

東海大學資訊工程學系研究所

碩士論文

Department of Computer Science

Tunghai University

以粒子群最佳化演算法加速之 Zigbee Access Point 布置規

劃

Zigbee Access Point Placement Planning Acceleration by

Particle Swarm Optimization

研究生：陳信文

Graduate Student : Shien-Wen Chen

指導教授：石志雄博士

Advisor : Chihhsiong Shih, Ph. D.

中華民國一〇二年一月

January, 2013

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 陳 信 文 所提之論文

以粒子群最佳化演算法加速之 Zigbee Access

Point 布置規劃

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人

應博安

簽章

委

員

孔崇旭

指 導 教 授

石志在

簽章

中華民國 102 年 1 月 15 日

## 摘要

利用無線電波的隨著傳輸距離而有不同程度能量衰減的特性，來達到定位目的，為室內定位可行性方法之一，而此定位參數即為已知的接收信號強度指標(Received Signal Strength Indicator, RSSI)。

地區特徵指紋定位法是常用的 Zigbee 室內定位程序。其原理是預先量測生活場地中之 RSSI，並建成一個指紋資料庫。當上線使用時，由 AP 之 RSSI 值，比對資料庫之每點位置之訊號強度，進而找到其真實位置。

然而 AP 擺放位置，對地區特徵指紋定位法比對之精確度有很大影響，本研究探討 AP 位置選定，使用粒子群最佳化演算法 (Particle Swarm Optimization, PSO) 找出最佳位置，分別從一維理論延伸至二維平面 AP 之分布，來取得最佳的覆蓋率以及辨識率。

在本文中，提出三種不同的 PSO 一維設計方法，方法一是一次移動一個 AP，計算每個位置的值後再利用 PSO 找到最佳位置，方法二是一次移動一個 AP，不計算各個位置，直接利用 PSO 找到最佳位置，方法三是直接將 AP 分布，利用 PSO 直接尋找最佳位置，並將其中兩種方法擴展到二維，用以加速電腦模擬 AP 最佳化的執行時間。

最後，將兩種方法與原來的的方法相比較，方法三方法的確能大幅縮短電腦模擬 AP 最佳化的執行時間，使這定位系統趨近於成熟，並且

依據其模擬結果於電腦教室佈置定位系統，進行實際定位，未來將可使用於大眾生活上，讓科技生活變得更便利。

關鍵字: 粒子群最佳化演算法 (Particle Swarm Optimization,PSO)，

Zigbee，室內定位

## Abstract

Indoor localization have been a major issue among the many popular mobile applications, such as personnel or cargo tracking. One of the feasible methods is to utilize the attenuation characteristics of radio wave with the transmission distance to achieve the positioning purpose. This positioning parameter is the well known receive signal strength indicator (Received Signal Strength Indicator, RSSI) of Zigbee.

The location fingerprinting is commonly used in the Zigbee indoor positioning method. The concept is through pre-measured footprints of the RSSI values at preset locations constituting a fingerprint database. When in on-line user mode, the user's true locations can be acquired through comparing the ratio of the signal strength of different access point(AP) with those in the database.

However, in the real world of radio transmission, the signal strength values (RSSI) are easily subjected to noises from surrounding environments, thus affecting the differentiating rate of finger print method. The locations of Access Points(APs) will affect the accuracy of location fingerprinting. One extreme is to place all APs in one location and causing all the signals to be the same, and target hard to track. So, this study was to investigate the AP position selections using particle swarm optimization algorithm (Particle Swarm Optimization, PSO) to find the optimal positions. The study starts from one-dimensional theory, then extends to the AP distribution on the two-dimensional plane to achieve the best coverage and recognition rate.

In this paper, three one-dimensional PSO design methods are proposed. The first method is to move one AP at a time, then calculate the value of each grid position, then find the best position by use PSO. And the second method is to move one AP at one time, but find the best position by using PSO directly. The third method is through using PSO for direct distribution of all APs, to find the best position combinations of all the APs. Due to the excellent performance of the third method to accelerate the optimized execution time of simulation, the third method is chosen to extend from one-dimension to the two-dimension AP distribution.

Third method is indeed able to substantially shorten the computer simulation time to optimize the execution time of AP. A computer classroom is selected for testing the simulation results. The positioning

system based on the simulation results have shown a reduced position estimation error in both average and total scales. Our method is thus proved feasible and can be used in real world localization applications.

## 致謝

首先要感謝我的指導教授 石志雄博士，在我就讀研究所時間給予許多指導與建議，讓我不僅在研究方面以及處事態度上接獲益良多，同時感謝台中教育大學資訊工程系 孔崇旭教授與中正大學資訊工程系 熊博安教授，在百忙之中撥空擔任口試委員，給予學生寶貴的意見，使本論文更加充實完備。

另外，要感謝實驗室的正曜學長、博丞學長在論文方面給了我很多靈感與建議，有你們的幫助才有這篇論文的誕生。

最後，我要感謝我的家人，在就讀研究所時間給予大量的支持與鼓勵，也因此才能完成此論文。

陳信文

中華民國一〇二年一月

## 目錄

摘要.....	II
Abstract.....	IV
致謝.....	VI
圖錄.....	3
論文架構.....	5
第一章 緒論.....	6
1.1.研究動機：.....	6
1.2.擬探討解決之問題描述：.....	7
第二章 文獻探討.....	8
2.1. ZigBee 室內定位.....	8
2.2.粒子群最佳化演算法 ( Particle Swarm Optimization ) .....	9
2.3.高斯混合模型(Gaussian Mixtured Model, GMM).....	12
2.4.最佳化室內定位.....	13
第三章 研究方法.....	15
3.1.接收信號強度指標(Received Signal Strength Indicator, RSSI)運算定義...16	
3.2.一維原始空間定位演算法.....	18
3.3.粒子群最佳化演算法 ( Particle Swarm Optimization , PSO ) .....	21
3.4.方法一 (利用 PSO 尋找矩陣最佳值)：.....	22
3.4.1.粒子學習方法：.....	23
3.4.2. AP 分布設計：.....	23
3.4.3. AP 分布設計(3 個 AP 說明)：.....	23
3.4.4.適應函數(val)：.....	23
3.4.5.演算法描述：.....	23
3.4.6.演算法流程圖：.....	24
3.5.方法二(區域最佳)設計：.....	26
3.5.1.粒子學習方法：.....	27
3.5.2. AP 分布設計：.....	27
3.5.3. AP 分布設計(3 個 AP 說明)：.....	27
3.5.4.適應函數(val)：.....	28
3.5.5.適應函數(3 個 AP 說明)：.....	28
3.5.6.演算法描述：.....	29
3.5.7.方法二執行過程圖：.....	31
3.6.方法三(AP 排列組合分布)設計.....	33
3.6.1.粒子學習方法：.....	33
3.6.2.各個 AP 分布設計：.....	33



3.6.3.各個 AP 分布設計(3 個 AP 說明) :.....	34
3.6.4.適應函數(val) :.....	35
3.6.5.適應函數(3 個 AP 說明) :.....	35
3.6.6.演算法描述 :.....	36
3.6.7. 方法三執行過程圖 :.....	37
3.7. 四種程式複雜度與執行結果比較.....	40
3.7.1.四種程式複雜度比較表.....	40
3.7.2. 電腦模擬執行結果比較.....	41
3.8.二維原始空間定位演算法.....	47
3.9.將一維的方法二擴展到二維.....	48
3.9.1.演算法描述 :.....	49
3.10.將一維的方法三擴展到二維.....	50
3.10.1.演算法描述 :.....	50
4.1.模擬結果.....	52
4.2.實作結果.....	56
4.2.1.硬體設備.....	56
4.2.2. 建立指紋資料庫.....	57
4.2.3. 最佳化前定位實驗.....	60
4.2.4.最佳化後定位實驗.....	61
<b>第五章結論</b> .....	<b>63</b>
<b>使用符號說明</b> .....	<b>65</b>
<b>參考文獻</b> .....	<b>66</b>

## 圖錄

圖 1 粒子群最佳化演算法流程圖 .....	12
圖 2 粒子群搜尋最佳解過程 .....	12
圖 3 簡易流程圖 .....	15
圖 4 一維距離與 RSSI 強度關係圖(AP 位置在 100) .....	17
圖 5 網格點 400 對 3 個 AP 的 RSSI 值從上到下分別為 $rssi_3$ 、 $rssi_2$ 、 $rssi_1$ .....	17
圖 6 理想狀態的 RSSI 與實測差 .....	18
圖 7 程式一維模擬流程圖 .....	20
圖 8 使用 PSO 加速最佳化方法一 .....	22
圖 9 方法一流程圖 .....	25
圖 10 方法二使用 PSO 加速整體模擬的部份 .....	26
圖 11 方法二流程圖 .....	30
圖 12 一開始隨機分布 .....	31
圖 13 執行過程變化 1 .....	31
圖 14 執行過程變化 2 .....	31
圖 15 執行過程變化 3 .....	32
圖 16 方法 3 流程圖 .....	37
圖 17 開始隨機分布 .....	38
圖 18 慢慢收斂 1 .....	38
圖 19 慢慢收斂 2 .....	38
圖 20 慢慢收斂 3 .....	38
圖 21 最後收斂 .....	39
圖 22 最後收斂確認 .....	39
圖 23 測試範圍固定 1000m 時， .....	44
圖 24 AP 數量固定為 3 時， .....	47
圖 25 測試範圍固定時， .....	55
圖 26 最佳分布位置 .....	56
圖 27 訊號強化流程圖 .....	58
圖 28 實際擺放 AP1 .....	59
圖 29 實際擺放 AP2 .....	59
圖 30 實際擺放 AP3 .....	60
圖 31 未最佳化 AP 實測圖 .....	61
圖 32 最佳化 AP 實測圖 .....	62

## 表錄

表 1 本論文與其他論文之區別.....	14
表 2 粒子數與執行次數表(測試範圍 200，AP 個數為 3).....	34
表 3 四種程式複雜度比較表.....	40
表 4 測試範圍 1000m(公尺)時，AP 數量 3.....	41
表 5 測試範圍 1000m(公尺)時，AP 數量 4.....	42
表 6 測試範圍 1000m(公尺)時，AP 數量 5.....	42
表 7 測試範圍 1000m(公尺)時，AP 數量 6.....	43
表 8 AP 數量固定維 3 時，測試範圍 200.....	44
表 9 AP 數量固定維 3 時，測試範圍 500.....	45
表 10 AP 數量固定維 3 時，測試範圍 1000.....	46
表 11 AP 數量固定維 3 時，測試範圍 2000.....	46
表 12 範圍固定 20*20，AP 數量 2.....	52
表 13 範圍固定 20*20，AP 數量 3.....	53
表 14 範圍固定 20*20，AP 數量 4.....	54
表 15 範圍固定 20*20，AP 數量 5.....	55

## 論文架構

第一章 緒論。

第二章 將進行相關研究論文探討。

第三章 理論說明：

### RSSI 運算定義

原本程式如何取得一維 AP 的最佳位置，並將其由一維擴展到二維。

說明方法一、方法二、方法三的一維模擬流程，並將方法二、方法三

由一維擴展到二維。

第四章 為本研究實驗結果，利用模擬結果實際擺放 AP，並實際定位

測量。

第五章 結論。

# 第一章 緒論

## 1.1.研究動機：

近年來由於無線技術的成熟以及可攜式設備的廣泛使用，各種不同的定位系統已越趨成熟，不論是到達時間定位 (Time of Arrival, TOA)[1]、到達時間差定位法(Time Difference of Arrival ; TDOA)[2]、接收訊號角度法 (Arrival of Angle , AOA)[3]、超音波定位(Ultra Sonic)...都是室內定位[4]可使用的�方法。

而接收訊號強度指標(Received Signal Strength Indicator, RSSI)是利用電波訊號在空氣傳輸的過程中，隨著傳播距離而有不同程度之能量衰減，藉由此傳播距離對訊號能量衰減的物理特性即可用來估測發送端與接收端之間的距離，進而達到定位的目的。

利用RSSI搭配上地區特徵指紋定位法時，AP的覆蓋率以及辨識率將會影響整個定位的效果，因此AP的擺放位置將會變得相當重要。

倘若空間中每個位置都去計算RSSI值後，又去做最佳化運算，其運算量將會變得十分龐大，不僅使得整體的模擬工作變得相當困難，其所需要的電腦模擬時間更是無法估計。因此，我們希望能夠提出一種方法，能夠找到AP擺放的最佳位置同時又加速電腦模擬的時間，將會是我們要探討的主要課題。

## 1.2.擬探討解決之問題描述：

在原本的方法中[5]，先定義RSSI運算後，挑選一個AP，並在範圍內的每個網格點都去計算RSSI運算，找到此一AP的最佳位置，不斷的重複同樣的動作，直到所有AP位置固定為止。此一方法在測試範圍小與AP個數少時，其執行時間相當快速，但是當測試範圍變大或是AP個數多時，其運算量相當龐大，不僅消耗大量的儲存空間更需要大量的時間。

所以本研究希望在尋找到 AP 最佳分布的同時，將模擬時間這個重要因素考慮進去，希望藉由最佳化演算法尋找到 AP 最佳分布的同時，也能大幅縮短其模擬時間。

本研究所採取的方法是粒子群最佳化演算法，粒子群最佳化演算法運算速度快與穩定的特性，不僅能夠精準的找到最佳位置，更能大幅縮短電腦模擬所需之時間，這個結果，正是我們所需要的。

