

東海大學資訊管理學研究所  
碩士學位論文

無線感測網路節能叢集之研究-使用地理資訊之  
基因演算法

**A Study Of Energy-Efficient Clustering Algorithm Using  
Geography-Cognizant Genetic Algorithm For Wireless  
Sensor Networks**

指導教授：姜自強博士  
研究生：林韋辰撰

中華民國 101 年 6 月 26 日

# 東海大學資訊管理學系碩士學位

## 考試委員審定書

資訊管理學系研究所 林韋辰 君所提之論文

無線感測網路節能叢集之研究-使用地理資訊之基因演  
算法

A STUDY OF ENERGY-EFFICIENT CLUSTERING ALGORITHM  
USING GEOGRAPHY-COGNIZANT GENETIC ALGORITHM FOR  
WIRELESS SENSOR NETWORKS

經本考試委員會審查，符合碩士資格標準。

學位考試委員會 召集人：侯廷偉 (簽章)

委員：丁建文

丁以亭

林正澤

姜育強

中華民國 101 年 6 月 26 日

## 致謝

兩年的研所時光很快就接近尾聲，在我鍵入致謝的此刻，仍有些共同奮鬥的朋友們還在努力埋頭於研究中。我的求學之路還算平順，我也覺得自己很幸運，能在這路途中的每個時間點，遇到不同無私奉獻的老師，以及每位對我意義非凡的朋友們。最重要的一件事，就是感謝我的父母，因為他們這數十年來的辛苦付出，今天的我才有機會能從這裡畢業，往後，希望我能發揮所學，進入職場，許一個讓家人更好的未來。感謝評審委員，於口試過程中給了許多指導與建議。感謝姜自強教授，當時答應擔任我的指導教授，兩年間從老師那裡學到了許多，並希望在日後，老師的研究與家庭都圓滿順心。感謝紀宗衡教授，在老師的訓練下，我才有機會進入研所就讀。感謝老師們的付出，我學習到的事物無法一一列出，心中的感謝也無法以文字說明，但相信這數年間的學習經驗已經影響了我的一生，謝謝您。感謝我的朋友們，包括已經畢業的學長姐，努力中的學弟妹，以及求學過程中曾一起奮鬥的各位，這一路走來有你們的陪伴，我很開心，同時希望每個人未來的發展都能順利。

論文名稱：無線感測網路節能叢集之研究-使用地理資訊之基因演算法

校所名稱：東海大學資訊管理學系研究所

畢業時間：2012 年 9 月

研究生：林韋辰 指導教授：姜自強

論文摘要：

在國際上的不同研究領域中，因無線感測網路的廣泛應用而有各種探討。無線感測網路的感測器通常大量佈署於人力無法前往或危險性的地區，因此無線感測網路的生命周期與能源節省的方法便成為許多研究討論的重要議題。在許多研究中基因演算法常被用來解決 NP-難問題，他們提出了客製化的基因演算法來解決不同類型的問題獲得令人滿意的結果。因此當基因演算法的編碼方式改變時，我們通常需要修改交配與突變運算子，使得新的演算法能夠正常運作。當考慮地理位置相關的問題時，我們設計一個考慮地理資訊的基因演算法提供合適的染色體編碼與適應性函數。無線感測網路的相關研究中，有學者提出考慮二維染色體編碼的基因演算法，並以實驗證明此方法勝過傳統的基因演算法。但我們考慮到二維染色體編碼過於冗長，其中夾帶的多餘資訊將導致演算法執行的效率不佳，因此本研究提出透過考慮地理位置資訊的基因演算法，期能使求解的品質更勝於傳統的基因演算法，同時亦較二維編碼的基因演算法更快速獲得解答；若在更短的時間內獲得最佳的解答，我們的方法便可節省無線感測節點的能源消耗，進而達到優化的效果。最後我們使用 NS2 網路模擬器對地理資訊之基因演算產生的結果法進行網路的運作模擬，實驗結果也反映了本方法確實在無線感測網路的建構有所幫助。期望將來用於實際的無線感測網路時，此方法考慮了地理位置的資訊而在無線感測網路的資料傳輸與能源控制上將有所助益。

關鍵詞：能源節省、地理位置資訊、基因演算法、無線感測網路

Title of Thesis : A Study Of Energy-Efficient Clustering Algorithm Using  
Geography-Cognizant Genetic Algorithm For Wireless Sensor Networks

Name of Institute: Tunghai University, Institute of Information Management

Graduation Time : 9/2012

Student Name : Wei-Chen Lin Advisor Name : Tzu-Chiang Chiang

Abstract :

Wireless sensor networks have been discussed widely in recent international research areas, because a large number of wireless sensors may be deployed in dangerous locations where human cannot reach. Therefore, the issues of survival for wireless sensor networks are proposed in most of the studies for charging with energy savings. In considering with location issues for wireless sensors, we need to add more location information of those sensors for designing a genetic algorithm with appropriate encoding method. In this paper we propose a geography-cognizant genetic algorithm which takes into account with sensor location information for energy-efficient wireless sensor networks. Through the experiments show that our proposed method does help to get better in a shorter time for convergence which means the effect of energy savings. We simulate the network operation of geography-cognizant genetic algorithm in the NS2, experiment results showed that our approach is really helps when construction of wireless sensor networks. We expect this method applied to practical wireless sensor networks, will be helpful in data transmission, and energy control.

Keywords: energy saving, location information, geography-cognizant genetic algorithm, wireless sensor networks.

# 目次

第一章	前言 .....	1
第二章	文獻探討 .....	2
第一節	無線感測網路 .....	2
第二節	基因演算法 .....	3
第三節	基因演算法的運算子 .....	4
第四節	節能基因演算法 .....	7
第五節	考慮地理位置的演算法與路由協定 .....	7
第三章	地理資訊之基因演算法 .....	8
第一節	考慮地理位置的基因演算法之運作 .....	9
第四章	實驗設計與方法 .....	16
第一節	演算法求解進程實驗 .....	18
第二節	模擬耗能實驗 .....	21
第三節	NS2 網路模擬實驗 .....	21
第四節	演算法求解進程實驗設計 .....	21
第五節	模擬耗能實驗設計 .....	30
第六節	NS2 網路模擬實驗-NSG .....	32
第七節	NS2 網路模擬實驗結果 .....	41
第五章	結論 .....	51
	參考文獻 .....	52
	附錄 .....	56

# 表次

表 4-1	演算法編碼長度之比較 .....	12
表 4-1	實驗環境之配備 .....	16
表 4-2	題目參數 .....	16
表 4-3	演算法參數設定 .....	21
表 4-4	模擬耗能實驗參數設定 .....	30
表 4-5	封包遺失實驗數據 .....	42
表 4-6	平均封包延遲實驗數據 .....	44
表 4-7	最大封包延遲實驗數據 .....	44
表 4-8	平均抖動率實驗數據 .....	46
表 4-9	最大抖動率實驗數據 .....	46
表 4-10	Average rate 比較 .....	49
表 4-11	三種演算法 Peak rate 比較 .....	50

# 圖次

圖 3-1	基因演算法的基本流程 .....	4
圖 3-2	常見的編碼方式 .....	4
圖 3-3	常見的交配形式 .....	6
圖 3-1	二維染色體編碼之基因演算法於漸增節點規模運作耗時 .....	8
圖 3-2	無線感測網路的結構 .....	10
圖 3-3	二維編碼染色體編碼 .....	10
圖 3-4	節點移動後之無線感測網路 .....	12
圖 3-5	節點移動後之二維染色體編碼 .....	12
圖 3-6	GCGA 的交配運算 .....	15
圖 4-1	基因演算法實驗流程圖 .....	18
圖 4-2	F1 求解題目 Q1 截至目前最佳解 .....	22
圖 4-3	F1 求解題目 Q2 截至目前最佳解 .....	23
圖 4-4	F1 求解題目 Q3 截至目前最佳解 .....	23
圖 4-5	F1 求解題目 Q4 截至目前最佳解 .....	24
圖 4-6	F1 求解題目 Q4 截至目前最佳解 .....	24
圖 4-7	F1 中各演算法總實驗時間之比較 .....	25
圖 4-8	F2 求解題目 Q1 截至目前最佳解 .....	26
圖 4-9	F2 求解題目 Q2 截至目前最佳解 .....	26
圖 4-10	F2 求解題目 Q3 截至目前最佳解 .....	27
圖 4-11	F2 求解題目 Q3 截至目前最佳解 .....	28
圖 4-12	F2 求解題目 Q3 截至目前最佳解 .....	28
圖 4-13	F2 中各演算法總實驗時間之比較 .....	29
圖 4-14	模擬耗能實驗之演算法耗時比較 .....	31
圖 4-15	模擬耗能實驗之演算法進程比較 .....	31



圖 4-16	5 個時間區段中的能源存量比較 .....	32
圖 4-17	NSG2 劇本模擬 .....	34
圖 4-18	適應值函式 F1 所得最佳解之封包遺失實驗.....	42
圖 4-19	適應值函式 F2 所得最佳解之封包遺失實驗.....	42
圖 4-20	適應值函式 F1 之平均封包延遲.....	44
圖 4-21	適應值函式 F2 之平均封包延遲.....	45
圖 4-22	適應值函式 F1 之平均抖動率.....	46
圖 4-23	適應值函式 F2 之平均抖動率.....	47
圖 4-24	適應值函式 F1 之最大抖動率.....	47
圖 4-25	適應值函式 F2 之最大抖動率.....	48
圖 4-26	F1-Average rate 之比較 .....	49
圖 4-27	F2-Average rate 之比較 .....	49
圖 4-28	F1- Peak rate 比較 .....	50
圖 4-29	F2 -Peak rate 比較 .....	50

# 第一章 前言

因無線感測網路應用廣泛，且近年來國際上的各研究領域中也有所討論，且因感測器會大量的部屬於人力無法前往或具有危險的地域，故與無線感測網路存活有重要關係的能源節省便成為大多數研究中討論的重點。因基因演算法(Genetic Algorithm, GA)常被用來解決 NP 問題，且必須針對不同問題，制定不同的編碼方式，再依各種編碼方式，對交配與突變等運算子做不同的改變，以符合編碼，除此之外，仍有將 GA 與其他演算法結合如類神經演算法的新演算法產生。

因此，本研究旨在考慮以下問題：

- (1) 對整體無線感測網路的結構與能源節省最佳化。
- (2) 使用基因演算法求解地理位置相關問題時，適當的將地理位置資訊融入到編碼中以增進求解品質。
- (3) 對基因演算法中各種運算子做更好的選擇，使得演算法能更快速獲得解答。
- (4) 在演化過程中考慮在不同時間區段中，實際的感測器能源存量並做為決策依據。

並提出對應的解決方法：

- (1) 因無線感測網路的問題是一個 NP 問題，所以採用基因演算法做為解題的工具時；設計一個考慮地理位置的染色體編碼。
- (2) 依照問題特性，對選擇、交配與突變等運算子做出正確的改善。
- (3) 於基因演算法的進程中，對下一個時間區段的能源存量進行模擬，比較不同適應性函數產生之最佳解，並且依照模擬的結果進行各種決策。

根據實驗結果表示在基因演算法的求解能力部分，本研究提出的地理資訊之基因演算法，在求解能力、速度與最佳解的品質有顯著的優良表現；對應到 NS2 網路模擬實驗的結果後，我們知道，搭配求解問題的類型，我們在基因演算法上必須要搭配的良好適應值函式設計，才能讓基因演算法的演化方向正確並得到符合模擬實驗所需的良好解答。

我們將結合這近期研究案成果進行加入地理問題相關研究，如叢集分群傳輸與新的空洞迴避方法，探討無線感測網路因應地理位置不同，而產生出的叢集分群與空洞迴避，或是結合兩者後更進一步的探討深入相關議題。

## 第二章 文獻探討

### 第一節 無線感測網路

無線感測網路的應用相當廣泛，如工業控制和監測、家庭自動化、消費類電子產品、安全和軍事監測，資產跟踪和供應鏈管理，智能農業，導彈指揮，火災報警，滑坡預警，環境監測及健康監測和商業應用等[18]。

感測器(Sensor)最早是美國加州柏克萊大學的研究人員所開發，是體積小、低電力、低成本的微積體電路，可以感測外在環境的溫度、濕度、聲音與影像等，且具有運算與通訊的功能，其硬體設備有四大單元，分別為感測單元(Sensing Unit)、處理單元(Processing Unit)、傳輸單元(Transceiver Unit)及電力供應單元(Power Unit)，感測單元又包括感測元件及訊號轉換元件，處理單元又包括儲存元件及處理器。

感測器散布在需要被監控的地域中，並互相傳遞訊息與封包資料，資料最後匯集到 sink 點，再傳送到後端伺服器，這過程稱為資料聚集(Data Aggregation)，這樣的網路稱為無線感測網路，由於感測器的電力有限，所以延長無線感測網路的生命週期是許多專家學者探討的議題，且通常注重於感測器的休眠機制與備援路線等議題[3,5,8,10,12,13,14,19,22,

23,24,25,27,28,29]。

在節能議題與延長網路壽命的一部分，相關文獻提到了正確選擇匯集節點的配置與最低限度的必要的跳數，可顯著延長網路的壽命[17]，還有消除節點閒置問題並實驗自我穩定、低延遲和低成本的路由協定[11]，以及提高傳輸效率與降低能源消耗的方法[15]，提出高節能且自我穩定的無線感測網路控制協議並證明其效率[6]，提出可得到更多頻寬、更可靠與降低效能的方法[16]，與增加總吞吐量並使得各感測節點間的耗能幾乎相等的方法[1]。

## 第二節 基因演算法

基因演算法(Genetic Algorithm, GA)是由 John Holland 以達爾文"物競天擇，適者生存"的理論為基礎所提出，並在人工智慧的領域中持續發展，到了 1989 年，David E. Goldberg 在其著作中說明了基因演算法的基本運算子，如選擇/複製、交配、突變，後來基因演算法便在各個領域中有廣泛的應用，如求解最佳化問題、NP 或非線性特性的問題等。

基因演算法的基本流程如圖 2-1 所示，個體由遺傳所組成，也稱染色體(Chromosome)，許多個體組成為族群(Population)，根據適應性函數產生的適應值進行選擇，接著進行交配與突變後便產生子代，演算法周而復始的運作，直到達成終止條件後結束，且在良好的參數設定之下，能夠獲得一個趨近於最佳解答的近似解。

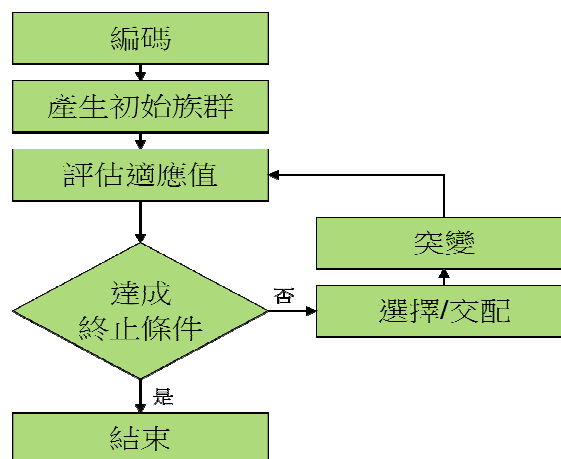


圖2-1 基因演算法的基本流程

### 第三節 基因演算法的運算子

#### 壹、編碼

常見的編碼(Encoding)方式有二進位編碼(Binary-coded)、實數編碼(Real-coded)、符號編碼(Symbol-coded)等，如圖 2-2 所示，且不同的編碼方式會影響之後的交配、突變等運算子。

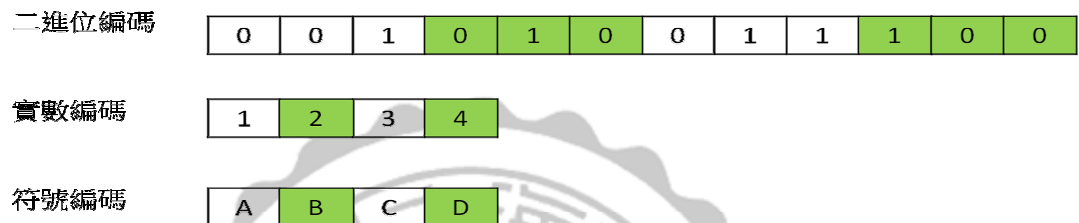


圖2-2 常見的編碼方式

#### 貳、初始化

染色體的遺傳值初始化(Initialization)主要由電腦隨機產生，能讓整個族群均勻的散布在樣本空間中。

#### 參、適應性函數

適應值函數(Fitness function)會計算染色體在符合限制式的情況下，透過事先定好的評分機制，給染色體一個代表其優劣的數值，稱為適應值，以做為選擇的參考依據。

#### 肆、選擇

透過選擇(Select)運算，我們可以決定採用哪種方式來保留上一代較優秀的染色體，並且淘汰掉不適合的染色體，進而提升族群的品質，下面我們對常見的選擇方式做個簡介：

