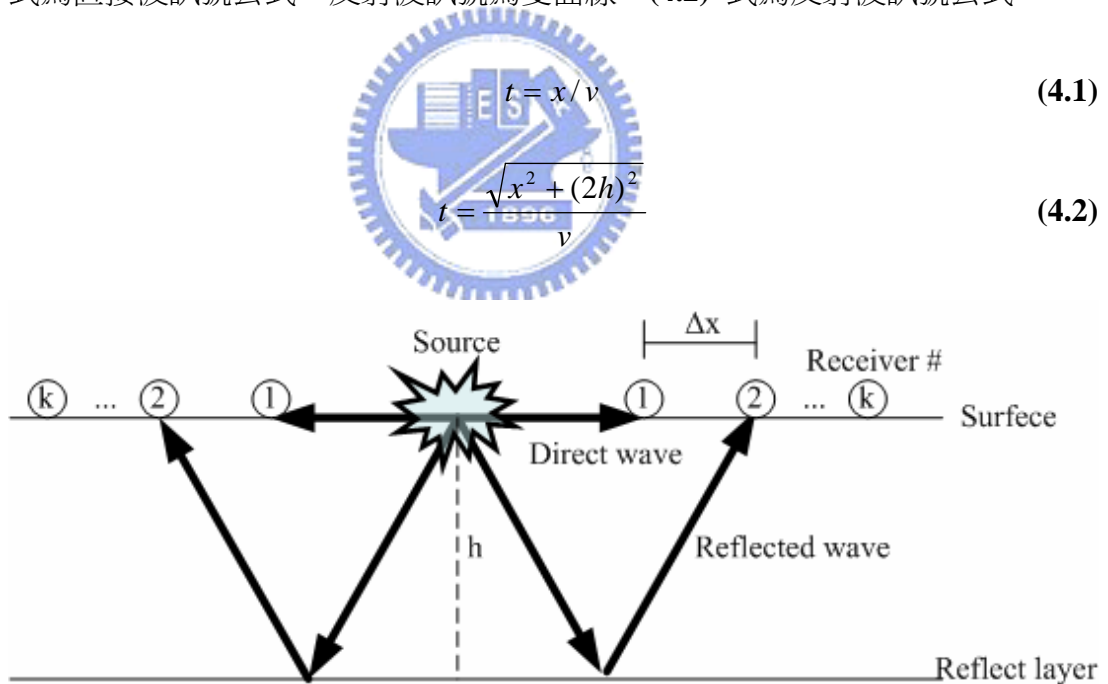


圖 4.8. 水平地層震測訊號。

我們計算出接收站接收到直接波與反射波的時間後，將訊號與 25 Hz 的 Ricker Wavelet 作 convolution，得到 impulse 訊號，再加上 random white band 10.2539 Hz 至 59.5703 Hz 的 noise，uniform distribution over interval (-0.2,0.2)。最後得到的震測訊號如圖 4.9(a)。圖中間上方 0 點位置為爆炸點，橫軸為接收器的編號，左右各 64 個，縱軸由上往下，為每一個接收器接收到直接波與反射

波訊號的時間。

我們將震測訊號作 Envelope 處理，得到圖 4.9(b)。接著將振幅超過 0.2 的點取出，共取出 432 點，作為偵測圖形系統的輸入，每個點的值是 $[x, i]$ ， x 是接收器的編號， i 是接收到訊號的時間除以 sampling interval，代表該接收站到接收到的第幾個訊號。我們得到的輸入點如圖 4.9(c)。我們設定偵測圖形數目 $k = 2$ ，偵測出兩條雙曲線，如圖 4.9(d)，將結果顯示在原始訊號上，如圖 4.9(e)，4.9(f) 為 $gBest$ 的適應值 vs. iterations。表 4.7 為參數偵測結果。