## 第二章 文獻探討

### 2.1. ZigBee 室內定位

ZigBee[7]為一由ZigBee Alliance制定之無線網路協定，使用 IEEE 802.15.4 標準的媒體存取層與實體層。相對其它網路傳輸有低速、低功耗、低成本的特色、支援大量網路節點與多種網路拓撲。

ZigBee網路能夠在小範圍內能夠將所有的設備互連，是藉由在各種物品上置入一個低功耗以及低成本的ZigBee無線模組來達到存取資訊以及進行控制的技術。

優點：

- (1) 省電：在非工作模式時，ZigBee 處於睡眠模式。
- (2) 可靠度高：媒體存取層(Medium Access Control layer, MAC)採用 talk-when-ready 之碰撞避免機制，此機制為當有資料傳送需求時則立即傳送，每個發送的資料封包都由接收方確認收到，並進行確認訊息回覆，若沒有得到確認訊息的回覆就表示發生了碰撞，將再傳一次，以此方式大幅提高系統資訊傳輸之可靠度。
- (3) 高度擴充性：一個是 Master 設備，其餘則是 Slave 設備。

缺點：

- (1) 節點有效距離短：傳輸範圍約只介於 10~75m 之間。
- (2) 資料傳輸速率低：只有 10kbps 到 250kbps，相對藍芽等其他

種類無線傳輸速率低上許多。

## 2.2. 粒子群最佳化演算法 (Particle Swarm Optimization)

粒子群最佳化演算法 (Particle Swarm Optimization) (以下簡稱 PSO) 是由 James Kennedy 與 Russell C. Eberhart[8] 於 1995 年所提出，以粒子 (particle) 的移動來模擬鳥群的覓食動作，每顆粒子與每顆粒子之間形成一個完全網路，每顆粒子之間訊息相互溝通，彼此分享最佳解的資料，且該粒子受到過去探索過最佳解的位置影響，使得每一個粒子皆往最佳解的位置移動，稱之為收斂。

PSO 利用一群可能的問題解在解空間 (solution hyperspace) 中移動以找尋問題的最佳解或近似最佳解，在 PSO 中，每個優化問題的解都是搜索空間中的一個粒子。所有的粒子都有一個由適應函數 (fitness function) 決定的適應值 (fitness value)，每個粒子還有一個速度決定他們飛翔的方向和距離。

與基因演算法相較，PSO 並沒有基因演算法的交配 (crossover) 與突變 (mutation)，初始化為一群隨機粒子 (隨機解)，然後通過疊代找到最佳解，在每一次疊代中，粒子通過跟蹤兩個“極值”來更新自己。第一個就是粒子本身所找到的最佳解，這個解叫做個體最佳值 (Pbest)，另一個極值是整個粒子目前所找到的最佳解，這個極值是全體最佳值 (Gbest)，當粒子找到這二個最佳值之後，然後粒子們



就追隨當前的最優粒子在解空間中搜索，最終往最佳解的位置收斂。

與基因演算法相較，PSO 的優勢在於概念簡單、實作容易、沒有太多的參數需要調整，在許多的情況下所有的粒子能夠更快的收斂於最佳解。

相當多的文獻指出，PSO 可廣泛的運用在各種最佳化領域，不論是類神經網路訓練、電腦遊戲[9,10,11,12]、模糊系統控制，或者應用在其它基因演算法所應用的領域[13]，都能有相當優異的效能。

以下說明粒子群體最佳化的演算法：

步驟一：將粒子初始化。給予每個粒子初速度以及初始位置。

步驟二：將粒子位置代入適應函數以求得每個粒子的適應值。

步驟三：每一個粒子的適應值與該粒子所經歷的最佳適應值比較，假若新的適應值比粒子的最佳適應值佳，則以新的位置及評估值取代粒子的最佳解位置及適應值，即找出每一粒子 到目前為止的搜尋過程中最佳解。

步驟四：每一個粒子的最佳適應值與群體的最佳適應值比較，假若粒子的最佳適應值比群體的最佳適應值佳，則以粒子的最佳解位置及適應值取代所有粒子的最佳解位置及適應值，即找出所有Particle 到目前為止所搜尋到的整體最佳解。

步驟五：利用(2.1)、(2.2)式子來更新粒子的速度：

$$v = w \times v + c_1 \times rand \times (Pbest - x) + c_2 \times rand \times (Gbest - x) \quad (2.1)$$

$$x = x + v \quad (2.2)$$

$x$  為粒子的位置，

$v$  為粒子的速度，

$Pbest$  為粒子最佳解位置，

$Gbest$  為所有粒子的最佳解位置，

$c_1 = c_2 = 2.0$ ，

$rand$  為  $-1$  至  $1$  的亂數值，

$w$  為權重值

步驟六：重覆 Step 3 到 Step 6，直到群體的最佳評估值符合我們的

需求。

圖 1 為 PSO 程式流程圖，由一開始設定執行次數，到達到停止條件結束。

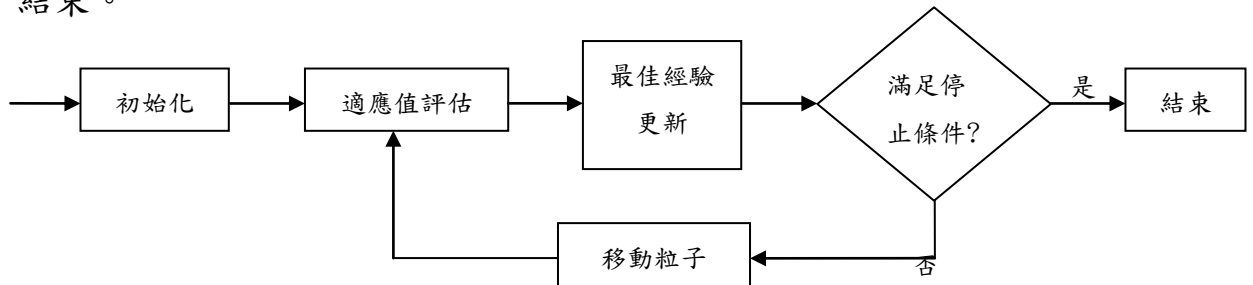


圖 1 粒子群最佳化演算法流程圖

圖 2 為粒子群演算法執行結果，測試函數為  $x \times \sin(x)$ ，X 範圍為 50，粒子由一開始的隨機分布，到最後的收斂結果。

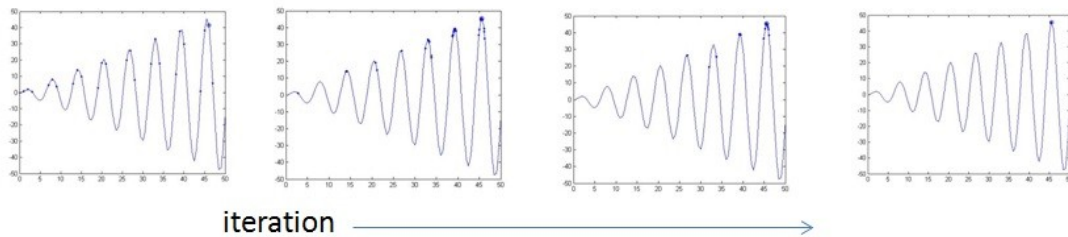


圖 2 粒子群搜尋最佳解過程

### 2.3.高斯混合模型(Gaussian Mixture Model, GMM)

高斯隨機變數也被稱為常態隨機變數(Normal random variables)為一著名的模型，由平均值與標準差兩個參數所決定，其相對應的數學表示法如方程式(2.3)與方程式(2.4)所示，而高斯混合模型(Gaussian Mixture Model, GMM)是由高斯模型所衍伸出來，高斯混合模型乃是由若干個不同平均值及標準差之高斯模型所合成。

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad (2.3)$$

$$F(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sigma} \phi\left(x - \frac{\mu}{\sigma}\right) \quad (2.4)$$

在室內定位的系統設計與實作上戴弘棋[14]提到:若只是使用高斯模型來描述實際訊號接收情況,則最後會因為對實際接收狀況所描述的不夠精確而導致所提供給定位系統的機率值失去參考意義,最終影響定位結果的精確度,也是本篇論文實作部分使用高斯混合模型之主要原因。

## 2.4.最佳化室內定位

解決Zigbee室內無線網路定位的問題,大多多的文獻都指向AP擺放好後,利用接收信號強度指標(Received Signal Strength Indicator, RSSI),搭配各種最佳化演算法來尋找最佳位置,而其定位結果也確實比原始常用的幾何代數求解定位點運算法有較高的準確性與容錯性,如曾光正[15]、Wa Si[16]、Janire Larranaga[17]。

其中,AP的擺放位置將會對整體定位結果有著極大的影響,若AP的分布過於集中或是過於分散,可想而知,其定位結果肯定不如預期。因此,在一個空間中該如何正確分布AP及加速便是本論文所要探討的地方。表一為本研究特色跟其他相關研究之比較結果,由此表可知本研究使用PSO加速AP分布之特色為其他研究所無。請參考表

一之加色部份。

相關研究	AP位置最佳化	定位(用PSO於指紋動態追蹤)
曾光正[15]	無	有
Wa Si[16]	無	有
Janire Larranaga[17]	無	有
本論文	有	有

表 1 本論文與其他論文之區別

### 第三章 研究方法

#### 簡易流程圖

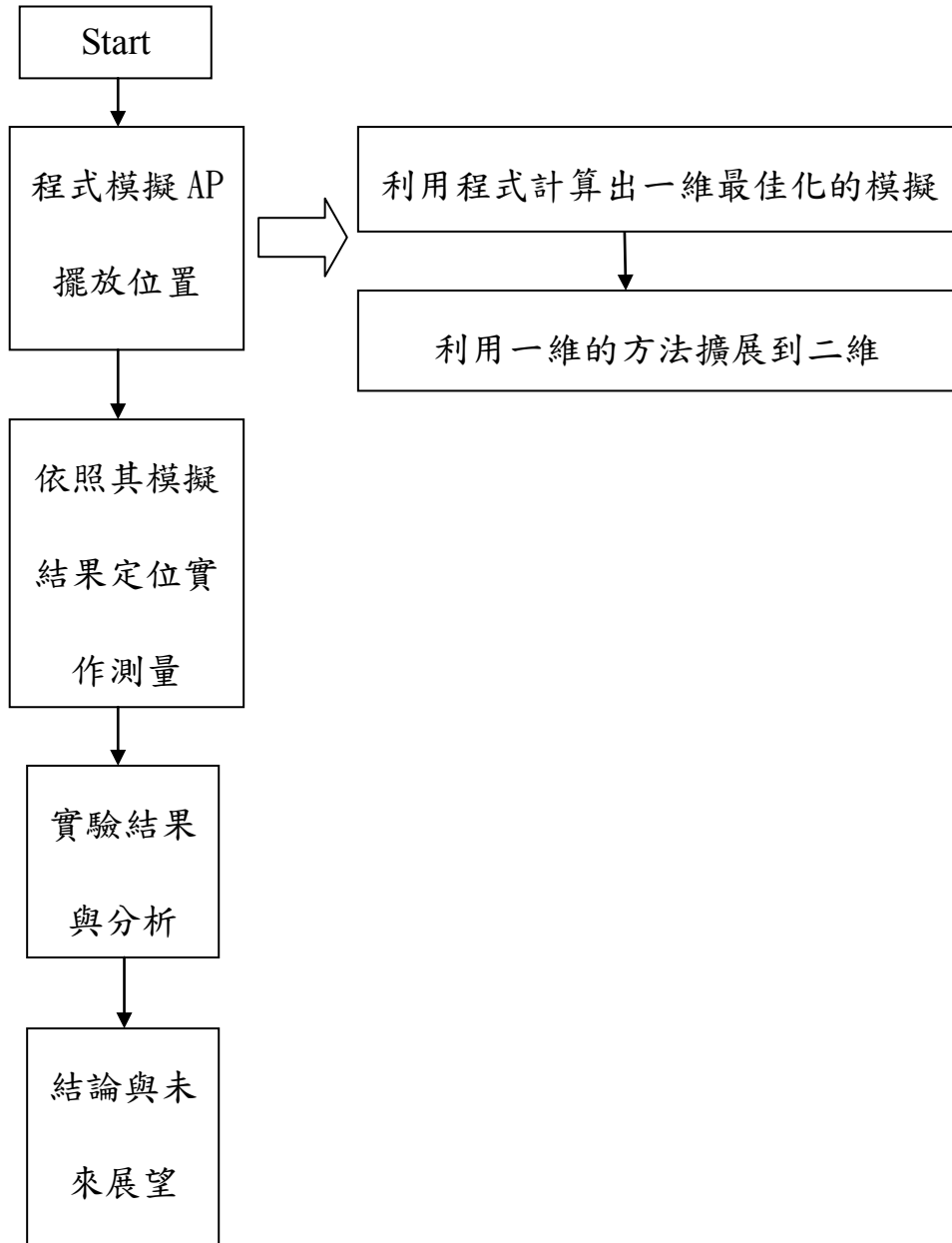


圖 3 簡易流程圖

藉由程式模擬，找到 AP 最佳擺放位置，進而在實際空間擺放，並測量其定位效果。

### 3.1.接收信號強度指標(Received Signal Strength Indicator, RSSI)運算定義

就一維單一一個 AP 而言，RSSI 值越大表示該位置其辨識率越佳，亦表示此位置越容易被定位，但是在使用地區特徵指紋定位法時，在建立指紋資料庫的部分，由一維距離與 RSSI 強度關係圖(圖 4)得知，同一個 RSSI 值卻會出現兩個位置，這是因為與 AP 距離相同的位置坐標有兩個，一個在 AP 左邊，一個在 AP 右邊，所以需要加入其他的 AP 來定位，這樣只要 AP 的位置不同，範圍內各個網格點的位置就可以藉由 RSSI 值的不同來進行定位。