##### (1) 輪盤法(Roulette Wheel)

輪盤法是根據適應函數所計算出適應值的好壞，對不同染色體給定一個機率

值，適應值越好的染色體，被挑選出的機會越高；適應值越差則機會越低，其公式如下：

$$P_k = \frac{f_j}{\sum_{j=1}^N f_j} \times 100\%$$

$f_j$ : 第  $j$  個染色體適應值

$P_k$ : 輪盤機率百分比

$N$ : 群體大小

## (2) 競賽法(Tournament)

由 Goldberg 等人所提出，其做法是從族群中挑出兩個或以上的染色體進行適應值比較，較好的染色體可以保留至下一代中，此方法有減少個體影響力，避免往一個染色體收斂的特色。

## (3) 菁英法(Elitism)

菁英法的概念是強迫在每一代中保留一些最好的染色體直接複製到下一代，避免優良染色體經過交配或突變後而消失。此方法可以有效的保留好的染色體，進而能夠加速收斂之效果，進而達到最佳解。

## 伍、 交配

交配(Crossover)是為族群注入新生命的重要過程，藉此可從前代族群中，產生具有其特徵的新族群，並有機會產生更好的染色體，常見的交配形式有(1)單點交配、(2)多點交配、(3)均一交配，如圖 2-3 所示。

### (1) 單點交配

單點交配是透過隨機產生單一個交配點，同時分割親代染色體，經過遺傳交換後產生新的子代。

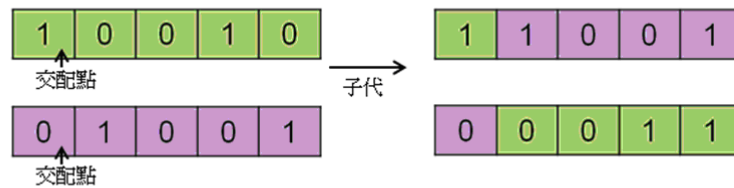
### (2) 多點交配

多點交配則是隨機找出兩個或以上的交配點，同時交錯的分割親代染色體，交換遺傳後產生子代。

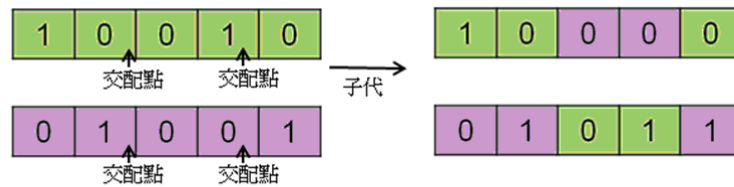
### (3) 均一交配

均一交配又稱"字罩"交配，有一長度與染色體相同的字罩，且由0與1所組成，親代染色體便依據此字罩交換遺傳，字罩為1時交換，0則維持原狀。

#### 單點交配



#### 多點交配



#### 均一交配

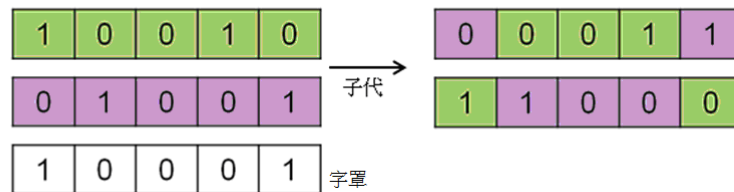


圖2-3 常見的交配形式

## 陸、突變

突變(Mutation)是一個探索的過程，能夠幫助收斂或無法前進的族群產生新的染色體，同常會給定一個較低的機率，避免對整個族群產生太大的變異。

## 柒、終止條件

常見的終止條件(Termination)有執行固定代數後終止與族群收斂，若沒有正確的設定終止條件，可能導致演算法成為無窮迴圈，或找尋不到適當的解答。

## 第四節 節能基因演算法

相關文獻有提出了基於基因演算法的收斂分析，對於進化與自主系統的決策過程，於移動式隨建即連網路進行自適應空間配置與重新配置[21]。以及基於生物地理學優化算法建立了一個最佳功率配置並減少花費總功率的算法[7]，與其他相關的算法[4,20,30]。

## 第五節 考慮地理位置的演算法與路由協定

在無線感測網路的領域中，已有相關文獻將地理位置的資訊納入染色體編碼的考量，如使用二維的染色體編碼對應實際無線感測網路的基因演算法[2]，以及根據感測器的實際地理位置，進一步藉此劃分出分群叢集的演算法，或是發現空洞位置的演算法等；在物流網路的相關研究中，也有文獻討論了地理位置資訊的重要性，並提出改善後的基因演算法，證實需要考慮地理位置的重要性。而在路由協定方面，有研究指出，若叢集內節點的位置較為相近，使用少量節點進行監測與溝通，便能有效節省能源[26]，與考慮表面形狀的幾何形狀與地理訊息相結合，建構出表面且確定節點的虛擬高度能做出有效的決策[17]。



### 第三章 地理資訊之基因演算法

當使用基因演算法求解無線感測網路問題，在編碼時若不考慮實際的地理位置資訊，可能在最好的參數設定下，卻無法獲得最佳的解答，相關文獻透過將地理位置資訊納入二維染色體編碼的方式證實能有效改善基因演算法，但過長的編碼可能因在演算法的運算(如選擇、交配與突變)不斷的走訪未攜帶資訊的多餘基因，而影響求解效率(如圖 3-1)，在總體無線感測網路的能源節省上無法達到最佳化，所以我們提出更能有效節約能的地理資訊之基因演算法(Geography-Cognizant Genetic Algorithm, GCGA)。

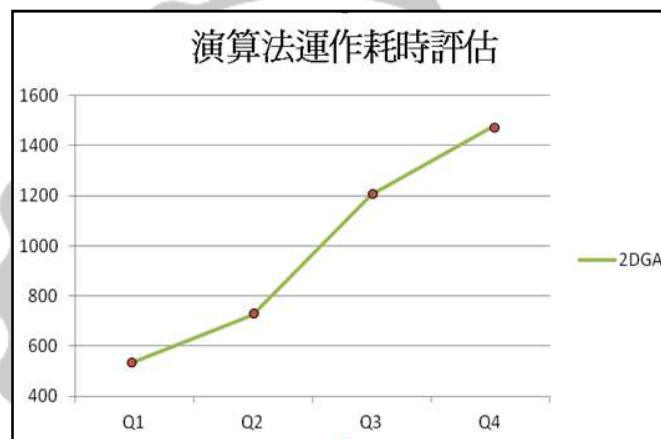


圖3-1 二維染色體編碼之基因演算法於漸增節點規模運作耗時

因此，本研究旨在考慮以下問題：

- (1) 對整體無線感測網路的結構與能源節省最佳化。
- (2) 使用基因演算法求解地理位置相關問題時，適當的將地理位置資訊融入編碼中以增進求解品質。
- (3) 對基因演算法中各種運算子做更好的選擇，使得演算法能更快速獲得解答
- (4) 在演化過程中，考慮在不同時間區段中，實際的感測器能源存量並做為決策依據。

並提出對應的解決方法：

- (1) 因無線感測網路的問題是個 NP 問題，所以採用基因演算法做為解題的工具。
- (2) 設計一個考慮地理位置的染色體編碼。
- (3) 依照問題特性，對選擇、交配與突變等運算子做出正確的改善。
- (4) 於基因演算法的進程中，對下一個時間區段的能源存量進行模擬，並且依照模擬的結果進行各種決策。

本節將解釋我們提出的演算法，用於節省能源無線感測網路的地理資訊之基因演算法，並解釋地理位置資訊如何納入染色體的編碼，進而改善現有的基因演算法。

本研究的基本假設如下：

- (1) 無線感測網路的節點位置固定
- (2) 每一節點至少有一鄰近節點在傳輸範圍內

## 第一節 考慮地理位置的基因演算法之運作

在無線感測網路中地域被叢集首領與其成員分割成許多區域，因此形成了各個叢集。每個感測節點都會搜尋離本身最近的叢集首領，藉此了解自己屬於哪一個叢集，節點會傳輸資料給叢集首領，而叢集首領則會將資料彙總並傳送給 sink 節點，如圖 3-2 所示。本文中所有的節點位置都固定，同時假設資料傳輸時不會遭遇任何干擾或傳送失敗。GCGA 的編碼方式與傳統的編碼不同，因此在進行交配與突變運算時的操作也不同，這個差異將在後面的段落進行詳細的討論。

### 無線感測網路

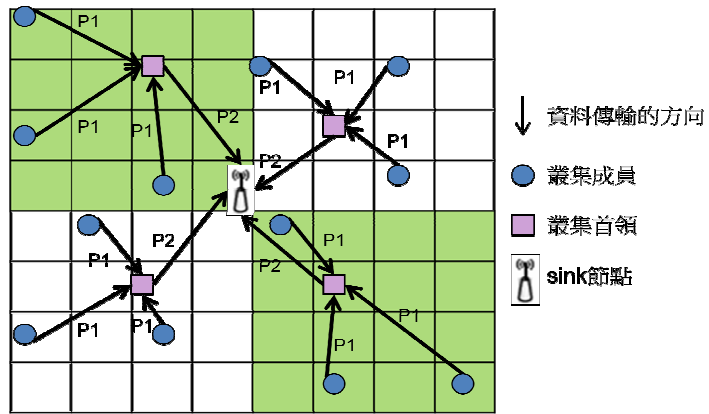


圖3-2 無線感測網路的結構

### 二維染色體編碼

	(1,1)				(2,1)			
	1	2	3	4	1	2	3	4
1	1	0	0	0	0	0	0	0
2	0	0	2	0	1	0	1	0
3	1	0	0	0	0	2	0	0
4	0	0	1	0	0	0	1	0
1	0	1	0	0	1	0	0	0
2	0	0	2	0	0	2	0	0
3	1	0	1	0	0	0	0	0
4	0	0	0	0	0	1	0	1

(1,2)      (2,2)

圖3-3 二維編碼染色體編碼

<b>V</b>																
X1,Y1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
Cx1,Cy1	1,2 3,2	2,1 3,4	2,2 4,4	1,1 3,4	1,2 1,3	1,2 2,1	1,1 1,1	1,2 3,3	2,2 2,2	2,1 2,3	2,2 1,1	1,1 1,3	2,1 3,2	2,1 1,2	2,2 2,4	1,1 3,2
(A)	(B)															

圖3-4 GCGA 染色體編碼

## 第二節 基因表示法

我們將目標無線感測網路分割成許多方格，如圖 3-2 所示，其邊長都相等，其中最多只有一個感測節點，因而直覺產生如圖 3-3 之二維編碼染色體編碼，除了需紀錄分割為四個叢集方位資料外，叢集成員標示為 1，叢集首領標示為 2。

GCGA 的染色體將記錄所有感測節點的狀態與所屬方格，圖 3-4 中的(A)是編碼的表示法，其中  $V$  是基因值，每個感測節點用 0 表示，染色體經過初始化後會隨機從中選出定量的叢集首領，以 1 表示，而  $X1$  與  $Y1$  表示所屬叢集，對應到圖 3-3 中分割出的四個區域，左上為(1,1)、左下是(1,2)、右上(2,1)與右下(2,2)， $Cx1$  和  $Cy1$  代表節點在各叢集中所對應的方格位置，以圖 3-3 中(1,1)叢集中最左上的叢集首領為例，其位置是 3,2；叢集(2,2)最右下的叢集首領位置是 2,2。我們透過記錄節點的所屬方格的方式，將實際地理位置資訊納入染色編碼，同時比現有的二維編碼方式更為精簡，圖 3-2 中描述同一無線感測網路的二維染色體的長度是 64，而圖 3-4 (B)中 GCGA 的染色體長度僅有 16。

現在讓我們進一步的討論編碼長度的問題，因二維染色體的編碼規則是將無線感測網路分割成等大小的方格，且方格中只能存在一個節點，若節點均勻分布時，無線感測網路的二維染色體編碼與 GCGA 染色體的長度之比較結果如表 1 所示，之間的關係呈現正向關係。若節點不是均勻分布，二維染色體編碼與 GCGA 染色體長度的差距可能會更加明顯，若圖 3-2 的無線感測網路分布變為圖 3-5，則對應的二維染色體將如圖 3-6 所示，長度為，同一條 GCGA 的染色體與二維染色體的長度比例將從 16:64 變為 16:100，差異將更加的顯著。

表3-1 演算法編碼長度之比較

編碼長度									
GCGA	2	3	4	5	...	10	...	100	...
2DGA	4	9	16	25	...	100	...	10000	...

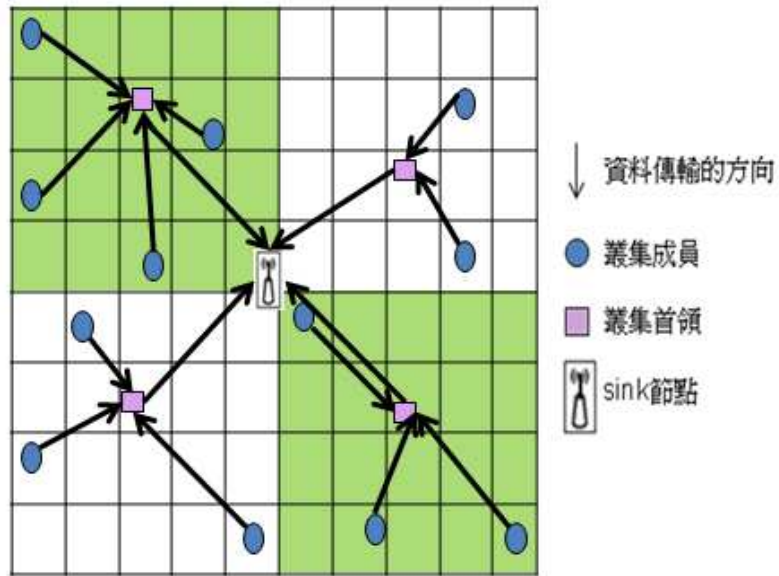


圖3-5 節點移動後之無線感測網路

1	0	0	0	0	0	0	0	0	0
0	0	2	1	0	0	0	0	1	0
1	0	0	0	0	0	0	2	0	0
0	0	1	0	0	0	0	0	1	0
0	1	0	0	0	1	0	0	0	0
0	0	2	0	0	0	0	2	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	0	1

圖3-6 節點移動後之二維染色體編碼

### 第三節 族群初始化

從能源節省的觀點來看，從所有感測節點選出 5% 的叢集首領可以達到此目的[9]。族群的初始化如果有好的開始，我們離找到最佳解就更向前邁進一步，因此，進行模擬時，我們將研究在選擇叢集首領的機率上，是否有能幫助我們得到更早找到最佳解的良好設定。

### 第四節 適應值函數

無線感測網路的最佳化的定義，是透過減少每個感測節點能消耗，並增加整體網路的持續性。在相關研究中，基因演算法的適應值函式並沒有考慮到傳輸節點的跳數，所以我們定義了兩個適應值函式，並比較以這兩個方向為演化目標時，所求得的無線感測網路之優劣。

適應值函式 F1 的詳細介紹如下所述：我們將無線感測網路分為兩個部分，在圖 5 中，第一部分是從感測節點到各叢集首領，以 P1 表示，第二部分是叢集首領到 sink 節點，用 P2 表示，並且將各部分節點間的距離數值平方後加總，視為傳輸時的總能源消耗，如公式，其中  $j$  為各叢集中成員的數量， $i$  為共有多少個群集首領， $d$  是兩點間的距離，越小的總花費表示此網路越接近最佳化。在此無線感測網路中，叢集首領的數量與位置是影響成本多寡的重要因素，整體的網路成本可能會因為叢集首領數量過少，叢集首領至感測節點的能源消耗增加，造成第一部分的成本提高，或是叢集首領的數量過多，叢集首領至 sink 節點的能源消耗變多，導致第二部分的成本提高，因此，找到合適數量的叢集首領，可以促進總體無線感測網路的成本最小化，此外，在每一代的演化中，GCGA 會找出最低成本的染色體，做為此代的最佳解答。

適應值函式 F2 較 F1 多考慮了一個參數 HOP，我們把從資料的來源點至目標點所經過的無線感測節點的傳輸跳數定義為 HOP，此參數乘以一個權重係數  $W$ ，做為演化目標，公式如下所述：

F1:

$$Cost_{total} = \sum_i \sum_j d_{P1(i,j)}^2 + \sum_i d_{P2(i)}^2 \quad (1)$$

F2:

$$Cost_{total} = F1 + W * HOP \quad (2)$$

根據適應值函數得到的適應值，我們會計算無線感測網路的能源消耗，並定義了三個參數：Sum、CpD(Cost per Distance)與 CpT(Cost per Timeslot)，程式會計算每一個叢集首領所連接的叢集成員與 sink 節點距離之總和(Sum)，除以 CpD(每距離單位會消耗能源)，再乘以 Cpt(每距離單位於一時間區段中會消耗的能源百分比)作為此叢集首領的能源消耗，舉例來說，若 Sum=800、

CpD = 100、Cpt = 3，在一時間區段中，若一個叢集首領連接的叢集成員與 sink 節點距離之總和為 850， $(850/100)3=25.5$ ，得到能源消耗量為 25.5%。

## 第五節 選擇

我們在進行實驗後決定同時使用了競賽法與菁英法。透過競賽法選擇，我們可以比使用輪盤法更能保存優秀的染色體。使用菁英法，也可以確保到目前為止最佳的染色體能持續存在於族群中，藉此提升族群演化的效果。