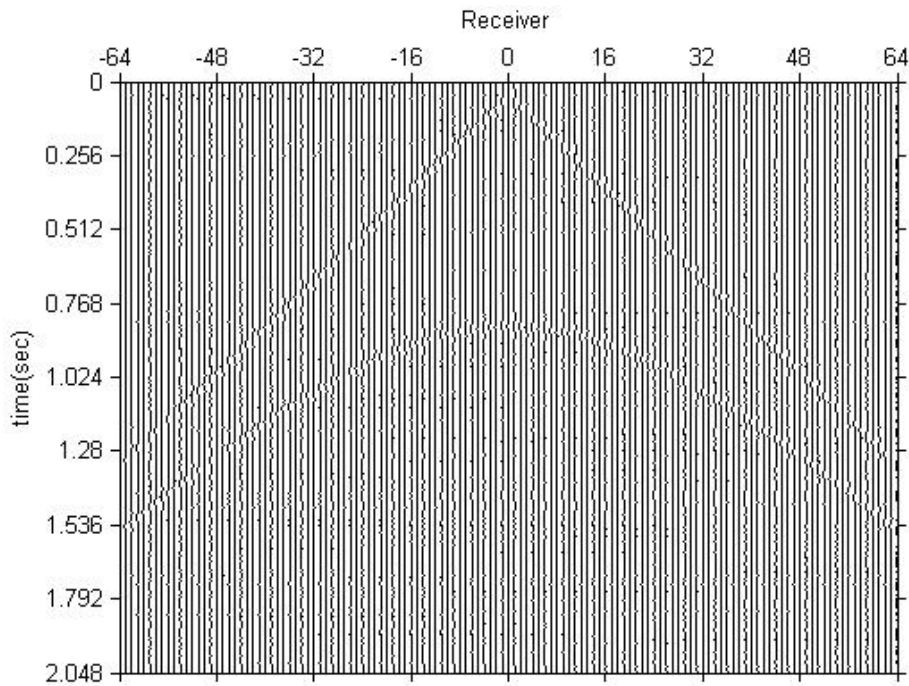


圖 4.9. (a) 模擬水平地層單炸點震測訊號圖。

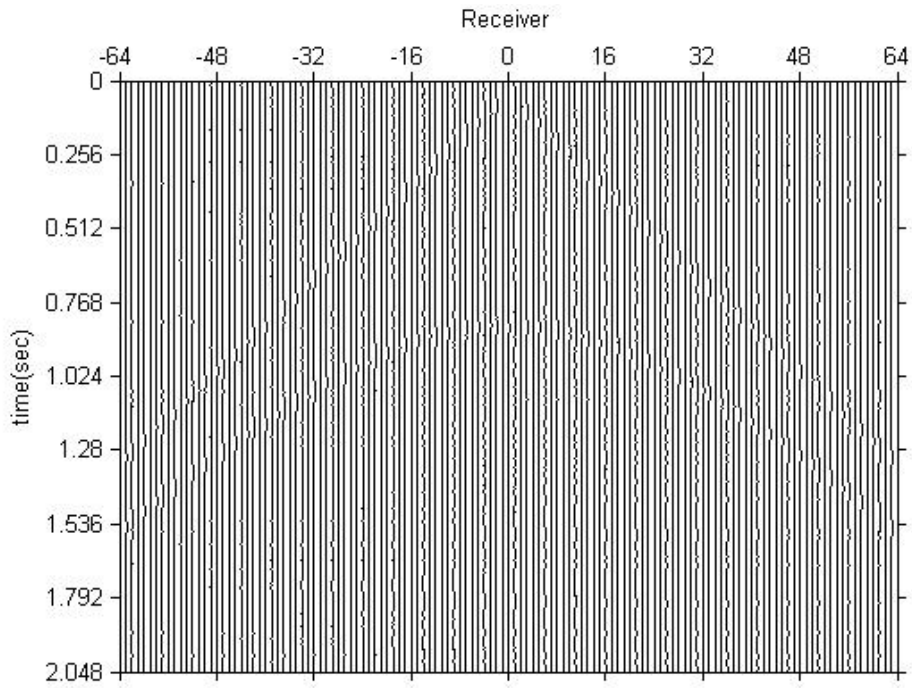


圖 4.9. (b) 模擬單炸點震測訊號圖經過 Envelope 處理。

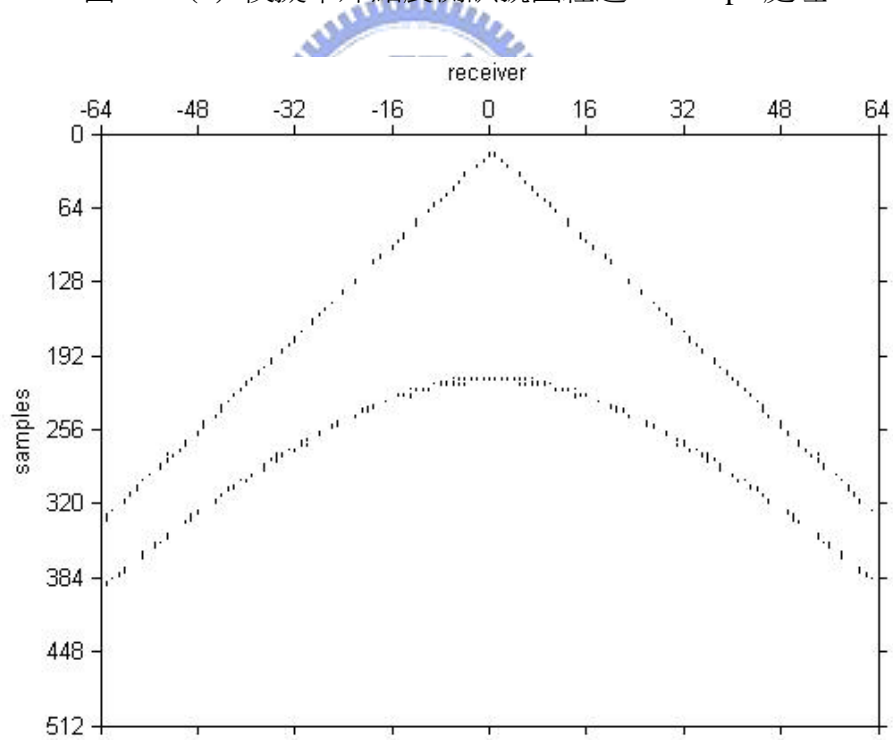


圖 4.9. (c) 經過 thresholding 後的模擬水平地層震測訊號輸入點。

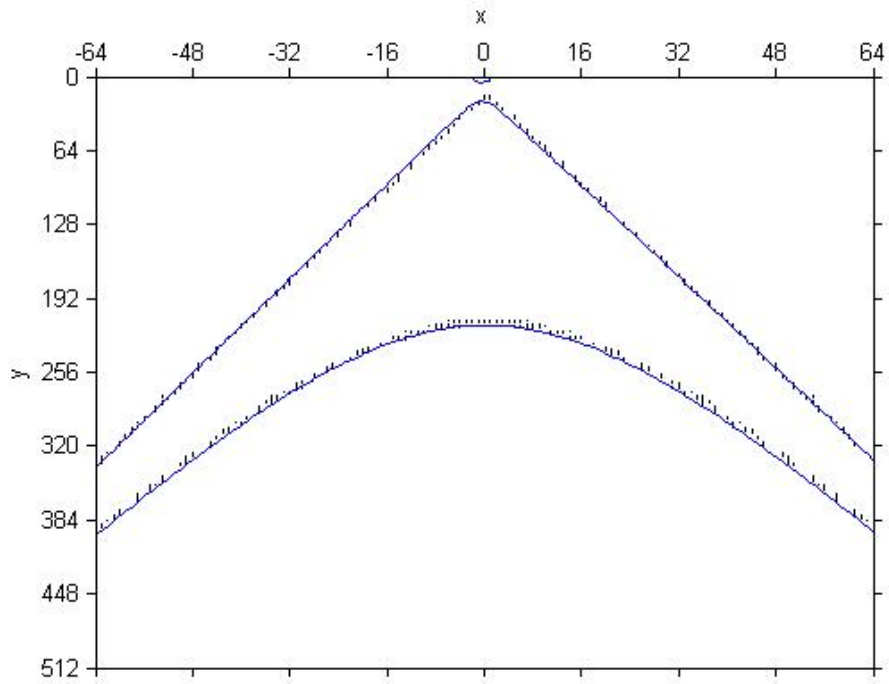


圖 4.9. (d) 模擬水平地層單炸點震測訊號偵測結果。

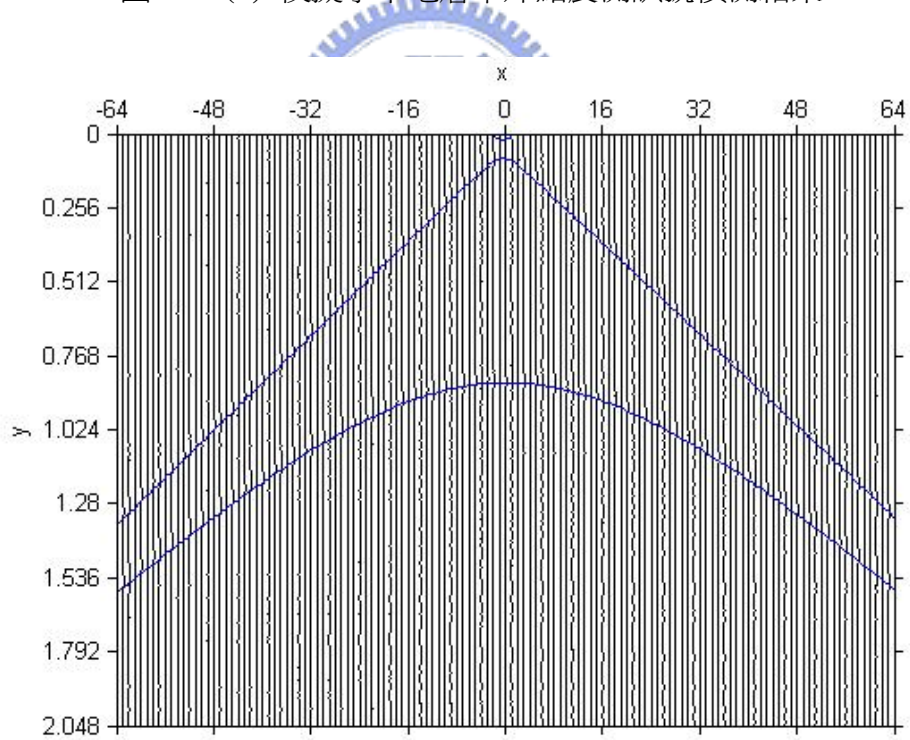


圖 4.9. (e) 模擬水平地層單炸點震測訊號偵測結果與原始訊號。

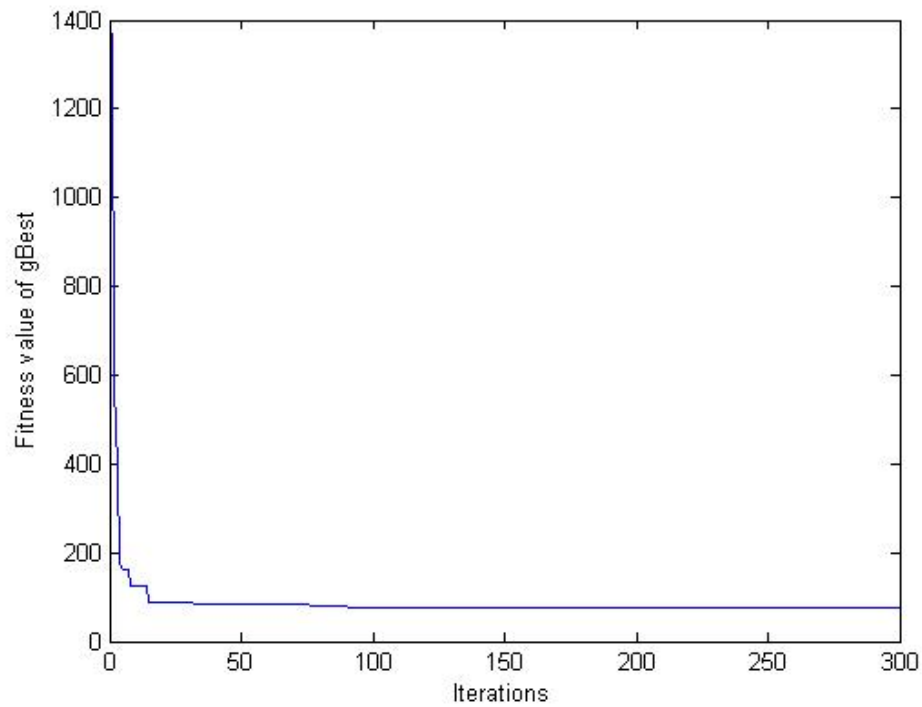


圖 4.9. (f) $gBest$ 的適應值 vs. iterations。

表 4.7. 模擬水平地層單炸點震測訊號圖參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
直接波	-0.24	-33.19	-4.83	0.20	6,636.69	-0.13°	112.21
反射波	-0.42	-12.93	5.02	-0.19	-11.04	-0.19°	

由震測訊號的公式，我們可以知道直接波的圖形，實際上為反射波圖形的漸近線，因此我們可以設定直接波圖形的係數 $r = 0$ ，同時它的方向是南北開口，可以設定 $\theta = 0^\circ$ 。因此對於直接波圖形，只需要 4 個參數 $[c_x \ c_y \ a \ b]$ 。我們以這樣設定的參數去偵測圖形，結果如圖 4.9(g)，直接波圖形的偵測結果為漸近線，4.9(h)為 $gBest$ 的適應值 vs. iterations。表 4.8 為直接波圖形使用 4 個參數的偵測結果。

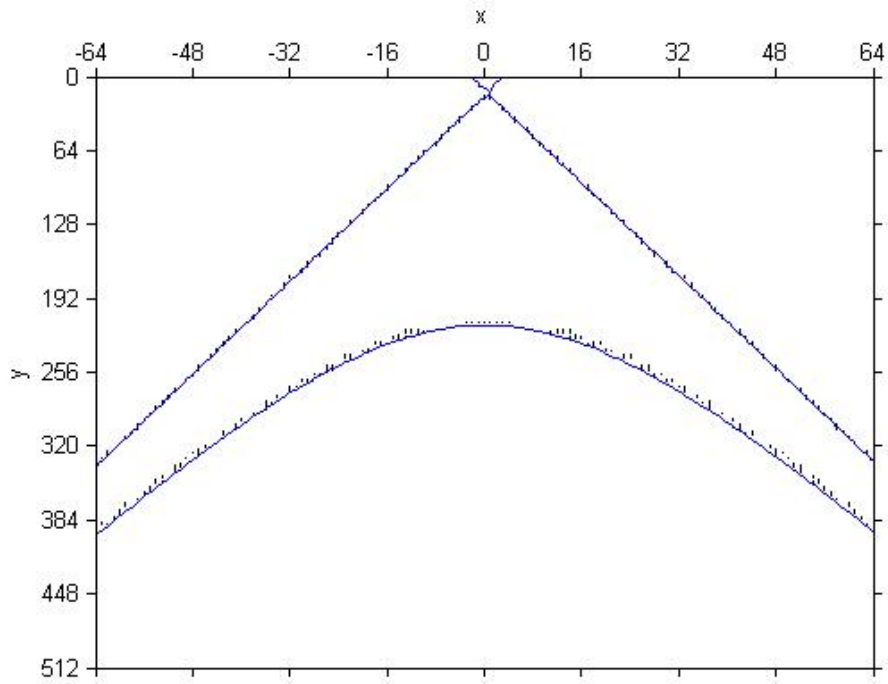


圖 4.9. (g) 直接波圖形使用 4 個參數的偵測結果。

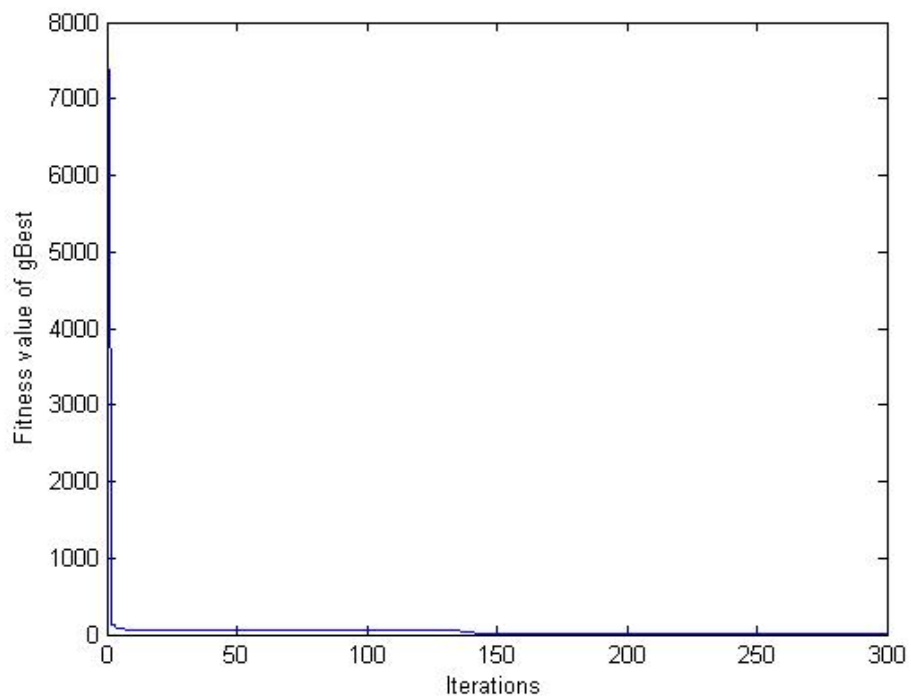


圖 4.9. (h) 直接波圖形使用 4 個參數的 $gBest$ 的適應值 vs. iterations。

表 4.8. 直接波圖形使用 4 個參數的偵測結果。

	c_x	c_y	a	b	r	θ	執行時間 (sec)
直接波	-0.5	-14.02	-5.02	0.19	0 (Preset)	0° (Preset)	92.43
反射波	-0.41	-12.85	5.00	-0.19	-11.23	-0.20°	

4.2.2 偵測傾斜地層單炸點震測訊號

傾斜地層單炸點震測訊號如圖 4.10 所示，共有兩層地層，反射層傾斜角度為 15° 。地面上 Source 為爆炸點，在爆炸點左右各有 64 個接收站，加上爆炸點處 1 個，共有 129 個接收站，每個接收站之間距離 Δx 為 50 meters，接收站與爆炸點之間距離 $x = k * \Delta x$ ， k 為接收站編號。接收站之 sampling interval 為 0.004 sec，總共有 512 個 samples。地表與反射層的垂直深度 h 為 2,500 meters，為爆炸點到地下地層的垂直深度。直接波與反射波速度 v 皆為 2,500 meters/sec，反射係數為 0.2。由圖 4.10 可知直接波及反射波的行進方向，由計算可知直接波訊號為直線，(4.3) 式為直接波訊號公式，反射波訊號為雙曲線，(4.4) 式為反射波訊號公式。

$$t = x / v \quad (4.3)$$

$$t = \frac{\sqrt{x^2 + (2h)^2} - 2 * x * \cos(90^\circ + \theta)}{v} \quad (4.4)$$