根據 IEEE802.15.4 協定標準[7]，RSSI 的數學表示式為：

$$RSSI = -(10 \times n \times \log_{10} d + A) \quad (3.1)$$

其中:參數A被定義為用dBm表示距離AP 1m處收到信號強度絕對值，本論文中測定為45，參數n被定義為路徑損失指數，本論文中測定為3，d則指的是目標與AP之間的距離。

而就一個網格點而言，如何才算有效率的分布 AP，又可以達到最佳的辨識率？就是該網格點對某個 AP 的 RSSI 值遠大於該網格點對其他 AP 的 RSSI 值，圖 5 中網格點 400 對 3 個 AP 的 RSSI 值從左到右分別為  $rssi_3$ 、 $rssi_2$ 、 $rssi_1$ ，RSSI 差值即是絕對值的  $rssi_3-rssi_2$ 、 $rssi_2-rssi_1$ 、 $rssi_1-rssi_3$ ，也就是說 RSSI 的差值越大，對該網格點既能有高的辨識

率，對 AP 也能有效率的分布。

對此，本論文中 AP 位置最佳化的 RSSI 運算是採用讓各個網格點的 RSSI 的差值越大，也就是讓各個網格點總合的 RSSI 的差值有最大的 AP 分布為 AP 的最佳位置。

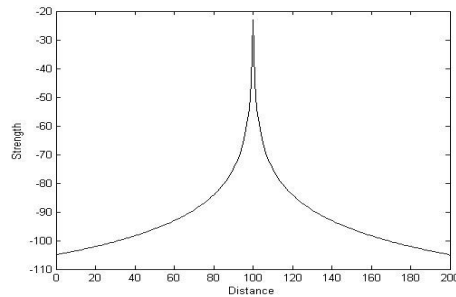


圖 4 一維距離與 RSSI 強度關係圖(AP 位置在 100)

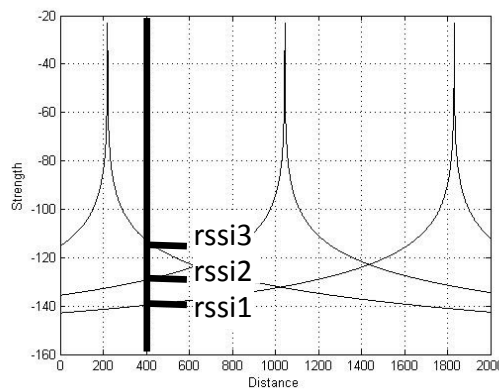


圖 5 網格點 400 對 3 個 AP 的 RSSI 值從上到下分別為  $rssi_3$ 、 $rssi_2$ 、 $rssi_1$

由於 RSSI 是訊號強度，也因此相當容易受到干擾，圖 6 展現了理想狀態與其實測的落差。



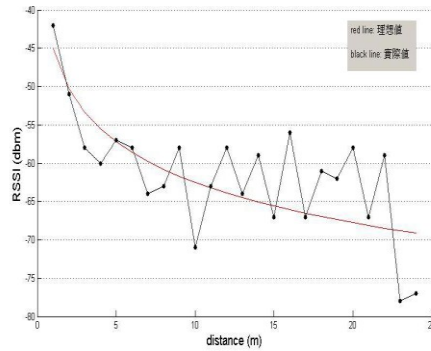


圖 6 理想狀態的 RSSI 與實測差

### 3.2.一維原始空間定位演算法

由程式內設定 AP 的數量以及 AP 分佈的一維測試範圍，先將 AP 平均分佈到空間中，再來做最佳化的動作。

為了避免 AP 的位置重疊而造成同心圓效應，而在 AP 的可能位置上做限制，也就是只在 AP 與 AP 之間，也就是說，最左邊的 AP 的可能位置是在最小 x 位置與往右一個 AP 之間，在中間的 AP 的可能位置是在往左一個 AP 與往右一個 AP 之間，而最右邊的 AP 可能位置是在最大 x 位置與往左一個 AP 之間。

最佳化的過程中使用前面所述之 RSSI 運算定義，針對每個 AP 的可能位置做 RSSI 運算，當其中某一個 AP 的位置改變時，其他的 AP 位置也有可能變動，所以從最左邊的 AP 開始，在其可能的位置上做 RSSI 運算，此時其他的 AP 位置固定不動，當找到 RSSI 運算最大的位置時，則將此 AP 移動到該位置，直到所有 AP 都移動過。

重複此動作直到所有 AP 的位置都固定，表示目前所有 AP 的位置就是我們所要尋找的最佳位置。

以下列出一維 AP 最佳位置實際撰寫程式步驟

步驟一：輸入AP個數。

步驟二：輸入測試範圍。

步驟三：計算平均分佈AP後各個位置的RSSI運算。

步驟四：判斷AP位置是否固定，若是，則表示AP到達最佳位置，若否，則繼續執行步驟五。

步驟五： 判斷AP的位置

步驟六： 依照AP位置的不同來進行不同範圍的RSSI運算

步驟七：找到各個AP最佳位置

步驟八：移動AP到該位置

步驟九： 重複步驟四

圖 7 為一維模擬流程圖，由一開始輸入的 AP 個數以及測試範圍，一次移動一個 AP，做最佳化動作，所有的 AP 不斷重複此動作，直到所有的 AP 位置固定為止。

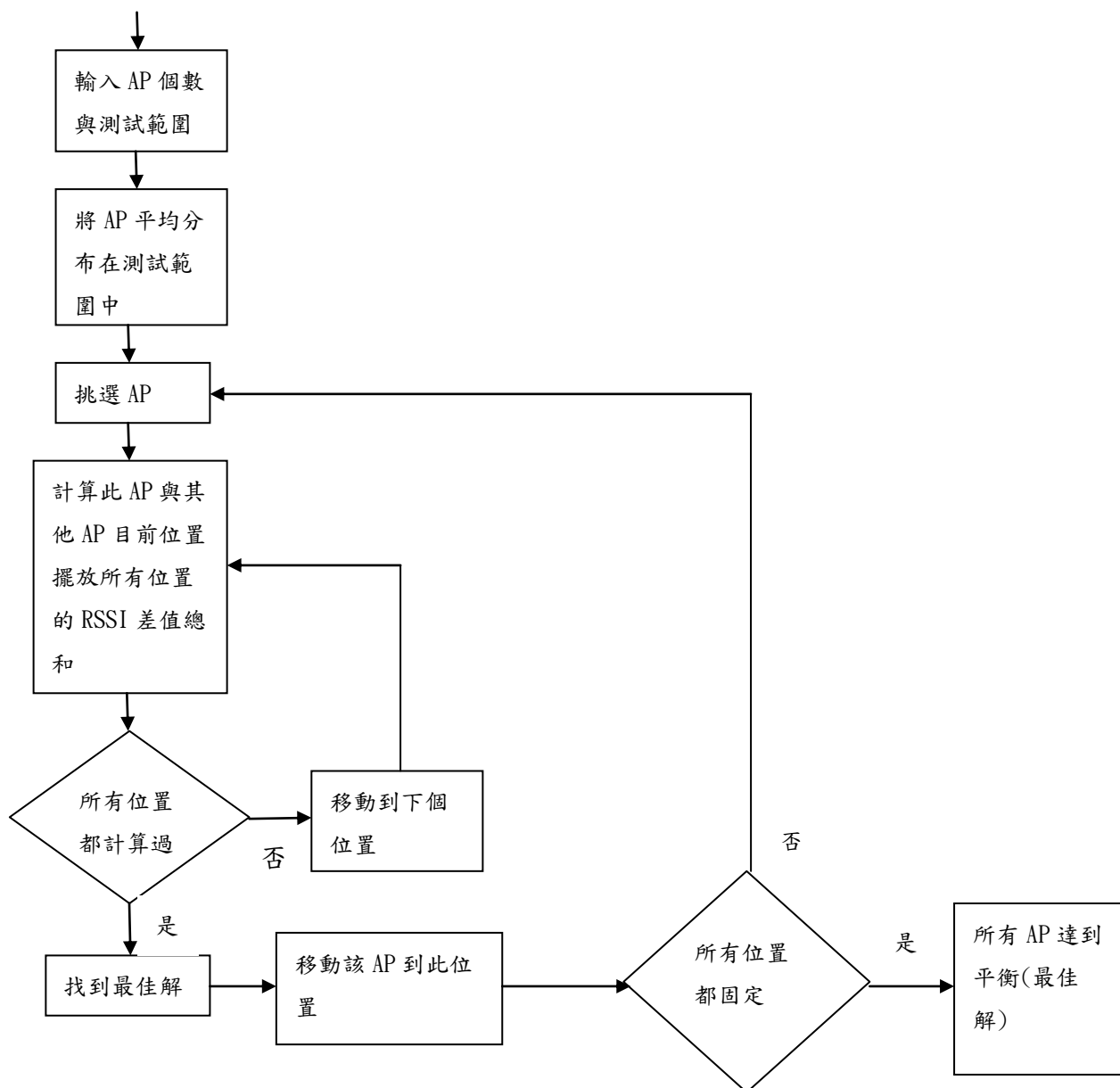


圖 7 程式一維模擬流程圖

### 3.3. 粒子群最佳化演算法 (Particle Swarm Optimization, PSO)

將尋找最佳位置問題與粒子結合，在 PSO 中，群體藉由粒子與粒子間之跳躍而在範圍空間中尋找最佳解，將每個位置所計算的 RSSI 差值總和當作一個整體，粒子為各個位置，演化收斂之後，根據 PSO 的特性，粒子與粒子間之跳躍會尋找到範圍內 RSSI 差值總和最大值以及其產生最大值的位置，換句話說，每個粒子皆代表可能的一個解，粒子與周圍粒子藉由交換其過去曾經找到的最佳解位置，分享其搜尋最佳解的經過，所以收斂時每個粒子將會有近似的解，此解即是我們所要的最佳位置。

在演算法執行過程中，群體最佳位置及個體最佳位置永遠保持最新變數值，然後這二個變數不斷的比較，不斷的更新群體最佳位置。首先，每個粒子在此族群現在位置上比較，以便決定群體最佳位置。然後，每一粒子比較自己歷史個人最佳位置，決定個體最佳位置。這兩位置，影響到每一粒子的下一代速度計算方式。