## 第六節 交配

John Holland 曾提出綱要理論(schema theory)以描述演化的過程與效果 [12]，說明帶有高於平均適應值的短小低秩(order)綱要可以顯著提升下一代族群的平均適應值。Goldberg 則根據綱要理論提出砌塊假說，說明經過選取、重組與再選取高品質與短小低秩的綱要，能有較大的機會形成具有更好適應值的染色體。問題可以拆解成許多子問題，每個子問題的解答可視為一個砌塊。我們為了不破壞子問題所產生的砌塊，因此不使用一般的單點、多點或均一交配，而是依照無線感測網路中的所屬叢集進行交配，如圖 3-7 所示，親代染色體交換了叢集 2,2 的遺傳而產生子代。

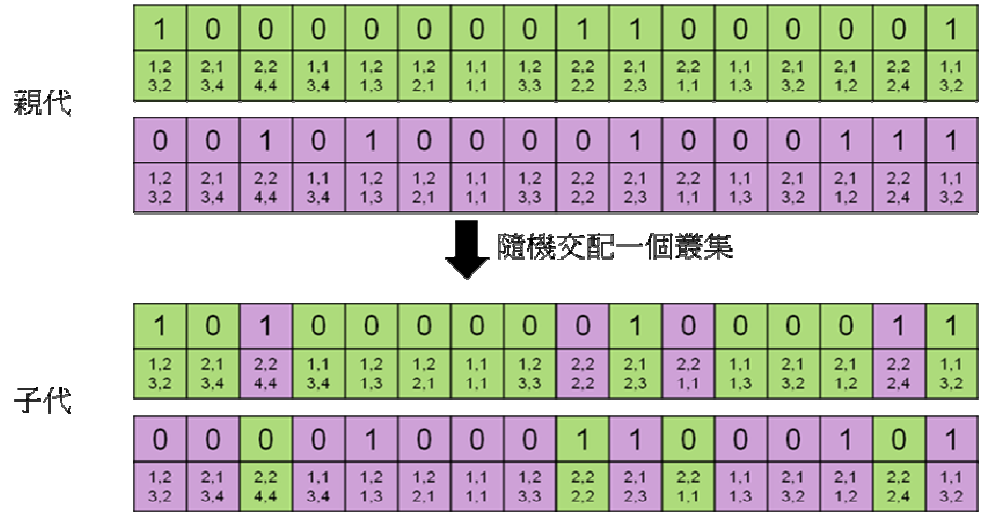


圖3-7 GCGA 的交配運算





## 第四章 實驗設計與方法

本研究實驗環境的硬體配備如表 4-1 所示，使用 Intel 廠牌 i5 的中央處理器，並配置有 8.00GB 之記憶體，且在 Windows7 作業系統中運作：

表4-1 實驗環境之配備

	規格
CPU	Intel® Core™ i5-2500 CPU @ 3.30GHz 3.30GHz
記憶體	8.00GB
作業系統	64 位元作業系統 Windows7 旗艦版

本研究的實驗題目共有五個，題目的細節如表 4-2 所示，分別命名為 Q1、Q2、Q3、Q4 與 Q5，當中配置了一個資料的來源端點與資料的匯集點，除了上述兩的節點之外，其餘的節點皆為隨機分佈。

從 Q1 至 Q5，點數從 20 遞增為 180；最大邊長從 200 至最大值 1800；叢集數量從 1x1 增加為 5x5；最大格子數量為總點數隨機分佈後，依照等距離切割成符合大小相等、且每個格子中最多僅會有一個點之條件的參數，其值從 5 至 25；叢集格子數量是最大格子數量除以叢集數，將此參數平方計算出每個叢集中最多可容納幾個節點，其值從 5，從此可計算叢集中可容納的最大點數為  $5 \times 5 = 25$ 。

表4-2 題目參數

題目編號	Q1	Q2	Q3	Q4	Q5
最大邊長	200	400	600	800	1000
叢集數量	1x1	2x2	3x3	4x4	5x5
邊長格子數量	5	10	15	20	25
叢集內格子數	5x5	5x5	5x5	5x5	5x5
總點數	20	40	60	80	100

以題目 Q1 為例，因其最大邊長為 200，所以資料來源點  $n_0$  的座標設為(0,0)，資料匯集點  $n_{19}$  的座標則為(200,200)。因 Q1 僅有一個叢集，在最大格子數量為 5、叢集格子數量為 5 的設定之下，此叢集中剩餘的 18 個節點將隨機分佈於 5x5 的空間內。

若以題目 Q5 為例，其最大邊長為 1000，所以資料來源點  $n_0$  的座標設為(0,0)，資料匯集點  $n_{179}$  的座標則為(1000,1000)。因 Q5 的長寬叢集數量設定為 5，因此共有 25 個叢集，在最大格子數量為 25、叢集格子數量為 5 的設定之下，題目剩餘的 98 個節點將隨機分佈於 25x25 的空間內。

我們實驗的目的在於使用較單純的基因演算法(GA)、二維染色體之基因演算法(2DGA)、與本研究提出的地理資訊之基因演算法(GCGA)間，求解無線感測網路時，所得的解答品質與花費的求解時間，藉此辨別出哪一種演算法較為適合求解 WSN 問題。

為了進一步的比較這三種演算法，我們將實驗分為三個部分做更深入的探討：

- (1) 演算法求解進程實驗
- (2) 模擬耗能實驗
- (3) NS2 網路模擬實驗

本實驗中的基因演算法(GA)是採用競賽法進行選擇，並使用單點交配，二維染色體編碼之基因演算法(2DGA)是使用競賽法與菁英法，使用叢集交配，地理資訊之基因演算法(GCGA)是使用競賽法與菁英法，使用叢集交配。

## 第一節 演算法求解進程實驗

演算法求解進程實驗是為了比較三種演算法，在不同題目間決定叢集式無線感測網路的叢集首領時，需要花費的時間、時間內所能找到的最佳解答與求解過程，以鑑別出是否適合求解叢集式無線感測網路之問題。

本部分的實驗環境，是透過 Visual Studio 2005 程式，使用 C++ 程式語言撰寫程式，編譯後輸出的執行檔便是實驗所用的基因演算法程式，流程如圖 11 所示。

程式一開始會讀取包含題目節點座標的文字檔案，接著判斷是否執行至終止時間區段（演算法求解進程實驗設定為 1；模擬耗能實驗則大於 1），為否則進入基因演算法的程式區塊，首先產生初始族群，評估適應值後判斷是否執行至終止代數，為否則進行交配與突變，重複基因演算法的程式區塊至達成終止代數，接著程式執行至終止時間區段後，輸出實驗結果與 NS2 網路模擬實驗用的最佳解 tcl 檔案後結束。

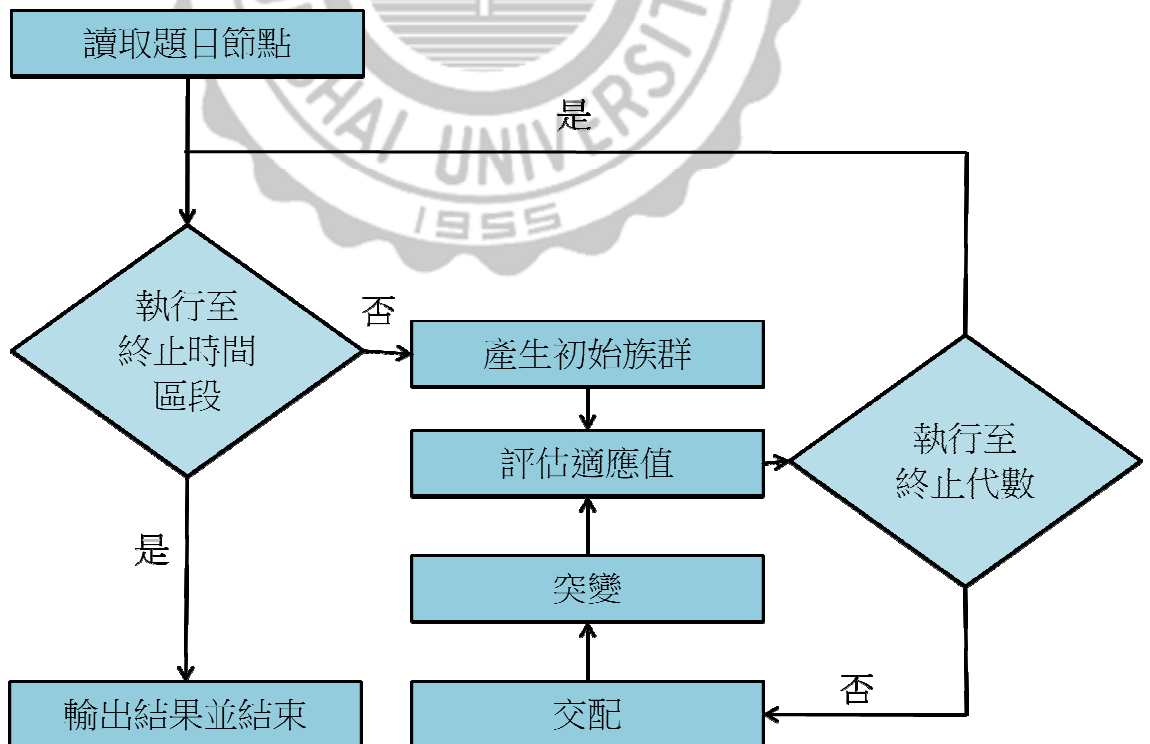


圖4-1 基因演算法實驗流程圖

本實驗的虛擬碼如下所示：

```
Main()
{
    TimeSlots //TimeSlots of Simulation
    Population //population of GA
    Generation //generation of GA
    LoadQuestion();

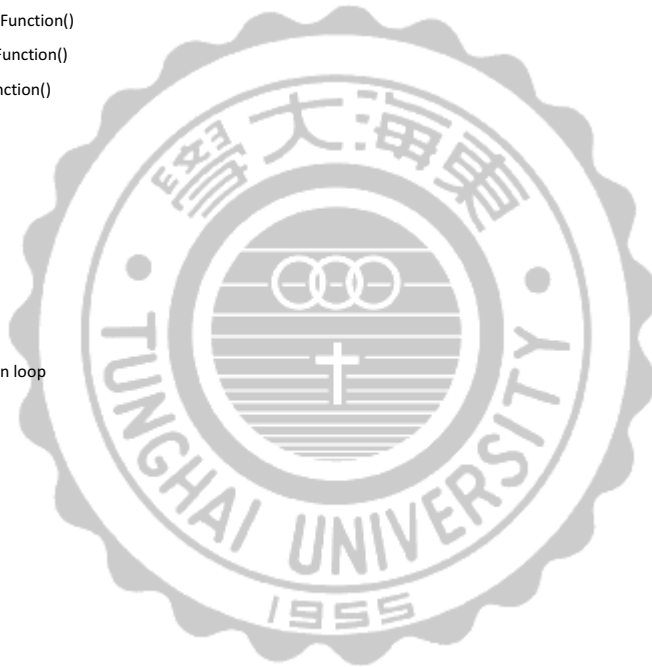
    For(;;) //
    {
        Initial()
        FitnessFunction()

        For(;;) //generation loop
        {
            CrossOverFunction()
            MutationFunction()
            FitnessFunction()
        }
    }
    StatisticsFunction()
}

CrossOverFunction()
{
    For(;;) //population loop
    {
        Select_Parent
        CrossOver
    }
}

MutationFunction()
{
    For(;;) //population loop
    {
        Mutation
    }
}

StatisticsFunction()
{
    Compute_BestChromosome_Fitness
    Compute_Generation_BestChromosome
    _Average_Fitness
}
```



```

FitnessFunction() //Function1
{
    For(;;) //population loop
    {
        Compute_Distance_Source_To_Destination
        Compute_ClusterHeader_Punishment
    }
}

```

```

FitnessFunction() //Function2
{
    For(;;) //population loop
    {
        Compute_Distance_Source_To_Destination
        Compute_ClusterHeader_Punishment
        Compute_Hop_Punishment
        Compute_Fitness
    }
}

```

```

LoadQuestion()
{
    Load_Points_File
    Compute_Points_Distance
}

```

```

Initial()
{
    For(;;) //generation loop
    {
        Initial_GA_Parameter
    }
}

```



## 第二節 模擬耗能實驗

模擬耗能實驗是比較三種演算法於同一題目，在數個時間區段中，符合可接受的最低剩餘能源參數，且決定每個時間區段中的叢集首領。我們將比較各演算法所花費的時間、求解過程與各時間區段的能源消耗，判斷演算法對於節省能源是否能發揮功效。

## 第三節 NS2 網路模擬實驗

NS2 網路模擬實驗，是透過知名的網路模擬程式 NS2，將演算法求解過程實驗的最佳解答，輸出成為 NS2 可接受的檔案格式並進行模擬，模擬的參數如表。我們將模擬實驗的結果輸出並比較各演算的封包遺失、網路延遲、抖動率與流量，判斷演算法所求得的最佳解答，在實際的網路運作上是否符合期望。

## 第四節 演算法求解過程實驗設計

各演算法的參數設定如表，每個實驗皆重複進行 20 次的測試，為了公平的比較三種演算法，我們採用了相同的參數設定。顧慮到實際運作時，感測器的運算能力，我們今染色體數目設定為 40，且有著較高的 80% 交配率，促進族群的演算速率，與較高的 8% 突變率，使得族群不易過早收斂而陷入區域最佳解，終止條件為演算法執行 300 個世代後便結束。

表4-3 演算法參數設定

演算法	GA	2DGA	GCGA
染色體數目	40	40	40
代數	300	300	300
交配率	80%	80%	80%
突變率	8%	8%	8%

## 壹、適應值函式 F1 之實驗結果

從三種演算法在 Q1 實驗題目的截至目前最佳解(圖 4-2)，圖中可看出 GA 的表現最不理想，而將無線感測網路的地理位置納入編碼考慮的 GCGA 與 2DGA 明顯勝過 GA，而 GCGA 的求解進度比 2DGA 更好，此時 2DGA 略勝過 GCGA 執行結果。我們推斷，在此實驗中，GA 可能因族群過早收斂而陷入區域最佳解，而 2DGA 雖勝過 GCGA，但差異並不大，可能因為題目的節點數量較少且分布較為稀疏，使得 GCGA 與 2DGA 都能有不錯的表現。

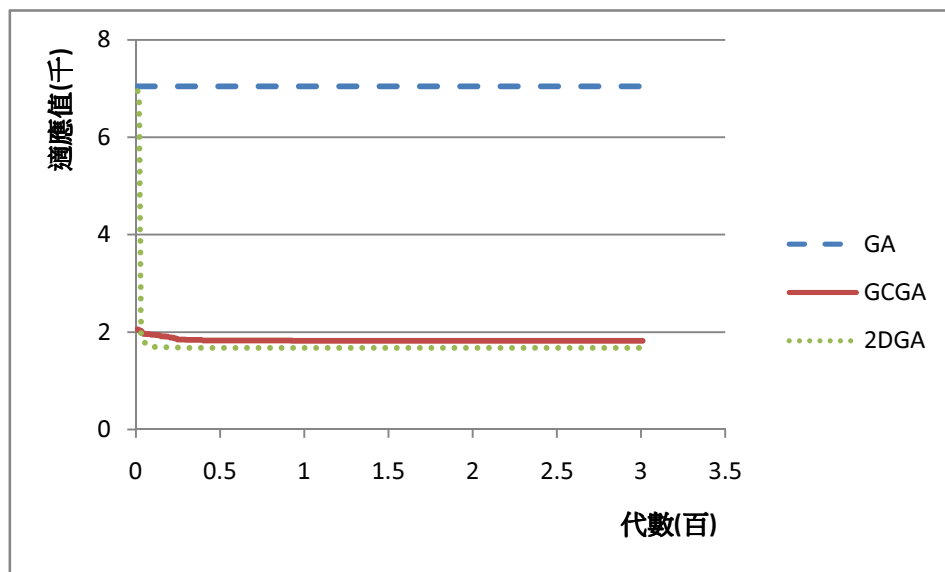


圖4-2 F1 求解題目 Q1 截至目前最佳解

求解實驗題目 Q2 時，從圖 4-3 中可以看出 GCGA 與 2DGA 勝過 GA。於本實驗中，因題目的節點數量提升而變得稍加複雜，我們推估，可能因問題規模的增加，使得 GCGA 在此問題中略勝過 2DGA。