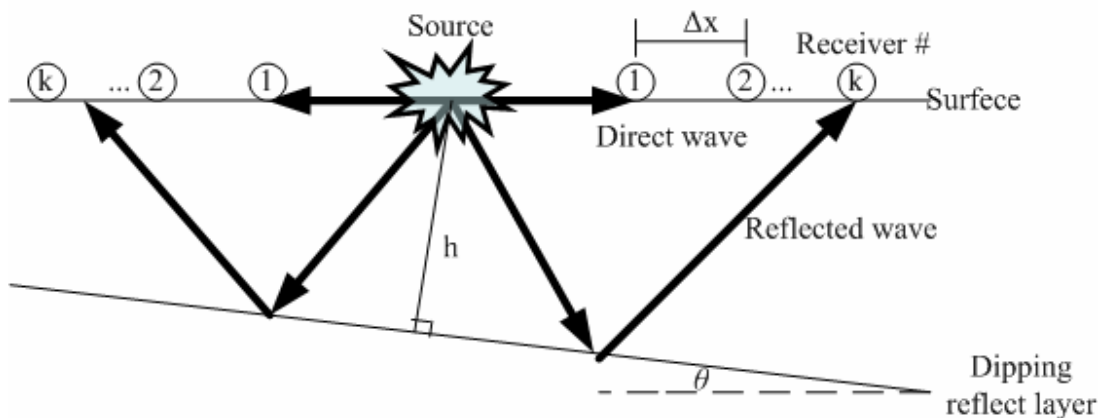


圖 4.10. 傾斜地層震測訊號。

我們計算出接收站接收到直接波與反射波的時間後，將訊號與 25 Hz 的 Ricker Wavelet 作 convolution，得到 impulse 訊號，再加上 random white band 10.2539 Hz 至 59.5703 Hz 的 noise，uniform distribution over interval (-0.2,0.2)。

最後得到的震測訊號如圖 4.11(a)。圖中間上方 0 點位置為爆炸點，橫軸為接收器的編號，左右各 64 個，縱軸由上往下，為每一個接收器接收到直接波與反射波訊號的時間。

我們將震測訊號作 Envelope 處理，得到圖 4.11(b)。接著將振幅超過 0.2 的點取出，共取出 352 點，作為偵測圖形系統的輸入，每個點的值是 $[x, i]$ ， x 是接收器的編號， i 是接收到訊號的時間除以 *sampling interval*，代表該接收站到接收到的第幾個訊號。我們得到的輸入點如圖 4.11(c)。我們設定偵測圖形數目為 2，偵測出兩條雙曲線，如圖 4.11(d)，將結果顯示在原始訊號上，如圖 4.11(e)，圖 4.11(f) 為 *gBest* 的適應值 vs. iterations。表 4.11 為參數偵測結果。