### 3.4.方法一 (利用 PSO 尋找矩陣最佳值)：

Input：計算各個位置後的 RSSI 矩陣

Output：該 AP 最佳位置

圖 8 展示了欲利用 PSO 加速其原本程式的部分。

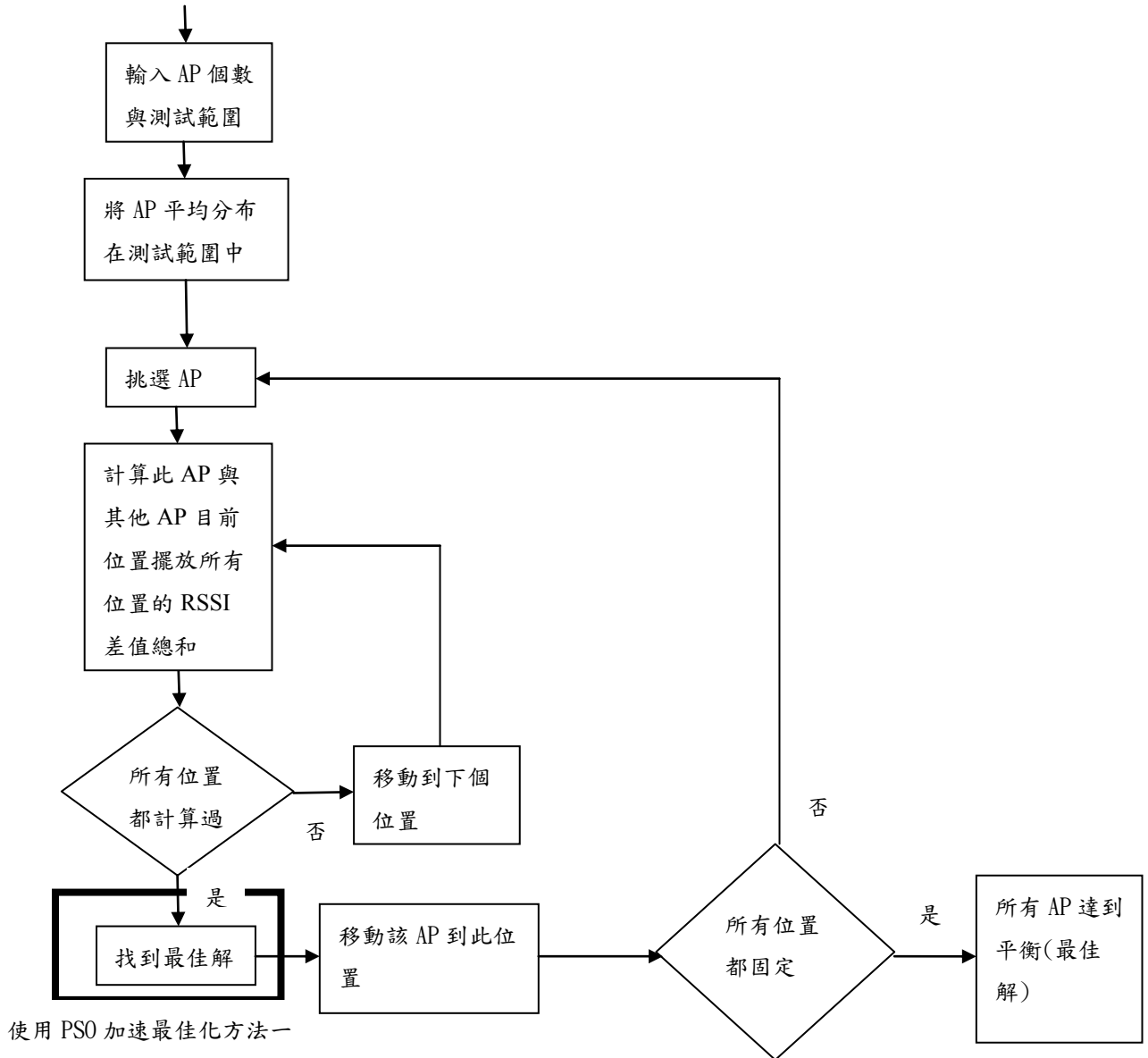


圖 8 使用 PSO 加速最佳化方法一

### 3.4.1. 粒子學習方法：

PSO 中的一個粒子代表該 RSSI 矩陣的值，利用 PSO 快速收斂的特性找到 RSSI 矩陣的最大值位置，及為該 AP 的最佳位置

### 3.4.2. AP 分布設計：

每個粒子的一個維度代表矩陣的 index，因此，粒子的可移動範圍為 1 到 RSSI\_length

### 3.4.3. AP 分布設計(3 個 AP 說明)：

程式執行之後各個粒子會不斷的疊代，在每一次疊代中，粒子通過 Pbest 以及 Gbest 來更新自己，最後往適應函數最大值的位置收斂，並且找到該位置

### 3.4.4. 適應函數(val)：

$val(i) = Ident\_Rate(x);$

最後收斂結果：各個粒子會往 RSSI 矩陣最大值的位置移動，output 在最大值的 AP 位置

### 3.4.5. 演算法描述：

步驟 1、傳入 RSSI 矩陣。將 1 個參數最佳化的問題視為一個 1 維的求解空間，定義一個具有 10 個粒子的族群，每個粒子的位置及速率

均為 1 維的向量。

$P=\{P_1,P_2\dots P_{10}\}$ ; // 粒子 1~ 粒子 10

步驟 2、初始化。以隨機方式設定每個粒子的位置及其速率。

$X=\{X_1,X_2\dots X_{10}\}$ ; // 粒子 1~ 粒子 10 的位置。

$V=\{V_1,V_2\dots V_{10}\}$ ; // 粒子 1~ 粒子 10 的速度。

步驟 3、將粒子的位置代入求解問題適應函數以求得每個粒子的適應值。

步驟 4、每一個粒子的適應值與該粒子所經歷的最佳適應值比較，假若新的適應值比粒子的最佳適應值大，則以新的位置及適應值取代粒子的最佳解位置及適應值。

$Pbest=\{Pbest_1,Pbest_2\dots Pbest_{10}\}$ ; // 粒子 1~ 粒子 10 的最佳適應值。

步驟 5、每一個粒子的最佳適應值與群體的最佳適應值比較，假若粒子的最佳適應值比群體的最佳適應值大，則以粒子的最佳解位置及評估值取代所有粒子的最佳解位置及適應值。

步驟 6、使用(2.1)、(2.2)式改變個體的速率並移動粒子位置

步驟 7、重複步驟 3~ 步驟 6，直到群體的最佳適應值符合我們的需求

### 3.4.6. 演算法流程圖：

圖 9 為方法一流程圖，由一開始輸入 AP 個數以及測試範圍，一次移動一個 AP，此一 AP 可移動範圍內的各點均以計算過 RSSI 運

算，之後利用 PSO 尋找到 AP 最佳位置，將此 AP 移動到該位置，重覆此動作直到 AP 位置固定為止。

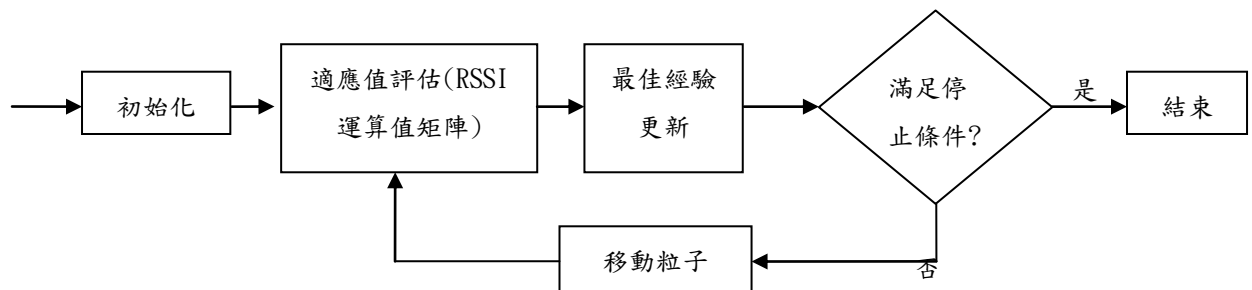


圖 9 方法一流程圖



### 3.5.方法二(區域最佳) 設計:

Input: 該 AP 可以移動範圍,其他 AP 的各個網格點 RSSI

Output: 該 AP 最佳位置

圖 10 顯示了方法二使用 PSO 加速整體模擬的部份

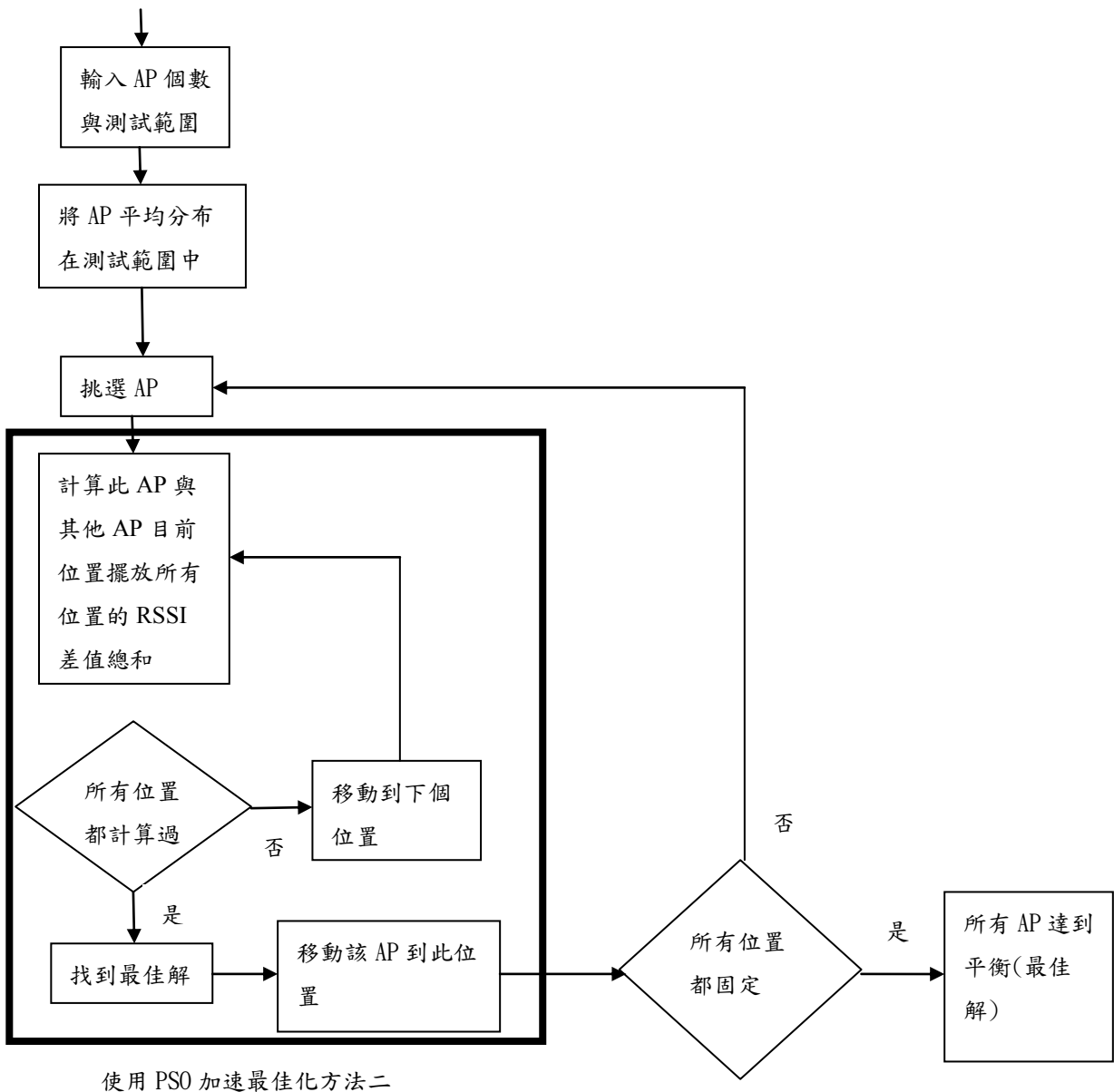


圖 10 方法二使用 PSO 加速整體模擬的部份

### 3.5.1. 粒子學習方法：

方法二中的一個粒子  $P_i$  代表該 AP 分布，而該 AP 的分布相當於是一個粒子群，粒子數為  $n$ ， $P = \{P_1, P_2, \dots, P_n\}$ ，演化的過程中，讓粒子不斷的嘗試各種 AP 不同的組合，並根據適應函數來評斷粒子的每個粒子的結果。

換句話說，每個粒子皆代表可能的一個解，粒子與周圍粒子藉由交換其過去曾經找到的最佳解位置，分享其搜尋最佳解的經過，所以收斂時每個粒子將會有近似的解。

### 3.5.2. AP 分布設計：

每個粒子的一個維度代表一個 AP，也就是說，AP 有自己的維度，若沒有移動範圍限制，則每個 AP 的可以移動範圍為 0(最小測試範圍)到  $length\_max$ (最大測試範圍)

此外為了避免 AP 有位置重疊造成同心圓現象，在 AP 可能範圍加以限制，從右邊一個 AP 的位置  $xmin$ (若無右邊 AP 則設為 0)到從左邊一個 AP 的位置  $xmax$ (若無左邊 AP 則設為  $length\_max$ )

$$x_{min} < AP < x_{max}$$

### 3.5.3. AP 分布設計(3 個 AP 說明)：

AP 的移動範圍限制  $x_{min} < AP < x_{max}$

各個粒子一開始的隨機分布必須符合上式不等式。

程式執行之後各個粒子會不斷嘗試各種符合上面不等式的各個 AP 的排列組合，在每一次疊代中，粒子通過 Pbest 以及 Gbest 來更新自己，最後往適應函數最大值的位置收斂，並且找到該位置的 AP 分布，也就是 AP<sub>1</sub>、AP<sub>2</sub>、AP<sub>3</sub>的最終位置

### 3.5.4.適應函數(val)：

在該種 AP 分布之下的 RSSI 運算

利用(3.1)式計算出如果AP分布在該位置時各個網格點的RSSI值為

y(x(i))，在配合上其他AP的各個網格點RSSI

$$val(x(i)) = \sum_{j=1}^{Inputreader-1} \sum_{k=j+1}^{Inputreader} abs(y_j(x(i)) - y_k(x(i))), \quad (3.2)$$

對所有 AP 組合之適應值

$$val = \sum_{i=1}^g \sum_{j=1}^{Inputreader-1} \sum_{k=j+1}^{Inputreader} abs(y_j(x(i)) - y_k(x(i))), \quad (3.3)$$

g 為所有網格點個數。

### 3.5.5.適應函數(3個AP說明)：

```
[tmpx,tmpy] = zigbee_infncurve20(p, 0, max(length_1));  
//AP 分布在該位置時，各個網格點的RSSI值  
x(move_reader,:) = tmpx; y(move_reader,:) = tmpy;  
//合併成一個矩陣  
Val(i) = Zigbee_move_reader(input_reader,x,y);  
//計算各個網格點RSSI相減的動作 rssi1 - rssi2， rssi2 - rssi3，
```

$\text{rssi3} - \text{rssi1}$

最後收斂結果：各個粒子會往 RSSI 運算產生最大值的位置移動，  
output 在最大值的 AP 位置

### 3.5.6. 演算法描述：

Input：該 AP 可以移動範圍,其他 AP 的各個網格點 RSSI

Output：該 AP 最佳位置

步驟 1、傳入 AP 的可能範圍(xmin to xmax)。正如 3.5.2 中所設計，  
將 1 個參數最佳化的問題視為一個 1 維的求解空間，定義一個具有  
10 個粒子的族群，每個粒子的位置及速率均為 1 維的向量。