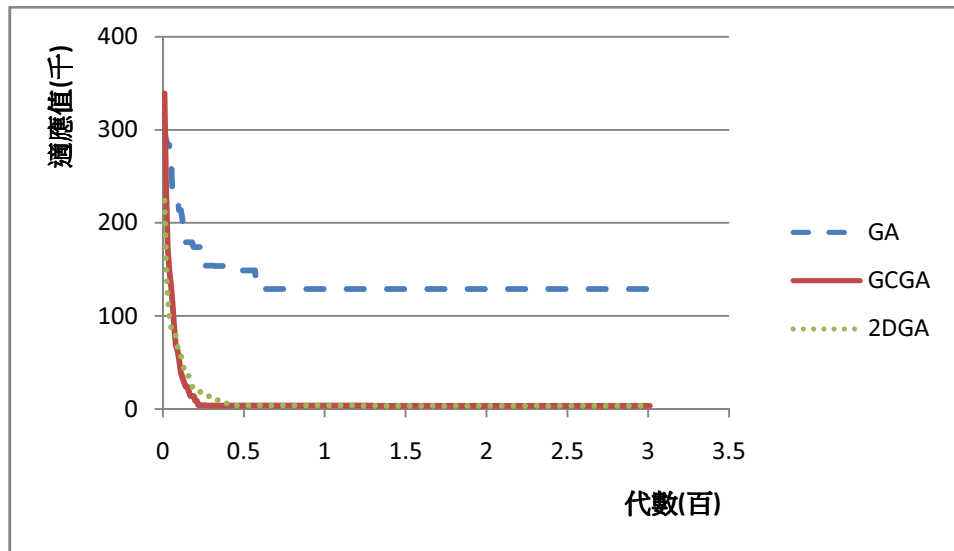


圖4-3 F1 求解題目 Q2 截至目前最佳解

當實驗題目為 Q3 與 Q4 時，我們可看到 GA 的表現仍然最差(圖 4-4, 4-5)，而至此問題規模，相較於 2DGA，GCGA 的曲線有著更明顯的陡降，顯示了探索較佳解答的能力差異，這種情況在前 100 代特別明顯。

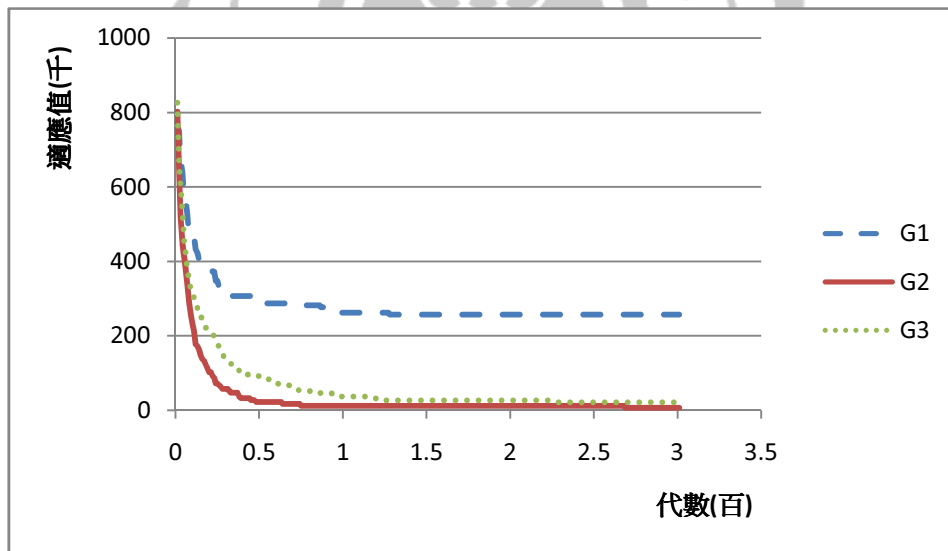


圖4-4 F1 求解題目 Q3 截至目前最佳解



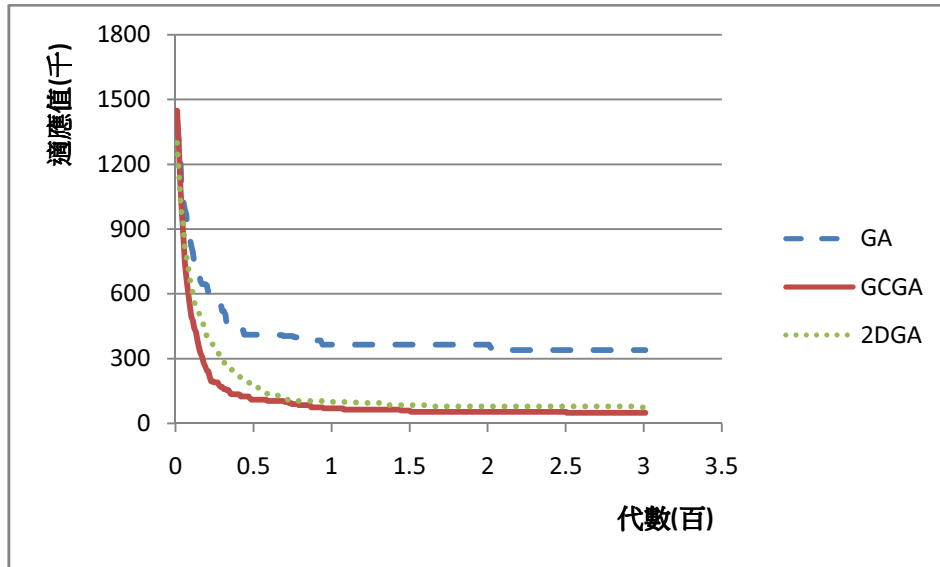


圖4-5 F1 求解題目 Q4 截至目前最佳解

在問題 Q5 中，因為題目規模的增加，我們可以看到在前 100 代 GCGA 最佳解適應值的陡降率比 2DGA 高出許多(圖 4-6)。而 2DGA 的求解能力在前 30 代與較差的 GA 差距較小，我們推估是冗長的編碼夾帶的無效資訊所導致。

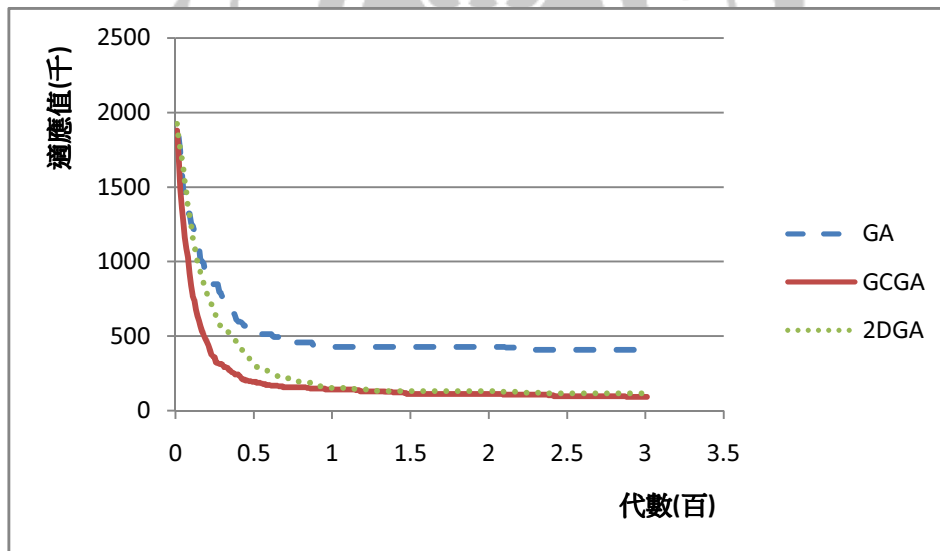


圖4-6 F1 求解題目 Q4 截至目前最佳解

在總實驗耗時的比較方面，如圖 4-7 所示 GCGA 與 GA 的耗時略相等，但兩者都遠低於 2DGA 的耗時，只需花到 1/34 至 1/11 之時間，GCGA 就可得到優於 2DGA 之叢集式無線感測網路之解答。

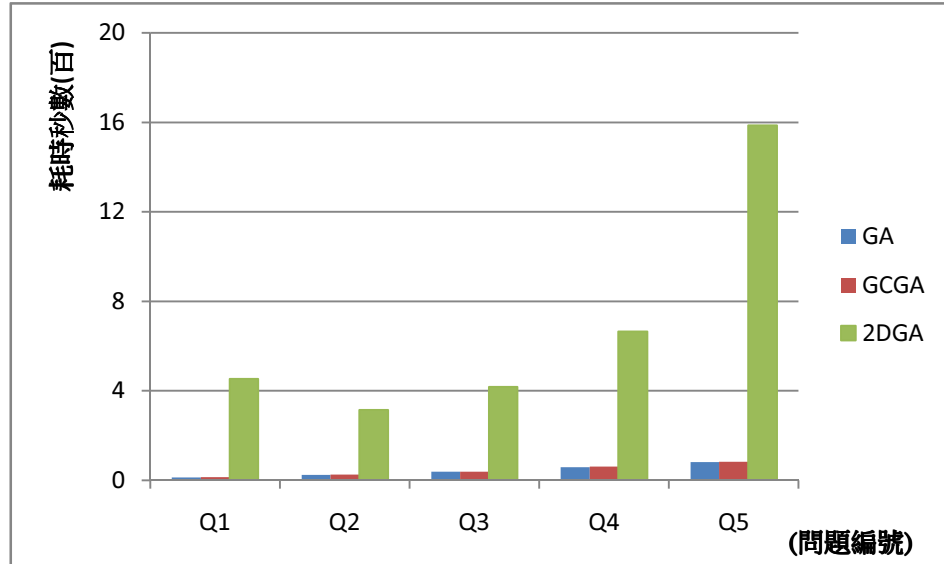


圖4-7 F1 中各演算法總實驗時間之比較

## 貳、適應值函式 F2 之實驗結果

從三種演算法在 Q1 實驗題目的截至目前最佳解(圖 4-8)，圖中可看出 GA 的表現最不理想，而將無線感測網路的地理位置納入編碼考慮的 GCGA 與 2DGA 明顯勝過 GA，而 GCGA 的求解進度比 2DGA 更好，此時 2DGA 略勝過 GCGA 執行結果。我們推斷，在此實驗中，GA 可能因族群過早收斂而陷入區域最佳解，而 2DGA 雖勝過 GCGA，但差異並不大，可能因為題目的節點數量較少且分布較為稀疏，使得 GCGA 與 2DGA 都能有不錯的表現。

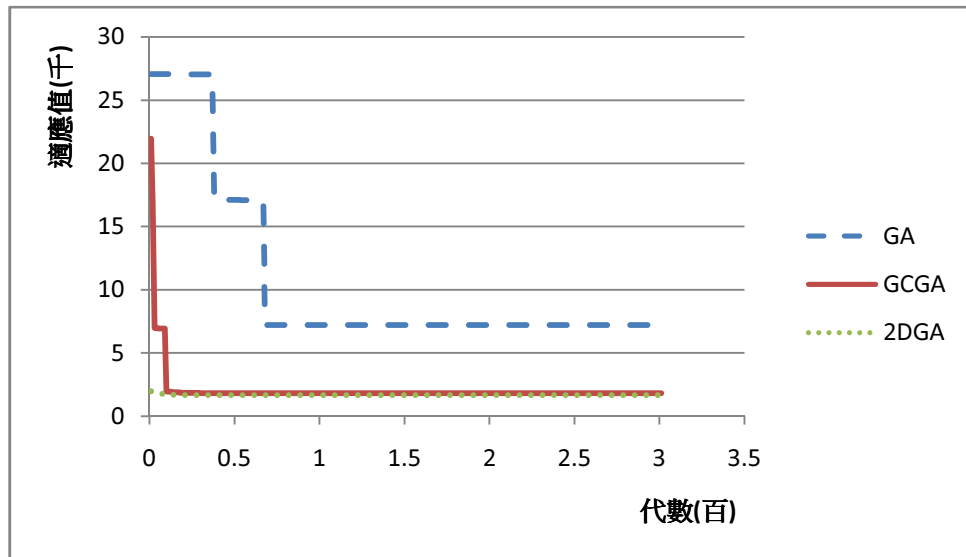


圖4-8 F2 求解題目 Q1 截至目前最佳解

求解實驗題目 Q2 時，從圖 4-9 中可以看出 GCGA 與 2DGA 勝過 GA，自此實驗開始，GCGA 便勝過 2DGA，此情況在前 30 代較為明顯。在題目規模稍小的情況下，GCGA 與 2DGA 的差距還不夠明顯，但沒考慮地理位置資訊的 GA 始終都處於落後的位置。

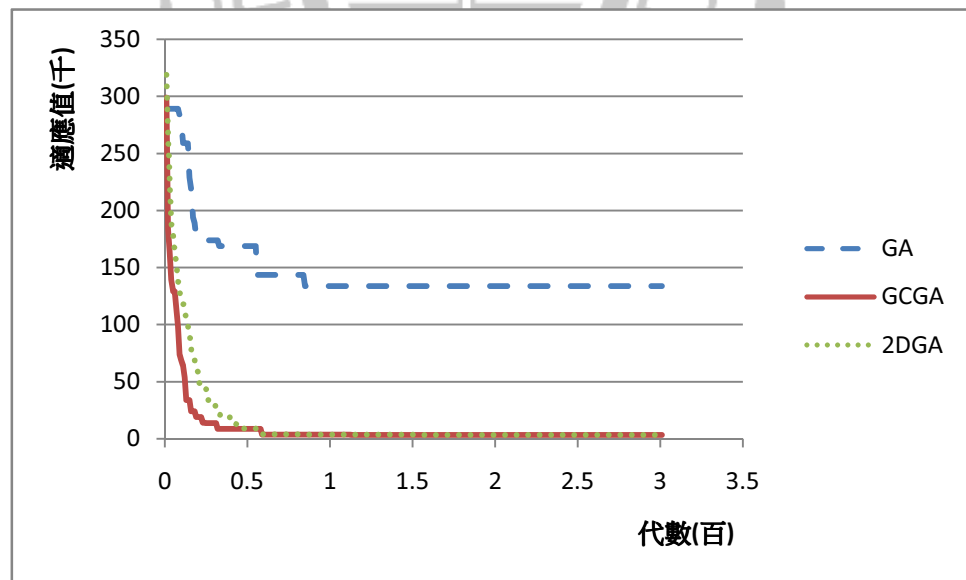


圖4-9 F2 求解題目 Q2 截至目前最佳解

求解實驗題目 Q3 時，首先我們可以看到隨著題目難度的增加(圖 4-10)，GA 的表現也越來越落後，其初始解的適應值高出許多，而最佳解的趨勢也無法跟上 GCGA 與 2DGA。且在前 150 代的演化過程中，2DGA 得到的最佳解都無法趕上 GCGA。

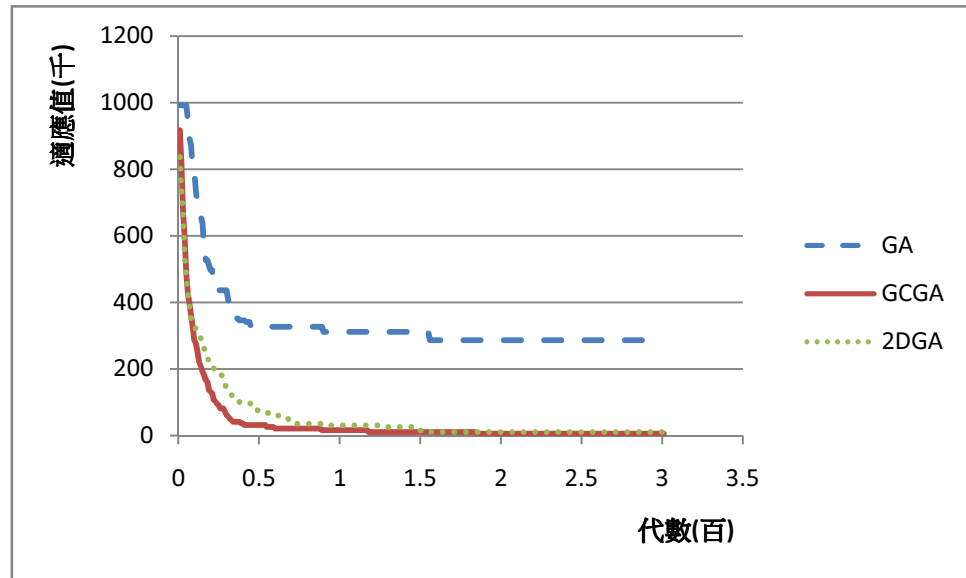


圖4-10 F2 求解題目 Q3 截至目前最佳解

在實驗題目 Q4 與 Q5 中，分別在圖 4-11 與圖 4-12 可以看到其結果幾乎一致，這說明了題目規模若達到一個程度，可看到一個較固定的趨勢，GA 的初始解與演化過程呈現明顯落後且經過更多演化也無法趕上 GCGA 與 2DGA，且在 140 代之前，GCGA 得到的最佳解答也勝過 2DGA。