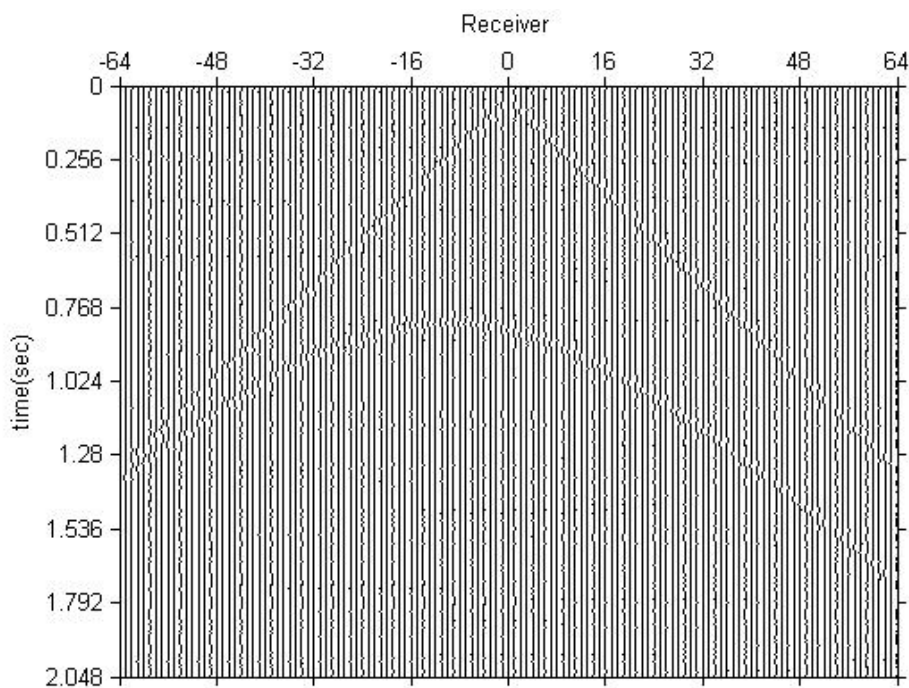


圖 4.11. (a) 模擬傾斜地層震測訊號。

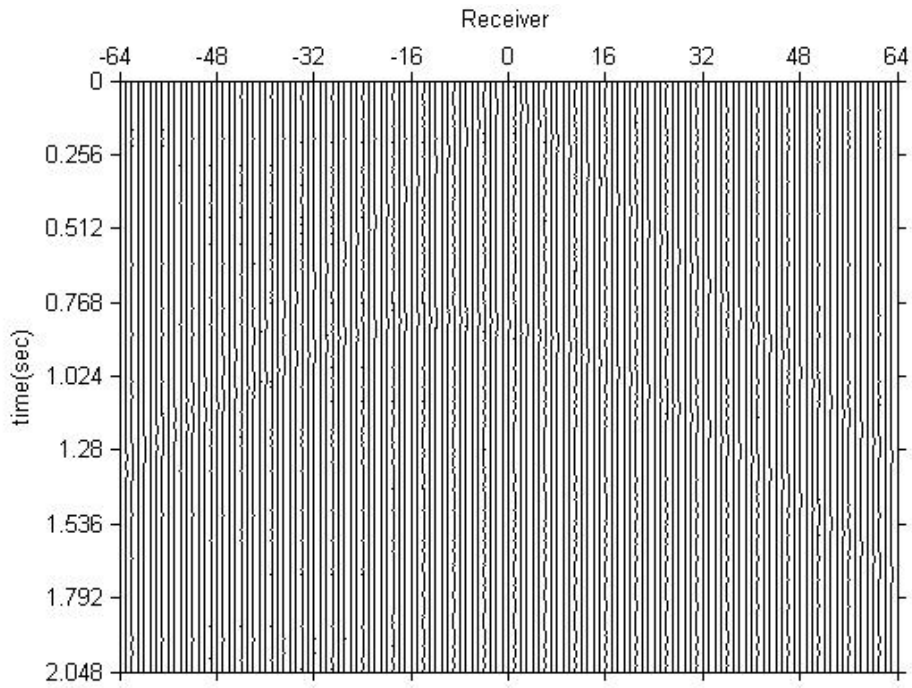


圖 4.11. (b) 模擬傾斜地層震測訊號圖經過 Envelope 處理。

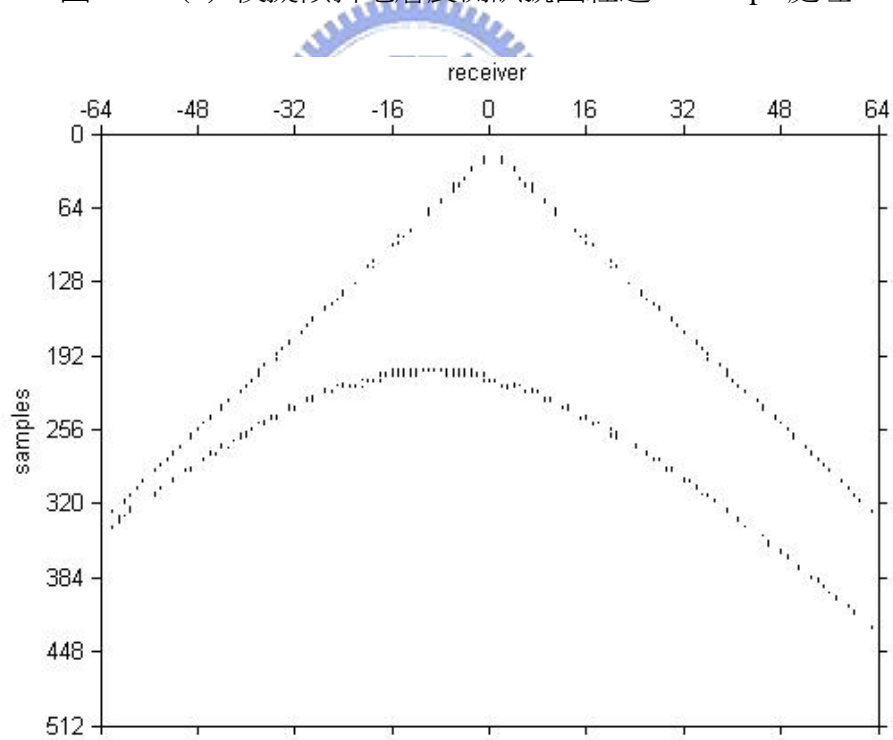


圖 4.11. (c) 經過 thresholding 後的模擬傾斜地層震測訊號輸入點。

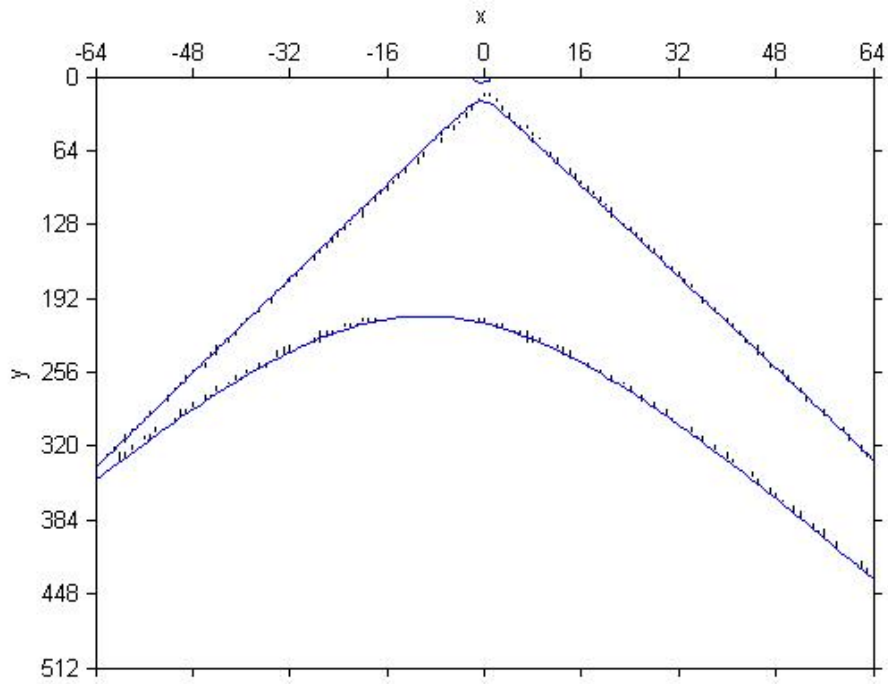


圖 4.11. (d) 模擬傾斜地層震測訊號偵測結果。

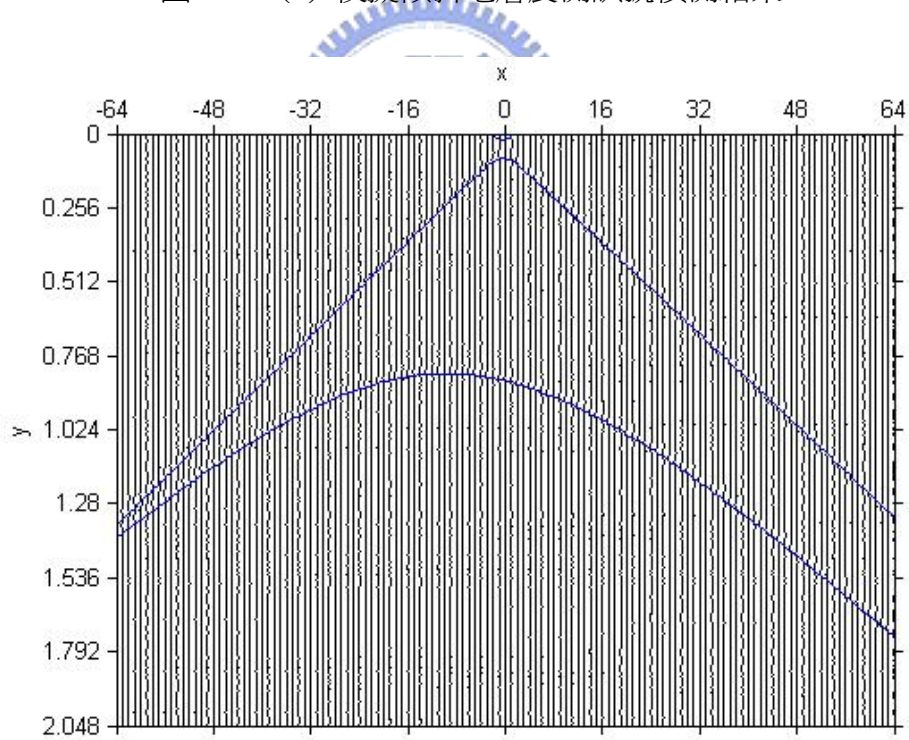


圖 4.11. (e) 模擬傾斜地層單炸點震測訊號偵測結果與原始訊號。

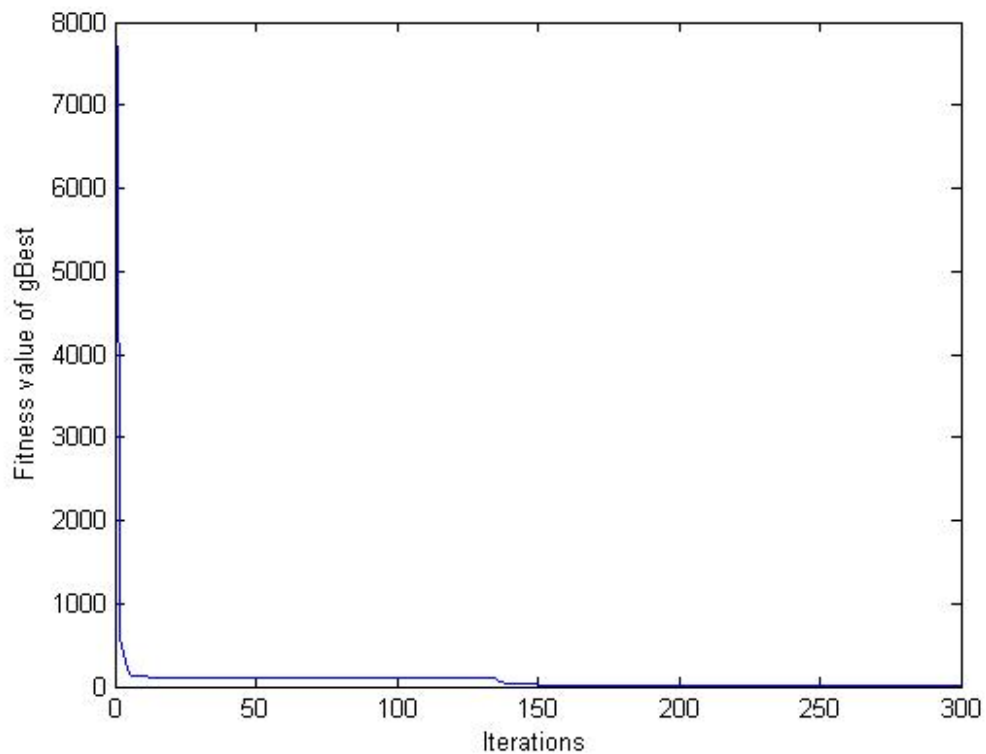


圖 4.11. (f) $gBest$ 的適應值 vs. iterations。

表 4.9. 模擬傾斜地層單炸點震測訊號圖參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間(sec)
直接波	-0.39	-13.02	5.08	-0.20	-12.15	0.13°	94.36
反射波	-9.34	-26.74	-4.89	-0.19	6,358.34	-0.09°	

同樣的，我們設定直接波的參數 $r = 0$ ， $\theta = 0^\circ$ 。因此對於直接波圖形，只需要 4 個參數 $[c_x \ c_y \ a \ b]$ 。我們以這樣設定的參數去偵測圖形，結果如圖 4.11(g)，直接波圖形的偵測結果為漸近線，4.11(h) 為 $gBest$ 的適應值 vs. iterations。表 4.10 為直接波圖形使用 4 個參數的偵測結果。

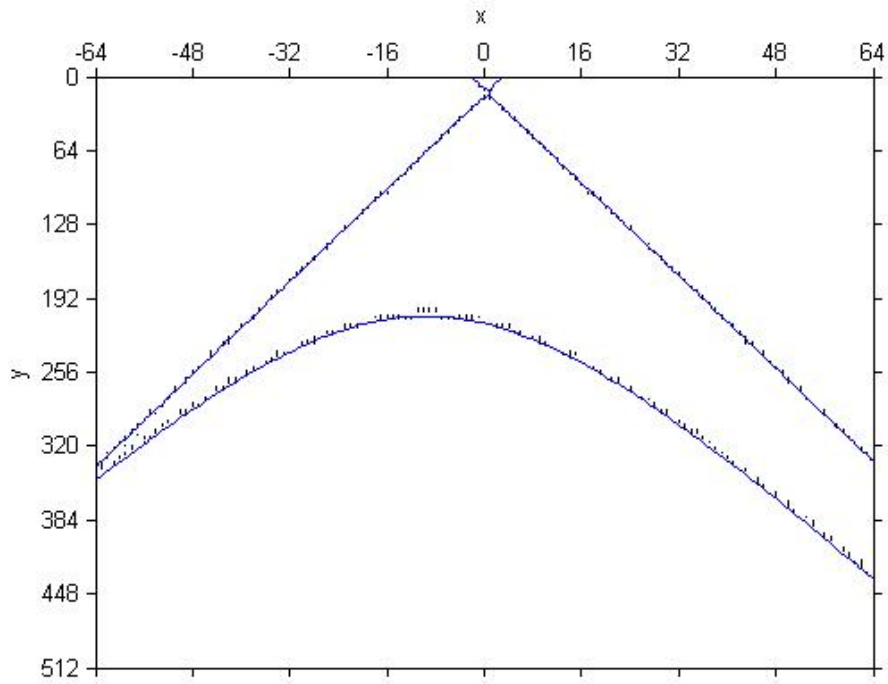
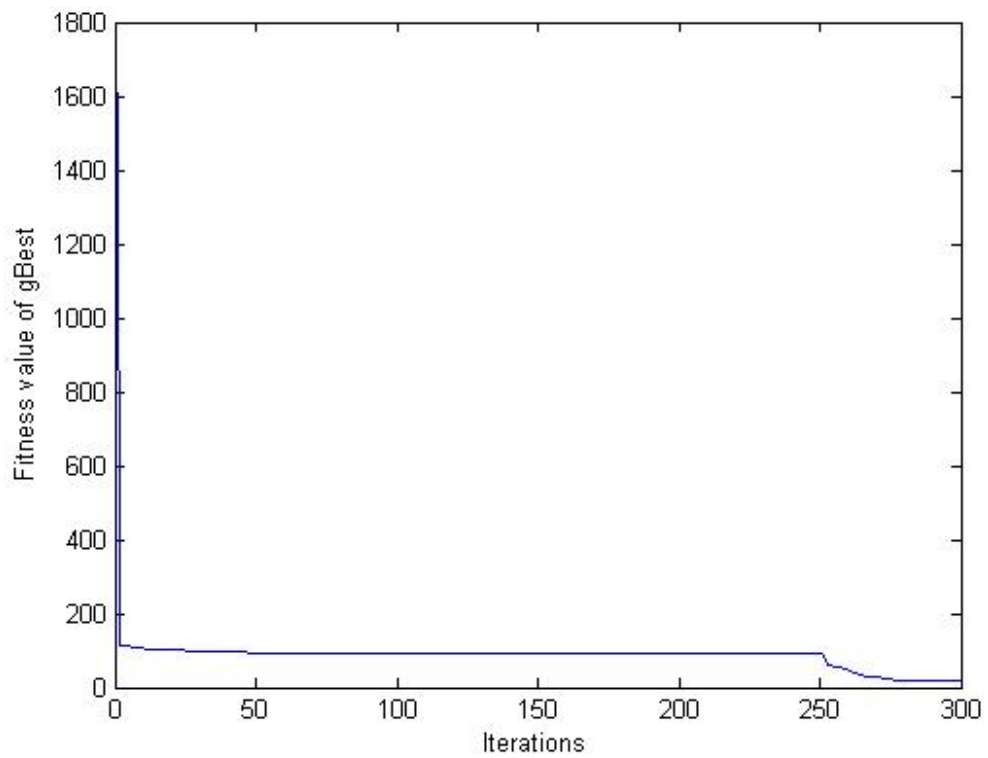


圖 4.11. (g) 直接波圖形使用 4 個參數的偵測結果。



4.11. (h) 直接波圖形使用 4 個參數的 $gBest$ 的適應值 vs. iterations。

表 4.10. 直接波圖形使用 4 個參數的偵測結果。

	c_x	c_y	a	b	r	θ	執行時間 (sec)
直接波	-0.51	-14.32	-5.34	0.22	0 (preset)	0° (preset)	97.23
反射波	-0.31	-12.14	5.12	-0.17	-12.43	-0.51°	

4.3 真實單炸點震測訊號圖形偵測

有兩張真實的水平地層單炸點震測訊號圖形，由 Colorado School of Mine 所建立的 Seismic Unix System 所取得，我們以同樣方法偵測其直接波與反射波訊號圖形。

4.3.1 偵測 Canadian Artic 的單炸點震測訊號

第一張震測訊號圖形地點來自 Canadian Artic，共有 48 個接收站，每個接收站接收 3,100 個訊號，接收站之 sampling interval 為 0.002 sec。原始的震測訊號圖形如圖 4.12(a)。橫軸由左到右為接收站的號碼，縱軸由上到下為接收到訊號的時間。

我們同樣的將訊號作 envelope 處理，如圖 4.12(b)。留下震幅超過 0.2 的 samples，此外我們去除底下多餘的圖形 (1.7 秒以下)，以免對偵測結果造成太大干擾。最後共得到 106 點，作為偵測系統的輸入點，如圖 4.12(c)。每個輸入點的值是 $[x, i]$ ， x 是接收器的編號， i 是接收到訊號的時間除以 sampling interval，代表該接收站到接收到的第幾個訊號。我們設定偵測圖形數目 $k = 3$ ，偵測出 3 條雙曲線，如圖 4.12(d)，將結果顯示在原始訊號上，如圖 4.12(e)。表 4.11 為震測訊號參數偵測結果。圖 4.12(f) 為 $gBest$ 的適應值與 iteration 的關係圖。