$P = \{P_1, P_2 \dots P_{10}\}$ ; // 粒子 1~ 粒子 10

步驟 2、初始化。以隨機方式設定每個粒子的位置(x)及其速率(v)。

$X = \{X_1, X_2 \dots X_{10}\}$ ; // 粒子 1~ 粒子 10 的位置。

$V = \{V_1, V_2 \dots V_{10}\}$ ; // 粒子 1~ 粒子 10 的速度。

步驟 3、將粒子的位置代入求解問題適應函數以求得每個粒子的適應  
值。

步驟 4、每一個粒子的適應值與該粒子所經歷的最佳適應值( $P_{best_i}$ )  
比較，假若新的適應值比粒子的最佳適應值大，則以新的位置及適應  
值取代粒子的最佳解位置及適應值。

$P_{best} = \{P_{best_1}, P_{best_2} \dots P_{best_{10}}\}$ ; // 粒子 1~ 粒子 10 的最佳適應值。

步驟 5、每一個粒子的最佳適應值與群體的最佳適應值比較 (Gbest)，假若粒子的最佳適應值比群體的最佳適應值大，則以粒子的最佳解位置及評估值取代所有粒子的最佳解位置及適應值。

步驟 6、使用(2.1)、(2.2)式改變個體的速率並移動粒子位置

步驟 7、修正粒子位置，重複步驟 3~步驟 6，直到群體的最佳適應值符合我們的需求。

圖 11 為方法二流程圖，由一開始輸入 AP 個數以及測試範圍，一次移動一個 AP，此一 AP 可移動範圍內的各點均未計算過 RSSI 運算，直接利用 PSO 尋找到 AP 最佳位置，將此 AP 移動到該位置，重覆此動作直到 AP 位置固定為止。

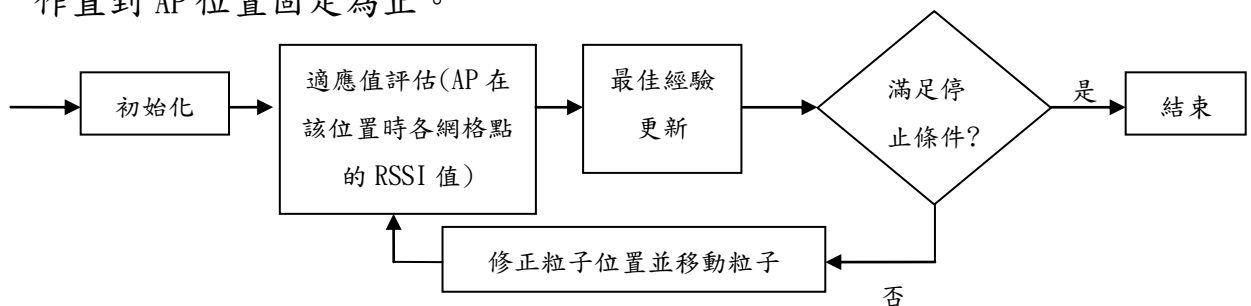


圖 11 方法二流程圖

### 3.5.7.方法二執行過程圖：

圖 12、圖 13、圖 14、圖 15 為使用方法二，測試範圍 2000，AP 個數為 3 的執行過程圖，由一開始的隨機分布，一次移動一個 AP，利用 PSO 尋找此一 AP 的最佳位置，到最後的穩定結果。

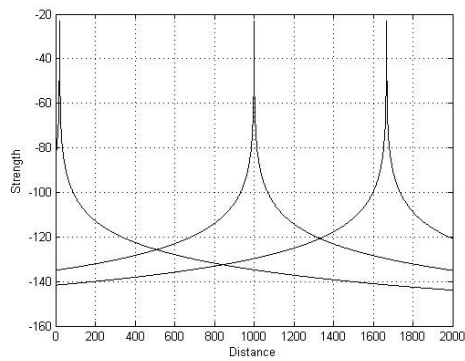


圖 12 一開始隨機分布

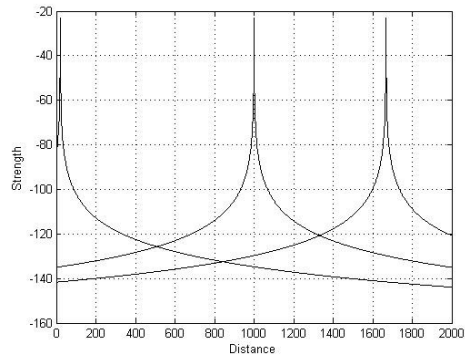


圖 13 執行過程變化 1

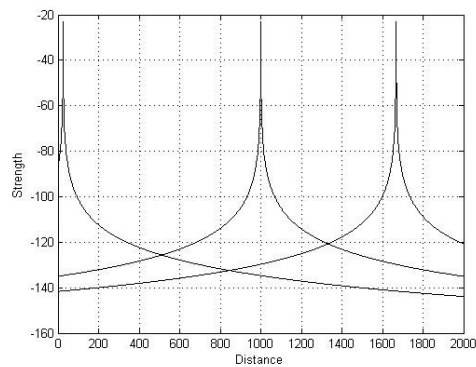


圖 14 執行過程變化 2

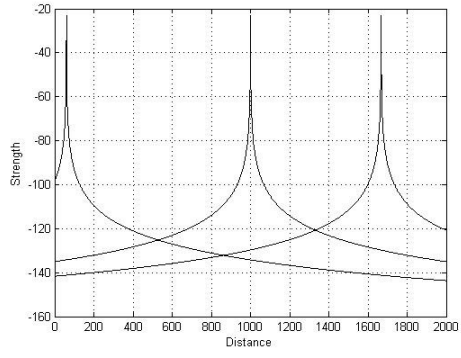


圖 15 執行過程變化 3

### 3.6.方法三(AP 排列組合分布) 設計

Input : AP 數量 , 測試範圍

Output : 各個 AP 分布

直接使用 PSO 尋找到 AP 的最佳分布

#### 3.6.1.粒子學習方法 :

方法三中的一個粒子  $P_i$  代表一種 AP 分布，而整個 AP 分布的排列組合相當於是一個粒子群，粒子數為  $n$ ， $P = \{P_1, P_2, \dots, P_n\}$ ，每個粒子都有  $r$  個維度( $r$  代表 AP 個數)，演化的過程中，讓粒子不斷的嘗試各種 AP 不同的組合，並根據適應函數來評斷粒子的每個粒子的結果。

換句話說，每個粒子皆代表可能的一個解，粒子與周圍粒子藉由交換其過去曾經找到的最佳解位置，分享其搜尋最佳解的經過，所以收斂時每個粒子將會有近似的解。

粒子在跳動時，可能會跳脫其限制範圍，所以必須在最位置更新之後，在做位置修正，確保 AP 位置是在我們所期望的空間。

#### 3.6.2.各個 AP 分布設計 :

為了確定原始方法中所找到的最佳位置在接下來之設計可以找到，再測試適應函數時，使用了大量的粒子數以減少其疊代次數，相對消耗執行空間變多，但是讓其可以使用較少的疊代次數找到最佳位



置，其測試結果如表 2。

粒子數 執行次數	10	50	100	150	200
5	未收斂	未收斂	未收斂	收斂	收斂
10	未收斂	未收斂	收斂	收斂	收斂
20	未收斂	收斂	收斂	收斂	收斂
50	收斂	收斂	收斂	收斂	收斂

表 2 粒子數與執行次數表(測試範圍 200，AP 個數為 3)

一個粒子的一個維度代表一個 AP 的可以移動範圍，n 個 AP 也就有 n 個維度，也就是說，每個 AP 自己有自己的維度而且有相同的粒子數，若沒有移動範圍限制，則每個 AP 的可以移動範圍為 0(最小測試範圍)到  $length\_max$ (最大測試範圍)。

此外為了避免重複計算，在 AP 位置上做了限制，

$$0 \leq AP_1 < AP_2 < AP_3 < \dots < AP_n \leq length\_max$$

若  $AP_1$  的位置超過了  $AP_2$ ，則將  $AP_1$  的位置定為  $AP_2-1$  網格點單位，若  $AP_1$  的位置超過了 0，則將  $AP_1$  的位置定為 0，以此類推。

### 3.6.3.各個 AP 分布設計(3 個 AP 說明):

粒子  $p(AP_1, AP_2, AP_3)$ ， $AP_1$ 、 $AP_2$ 、 $AP_3$  的移動範圍限制如上式，也就是說粒子 p 的  $AP_1$  維度可移動範圍  $0 \leq AP_1 < AP_2$

$AP_2$  維度可移動範圍  $AP_1 < AP_2 < AP_3$

$AP_3$  維度可移動範圍  $AP_2 < AP_3 \leq length\_max$

各個粒子一開始的隨機分布必須符合上面各個不等式。

程式執行之後各個粒子會不斷嘗試各種符合上面不等式的各個 AP 的排列組合，在每一次疊代中，粒子通過 Pbest 以及 Gbest 來更新自己，最後往適應函數最大值的位置收斂，並且找到該位置的 AP 分布，也就是 AP<sub>1</sub>、AP<sub>2</sub>、AP<sub>3</sub> 的最終位置

### 3.6.4.適應函數(val) :

在該種 AP 分布之下的 RSSI 運算

利用(3.1)式計算出如果 AP 分布在該位置時各個網格點的 RSSI 值為 y(i)，並利用(3.3)式計算出對所有 AP 組合之適應值

### 3.6.5.適應函數(3 個 AP 說明) :

```
[tmpx1,tmpy1] = zigbee_infncurve20(r1, 0, length_max);  
//AP 1 分布在該位置時，各個網格點的RSSI值  
[tmpx2,tmpy2] = zigbee_infncurve20(r2, 0, length_max);  
//AP 2 分布在該位置時，各個網格點的RSSI值  
[tmpx3,tmpy3] = zigbee_infncurve20(r3, 0, length_max);  
//AP 3 分布在該位置時，各個網格點的RSSI值  
  
y = [tmpy1; tmpy2; tmpy3];  
//合併成一個矩陣  
val(i) = Zigbee_move_reader(length_l,input_reader,x,y);  
//計算各個網格點rssI相減的動作 rssi1 - rssi2， rssi2 - rssi3， rssi3 - rssi1
```

最後收斂結果：各個粒子會往 RSSI 運算產生最大值的位置移動，

output 在該位置的 AP 分布

### 3.6.6.演算法描述：

Input：AP 數量，測試範圍

Output：各個 AP 分布

步驟 1、輸入 AP 個數(n)以及測試範圍(length\_max)。正如 3.6.2 中所設計，定義一個具有 10 個粒子的族群，並將 n 個參數最佳化的問題視為一個 n 維的求解空間，每個粒子均有 n 維的空間。

$P=\{P_1, P_2 \dots P_{10}\}$ ; //粒子 1~粒子 10

步驟 2、初始化。以隨機方式設定每個粒子的位置及其速率。

$X=\{X_1, X_2 \dots X_{10}\}$ ; //粒子 1~粒子 10 的 10 種 AP 組合

$X_1=\{X_{11}, X_{12} \dots X_{1r}\}$ ; //粒子 1 內的各種 AP 位置

$V=\{V_1, V_2 \dots V_{10}\}$ ; //粒子 1~粒子 10 的速度

$V_1=\{V_{11}, V_{12} \dots V_{1r}\}$ ; //粒子 1 內的各種 AP 速度

步驟 3、將粒子的位置代入求解問題適應函數以求得每個粒子的適應值。

步驟 4、每一個粒子的適應值與該粒子所經歷的最佳適應值比較，假若新的適應值比粒子的最佳適應值大，則以新的位置及適應值取代粒子的最佳解位置及適應值。

$Pbest=\{Pbest_1, Pbest_2 \dots Pbest_n\}$ ;

步驟 5、每一個粒子的最佳適應值與群體的最佳適應值(Gbest)比較，假若粒子的最佳適應值比群體的最佳適應值大，則以粒子的最佳解位置及評估值取代所有粒子的最佳解位置及適應值。