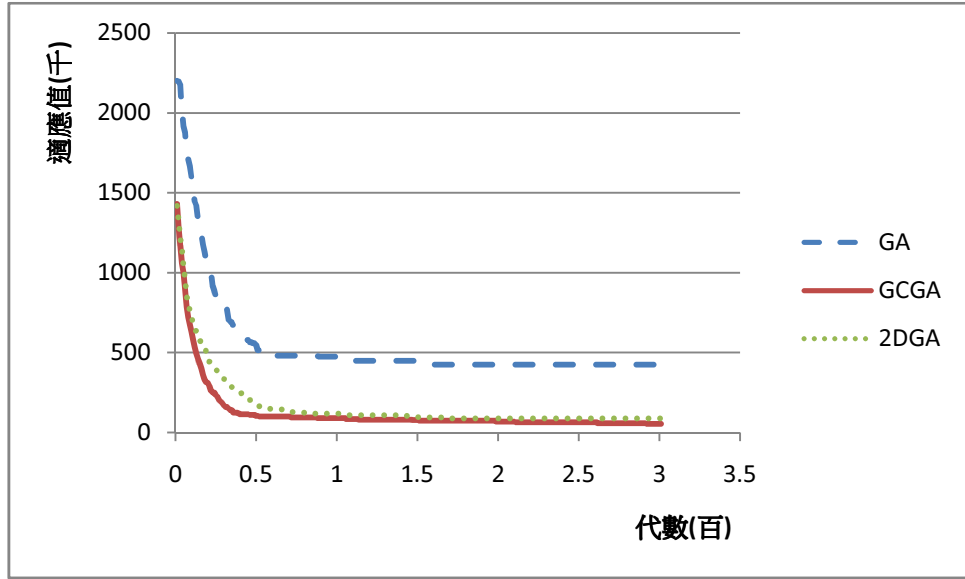


圖4-11 F2 求解題目 Q4 截至目前最佳解

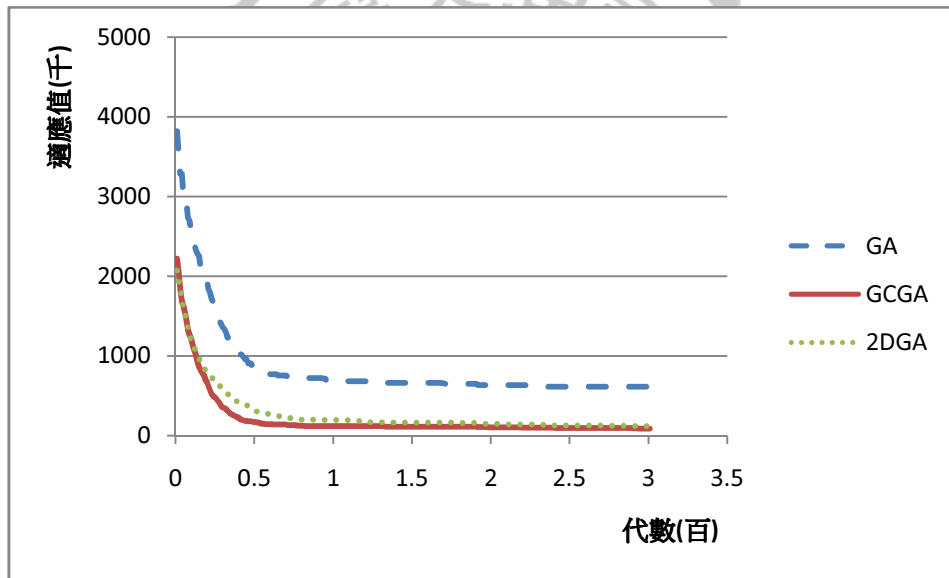


圖4-12 F2 求解題目 Q5 截至目前最佳解

在總實驗耗時的比較方面，如圖 4-13 所示 GCGA 與 GA 的耗時略相等，但兩者都遠低於 2DGA 的耗時，只需花到 1/33 至 1/11 之時間，GCGA 就可得到優於 2DGA 之叢集式無線感測網路之解答。

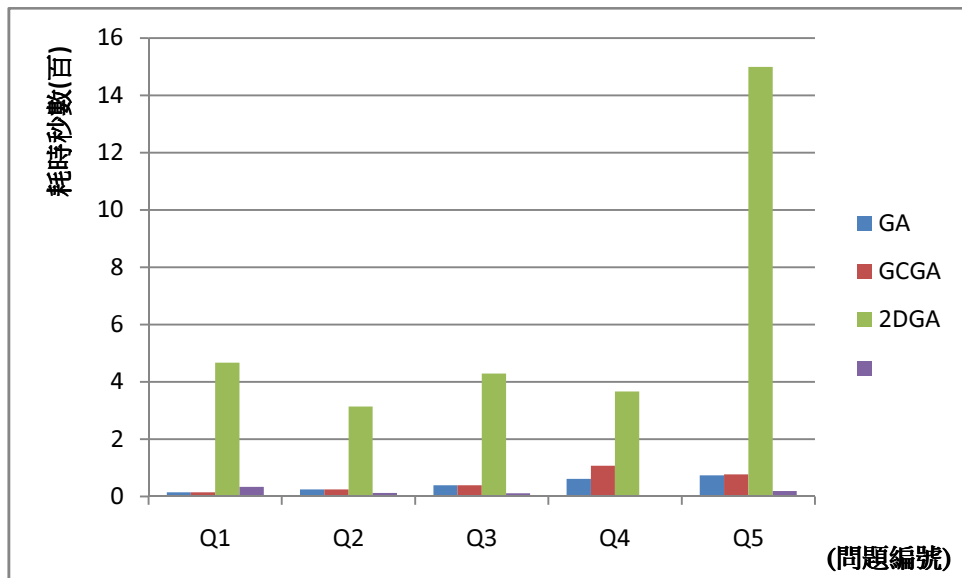


圖4-13 F2 中各演算法總實驗時間之比較

### 參、小結

在演算法的求解進程實驗中，我們知道，當問題規模增大時，GA 將越來越難找出好的解答，這是沒有考慮地理位置資訊產生的影響，且當族群較小時，若每次交配時破壞解答的短小低秩，將拖累演化速度與結果。雖然 2DGA 所得的結果較接近 GCGA，但所花費的時間遠遠超過 GCGA，也無法得到更好的結果。在本部分的實驗中，我們提出的 GCGA 確實能比 GA 得到更好的解答，且花費的時間遠低於 2DGA，證明有不錯的求解能力。

## 第五節 模擬耗能實驗設計

本部分的實驗主要是測試於一段時間內，在固定的無線感測網路中，若考慮感測節點的能源消耗時，GA、GCGA 與 2DGA 這三種演算法的求解進程、花費時間與總體能源殘量的變化。並希望透過上述的實驗數據推斷，若考慮感測器能源時，三種演算法所得結果之優劣。模擬耗能實驗的參數如表 4-4，時間區段設定為 5，代表實驗將模擬需要每隔一段時間，評估節點能源存量，並找出新的叢集首領的次數。

表4-4 模擬耗能實驗參數設定

演算法	GA	2DGA	GCGA
染色體數目	40	40	40
代數	300	300	300
交配率	80%	80%	80%
突變率	8%	8%	8%
時間區段	5	5	5
長度單位耗能	20	20	20
耗能量	1	1	1

### 壹、 模擬耗能實驗結果

在本部分的實驗結果中，模擬耗能實驗之演算法進程比較與耗時比較仍與前一階段的實驗相同(圖 4-14)。演算法進程比較方面(圖 4-15)，是 GCGA 優於 2DGA 與 GA；演算法耗時比較(圖 4-16)則是 GCGA 略同於 GA 而勝過 2DGA。最後在五個時間區段的能源存量比較中，我們可以看到隨著時間的推進，GCGA 的能源存量也高於其他兩者，且差距愈趨明顯，我們推論，若考慮了地理位置之資訊，的確能幫助整體無線感測網路的能源節省效果。

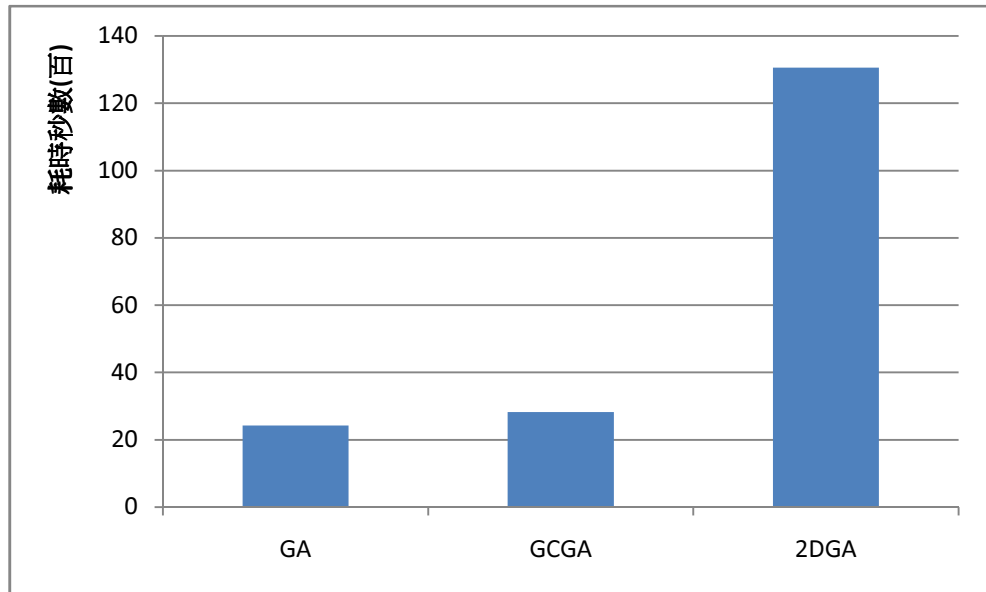


圖4-14 模擬耗能實驗之演算法耗時比較

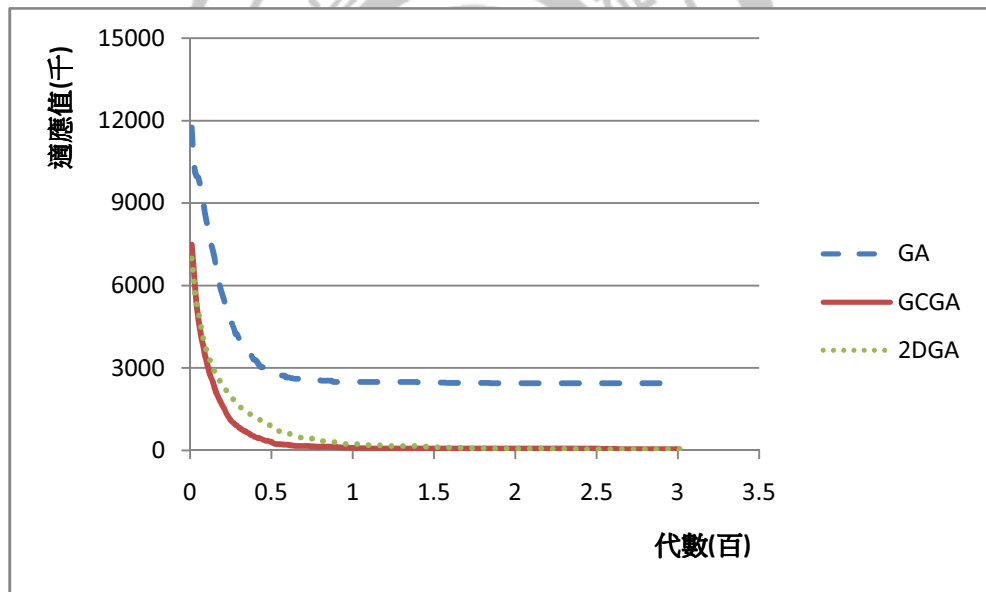


圖4-15 模擬耗能實驗之演算法進程比較



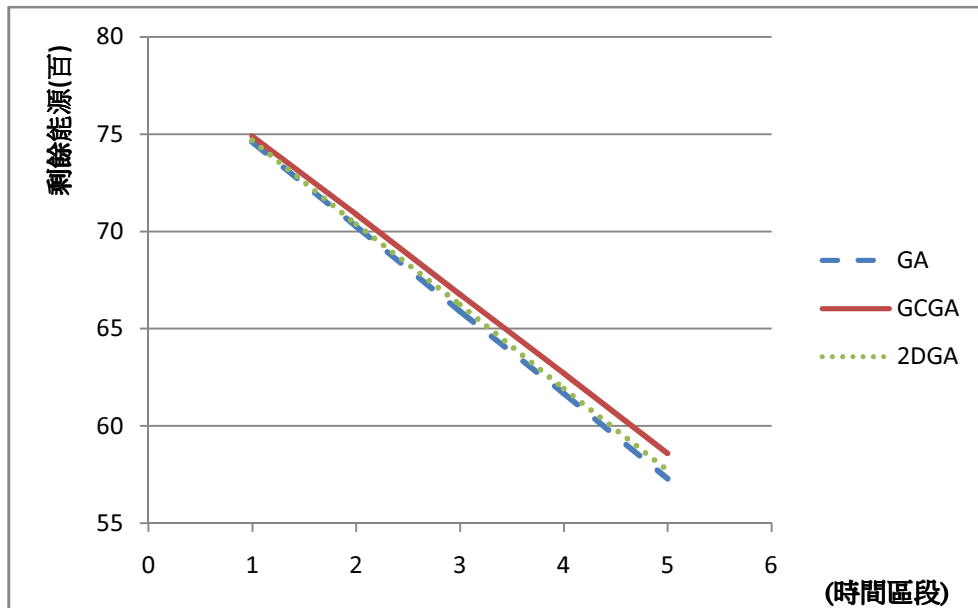


圖4-165 個時間區段中的能源存量比較

## 第六節 NS2 網路模擬實驗-NSG

Network Scenario Generator (NSG)是一個專為 NS2 所設計的劇本產生器，使用 NSG2 時，主要分成五個模式，這些模式大多都是依照 NS2 中主要的幾種物件來設計。NSG2 中包含以下五種模式，我們接著做進一步的說明：

- (1) Hand mode
- (2) Node mode
- (3) Agent mode
- (4) Application mode
- (5) Parameters

### 壹、 Hand mode

在 Hand mode 模式下的功能最為單純，主要是讓使用者可以調整要觀看的範圍。當建置一個大型的劇本時，可能會包含大量的 Node，畫面可能容納不下所有的物件，這時可利用 Hand mode 來調整要觀看的範圍，使用時只要按住滑鼠的左鍵，然後拖曳即可。

## 貳、 Node mode

在 Node mode 模式下，使用者可配置 Node 的位置，在設計無線網路劇本時，Node 的座標就變的非常重要，因為每一個 Node 的座標將會影響到傳輸範圍、訊號強度以及 Routing protocol 等，所以在建置無線網路劇本時，NSG2 會在畫面下方顯示出座標的資訊作為參考。並且 NSG 會假設無線網路 Node 的傳輸距離為 250m，所以當二個 Node 的傳輸距離在 250m 以內時，NSG2 會在二者之間畫一條連線，使用者可以透過滑鼠點選 Node 並拖曳來調整所要擺放的 Node 位置。

## 參、 Agent mode

在 Agent mode 中，使用者可設定要附加在 Node 上的通訊協定（在 ns2 中稱為 Agent），目前 NSG2 主要支援 TCP（包含多種變形）和 UDP 通訊協定。在 Agent mode 除了設定 Agent 所屬的 Node 之外，還可以設定傳送端和接收端的相關資訊。

## 肆、 Application mode

在 Application mode 模式底下可以用來設定最上層的應用程式，例如 CBR 與 FTP，設定時只要在要附加的 Agent 上按一下滑鼠左鍵，然後再點選要擺放 Application 的位置即可，同樣的 Application 的重點在於附加的 Agent 而不是在於其位置，所以位置的設定只要在清楚容易辨識的位置即可。

## 伍、 Parameters

另外還有參數設定的功能，在建立無線網路劇本時，則另外多了無線網路（Wireless）及頻道特性（Channel）的相關設定。當一切都設定完畢之後，按下 TCL 按鈕即可產生 TCL 劇本檔，使用者這時候還可以自行在劇本檔上作一些調整，調整完畢後便可以將檔案儲存並利用 NS2 去執行。