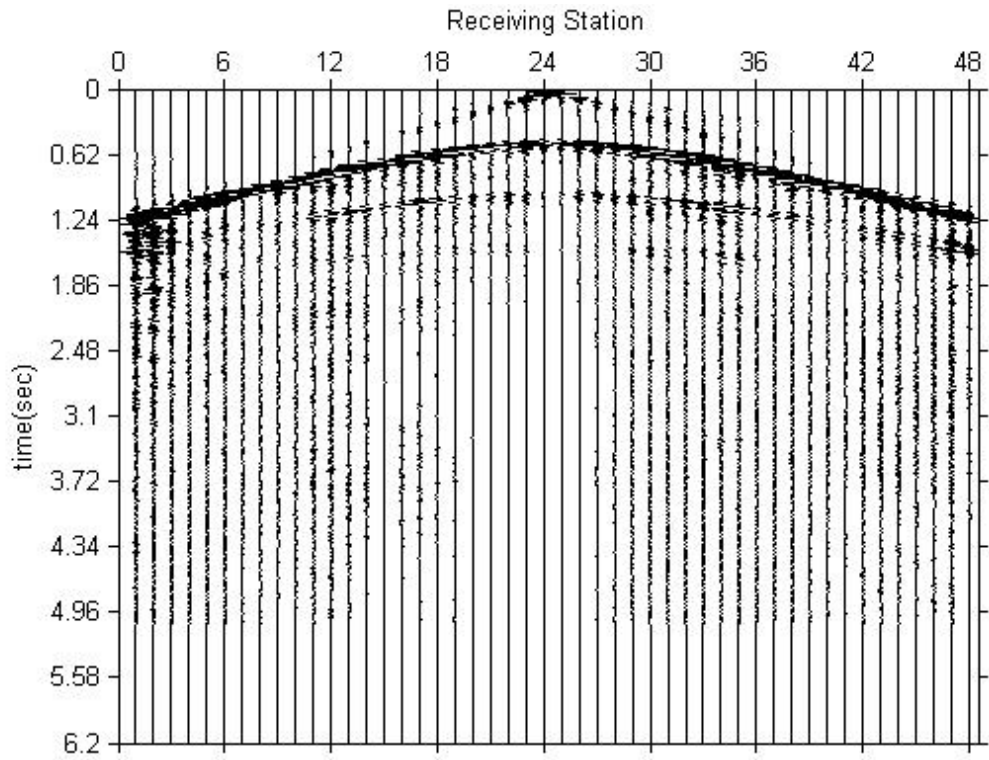


圖 4.12. (a) Canadian Arctic 的單炸點震測訊號。

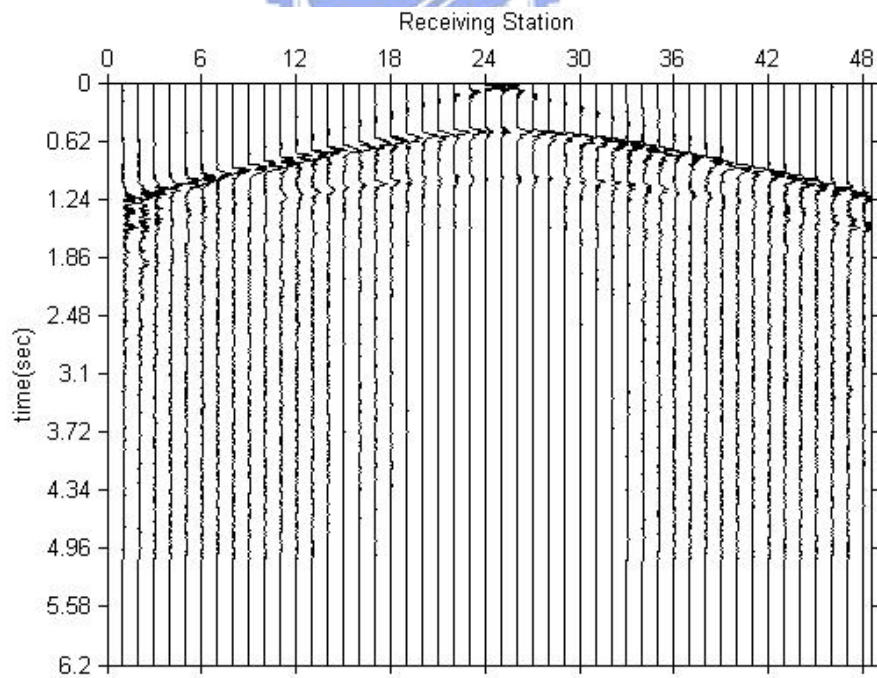


圖 4.12. (b) 震測訊號經過 Envelope 處理。

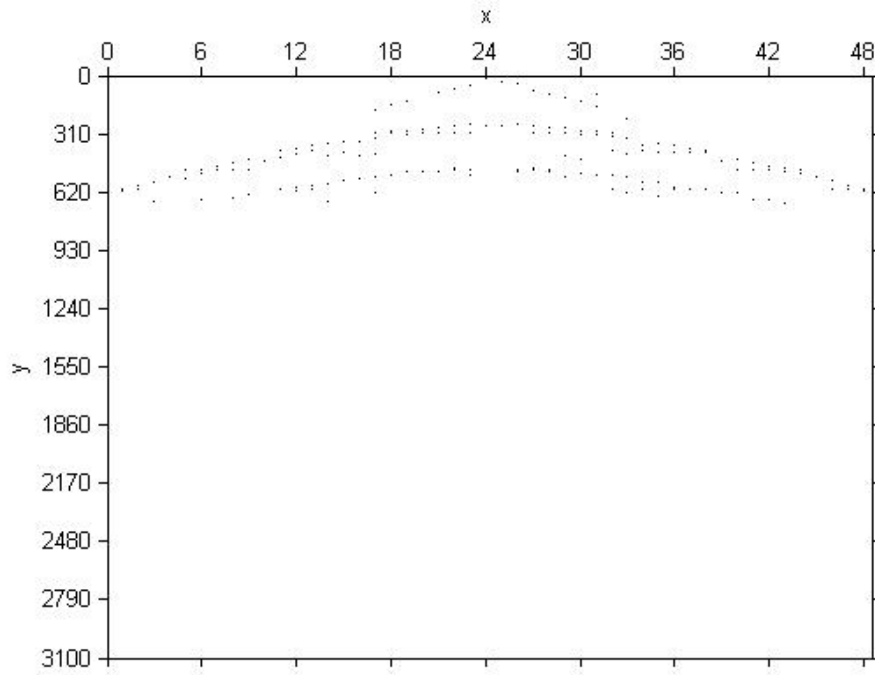


圖 4.12. (c) 經過 thresholding 後的偵測系統輸入點。

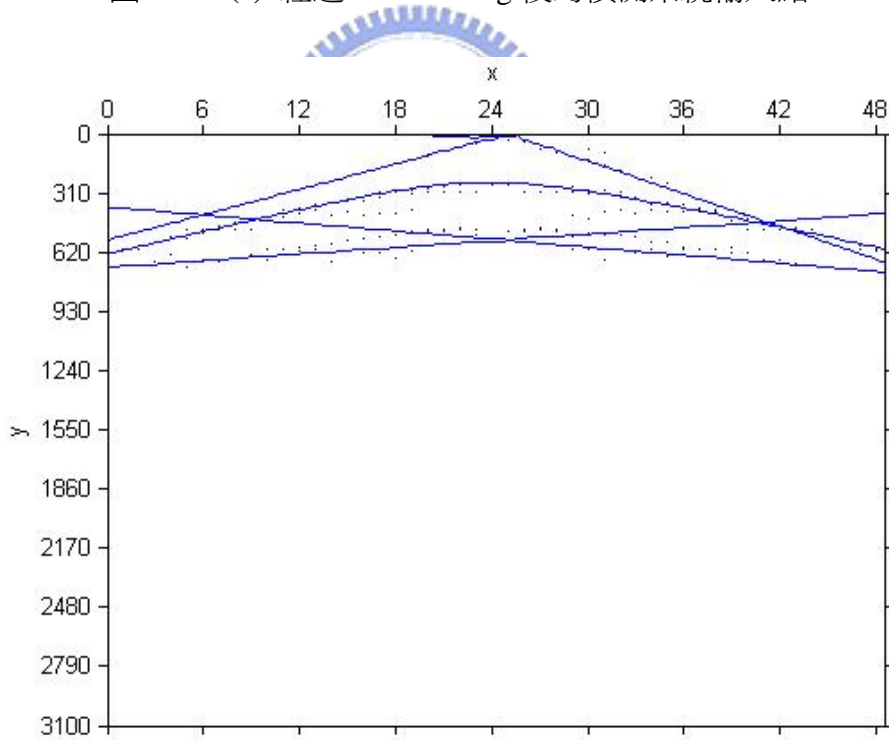


圖 4.12. (d) 震測訊號偵測結果。

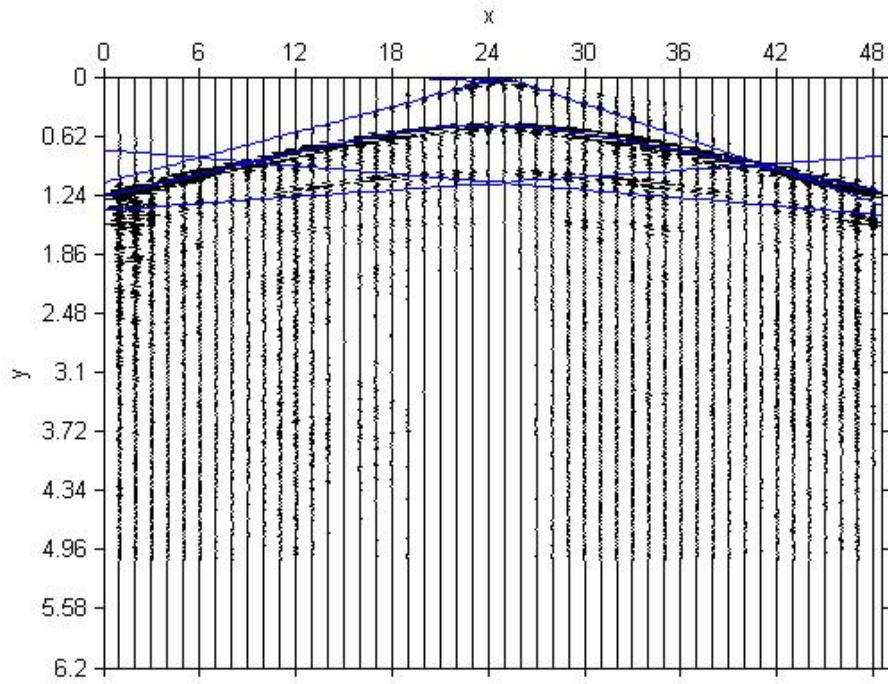


圖 4.12. (e) 震測訊號偵測結果與原始訊號。

表 4.11. 震測訊號參數偵測結果。

	c_x	c_y	a	b	r	θ	執行時間 (sec)
Pattern 1	23.35	-135.62	-19.11	0.05	714.82	-0.19°	219.69
Pattern 2	25.90	17.04	-26.85	0.03	0.01	0.43°	
Pattern 3	24.68	-556.45	0.15	-6.33	-2.19	90.80°	