步驟 6、使用(2.1)、(2.2)式改變個體的速率並移動粒子位置

步驟 7、修正粒子位置，重複步驟 3~步驟 6，直到群體的最佳適應值符合我們的需求。

圖 16 為方法三程式流程圖：

由一開始輸入 AP 個數與測試範圍，利用 PSO 直接尋找 AP 的最佳分布。

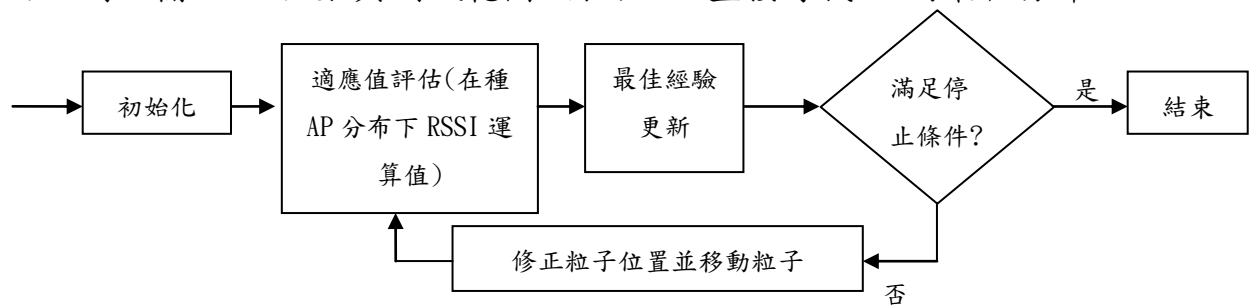


圖 16 方法 3 流程圖

### 3.6.7. 方法三執行過程圖：

圖 17、圖 18、圖 19、圖 20、圖 21、圖 22 為使用方法三，測試範圍 2000，AP 個數為 3 的執行過程圖

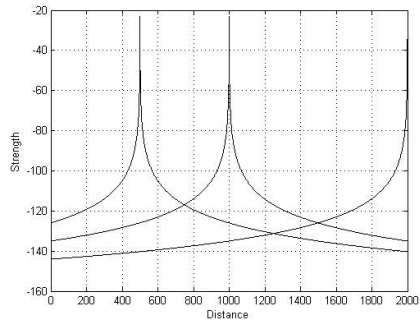


圖 17 開始隨機分布

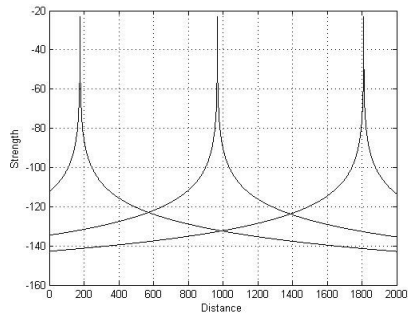


圖 18 慢慢收斂 1

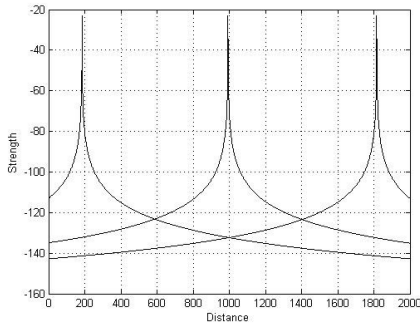


圖 19 慢慢收斂 2

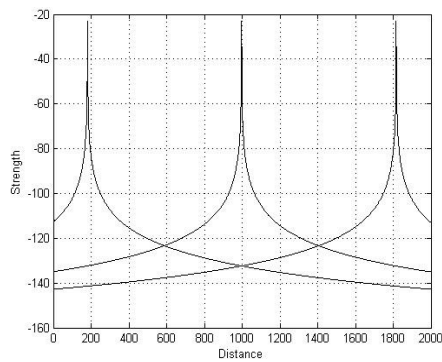


圖 20 慢慢收斂 3

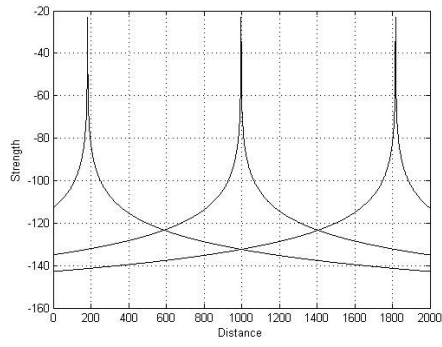


圖 21 最後收斂

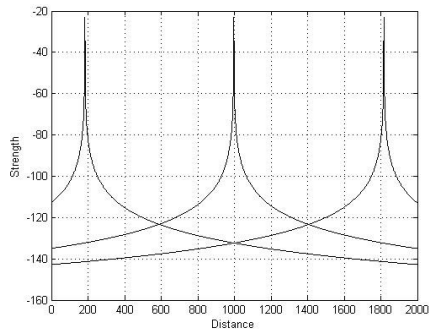


圖 22 最後收斂確認

### 3.7. 四種程式複雜度與執行結果比較

#### 3.7.1. 四種程式複雜度比較表

若測試範圍為  $s$ ，AP 個數為  $r$ ，粒子數為  $n$ ，疊代次數為  $m$ 。

在原本程式中，因為一次移動一個 AP，而 1 個 AP 可能移動的範圍約為  $s$ ，因此其複雜度為  $O(s^r)$ 。

在方法一中，因為只用 PSO 尋找矩陣最大值，整體複雜度還是集中在原本程式，因此其複雜度為  $O(s^r)$ 。

在方法二中，因為用 PSO 尋找 1 個 AP 的最佳位置， $n$  個粒子維度為 1，並且執行  $m$  次，其複雜度為  $O(n \times m)$ ，所以  $r$  個 AP 其複雜度為  $O((n \times m)^r)$ 。

在方法三中，使用 PSO 尋找所有 AP 的最佳分布位置， $n$  個粒子維度為  $r$ ，並且執行  $m$  次，其複雜度為  $O(n \times m \times r)$ 。

表 1 比較各個程式的時間複雜度

程式名稱	複雜度 bigO
原本程式	$O(s^r)$ 測試範圍 $s$ ，AP 個數 $r$
方法一	$O(s^r)$ 測試範圍 $s$ ，AP 個數 $r$
方法二	$O((n \times m)^r)$ 測試範圍 $s$ ，AP 個數 $r$ ，疊代次數 $m$
方法三	$O(n \times m \times r)$ 疊代次數 $m$ ， $r$ 維度

表 3 四種程式複雜度比較表

### 3.7.2. 電腦模擬執行結果比較.

電腦規格:

CPU : DualCore Intel Celeron E3200, 2400 MHz

Memory : 2048MB

程式語言:MATLAB 7.10.1(R2010a)

#### 3.7.2.1.測試範圍 1000m(公尺)時，AP 數量改變時所做的測

試

表 2 為 AP 數量為 3，而且測試範圍為 1000 m(公尺)時，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
Original program	3	(91, 500, 909)	5.9783e+004	86.2200
Method 1	3	(91, 500, 909)	5.9783e+004	84.1890
Method 2	3	(91, 500, 909)	5.9783e+004	139.1126
Method 3	3	(91, 500, 909)	5.9783e+004	8.9648

表 4 測試範圍 1000m(公尺)時，AP 數量 3

由表 2 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。



表 3 為 AP 數量為 4，而且測試範圍為 1000 m(公尺)時，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
Original program	4	69,314,685,931	1.1188e+005	107.361
Method 1	4	69,314,685,931	1.1188e+005	107.951
Method 2	4	69,314,685,931	1.1188e+005	226.144
Method 3	4	69,314,685,931	1.1188e+005	12.1159

表 5 測試範圍 1000m(公尺)時，AP 數量 4

由表 3 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 4 為 AP 數量為 5，而且測試範圍為 1000 m(公尺)時，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
Original program	5	56,220,499,779,943	1.7915e+005	108.7205
Method 1	5	56,220,499,779,943	1.7915e+005	148.3088
Method 2	5	56,220,499,779,943	1.7915e+005	285.3939
Method 3	5	56,220,499,779,943	1.7915e+005	18.7170

表 6 測試範圍 1000m(公尺)時，AP 數量 5

由表 4 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 5 為 AP 數量為 6，而且測試範圍為 1000 m(公尺)時，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
Original program	6	49,167,375,624,832,951	2.6155e+005	132.1672
Method 1	6	49,167,375,624,832,951	2.6155e+005	133.1927
Method 2	6	49,167,375,624,832,951	2.6155e+005	414.4512
Method 3	6	49,167,375,624,832,951	2.6155e+005	99.6448

表 7 測試範圍 1000m(公尺)時，AP 數量 6

由表 5 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

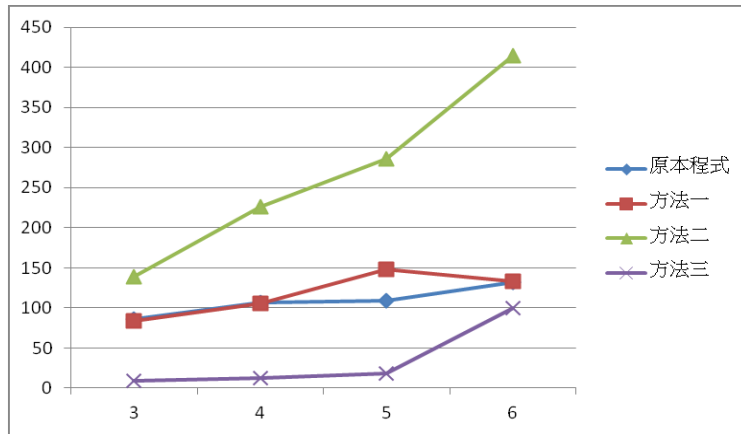


圖 23 測試範圍固定 1000m 時，  
AP 數量(x 座標)與執行時間(s)(y 座標)折線圖

經由上面表 2、表 3、表 4、表 5、表 6，將其整理成圖 23，方法一、方法二與方法三雖然都可以找到最佳位置但是在執行時間方面卻大不相同，方法一、方法二雖然也使用粒子群最佳化演算法但是卻反而增加了程式複雜度而影響了執行時間，而很明顯的方法三不僅在準確度上與執行時間都優於原始方法、方法一、方法二，因此使用方法三不僅可以大幅減少其程式的複雜度，又可得到準確的結果。

### 3.7.2.2. AP 數量固定維 3 時，測試範圍改變

表 6 為 AP 數量為 3，而且測試範圍為 200 m(公尺)時，其各個程式的執行結果。

	Test area	AP Position	RSSI value	Runtime(s)
Original program	200	18,100,182	1.1995e+004	2.6950
Method 1	200	18,100,182	1.1995e+004	2.7821
Method 2	200	18,100,182	1.1995e+004	21.4021
Method 3	200	18,100,182	1.1995e+004	1.6265

表 8 AP 數量固定維 3 時，測試範圍 200

由表 6 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 7 為 AP 數量為 3，而且測試範圍為 500 m(公尺)時，其各個程式的執行結果。

	Test area	AP Position	RSSI value	Runtime(s)
Original program	500	45,250,455	2.9916e+004	21.3023
Method 1	500	45,250,455	2.9916e+004	20.6105
Method 2	500	45,250,455	2.9916e+004	67.1850
Method 3	500	45,250,455	2.9916e+004	4.0497

表 9 AP 數量固定維 3 時，測試範圍 500

由表 7 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 8 為 AP 數量為 3，而且測試範圍為 1000 m(公尺)時，其各個程式的執行結果。

	Test area	AP Position	RSSI value	Runtime(s)
Original program	1000	(91,500,909)	5.9783e+004	86.2200

Method 1	1000	(91,500,909)	5.9783e+004	84.1890
Method 2	1000	(91,500,909)	5.9783e+004	139.112
Method 3	1000	(91,500,909)	5.9783e+004	8.9648

表 10 AP 數量固定維 3 時，測試範圍 1000

由表 8 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 9 為 AP 數量為 3，而且測試範圍為 2000 m(公尺)時，其各個程式的執行結果。

	Test area	AP Position	RSSI value	Runtime(s)
Original program	2000	183,1000,1817	1.1952e+005	432.741
Method 1	2000	183,1000,1817	1.1952e+005	434.006
Method 2	2000	183,1000,1817	1.1952e+005	557.613
Method 3	2000	183,1000,1817	1.1952e+005	17.7677

表 11 AP 數量固定維 3 時，測試範圍 2000

由表 9 可以很清楚的知道原本程式、方法一、方法二、方法三之間的差別，在準確度上，原本程式、方法一、方法二、方法三都可以很精準的尋找到最佳位置，但是在執行時間上，4 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

圖 24 是 AP 個數固定時，各個程式的執行時間與測試範圍所做的折線圖

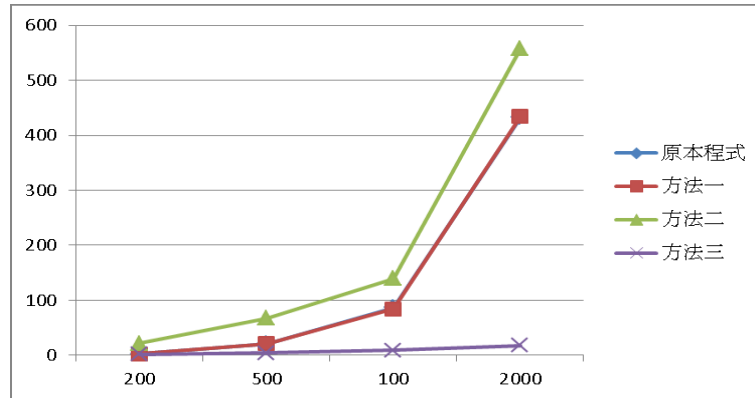


圖 24 AP 數量固定為 3 時，  
測試範圍(x 座標)與執行時間(s) (y 座標)折線圖

經由上面折線圖，方法一、方法二與方法三雖然都可以找到最佳位置但是在執行時間方面卻大不相同，方法一、方法二雖然也使用粒子群最佳化演算法但是卻反而增加了程式複雜度而影響了執行時間，而很明顯的方法三不僅在準確度上與執行時間都優於原始方法、方法一、方法二，因此使用方法三不僅可以大幅減少其程式的複雜度，又可得到準確的果。