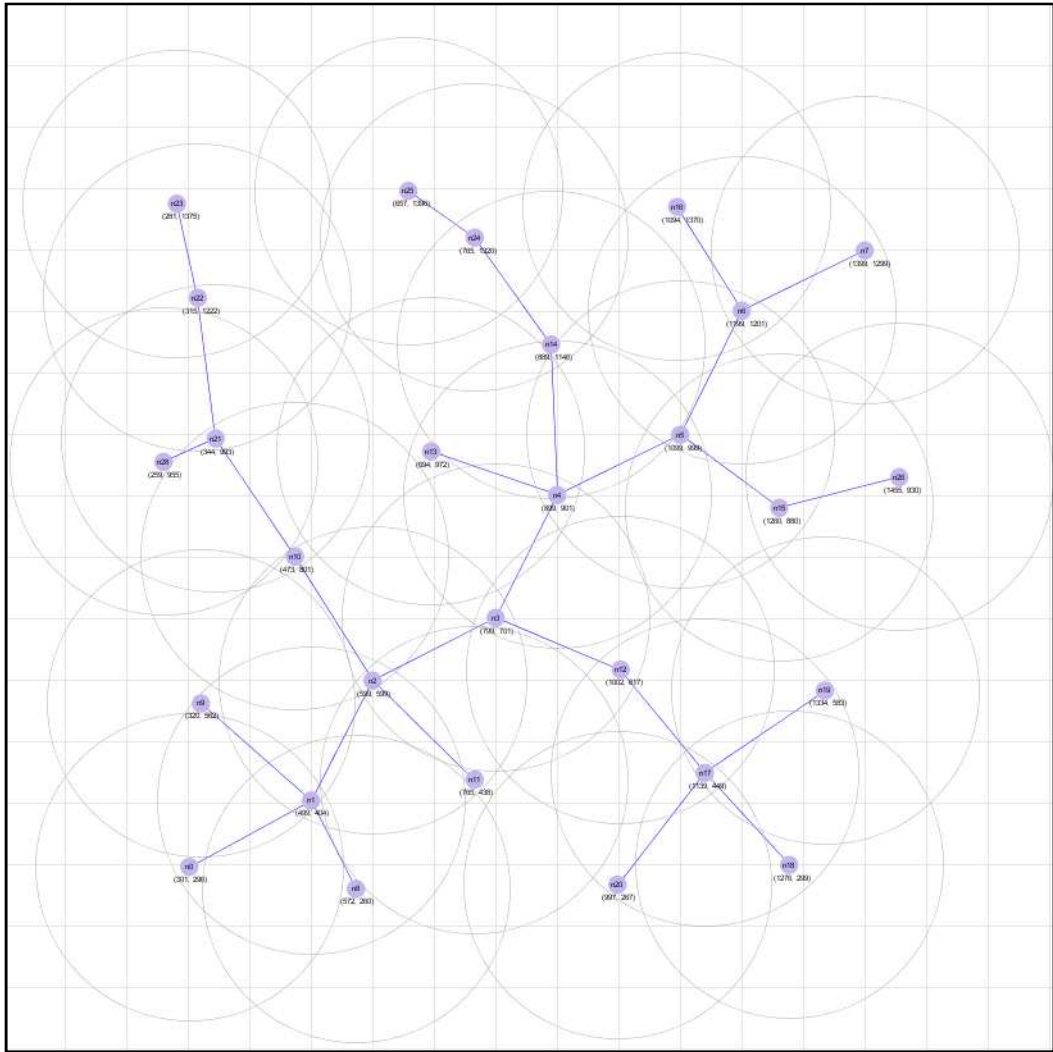


圖4-17 NSG2 劇本模擬

圖 4-17 即為運用 NSG2 劇本產生器所設計，並產生的 NS2 無線網路實驗模擬圖。再以產生的 TCL 劇本檔，進行調整與修改設計實驗。

## 陸、 NS2 網路模擬實驗-網路效能測量

本實驗的布第在於量測以 UDP 為傳輸協定的應用程式之封包遺失率 (packet loss rate)、封包延遲 (packet delay)、抖動率 (packet jitter)、和吞吐量 (throughput)。

## 柒、背景知識

當模擬結束後，我們通常會需要使用模擬過程中所產生的模擬過程記錄檔來分析以得到封包遺失率、封包延遲、抖動率和吞吐量。所採用的方法是當有一條以 UDP 為傳輸協定的應用程式開始發送資料時，當封包從應用程式層到 UDP 層時，我們就把封包的序號、時間、和大小記錄到一個檔案中（發送端記錄檔），而當封包到達接收端時，同樣地我們把收到的封包序號、傳送時間（在傳送端要發送封包時會把當時的時間放在封包的標頭檔[common header]中）、到達時間、封包延遲時間（= 到達時間 - 傳送時間）、和封包大小記錄到另一個檔案中（接收端記錄檔），有了這兩個記錄檔，我們要求得吞吐量、封包延遲、抖動率、或封包遺失率就變得很簡單了。

執行方式: `$ns xxx.tcl`

模擬結束後就會產生 `sd_udp` 和 `rd_udp` 兩個記錄檔。

## 捌、awk 語言

是一種程式語言，具有一般程式語言常見的功能。awk 語言具有某些特點，如：使用直譯器(Interpreter)不需先行編譯；變數無型別之分(Typeless)，可使用文字當陣列的註標(Associative Array)等，因此，使用 awk 撰寫程式比起使用其它語言更簡潔便利且節省時間。另外，awk 還具有一些內建功能，使得 awk 擅於處理具資料列(Record)、欄位(Field)型態的資料；最後，awk 內建有 pipe 的功能，可將處理中的資料傳送給外部的 Shell 命令加以處理，再將 Shell 命令處理後的資料傳回 awk 程式，這個特點也使得 awk 程式很容易使用系統資源。

## 玖、 gnuplot

做完數值分析後，最重要的就是要把數值的結果畫成圖呈現出來，在本實驗中，所採用的繪圖工具是 gnuplot。

Gnuplot 是一個命令導向的交談式繪圖程式(command-driven interactive function plotting program)。使用者輸入的每一項命令，可以逐步設定或修改繪圖環境。它以圖形表達數據或函數，使我們可以藉由圖形做更進一步的分析。

## 壹拾、 封包遺失率

對於封包遺失率(packet loss rate)，我們可以先計算發送端記錄檔中有多少筆記錄，每一筆記錄就代表一個送出封包的資訊，因此有多少筆記錄就代表有多少封包被送出，同理，我們去計算接收端記錄檔中有多少記錄，每一筆記錄代表每一個所接收封包的資訊，因此有多少筆記錄就代表有多少封包被接收，從這兩個數值的差值我們就可以得知會有多少的封包在傳送的過程被丟棄了，然後再把這個差值除以全部送出的封包量，就可以得到封包遺失率。

計算 CBR 的封包遺失率：從 sd\_udp 檔案中，可以得知共有幾筆封包傳送記錄；從 rd\_udp 檔案中，可以得知共多少筆封包接收記錄，相減後的差值即為封包遺失的數量，因此封包遺失率為封包遺失數/傳送封包數。

## 壹拾壹、 封包延遲

封包延遲 (packet delay) 即為封包到達時間與封包傳送時間的差值。對於封包延遲，則可以直接從接收端記錄檔的第四欄得到。

求得封包延遲時間執行方法:

```
$awk '{print $1, $4}' rd_udp>cbr_delay
```

使用 gnuplot 畫出 cbr\_delay

```
gnuplot> plot "cbr_delay" title 'cbr: packet delay' with linespoints 1
gnuplot> set xlabel 'packet sequence'
gnuplot> set ylabel 'delay time (sec)'
gnuplot> set terminal gif
gnuplot> set output "cbr_delay.gif"
gnuplot> replot
```

## 壹拾貳、抖動率

抖動率(jitter)就是延遲時間變化量(delay variance)，由於網路的流量隨時都在變化，當流量大的時候，許多封包就必需在節點的佇列中等待被傳送，因此每個封包從傳送端到目的地端的時間也就不一定會相同，而這個不同的差異就是所謂的抖動率。抖動率越大，則表示網路越不穩定。

對於抖動率(jitter)，則可以使用封包延遲時間差距除以封包序號差距得到 ( $\text{jitter} = ((\text{rcvtime}(j) - \text{sendtime}(j)) - (\text{rcvtime}(i) - \text{sendtime}(i))) / (j-i) = (\text{delay}(j) - \text{delay}(i)) / (j-i)$ ，其中  $j > i$ )。

計算抖動率：measure-jitter.awk

```
BEGIN{
    last_pkt_id = -1;
    last_e2e_delay = -1;
}
{
    pkt_id = $1;
    send_time = $2;
    rcv_time = $3;
    e2e_delay = $4;
    pkt_size = $5;

    if( last_pkt_id !=-1) {
        jitter = (e2e_delay - last_e2e_delay) / (pkt_id - last_pkt_id);
        printf("%f %f\n", send_time, jitter);
    }

    last_pkt_id = pkt_id;
    last_e2e_delay = e2e_delay;
}
{
}
```

執行方法:

```
$awk -f measure-jitter.awkrd_udp>cbr_jitter
```

使用 gnuplot 畫出 cbr\_jitter

```
gnuplot> plot "cbr_jitter" title 'cbr: packet jitter' with linespoints 1
```

```
gnuplot> set xlabel 'packet start time (sec)'
```

```
gnuplot> set ylabel 'jitter (sec)'
```

```
gnuplot> set terminal gif
```

```
gnuplot> set output "cbr_jitter.gif"
```

```
gnuplot> replot
```

### 壹拾參、吞吐率

吞吐率 (throughput) 為單位時間內，所能負荷的封包容量。對於吞吐率，則可以把所接收的封包大小總和除以所花費的時間就可以得到。

計算吞吐率:measure-throughput.pl



```
#使用方法: perl measure-throughput.pl <trace file> <granularity>
```

```
#記錄檔檔名
```

```
$infile=$ARGV[0];
```

```
#多少時間計算一次(單位為秒)
```

```
$granularity=$ARGV[1];
```

```
$sum=0;
```

```
$sum_total=0;
```

```
$clock=0;
```

```
$maxrate=0;
```

```
$init=0;
```

```
#打開記錄檔
```

```
open (DATA,"<$infile")
```

```
|| die "Can't open $infile $!";
```

```
#讀取記錄檔中的每行資料,資料是以空白分成眾多欄位
```

```
while (<DATA>) {
```

```
    @x = split(' ');
```

```
    if($init==0){
```

```
        $start=$x[2];
```

執行的方法:

```
$perl measure-throughput.pl rd_udp 1
```

執行的結果

Average rate: bps

## 第七節 NS2 網路模擬實驗結果

三種演算法解前面章節所敘述的五個問題之最佳解答，將透過 NS2 來進行網路的模擬實驗，並且比較以下的實驗數據，判斷網路的優劣，並回顧是否符合基因演算法的解答，從 GA、GCGA 與 2DGA 中找出適用於叢集式無線感測網路之上。我們更進一步探討兩種適應值函式，討論何者較適合使用於無線感測網路：

- (1) 封包遺失
- (2) 封包延遲
- (3) 抖動率
- (4) 吞吐量

### 壹、封包遺失

在本部分的實驗中，我們可以看到當適應值函式為 F1 時，封包遺失的數據與題目的難度增長並無顯著的相關性(表 4-5, 圖 4-18)，在實驗題目為 Q3 時，GCGA 與 2DGA 的封包遺失為五個實驗中的最高點；而 GA 的封包遺失最高點則是落在 Q4。

當適應值函式為 F2 時，我們就可看到三種演算法在五個實驗的難度增加之下，能從中看出題目難度與演化目標間之趨勢(圖 4-19)：GA 與 2DGGGA 的封包遺失率隨著題目難度增加而變多，而 GCGA 一直維持著較低的封包遺失率。

表4-5 封包遺失實驗數據

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	0	0	0	0	0	0
Q2	0	0	0	0.001024	0	0
Q3	0.001024	0.011521	0.004864	0.016385	0.010497	0.017921
Q4	0.016897	0.004608	0.004864	0.027555	0.019969	0.034306
Q5	0.000256	0.007168	0	0.03482	0.026626	0.040722

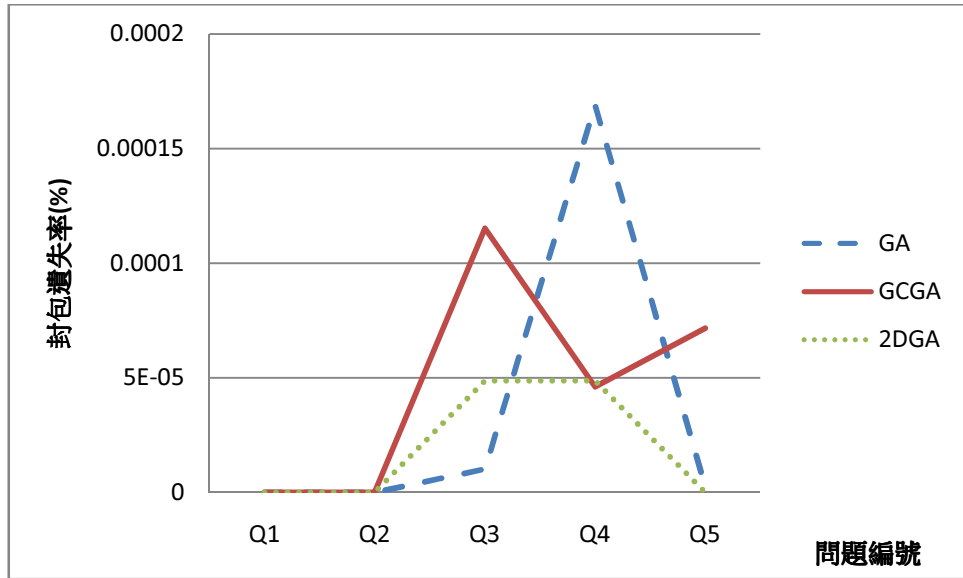


圖4-18 適應值函式 F1 所得最佳解之封包遺失實驗

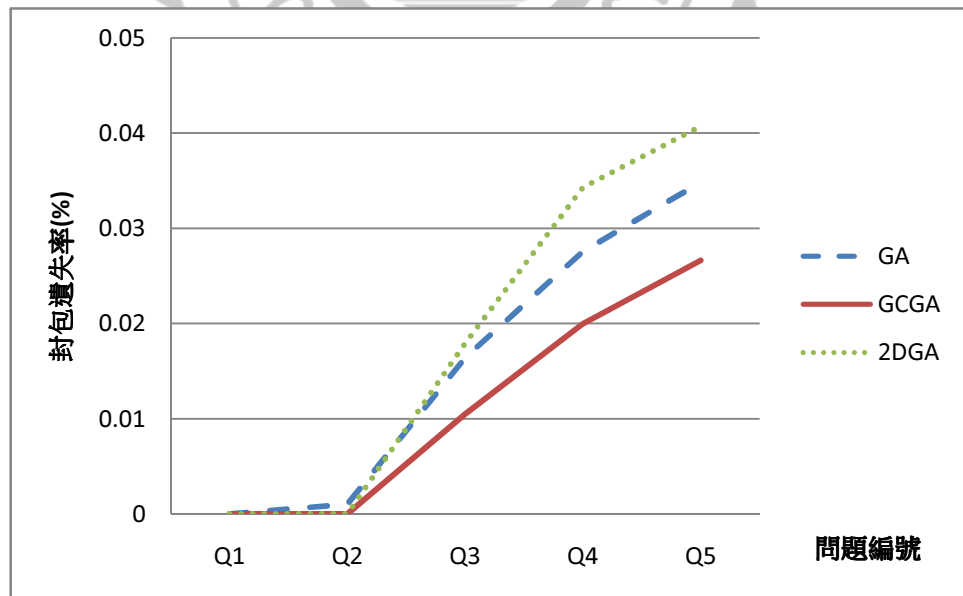


圖4-19 適應值函式 F2 所得最佳解之封包遺失實驗

## 貳、封包延遲

使用適應值函式 F1 時，三種演算法在題目 Q1 中因為都找到相同的最佳解答，因此有相同的封包延遲數據；當實驗題目為 Q2 時，僅 GA 有著較大的封包延遲數據，GCGA 與 2DGA 的則較低，可看出與 Q1 題目難度的區別；實驗題目 Q3 則是 2DGA 的平均延遲數據較低，GA 其次；當實驗到了 Q4，結果可看出 2DGA 的延遲較高，最差，GA 其次，而 GCGA 最佳；至實驗 Q5，2DGA 仍有最高的平均延遲，GA 其次，而 GCGA 仍最佳，從圖中我們較難看出題目規模與演算法解答之趨勢，可能是考慮的演化目標不同所導致。

使用適應值函式 F2 時，三種演算法在題目 Q1 中因為都找到相同的最佳解答，因此有相同的封包延遲數據；至 Q2 時，數據中可看出 GCGA 與 2DGA 相差無幾，而 GA 略好；到了實驗題目 Q3，結果則是 GCGA 與 2DGA 較近，而 GA 有較高的平均延遲則較佳；在實驗 Q4 中，則是 GCGA 比 2DGA 與 GA 更好一些；最後，在實驗題目 Q5 中，GA 的結果最差並有著三種演算法最高的延遲數據，2DGA 其次，仍有高延遲，GCGA 最好，有著較低的平均封包延遲實驗數據。此實驗中我們可以看到以 Q3 為分水嶺，較小的題目規模適合使用 GA，而當題目變得更加複雜時我們就需要使用考慮地理資訊之基因演算法求解。

在最大封包延遲的數據中，適應值函式 F1 所得的最佳解仍呈現較無趨勢的結果，相對地 F2 中則呈現以 Q3 實驗為基準，實驗題目較簡單，則最大延遲較低；實驗愈趨複雜，最大延遲則較高，題目越複雜，越適合使用 GCGA 解。

表4-6 平均封包延遲實驗數據

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	0.011458	0.011458	0.011458	0.011458	0.011458	0.011458
Q2	1.598729	0.017525	0.017515	1.416762	2.023732	1.923353
Q3	0.905238	2.259099	0.586997	0.763017	0.522139	0.275772
Q4	1.623447	0.931343	3.283821	1.63179	1.06622	1.244653
Q5	1.526414	1.160257	1.957723	1.91372	1.656529	1.837374

表4-7 最大封包延遲實驗數據

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	0.016055	0.016055	0.016055	0.016055	0.016055	0.016055
Q2	2.403056	0.052495	0.050075	2.031893	2.43582	2.410271
Q3	2.648783	3.674875	2.366475	2.004491	1.957064	2.041875
Q4	2.931397	1.991073	7.392177	4.044706	3.245746	3.511216
Q5	3.52888	3.509762	4.442126	4.392107	3.423838	4.091561