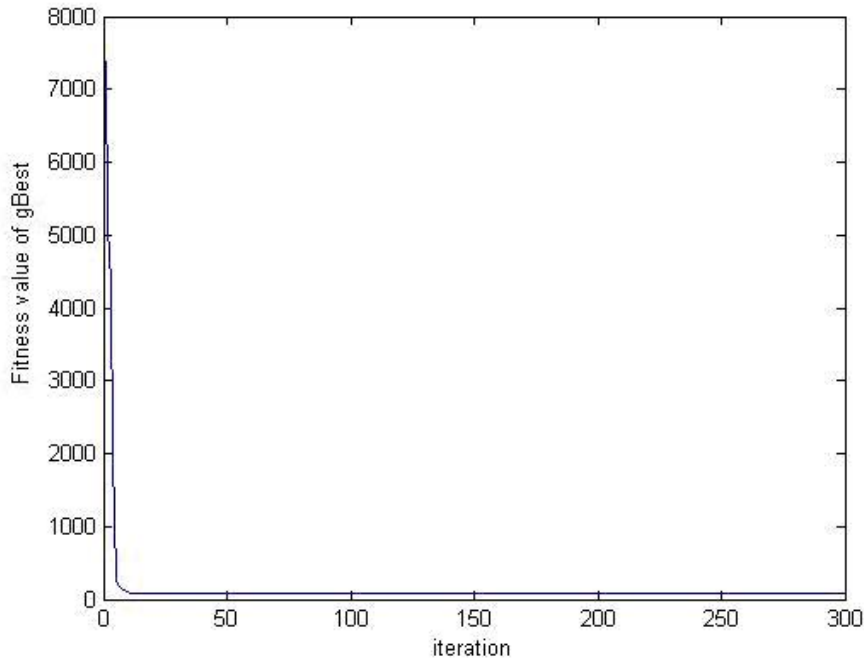


圖 4.12. (f) $gBest$ 的適應值 vs. iterations。

4.3.2 偵測 Gulf of Cadiz 的單炸點震測訊號

第二張震測訊號圖形地點來自 Gulf of Cadiz，共有 48 個接收站，每個接收站接收 2,050 個訊號，接收站之 sampling interval 為 0.004 sec。原始的震測訊號圖形如圖 4.13(a)，橫軸由左到右為接收站的號碼，縱軸由上到下為接收到訊號的時間。

我們同樣的將訊號作 envelope 處理，如圖 4.13(b)。留下震幅超過 0.2 的 samples，此外我們去除底下多餘的圖形 (3.075 秒以下)，以免對偵測結果造成太大干擾。最後共得到 136 點，作為偵測系統的輸入點，如圖 4.13(c)。每個輸入點的值是 $[x, i]$ ， x 是接收器的編號， i 是接收到訊號的時間除以 sampling interval，代表該接收站到接收到的第幾個訊號。我們設定偵測圖形數目 $k = 2$ ，偵測出兩條雙曲線，如圖 4.13(d)，比較下面的雙曲線的開口為南北向。將結果顯示在原始訊號上，如圖 4.13(e)。表 4.12 為震測訊號參數偵測結果。圖 4.13(f) 為 $gBest$ 的適應值與 iteration 的關係圖。