### 3.8.二維原始空間定位演算法

由程式內設定AP的數量以及AP分佈的二維測試範圍，先將AP隨機分佈到空間中，再來做最佳化的動作。

同樣的為了避免 AP 的位置重疊而造成同心圓效應，而在 AP 的可能位置上做限制，將隨機分布的 AP 與測試空間三角化後，AP 的

可移動範圍即被限制在與自身有接觸的三角形空間內，並利用 MATLAB 中的 Function : inpolygon 將 AP 的可移動範圍定義出來。

最佳化的過程中使用前面所述之 RSSI 運算定義，針對每個 AP 的可能位置做 RSSI 運算，當其中某一個 AP 的位置改變時，其他的 AP 位置也有可能會變動，所以將 AP 編號，從第一個 AP 開始，在其可能的位置上做 RSSI 運算，此時其他的 AP 位置固定不動，當找到 RSSI 運算最大的位置時，則將此 AP 移動到該位置，直到所有 AP 都移動過。重複此動作直到所有 AP 的位置都固定，表示目前所有 AP 的位置就是我們所要尋找的最佳位置。

方法二中粒子只有一個維度，可移動空間限制在 MATLAB Function inpolygon 所找到的範圍空間，並在此空間內找到最佳位置。

方法三中粒子的每個維度都各自獨立，不互相干擾，每個維度代表該 AP 的最佳位置，粒子在跳動時，可移動空間限制在 MATLAB Function inpolygon 所找到的範圍空間，並在此空間內找到最佳位置。

### **3.9.將一維的方法二擴展到二維**

Input：該 AP 可以移動範圍,其他 AP 的各個網格點 RSSI

Output：該 AP 最佳位置

### 3.9.1.演算法描述：

步驟 1、傳入 AP 的可能範圍。將 1 個參數最佳化的問題視為一個 1 維的求解空間，定義一個具有 10 個粒子的族群(P)，每個粒子的位置及速率均為 1 維的向量。

$P=\{P_1,P_2\dots P_{10}\}$ ; //粒子 1~粒子 10

步驟 2、初始化。以隨機方式設定每個粒子的位置(X)及其速率(V)。

$X=\{X_1,X_2\dots X_{10}\}$ ; //粒子 1~粒子 10 的位置

$V=\{V_1,V_2\dots V_{10}\}$ ; //粒子 1~粒子 10 的速度

步驟 3、將粒子的位置代入求解問題適應函數以求得每個粒子的適應值。

步驟 4、每一個粒子的適應值與該粒子所經歷的最佳適應值( $Pbest_i$ )

比較，假若新的適應值比粒子的最佳適應值大，則以新的位置及適應值取代粒子的最佳解位置及適應值。

$Pbest=\{Pbest_1,Pbest_2\dots Pbest_{10}\}$ ;

步驟 5、每一個粒子的最佳適應值與群體的最佳適應值( $Gbest$ )比較，假若粒子的最佳適應值比群體的最佳適應值大，則以粒子的最佳解位置及評估值取代所有粒子的最佳解位置及適應值。

步驟 6、使用(2.1)、(2.2)式改變個體的速率並移動粒子位置

步驟 7、重複步驟 3~步驟 6，直到群體的最佳適應值符合我們的需求。



### 3.10.將一維的方法三擴展到二維

Input : AP 數量 ,測試範圍

Output : 各個 AP 分布

#### 3.10.1.演算法描述 :

步驟 1、 輸入 AP 個數(n)以及測試範圍。將 n 個參數最佳化的問題視為一個 n 維的求解空間，定義一個具有 10 個粒子的族群，每個粒子的位置及速率均為 n 維的向量。

$P=\{P_1,P_2\dots P_{10}\}$ ; //粒子 1~粒子 10

步驟 2、初始化。以隨機方式設定每個粒子的位置及其速率。

$X=\{X_1,X_2\dots X_{10}\}$ ; //粒子 1~粒子 10 的 10 種 AP 組合

$X_1=\{X_{11}, X_{12}\dots X_{1n}\}$ ; //粒子 1 內各個 AP 的位置

$V=\{V_1,V_2\dots V_{10}\}$ ; //粒子 1~粒子 10 的速度

$V_1=\{V_{11}, V_{12}\dots V_{1n}\}$ ; //粒子 1 內各個 AP 的速度

步驟 3、將粒子的位置代入求解問題適應函數以求得每個粒子的適應值。

步驟 4、每一個粒子的適應值與該粒子所經歷的最佳適應值( $Pbest_i$ )

比較，假若新的適應值比粒子的最佳適應值大，則以新的位置及適應值取代粒子的最佳解位置及適應值。

$Pbest=\{Pbest_1,Pbest_2\dots Pbest_{10}\}$ ;

步驟 5、每一個粒子的最佳適應值與群體的最佳適應值( $Gbest$ )比較，

假若粒子的最佳適應值比群體的最佳適應值大，則以粒子的最佳解位

置及評估值取代所有粒子的最佳解位置及適應值。

步驟 6、使用(2.1)、(2.2)式子改變個體的速率並移動粒子位置。

步驟 7、重複步驟 3~步驟 6，直到群體的最佳適應值符合我們的需求。

## 第四章成果

### 4.1.模擬結果

在一維的實驗中，原始方法是對照組，方法一、方法二與方法三是實驗組，而實驗結果驗證了方法一、方法二與方法三的準確性，但執行時間各有快慢，也因此決定將方法二與方法三擴展到二維模擬。

下面各表為測試範圍固定時，AP 數量改變後其各個程式的執行結果與執行時間。

表 10 為測試範圍為 20\*20 m(公尺)時，而且 AP 個數為 2，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
原始空間(區域最佳化)定位演算法	2	9 8	559.4055	25.4212
		9 15		
粒子群區域最佳演算法 2	2	11 9	514.5727	32.1522
		10 17		
粒子群排列組合最佳演算法	2	7 11	573.1840	1.7147
		8 5		

表 12 範圍固定 20\*20，AP 數量 2

由表 10 可以很清楚的知道原本程式、方法二、方法三之間的差別，在準確度上，原本程式、方法二、方法三都可以尋找到最佳位置，但是在執行時間上，3 者卻有很大的差別，方法三很明顯可以快速又

準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 11 為測試範圍為 20\*20 m(公尺)時，而且 AP 個數為 3，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
原始空間(區域最佳化)定位演算法	3	10 7 3 10 5 1	1.4092e+003	37.4682
粒子群區域最佳演算法 2	3	12 6 2 13 6 15	959.9620	81.8430
粒子群排列組合最佳演算法	3	9 5 14 9 9 11	1.7092e+003	7.7163

表 13 範圍固定 20\*20，AP 數量 3

由表 11 可以很清楚的知道原本程式、方法二、方法三之間的差別，在準確度上，原本程式、方法二、方法三都可以尋找到最佳位置，但是在執行時間上，三者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 12 為測試範圍為 20\*20 m(公尺)時，而且 AP 個數為 4，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	Runtime(s)
原始空間(區域最佳化)定位演算法	4	13 12	2.9671e+003	47.6146
		6 12		
		10 5		
		3 7		
粒子群區域最佳演算法 2	4	11 8	2.2017e+003	273.4591
		5 9		
		7 19		
		2 8		
粒子群排列組合最佳演算法	4	6 12	3.3115e+003	24.2659
		10 6		
		11 11		
		5 7		

表 14 範圍固定 20\*20，AP 數量 4

由表 12 可以很清楚的知道原本程式、方法二、方法三之間的差別，在準確度上，原本程式、方法二、方法三都可以尋找到最佳位置，但是在執行時間上，3 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

表 13 為測試範圍為 20\*20 m(公尺)時，而且 AP 個數為 5，其各個程式的執行結果。

	Number of AP	AP Position	RSSI value	RunTime(s)
原始空間(區域最佳化)定位演算法	5	16 10	4.4706e+003	88.0013
		7 11		
		19 3		
		11 4		

		12	10		
粒子群區域最佳演算法 2	5	1	4	3.2451e+003	584.3372
		8	16		
		4	8		
		11	13		
		15	6		
粒子群排列組合最佳演算法	5	10	12	5.3911e+003	29.9488
		6	11		
		6	6		
		10	5		
		13	9		

表 15 範圍固定 20\*20，AP 數量 5

由表 13 可以很清楚的知道原本程式、方法二、方法三之間的差別，在準確度上，原本程式、方法二、方法三都可以尋找到最佳位置，但是在執行時間上，3 者卻有很大的差別，方法三很明顯可以快速又準確的找到最佳位置，而方法二雖然準確但是其執行時間卻比原始程式更久。

圖 25 為測試範圍固定 20\*20，AP 數量與執行時間折線圖

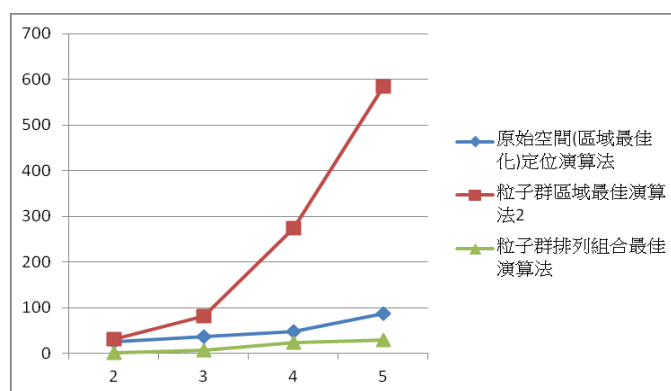


圖 25 測試範圍固定時，AP 數量(x 座標)與執行時間(s)(y 座標)折線圖

經由圖 24，方法二與方法三雖然都可以找到最佳位置但是在執行時間方面卻大不相同，方法二雖然也使用粒子群最佳化演算法但是卻

反而增加了程式複雜度而影響了執行時間，而很明顯的方法三不僅在準確度上與執行時間都優於原始方法與方法二，因此使用方法三不僅可以大幅減少其程式的複雜度，又可得到準確的結果，這正是本實驗所希望的，並且利用程式模擬的結果，並且實際擺放 AP。

根據以上理論，可取得最佳 AP 擺放位置，如圖 26

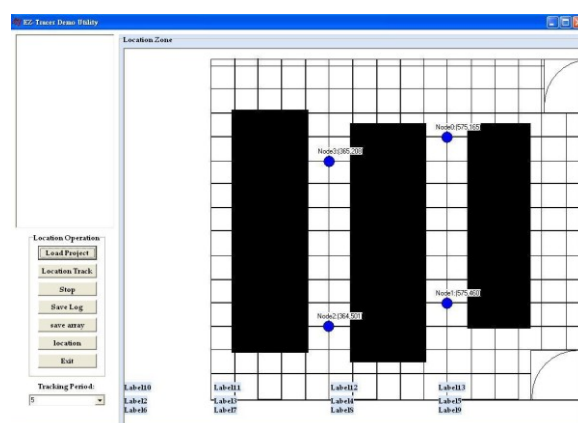


圖 26 最佳分布位置

## 4.2.實作結果

### 4.2.1.硬體設備

EZ-Tracer 包含組成一個定位系統所需要的基本硬體設備：

Master：負責採集原始定位資料並上報至上層管理軟體。Master 採用 USB 介面和上位機通信，無需外接電源，方便易用。

AP：使用 IEEE802.15.4 協定，組成資料獲取的骨幹網路，並作為定位的座標參考節點。以 Mesh 拓撲結構為基礎，若干個 AP 可以靈活組網，實現對各種應用環境的覆蓋，並有效避免通信死角的出現。

電子標籤(Tag)：作為定位目標的發射裝置，即時上報原始定位資料。電子標籤採用低功耗的 8 位微控制器，通過控制休眠喚醒時間有效地降低整體功耗。

以上硬體設備都內嵌赫立訊先進的 IP-Link 系列工業級無線通信模組，IP-Link 系列模組工作在 2.4GHz ISM 頻段符合 IEEE802.15.4/ZigBee 標準，具有介面豐富、工作穩定可靠、抗干擾能力強、支援大容量網路及路由級數等先進特性，並通過了 ZCP (ZigBee 相容性平臺)、FCC 等行業認證。

#### 4.2.2. 建立指紋資料庫

將所要測量的空間(電腦教室)，分成格網點，並模擬 AP 最佳擺放位置，實際擺放 AP，在每個格網點上測量 100~1000 個數據，取其平均及標準差。

當所有讀取器歷經接收電子標籤廣播出的訊號一段時間，亦即收集足夠的樣本後，接下來即需分析、模型化所有收集進來的資料，將它們從單純的資料經模型化後分析轉變為有用的資訊。將資料高斯混合模型化的目的是為了使用機率密度函數來描述真實的資料接收狀況，當讀取一訊號強度值時，由於資料已被轉化為機率密度函數 (Probability Density Function, PDF)，因此，每一讀取之訊號強度值將可對應出一個機率密度值，定位系統即是依據這個機率值來判



斷位置。

由於使用的是 ZigBee 的 Mesh 網路，各個 AP 間會花一短暫的時間做網路建立的動作，亦即自組網路的功能。在網路連線成功後，各個 AP 開始開啟 RF 接收器準備接收位於現場的電子標籤所廣播出來的辨認訊號。各個 AP 在接收電子標籤資料後，即存於自身之暫存器上，直到資料收集器以輪詢的方式向各個 AP 請求資料傳送後，AP 才將自身的暫存器清空，這樣的方式係為了避免資料在無線傳輸時發生碰撞而產生封包丟失的疑慮。將收取進來的訊號，刪除離中間值 (Medium) 太遠之訊號能量，其次，再將剩餘的訊號作均值化 (Averaging)，經過這兩個步驟後，每一組讀取器最終僅產生一個數值提供定位引擎做機率累計運算。