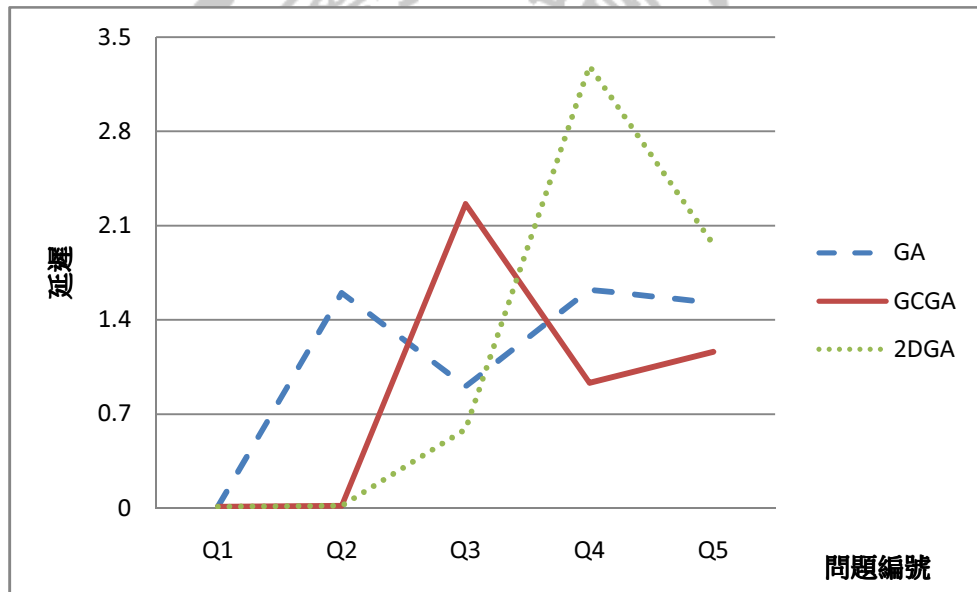


圖4-20 適應值函式 F1 之平均封包延遲

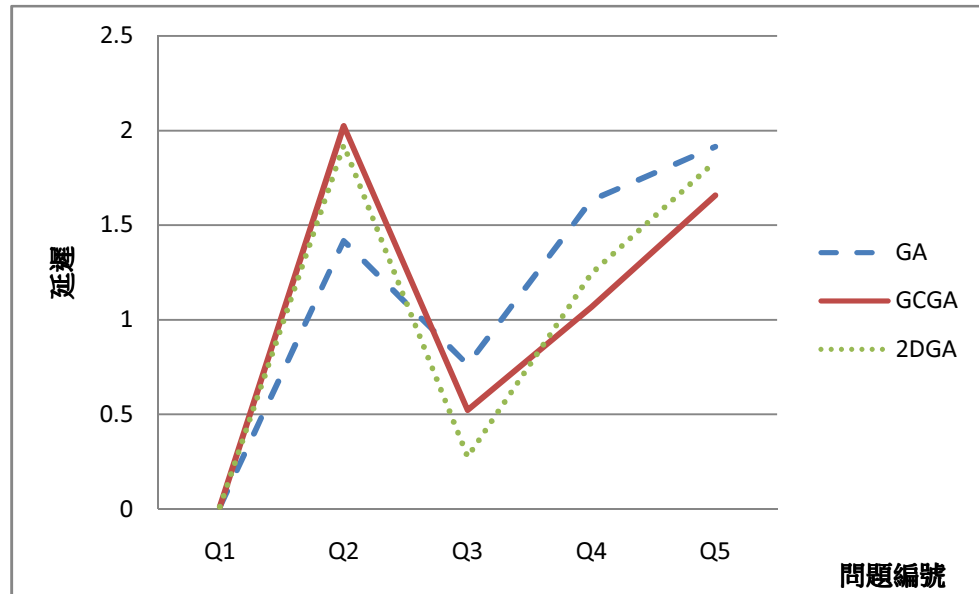


圖4-21 適應值函式 F2 之平均封包延遲

### 參、抖動率

在抖動率實驗數據的部分，當適應值函式為 F1 時，解題目 Q1 時三種演算法的抖動率較小，情況並不顯著；題目是 Q2 時，抖動率的數據也在可接受的範圍內；而在 Q3 題目中，2DGA 雖出現了小幅度的波動，但情況仍可接受；實驗題目 Q4 中，都動率也是可接受的，但可注意到的是 GA 的最低點與 2DGA 的最高點，似乎出現了較大的波動；最後的實驗題目 Q5，GCGA 可能因與適應值函式 F1 的搭配並不理想，使得峰值稍高外，並沒有嚴重的抖動情形發生。

當適應值函式為 F2 時，解題目 Q1 時三種演算法的抖動率較小，情況並不顯著；題目是 Q2 時，抖動率的數據顯示抖動率並不嚴重；當實驗題目為 Q3 時，三種演算法的結果顯示抖動率還在可接受的範圍內，但 GA 的抖動率明顯上升；而在實驗題目 Q4 中，GA 出現較高的峰值；在最後的實驗題目 Q5 中，除了 2DGA 有較低的低點之外，整體而言還是可接受的抖動率數據。

表4-8 平均抖動率實驗數據

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	-3E-06	-3E-06	-3E-06	0.016055	0.016055	0.016055
Q2	0.002459	-6E-06	-6E-06	2.031893	2.43582	2.410271
Q3	-0.00499	0.009161	-0.00152	2.004491	1.957064	2.041875
Q4	0.006324	0.003004	-0.00477	4.044706	3.245746	3.511216
Q5	0.000368	0.002441	0.010145	4.392107	3.423838	4.091561

表4-9 最大抖動率實驗數據

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
Q2	0.105001	0.006638	0.004597	0.1425	0.111108	0.143244
Q3	0.380917	0.452603	0.255944	0.336914	0.223734	0.232899
Q4	0.309619	0.419371	0.460603	0.584256	0.375886	0.418493
Q5	0.313737	0.690835	0.310284	0.726567	0.410028	0.563392

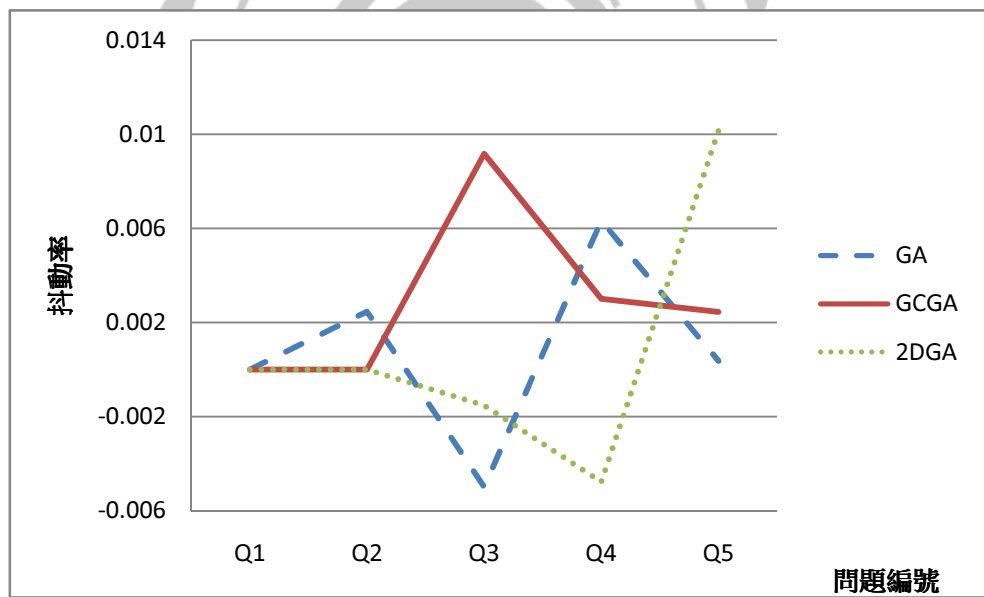


圖4-22 適應值函式 F1 之平均抖動率

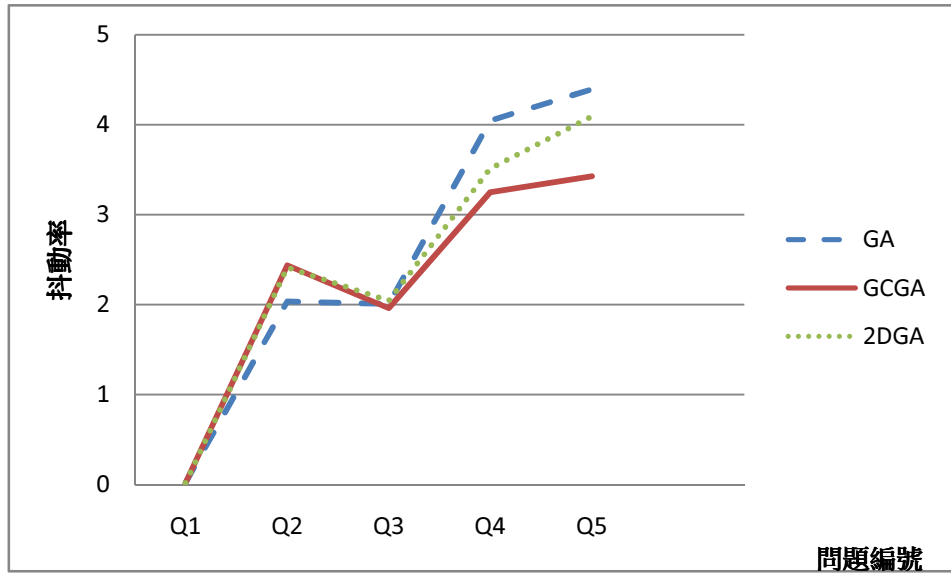


圖4-23 適應值函式 F2 之平均抖動率

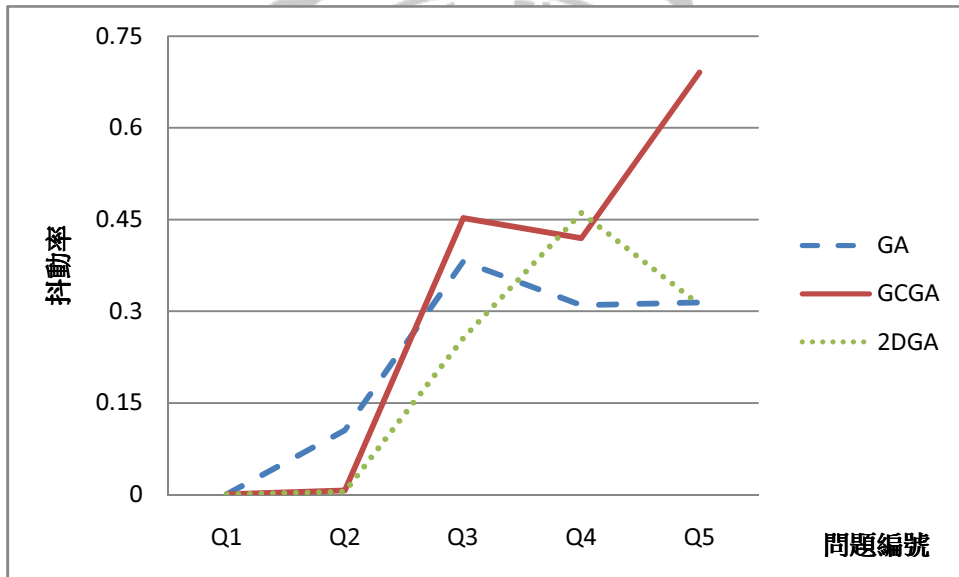


圖4-24 適應值函式 F1 之最大抖動率



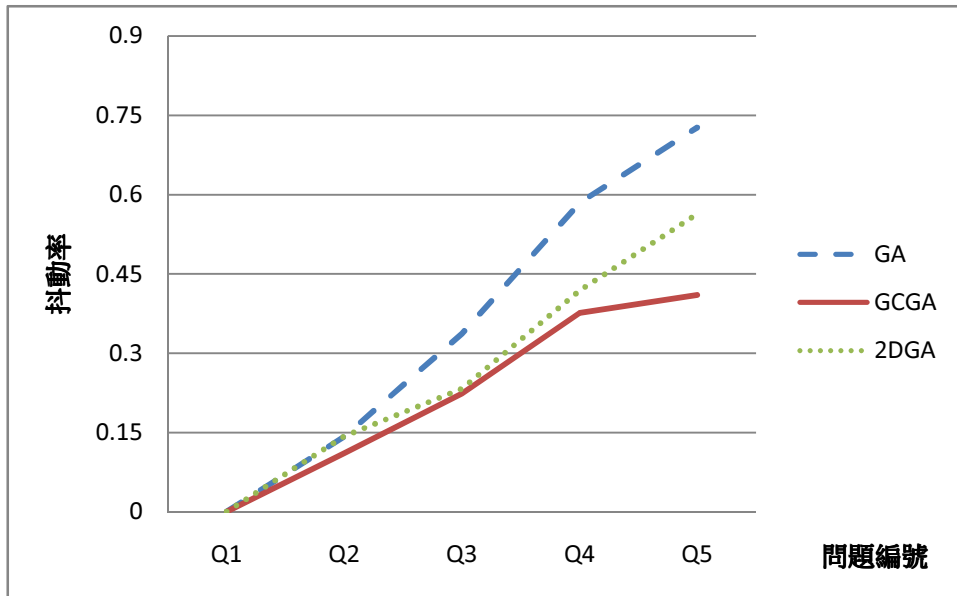


圖4-25 適應值函式 F2 之最大抖動率

## 肆、 吞吐量

在吞吐量的實驗中，我們同樣比較的適應值函式 F1 與 F2 的實驗數據，詳細的數據如表 7 所示。在圖 5 中，適應值函式 F1 的三種演算法執行結果，在 Average rate 的部分大致上為 2DGA 最佳，而 GCGA 次之，GA 最不理想。對照到 Peak rate 的數據，結果仍是 2DGA 最佳，GCGA 次之，而 GA 最差。

當適應值函式為 F2 時，我們可看出相較於 F1，三種演算法的 Average rate 趨勢較為明顯，而結果明顯地表示 GCGA 勝過 2DGA，又勝過 GA 的情況，但在實驗題目 Q4 時可能因為題目的節點分佈，使得在本實驗結果是 GA 與 2DGA 好過 GCGA。Peak rate 的部分則是 GCGA 較高，2DGA 次之，GA 最低。我們推論，因適應值函式 F2 的影響之下，GCGA 在這裡得到的結果呈現三種演算法中最佳的結果。

由此可知將地理位置資訊納入編碼的確可幫助叢集式無感測感測網路找到優良的網路架構，而適應值函式若設計適當則會讓演算法的整體目標朝著較適當的方向前進，在此節的實驗中，幾乎所有實驗若使用適應值函式 F2，

相較於 F1 都可得到擁有整體趨勢的實驗數據，我們推論，若適應值函式設計得當，所得的優良解答反映到實際的網路上才是好的解答。因此若演算法 GCGA 使用適應值函式 F2，應用於叢集式無線感測網路，結果將最理想。

表4-10 Average rate 比較

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	207898.7	207898.7	207898.7	207898.7	207898.7	207898.7
Q2	173757	207929.3	207929.6	171947.6	175655.9	173616.6
Q3	106776.2	78648.73	124268	79737.4	102343.6	68066.24
Q4	133242.7	61155.8	57039.21	64710.02	72736.8	56336.24
Q5	69266.1	58909.12	82819.24	55347.35	69429.98	64917.56