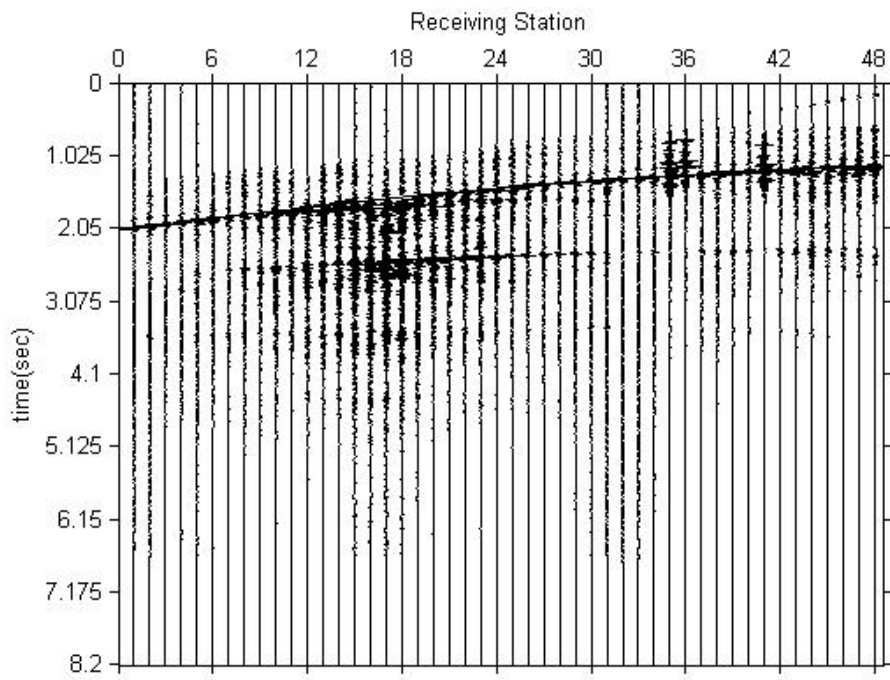


圖 4.13. (a) Gulf of Cadiz 的單炸點震測訊號。

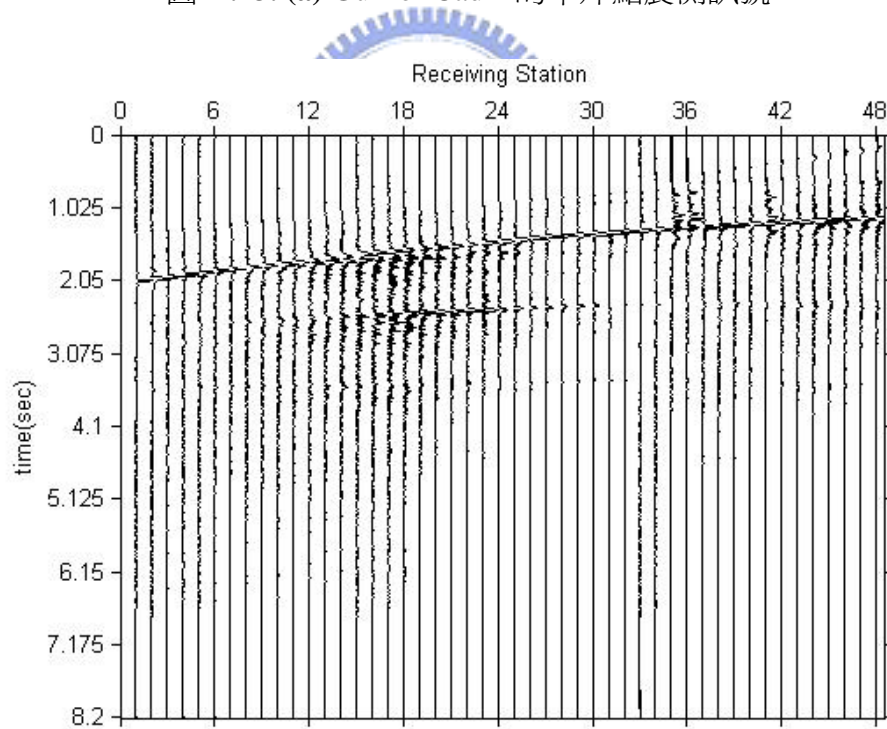


圖 4.13. (b) 震測訊號經過 Envelope 處理。

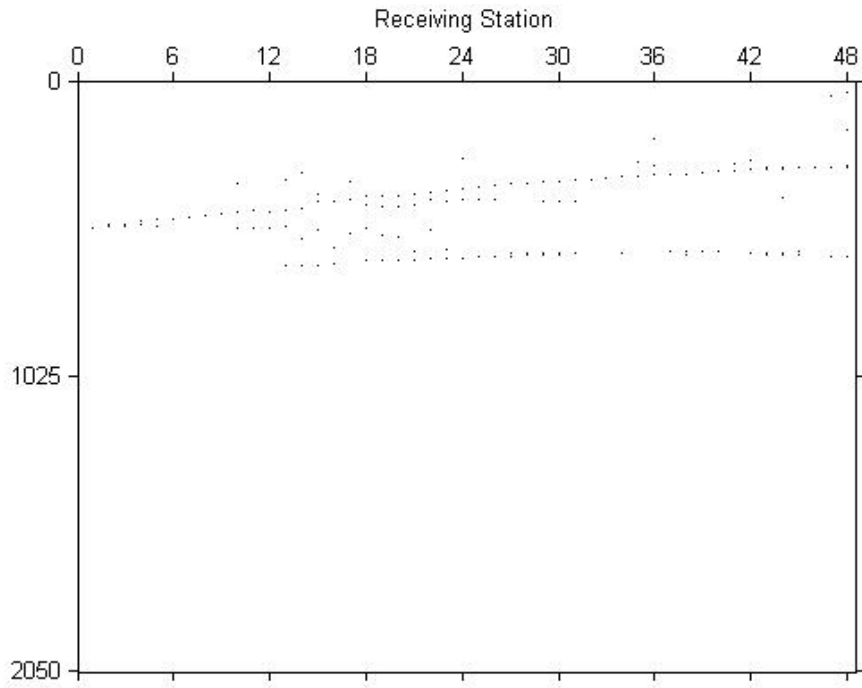


圖 4.13. (c) 經過 thresholding 後的偵測系統輸入點。

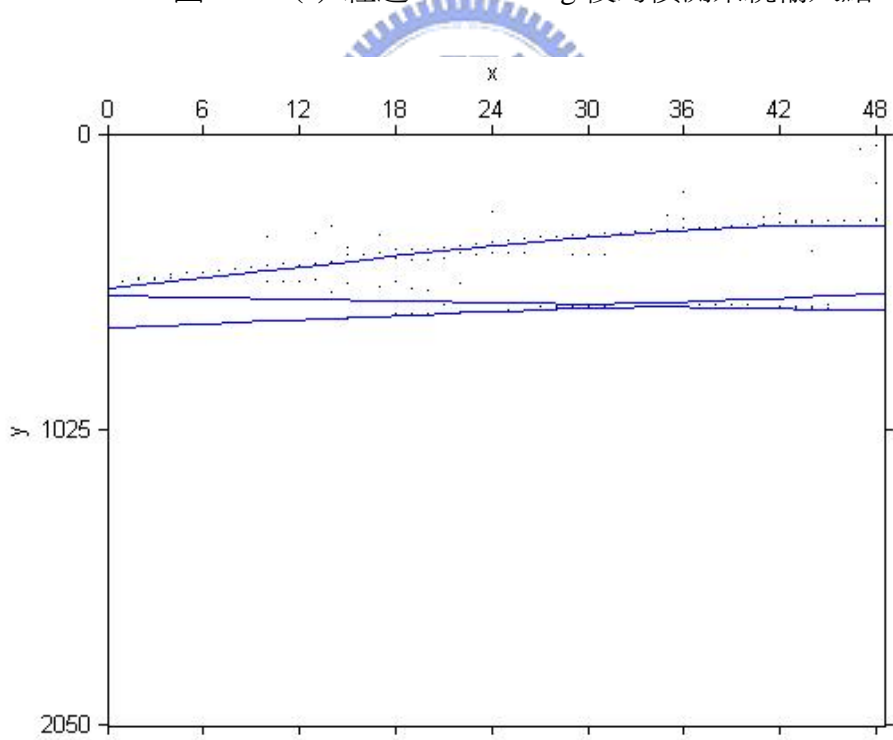


圖 4.13. (d) 震測訊號偵測結果。

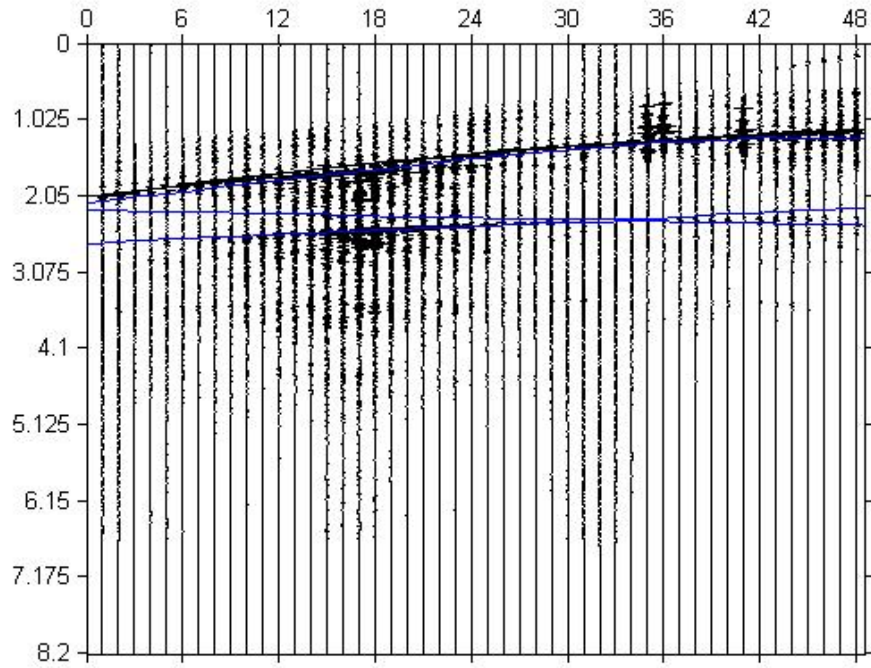


圖 4.13. (e) 震測訊號偵測結果與原始訊號。

表 4.12. 震測訊號參數偵測結果

	c_x	c_y	a	b	r	θ	執行時間 (sec)
Pattern 1	50.51	-163.26	8.26	0.10	232.10	-0.21°	95.02
Pattern 2	32.24	-594.45	1.54	-0.64	1.34	-80.48°	

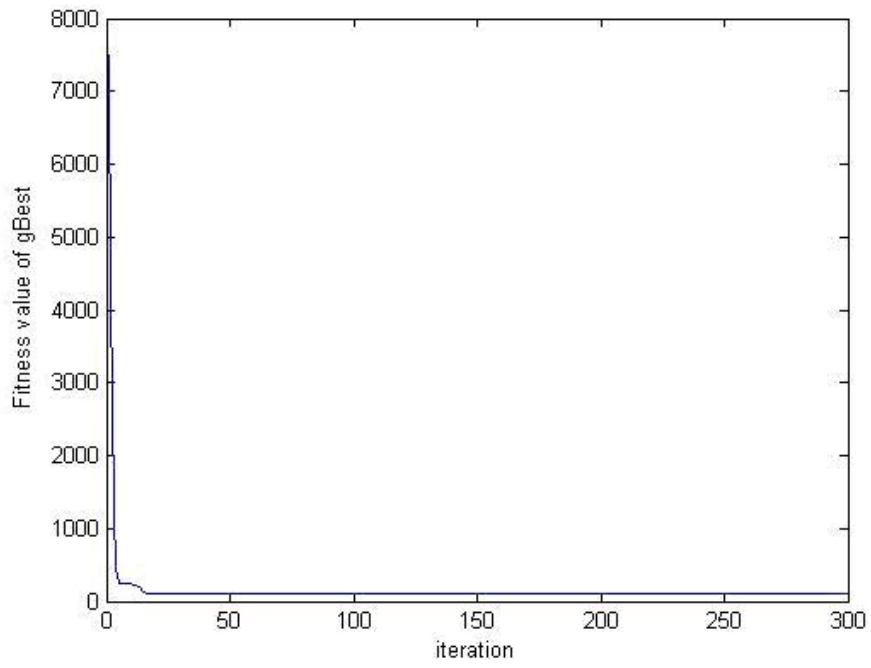


圖 4.13. (f) $gBest$ 的適應值 vs. iterations。



Chapter 5 結論

我們提出一種以粒子群演算法 (Particle Swarm Optimization, PSO) 為基礎的偵測系統，用於偵測參數圖形 (圓、橢圓、雙曲線、及雙曲線的漸近線) 的參數，包括圖形的中心點、大小、及傾斜角度。PSO 比起其它的最佳化方法簡單方便，同時也擁有廣域搜尋與尋找全域最佳解的能力。

我們定義了一個表示式來代表不同的參數圖形，並定義點到圖形的距離。影像上共有 n 個點，在多個圖形時，先計算一點 (x_i, y_i) 到 k 個圖形的個別距離 $d_{i,1}, d_{i,2}, \dots, d_{i,k}$ ，將這些距離相乘 $D_i = d_{i,1} * d_{i,2} * \dots * d_{i,k}$ ，作為一個點 (x_i, y_i) 到全部 k 個圖形的距離。同樣方法計算出每一個點到全部圖形距離 D_1, D_2, \dots, D_n ，再將所有的距離相加後除以全部的輸入點數目 n ， $f_j = (D_1 + D_2 + \dots + D_n) / n$ ，以此代表所有 n 個點到全部 k 個圖形的平均距離，並作為每一個粒子的適應值。我們以 PSO 中的一個粒子代表所有要偵測的 k 個參數圖形的參數，在參數空間中設定大量 m 個粒子，輸入欲偵測圖形上的點，計算出每個粒子的適應值 f_1, f_2, \dots, f_m ，再以找出適應值 (距離) 為最小的一個粒子，即為我們要偵測的所有 k 個圖形的參數。

我們測試了幾個模擬的參數圖形，並應用於偵測單炸點震測圖形的直接波 (直線) 與反射波 (雙曲線) 圖形上，包含模擬的單炸點震測訊號圖及真實的單炸點震測圖形，結果顯示 PSO 偵測系統能成功偵測出參數圖形的參數，有助於研究者對於震測訊號的進一步處理與解釋。

Chapter 6 討論

6.1 特殊圖形的參數偵測

6.1.1 單一特殊圖形的參數偵測

我們所提出的 PSO 參數偵測系統可用於偵測二次曲線中的圓、橢圓、雙曲線及雙曲線的漸近線，然而對於其他的參數圖形，如拋物線或其他的任意曲線，我們無法以 (3.5) 式來表示，也無法計算點到圖形的距離，因此無法以 PSO 參數偵測系統去偵測。

由於我們是以 (3.5) 式來表示參數圖形，因此對於一個直線的圖形，我們會將它視作爲一個雙曲線的漸近線的其中之一，因此偵測結果會是漸近線或是相當接近的雙曲線，會多出額外的偵測結果。因此如果我們已經知道所要偵測的圖形是單一的直線，可以改用直線的參數式，如 (6.1.1) 式，來定義直線的圖形，其中 ρ 爲直線到原點的法線距離， θ 爲法線與 x 軸的夾角。並以 (6.1.2) 式來定義點 (x_i, y_i) 到直線的距離，將粒子設定爲 $[\rho \ \theta]^T$ 。如此偵測的結果，會是單純的一條直線，沒有額外的偵測結果，如圖 6.1，表 6.1 爲原始輸入參數，表 6.2 以直線參數式定義圖形的偵測結果。

$$\rho = x \cos \theta + y \sin \theta \quad (6.1.1)$$

$$d_i = |x_i \cos \theta + y_i \sin \theta - \rho| \quad (6.1.2)$$

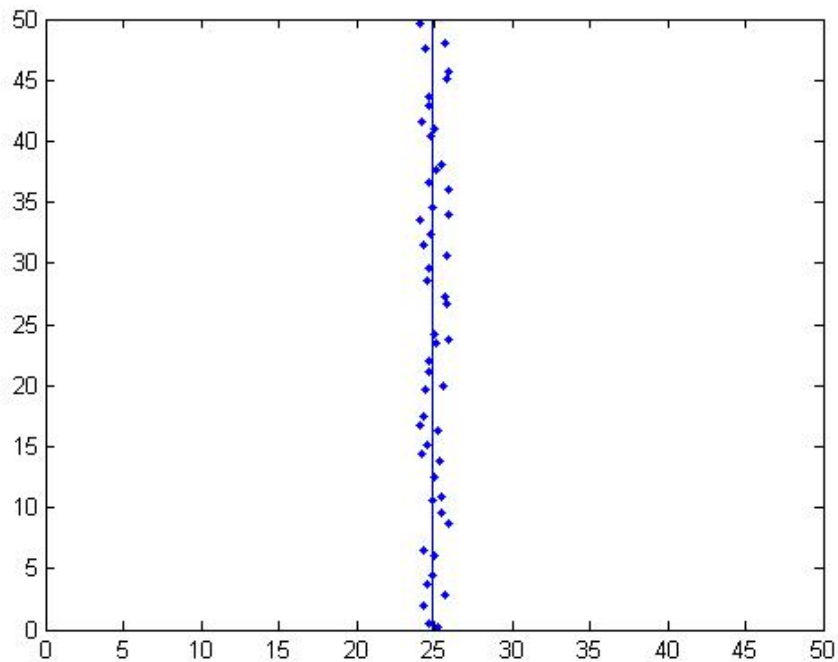


圖 6.1 以直線參數式來定義圖形的偵測結果。

表 6.1 直線的輸入參數。

$x = 25$

表 6.2 以直線參數式定義圖形的偵測結果。

ρ	θ
24.96	0.09°

同樣的，圓與橢圓可以用方程式 (6.2.1) 式來表示。 (c_x, c_y) 為中心座標， (a, b) 為長軸與短軸的長度。當圖形為圓形時， $a = b$ 且 $c = 0$ 。當圖形為圓形或是沒有旋轉的橢圓時，係數 $c = 0$ 。對於輸入影像中一點 (x_i, y_i) 到此圖形的距離 d_i 則定義為 (6.2.2) 式。粒子則設定為 $[c_x \ c_y \ a \ b \ c]^T$ 。

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} + cxy = 1 \quad (6.2.1)$$

$$d_i = \left| \frac{(x_i - c_x)^2}{a^2} + \frac{(y_i - c_y)^2}{b^2} + cx_i y_i - 1 \right| \quad (6.2.2)$$

雙曲線有兩種形式，左右開口的雙曲線可以用 (6.3.1) 式來表示，上下開

口的雙曲線以 (6.3.2) 式來表示。 (c_x, c_y) 為雙曲線中心， (a, b) 為實軸與虛軸的長度。當圖形是沒有旋轉的雙曲線時，係數 $c = 0$ 。對於輸入影像中一點 (x_i, y_i) 到雙曲線圖形的距離 d_i 則可以表示為 (6.3.3) 式及 (6.3.4) 式。粒子則設定為 $[c_x \ c_y \ a \ b \ c]^T$ 。

$$\frac{(x - c_x)^2}{a^2} - \frac{(y - c_y)^2}{b^2} + cxy = 1 \quad (6.3.1)$$

$$-\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} + cxy = 1 \quad (6.3.2)$$

$$d_i = \left| \sqrt{\frac{(x_i - c_x)^2}{a^2} - \frac{(y_i - c_y)^2}{b^2} + cx_i y_i} - 1 \right| \quad (6.3.3)$$

$$d_i = \left| \sqrt{-\frac{(x_i - c_x)^2}{a^2} + \frac{(y_i - c_y)^2}{b^2} + cx_i y_i} - 1 \right| \quad (6.3.4)$$