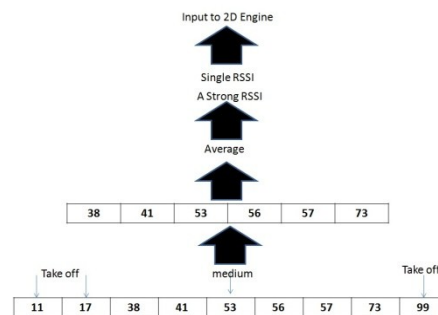


圖 27 訊號強化流程圖

當一電子標籤所發射出的訊號被各個讀取器解析出訊號強度後，定位系統會根據這些事前所記錄的高斯混合模型來累計機率，最後使用最大概率法則，即判定該電子標籤位於該格網點位置。

本次實驗則是將 4 組機率相乘得出最大機率值，判定最有可能的

格網點，達到定位功能。之後可再加入座標點限制，使定位系統只會搜尋上一定位點的附近座標，達到更快速的定位功能，而不必全部格網點再重新計算。

其實作結果如圖 28、圖 29、圖 30：



圖 28 實際擺放 AP1



圖 29 實際擺放 AP2

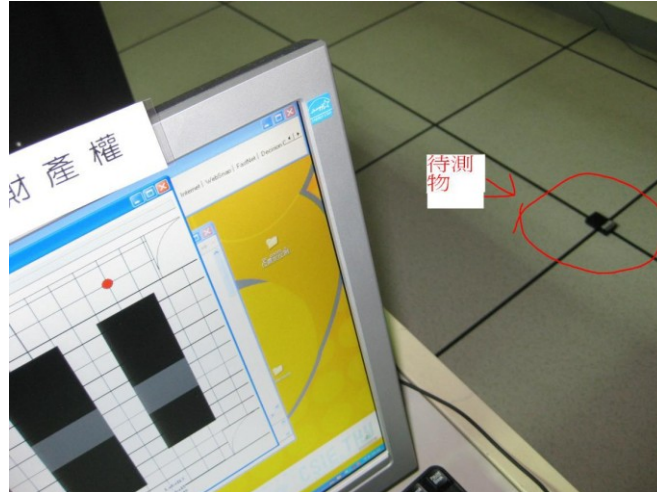


圖 30 實際擺放 AP3

#### 4.2.3. 最佳化前定位實驗

AP 位置待測位置的四個角落，其實測結果定位總誤差 433.591 cm，平均誤差 12.0442 cm，定位結果如圖 28，藍色實心圓圈代表 AP 擺放位置，於各個網格點進行測量。

原位置矩陣：A

最小平方差矩陣：T

測量位置矩陣：B

最終位置矩陣：C

$$A \times T = B$$

$$T = (B' B)^{-1}(B' A)$$

$$A \times T = C$$

紅色空心點為修正後座標。

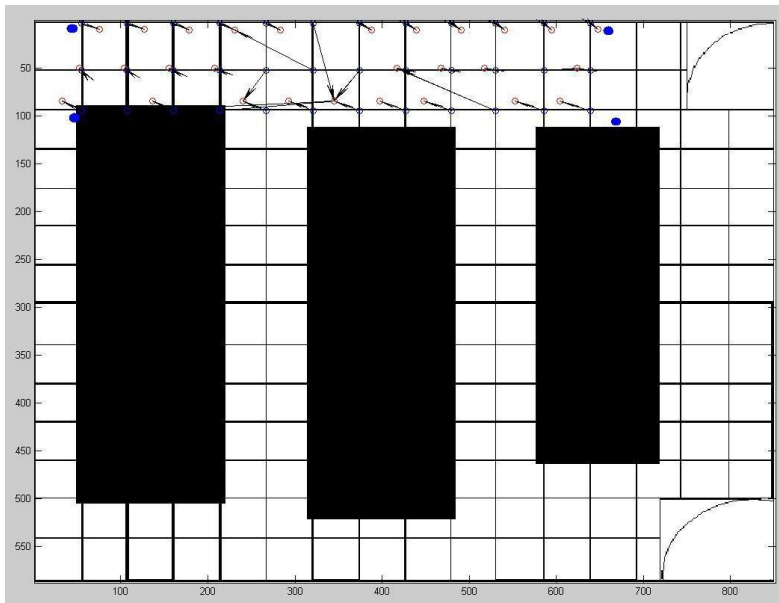


圖 31 未最佳化 AP 實測圖

#### 4.2.4.最佳化後定位實驗

利用程式模擬找出最佳 AP 位置，並且實際擺放，其實測結果定位總誤差 373.3865cm，平均誤差 10.37184cm，確實比未最佳化時所做之測量平均誤差更低，整體定位效果更佳準確，定位結果如圖 29，藍色實心圓圈代表 AP 擺放位置，於各個網格點進行測量。

原位置矩陣：A

最小平方差矩陣：T

測量位置矩陣：B

最終位置矩陣：C

$$A \times T = B$$

$$T = (B' B)^{-1}(B' A)$$

$$A \times T = C$$

紅色空心點為修正後座標。

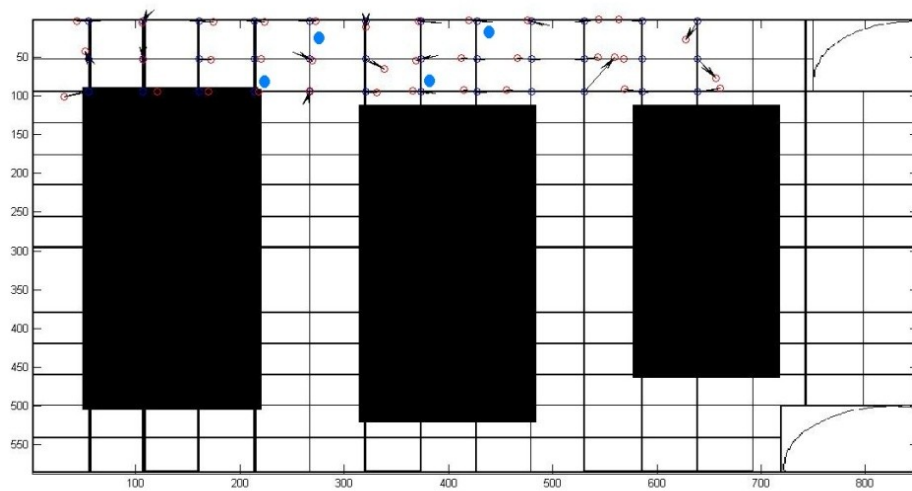


圖 32 最佳化 AP 實測圖

## 第五章結論

利用無線電波的隨著傳輸距離而有不同程度能量衰減的特性，定位參數即為接收信號強度指標(Received Signal Strength Indicator, RSSI)，並搭配上地區特徵指紋定位法，來完成整個定位系統，因此 AP 的分布會影響整體的定位效果，所以利用程式模擬來找出 AP 的最佳分布。

在本論文中提出了兩種使用粒子群最佳化演算法 (Particle Swarm Optimization, PSO) 來尋找最佳位置，透過粒子間的學習，使粒子間互相配合找出最佳位置，方法二雖然都使用粒子群最佳化演算法，並且在準確度上與原始方法相同，但是在複雜度上卻反而比原始方法高，所以將整個程式模擬進化到方法三，不僅在準確度上與原始方法相同，更大幅縮減了複雜度，使得程式可以快速又準確的計算出 AP 的最佳分布。

我們希望藉由完成這個研究，降低電腦模擬 AP 分布之時間，透過使用粒子群最佳化演算法，不僅能得到所需要的結果，更能大幅的降低程式複雜度，對於找到 AP 的最佳擺放位置，得到最佳的辨識率與覆蓋率，加速模擬室內定位得到相當大的成效，並且將電腦模擬之結果於實際空間內進行 AP 擺放，完成室內定位，建置小範圍內的精準定位系統。並且未來將利用 zigbee 低成本架構、低功率損耗、佈

建容易…等優勢條件，使這定位系統便趨近於成熟，可使用在大眾生活上，讓科技生活變得更便利。

## 使用符號說明

PSO：粒子群最佳化演算法 (Particle Swarm Optimization)

AP: Access Point

RSSI : Received Signal Strength Indicator

n : 粒子數

r : AP 數量

s : 測試範圍

m : 疊代次數

A : 原位置矩陣

T : 最小平方差

B : 測量位置

C: 最終位置



## 參考文獻

- [1] B. H. Wellenhoff, H. Lichtenegger and J. Collins, "Global Positioning System : Theory and Practice" , Fouth Edition Springer Verlag , 1997.
- [2] M. O. Sunay and J. Jekin , "Mobile Location Tracking in DS CDMA Networks Using Forward Link Time Difference of Arrival and Its Application to Zone-Based Billing" , Bell Labs , Lucent Technologies , 2001.
- [3] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, Andy Ward and Andy Hopper, "Implementing a sentient computing system." IEEE Computer Magazine, August 2001, 34(8):pp.50-56
- [4] 林景頌, 室內定位品質分析之研究, 國立成功大學電機工程學系碩士論文, 中華民國九十五年七月
- [5] 石志雄, 陳信文, Zigbee 無線感測定錨點(reader)分布位置之模擬與實測, 東海大學資訊工程系
- [6] Robert J. Orr and Gregory D. Abowd , "The smart floor: a mechanism for natural user identification and tracking" , Proceedings of Conference on Human Factors in Computing Systems , The Hague , (The Hague, Netherlands ) , April 2000.
- [7] ZigBee Alliance, <http://www.caba.org/standard/zigbee.html>.
- [8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proc. IEEE International Conference on Neural Networks, Vol. IV, pp 1942-1948, 1995.
- [9] 陳柏仲, 蔡清欉, 廖啟賢, 林灶生, "以粒子群最佳化演算法為基礎之

第一人稱射擊遊戲角色設計,” 智慧生活科技研討會,2006.

[10] 黃奐禎, 利用粒子群演算法在電腦遊戲訓練團隊策略, 中華民國

九十六 年一月

[11] S. Bakkes, P. Spronck, and E. Postma, “TEAM: The Teamoriented Evolutionary Adaptability Mechanism,” Entertainment Computer – ICEC 2004, volume 3166, Springer-Verlag, pp 273-282, 2004.

[12] S. Bakkes, P. Spronck, and E. Postma, “Best-Response Learning of Team Behaviour in Quake III,” IJCAI 2005 Workshop on Reasoning, Representation, and Learning in computer games, 2005.

[13] X. Hu, “Particle Swarm Optimization Tutorials”,  
<http://web.ics.purdue.edu/~hux/tutorials.shtml>, 2002.

[14] 戴弘棋, 基於高斯混合模型之室內定位系統設計與實作, 大同大

學通訊工程研究所碩士論文, 中華民國九十八年七月

[15] 曾光正, 智慧型室內定位演算法之研究, 樹德科技大學電腦與通

訊研究所碩士論文, 中華民國九十九年六月

[16] Wa Si, Energy Saving System for Office Lighting by Using PSO and

ZigBee Network, SEAL'10 Proceedings of the 8th international

conference on Simulated evolution and learning

[17] Janire Larranaga, An Environment Adaptive ZigBee-based Indoor

Positioning Algorithm, 2010 International Conference on Indoor

Positioning and Indoor Navigation (IPIN), 15-17 September 2010, Zürich, Switzerland

[18] CASE STUDY WITHIN TAIWAN UNIVERSITY CAMPUS, ,

Department of Information Management, Hsing-Wu College, Taipei, Taiwan 22

[19] 陳大鈞, Measure Point selection Algorithms for Wireless Indoor

Positioning Systems, 國立臺灣大學資訊管理研究所碩士論文, 2004

年七月。

[20] 陳嶽東, 以訊號強度為基礎利用高斯混合模型之 IEEE802.11 無

線區域網路定位演算法, 國立成功大學資訊工程研究所碩士論文,

2005 年六月。

[21] A.H. Sayed, A. Tarighat and N. Khajehnouri, “Network-based wireless location: challenges faced in developing techniques for accurate wireless location information,” IEEE Signal Processing Magazine, vol. 22, Issue 4, pp.24-40, July, 2005.

[22] D. Hahnel, W. Burgard, D. Fox, K. Fishkin and M. Philipose, “Mapping and localization with RFID technology,” IEEE Proceedings of International Conference on Robotics and Automation (ICRA2004), vol. 1, pp. 1015-1020, 2004.

[23] Y. Wu, J. Hu, and Z. Chen, “Radio Map Filter for Sensor Network Indoor Localization Systems.” 2007 5th IEEE International Conference, vol.1 23-27, pp.63-68, June 2007.

[24] V. Torres, J. Becerra, F. Esparza, A. Lopez, F. Falcone , Analysis of

Topology Considerations in Indoor ZigBee meshed Networks ,

Universidad Pública de Navarra , 2010

[25] CHIA-PING HUANG , ZIGBEE WIRELESS NETWORK  
APPLICATION RESEARCH