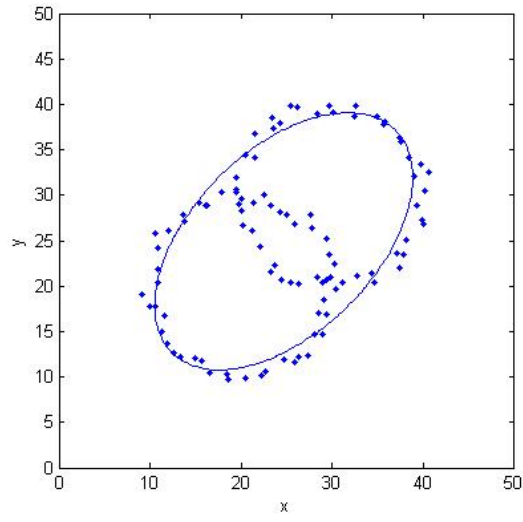
6.1.2 兩個以上特殊圖形的參數偵測

如果影像中有兩個以上的不同的參數圖形，例如一條直線與一個沒有旋轉的橢圓，則可以設定一個粒子為 $[\rho \ \theta \ c_x \ c_y \ a \ b]^T$ ，因為是沒有旋轉的橢圓，所以係數 $c = 0$ ，全部只需要考慮 6 個參數。

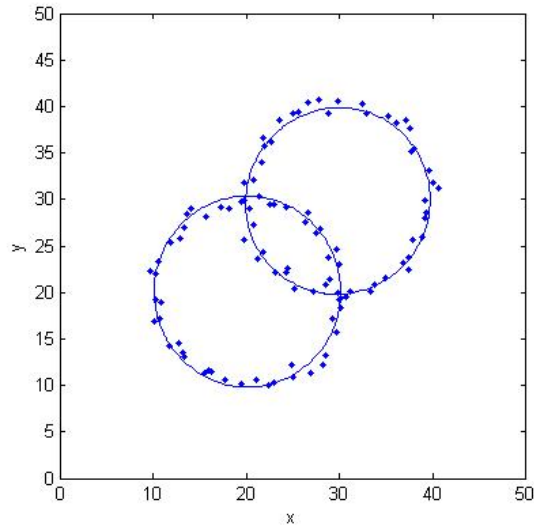
我們以 (6.1.2) 式去計算一個點 (x_i, y_i) 到直線的距離，得到 $d_{i,1}$ ；以 (6.2.2) 式去計算點到橢圓的距離，得到 $d_{i,2}$ 。計算 $D_i = d_{i,1} * d_{i,2}$ ，為一個點到全部圖形的距離。同樣方法計算出每一個點到全部圖形距離 D_1, D_2, \dots, D_n ，再將所有的距離相加後除以影像上點的數目 n ， $f_j = (D_1 + D_2 + \dots + D_n) / n$ ，以此代表所有點到這兩個圖形的平均距離，作為這個粒子的適應值。接著同樣設定大量的粒子，以 PSO 的方法去找出距離最小的一個粒子，即為我們要偵測的直線與橢圓的參數。

6.2 圖形數目的設定

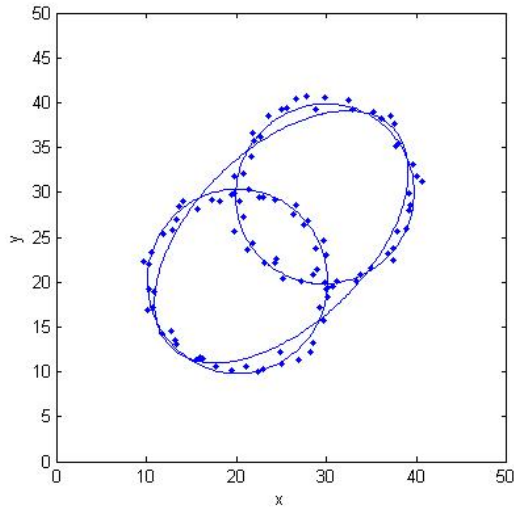
對於參數圖形數目的判斷，我們目前只能由改變不同的數目，計算出每一次實驗的適應值來判斷。如圖 6.2 與圖 6.3 所示，設定圖形數目 $k = 2$ 時，適應值最低，因此我們判斷圖形數目 $k = 2$ 。



(a) $k = 1$, 適應值 = 59.61 °。



(b) $k = 2$, 適應值 = 3.78 °。



(c) $k = 3$ ，適應值 = 23.43。

圖 6.2 設定不同圖形個數的偵測結果與適應值。

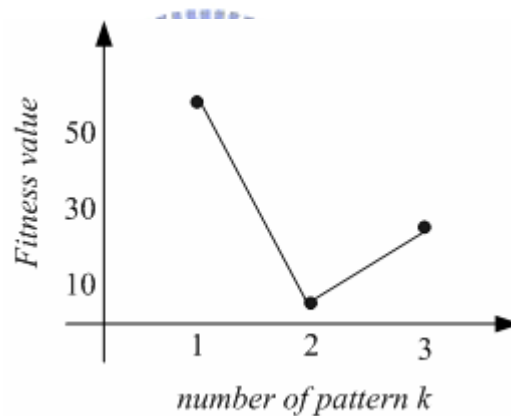


圖 6.3 不同的圖形個數 k 與適應值。

6.3 粒子數目的設定

關於粒子的數目與收斂速度，由於收斂的速度與初始的粒子分佈有很大的關係，如果一開始就有粒子很靠近最佳解位置，很快的，粒子就會收斂到最佳解位置附近。如果粒子都距離最佳解位置很遠，則需要多次的演化之後，粒子才有可能到達最佳解的位置。由於我們不知道最佳解的範圍，所以隨機分佈粒子在空間中，由機率上來說，粒子的數目越多，一開始就有粒子很靠近最佳解位置的機率也越高。但是由於粒子數目多，計算量也就越大，程式執行時間也會增加。因此如何在粒子數目和收斂速度兩者間找到平衡點是未來研究的課題。

6.4 學習參數的設定

關於學習參數的設定， α 為 $gBest$ 方向的學習參數， β 為 $pBest$ 方向的學習參數。在我們的實驗中，設定 $\alpha = \beta = 2$ ，從演化開始到結束都不會改變，有可能會使得粒子一直被 $gBest$ 方向的力量吸引，無法跳脫 $gBest$ 的範圍，不能達到廣域搜尋的能力。因此我們想法是隨時間改變學習參數，一開始將 α 設定較大，然後隨著演化次數漸漸變小。相反的在一開始將 β 設定較小，然後隨著演化次數漸漸變大。這樣使得粒子在演化初期 $gBest$ 方向的變動量大，達到廣域搜尋的目的；在演化後期，粒子都相當靠近時， $pBest$ 影響力變大，能有細部搜尋的能力，在未來的研究中可加入此一考慮因素。



References

1. P. V. C. Hough, "Method and means for recognizing complex pattern," U.S. Patent 3069654, 1962.
2. R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Comm. Assoc. Comput. Mach.*, vol. 15, pp. 11-15, 1972.
3. L. Xu., E. Oja, and P. Kultanena, "A new curve detection method : Randomized Hough transform (RHT)," *Pattern Recognition*, vol. 11, no. 5, pp. 331-338, 1990.
4. A. R. Hare and M. B. Sandler, "General test framework for straight-line detection by Hough transform," *IEEE International Symposium on Circuits and Systems, ISCAS'93*, vol. 1, pp. 239-242, 1993.
5. S. J. Perantonis, B. Gatos, and N. Papamarkos, "Block decomposition and segmentation for fast Hough transform evaluation," *Pattern Recognition*, vol. 32, pp. 811-824, 1999.
6. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.
7. J. Basak and A. Das, "A Hough transform network: learning conoidal structures in a connectionist framework," *IEEE Trans. On Neural Networks*, vol. 13, no.2, pp. 381-392, 2002.
8. K. Y. Huang, K. S. Fu, T. H. Sheen, and S. W. Cheng, "Image Processing of Seismograms: (A) Hough Transformation for the detection of seismic pattern. (B) Thinning processing in the seismogram," *Pattern Recognition*, vol. 18, no. 6, pp. 429-440, 1985.
9. K. Y. Huang, J. D. You, C. J. Chen, H. L. Lai, and A. C. Dong, "Hough transform neural network for seismic pattern detection," *International Joint Conference on Neural Networks (IJCNN)*, July 16-21, Vancouver, pp.2453-2458, 2006.
10. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ. pp. 1942-1948, 1995.
11. X. Hu, "Particle swarm optimization: An introduction and bibliography," <http://www.swarmintelligence.org/> , 2006.
12. K. P. Wang and L. Huang, "Particles swarm optimization for traveling salesman problem," *International Conference on Machine Learning and*

Cybernetics, vol. 3, pp 1583-1585, 2003.

13. Y. Fukuyama, S. Takayama, Y. Nakanishi and H. Yoshida, "A particle swarm optimization for reactive power and voltage control in electric power systems," Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, Florida, pp. 1523-1528, 1999.
14. R. C. Eberhart and Y. Shi "Tracking and optimizing dynamic systems with particle swarms," Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, pp. 94-97, 2001.
15. D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press Piscataway, NJ, 1995.
16. J. Kennedy, "Minds and cultures: particle swarm implications," Socially Intelligent Agents: Papers from the 1997 AAAI Fall Symposium, Menlo Park, CA. pp. 67-72, 1997.
17. R. C. Eberhart and Y. Shi, "Evolving artificial neural networks," Proceedings of International Conference on Neural Networks and Brain, Beijing, pp. PL5-PL13, 1998.
18. E. Ozcan and C. K. Mohan, "Analysis of a simple particle swarm optimization system," Intelligent Engineering Systems Through Artificial Neural Networks, pp. 253-258, 1998.
19. A. P. Engelbrecht and A. Ismail, "Training product unit neural networks," Stability and Control: Theory and Applications, vol. 2, no. 1-2, pp. 59-74, 1999.
20. A. Ismail and A. P. Engelbrecht, "Training product units in feedforward neural networks using particle swarm optimization," Proceedings of the International Conference on Artificial Intelligence, Durban, pp. 36-40, 1999.
21. F. Van den Bergh, "Particle swarm weight initialization in multi-layer perceptron artificial neural networks," Development and Practice of Artificial Intelligence Techniques, Durban, pp. 41-45, 1999.
22. Z. He, C. Wei, L. Yang, X. Gao, and S. Yao, "Extracting rules from fuzzy neural network by particle swarm optimization," Proceedings of IEEE Congress on Evolutionary Computation (CEC 1998), Anchorage, Alaska, pp.74-77, 1998.
23. R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, San Diego, CA, vol. 1447, pp. 611-616, 1998.
24. Y. Shi and R. C. Eberhart "A modified particle swarm optimizer," IEEE World Congress on Evolutionary Computation Proceedings, pp. 69-73, 1998.

25. J. J. Liang and A. K. Qin, "Particle swarm optimization algorithms with novel learning strategies," IEEE international Conference on System, pp. 3659-3664, 2004.
26. P. J. Angeline, "Using selection to improve particle swarm optimization," IEEE International Conference on Evolutionary Computation Proceedings, pp. 84-89, 1998.
27. J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," IEEE Congress on Evolutionary Computation, vol. 3, pp. 1938, 1999.
28. X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," IEEE Congress on Evolutionary Computation, pp.1666-1670, 2002.
29. M. T. Taner, F. Koehler, and R. E. Sheriff, "Complex seismic trace analysis," Geophysics, vol.44, no.6, pp. 1041-1063, June, 1979.
30. M. M. Slotnick, *Lessons in Seismic Computing*, The Society of Exploration Geophysicists, Tulsa, Oklahoma, 1959.