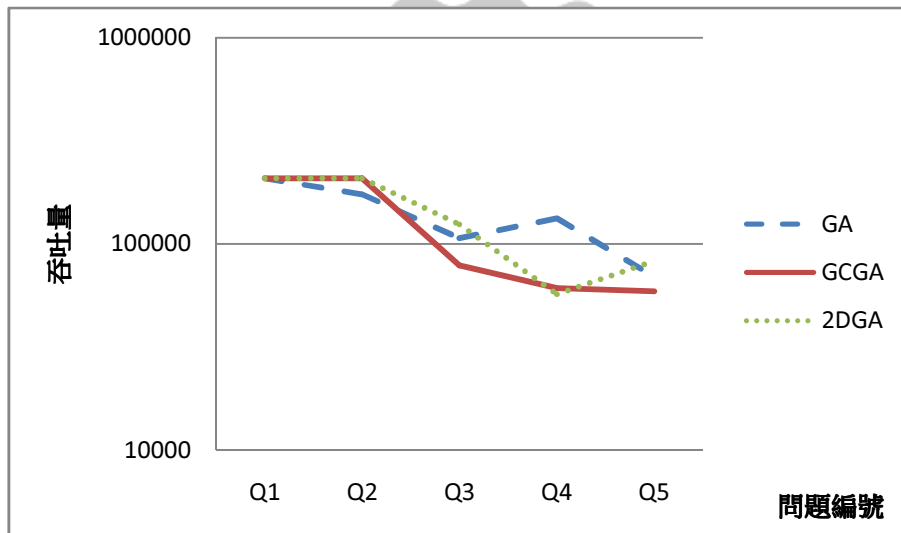


圖4-26 F1-Average rate 之比較

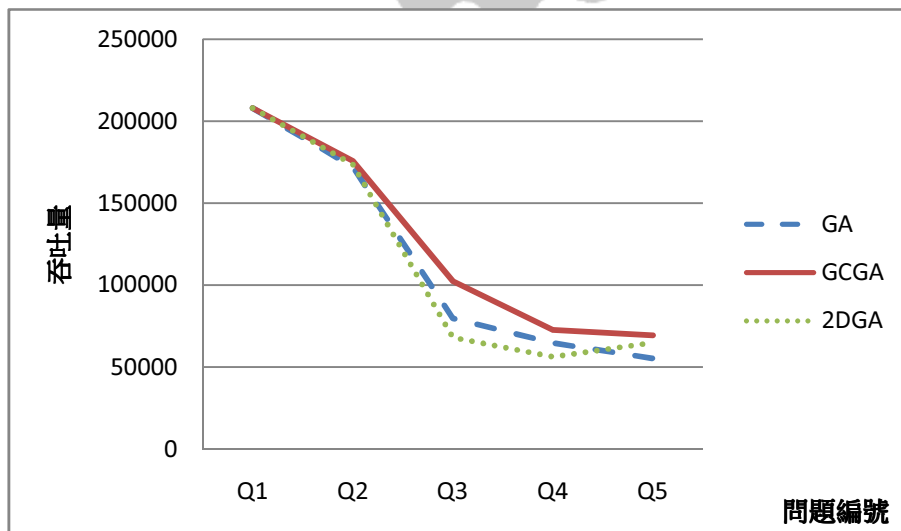


圖4-27 F2-Average rate 之比較

表4-11 三種演算法 Peak rate 比較

	GA-F1	GCGA-F1	2DGA-F1	GA-F2	GCGA-F2	2DGA-F2
Q1	212800	212800	212800	212800	212800	212800
Q2	238336	212800	212800	229824	238336	238336
Q3	314944	289408	187264	144704	161728	144704
Q4	204288	144704	238336	204288	161728	136192
Q5	170240	153216	153216	119168	136192	153216

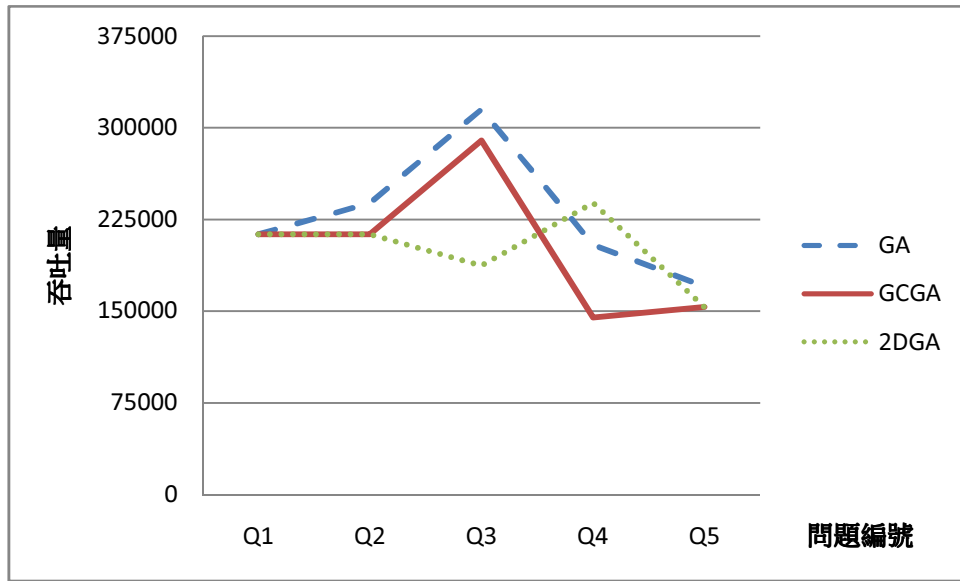


圖4-28 F1- Peak rate 比較

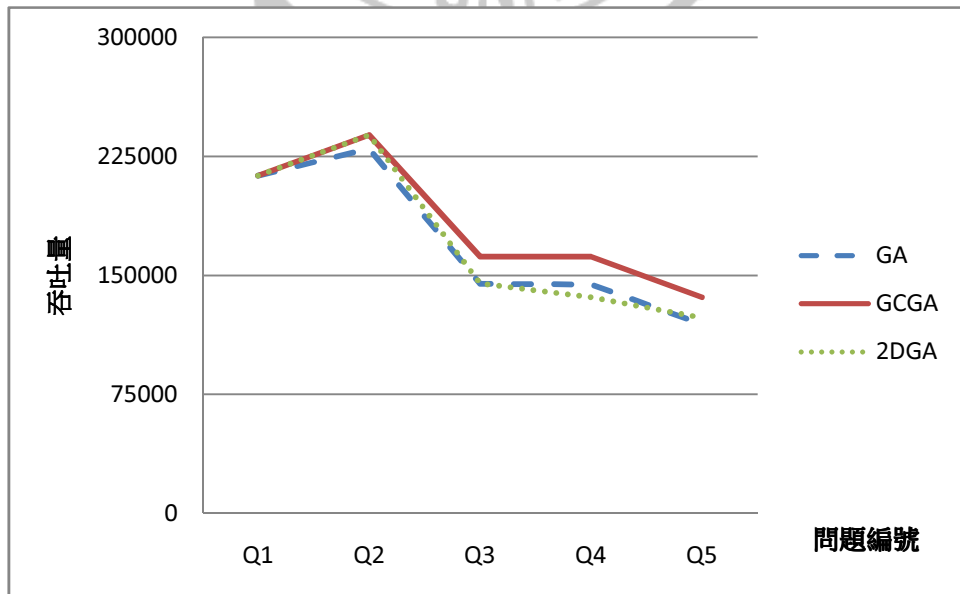


圖4-29 F2 -Peak rate 比較

## 第五章 結論

在基因演算法的求解進度比較時,我們提出的 GCGA 之求解品質能勝過傳統的基因演算法,同時比二維編碼的基因演算法更快速獲得解答,藉此證實將地理位置資訊納入染色體編碼,且對運算子做適當調整的 GA 是能對整體無線感測網路達到更佳的能源節省效果,期許能使用在實際的案例當中,並達成以下的目的:

- (1) 因無線感測網路的問題是個 NP 問題,所以採用基因演算法做為解題的工具;設計一個考慮地理位置的染色體編碼;
- (2) 依照問題特性,對選擇、交配與突變等運算子做出正確的改善;
- (3) 於基因演算法的進程中,對下一個時間區段的能源存量進行模擬,並且依照模擬的結果進行各種決策。

我們將三種基因演算法的實驗結果用於 NS2 實驗模擬時,不同的適應值函式的確產生出不通演化目標的叢集無線感測網路架構,我們雖然只考慮了兩種適應值函式,但可從中看出考慮地理位置資訊的編碼之外,仍要將適應值函式中的地理位置資訊做適當的考量,使得演化目標更為明顯。

我們在本研究中證明了提出的演算法—地理資訊之基因演算法,在演算法的求解進程、耗時、於模擬耗能實驗與 NS2 網路模擬實驗之結果皆令人滿意,是因為我們付出了比原本的基因演算法稍多的時間與較多的編碼長度,得到了比原本的基因演算法與二維染色體編碼之基因演算法更好的結果。希望未來能進一步設計更能計算出能源消耗的公式,與更能明確定義出無線感測網路之演化目標的適應值函數,甚至運用於實作的部分。在實驗的過程中,我們也察覺到空洞問題的存在,因此若能探討空洞的探索與迴避,必定能找出更優良的無線感測網路設定,做出更多貢獻。

## 参考文献

- [1] Di Wu, Gang Zhu, Dongmei Zhao, Lina Liu, "Energy balancing in an OFDM-based WSN", 2011.
- [2] Evolutionary Genetic Algorithm for Efficient Clustering of Wireless Sensor Networks, Hyun-Sik Seo, Se-Jin Oh and Chae-Woo Lee, 2009.
- [3] G. Hossein Ekbatani Fard, Reza Monsefi, Mohammad-R. Akbarzadeh-T. Mohammad H. Yaghmaee, "A Multi-objective Genetic Algorithm based Approach for Energy Efficient QoS-Routing in Two-tiered Wireless Sensor Networks", 5th International Symposium on Wireless Pervasive Computing (ISWPC), 2010.
- [4] H. Wright, M. D. Vose, and J. E. Rowe, "Implicit Parallelism, Proc. of the Genetic and Evolutionary Computation Conference", 2003, pp 1505-1517.
- [5] Hongbo Jiang, Shudong Jin, Chonggang Wang, "Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks", JUNE 2011.
- [6] Hsueh-Wen Tseng, Shiann-Tsong Sheu, Member, IEEE, and Yun-Yen Shih, "Rotational Listening Strategy for IEEE 802.15.4 Wireless Body Networks", 2010.
- [7] Ilhem Boussaïd, Amitava Chatterjee, "Hybridizing Biogeography-Based Optimization With Differential Evolution for Optimal Power Allocation in Wireless Sensor Networks", 2011.
- [8] J. Abouei, K.N. Plataniotis, S. Pasupathy, "Green modulations in energy-constrained wireless sensor networks", 2010.
- [9] Jalel Ben-Othman, Karim Bessaoud, Alain Bui, Laurence Pilard, "Self-stabilizing algorithm for energy saving in Wireless Sensor Networks",

2011.

- [10] JamshidShanbehzadeh, SaeedMehrzoo, AbdolhosseinSarrafzadeh, “An Intelligent Energy Efficient Clustering in Wireless Sensor Networks”, 2011.
- [11] KamilStaniec, GrzegorzDebita, ”2011 Studies on the Planarity, Reliability and Energetic Efficiency in Wireless Sensor Networks”, 2011.
- [12] Lu Xiang qing ,Tian Yan, “An Efficient Heuristic Control Algorithm for Wireless Sensor Networks”, International Conference on Educational and Network Technology (ICENT), 2010.
- [13] Mahdi Zareei1, RahmatBudiarto, Tat-Chee Wan, “Study of Mobility Effect on Energy Efficiency in Medium Access Control Protocols”, 2011.
- [14] Omar Asad, MelikeErol-Kantarci, Hussein Mouftah, “Sensor Network Web Services for Demand-Side Energy Management Applications in the Smart Grid”, 2011.
- [15] OnurSoysal, Sami Ayyorgun, Murat Demirbas, “PowerNap: An Energy Efficient MAC Layer for Random Routing in Wireless Sensor Networks”, 2011.
- [16] PuttipongMahasukhon, Hamid Sharif, Michael Hempel, Ting Zhou, Tao Ma, and PradhumnaLalShrestha, ”A STUDY ON ENERGY EFFICIENT MULTI-TIER MULTI-HOP WIRELESS SENSOR NETWORKS FOR FREIGHT-TRAIN MONITORING”, 2011.
- [17] Qinqin Tang, Hongjun Bi, Zhenjiang Zhang, Yun Liu, ”POTENTIAL BASED VIRTUAL NETWORK FOR ENERGY BALANCE IN WSNS”, 2010.
- [18] Rajeswari, P.T.Kalaivaani, “Energy Efficient Routing Protocol for Wireless Sensor Networks Using Spatial Correlation Based Medium Access Control Protocol Compared with IEEE 802.11”, 2011.

- [19] S. Bandyopadhyay, E.J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", 2003,
- [20] Saeed Mehrjoo, Hassan Aghaee, Hossein Karimi, "A Novel Hybrid GA-ABC based Energy Efficient Clustering in Wireless Sensor Network", Canadian Journal on Multimedia and Wireless Networks Vol. 2, No. 2, April 2011.
- [21] Stephen Gundry, Elkin Urrea, Cem S. Afak S. ahin, "Formal Convergence Analysis for Bio-inspired Topology Control in MANETs", 2011.
- [22] V. Mhatre and C. Rosenberg, "Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation", [J] Adhoc Networks Journal, Elsevier Science, 2004, 2(1):45-63.
- [23] V. Mhatre, C. Rosenberg, R. Mazurandar and N. Shroff, "A Minimum Cost heterogeneous sensor network with a lifetime constraint" IEEE Transactions on Mobile Computing, Vol. 4, No. 1, pp. 4-15, January 2005.
- [24] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks" in Proceeding of the 33rd Annual Hawaii International Conference on System Sciences, 2000, pp.3005-3014.
- [25] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An Applicationspecific Protocol Architecture for Wireless Microsensor Networks" IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670, 2002.
- [26] Wen-ya Zhang, Zi-ze Liang, Zeng-guang Hou and Min Tan, "A Power Efficient Routing Protocol for Wireless Sensor Network", 2007.
- [27] Xin Fei, Samer Samarah, Azzedine Boukerche, "A Bio-Inspired Coverage-Aware Scheduling Scheme for Wireless Sensor Networks", 2010

IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010.

[28] XU Jiu-qiang, WANG Hong-chuan, LANG Feng-gao, WANG Ping, HOU Zhen-peng, "Study on WSN Topology Division and Lifetime". 2011.

[29] Y. Chang, Z. Lin and J. Chen, "Cluster based self-organization management protocols for wireless sensor networks" IEEE Transactions on Consumer Electronics, Vol. 52, No. 1, pp. 75-80, February 2006.

[30] Zeynep Orman<sup>1</sup>, Ali Norouzi, Faezeh Sadat Babamir, "Intelligent Node Placement using GA (INPGA) Protocol in Wireless Sensor Networks", 2011.



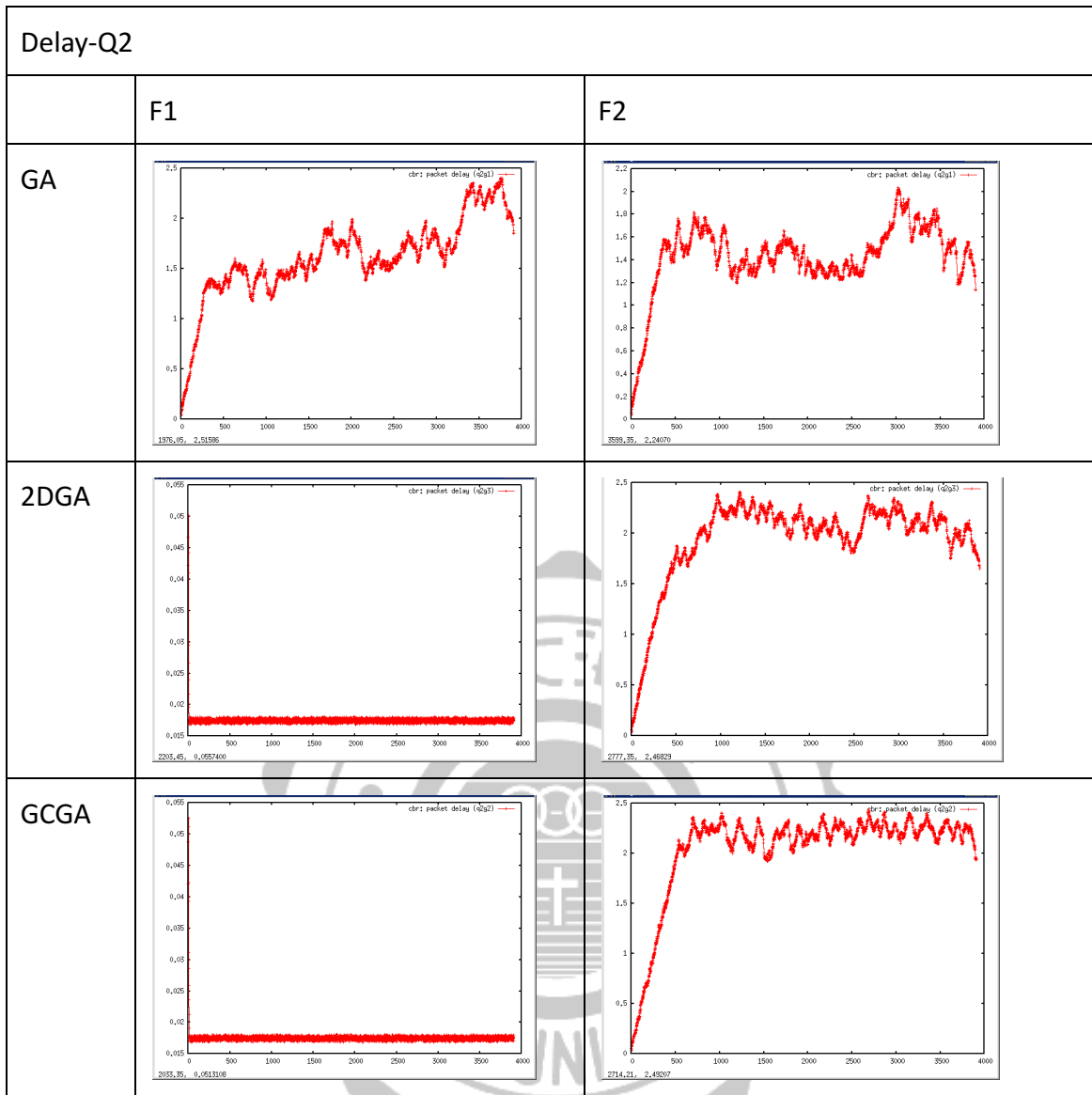


# 附錄

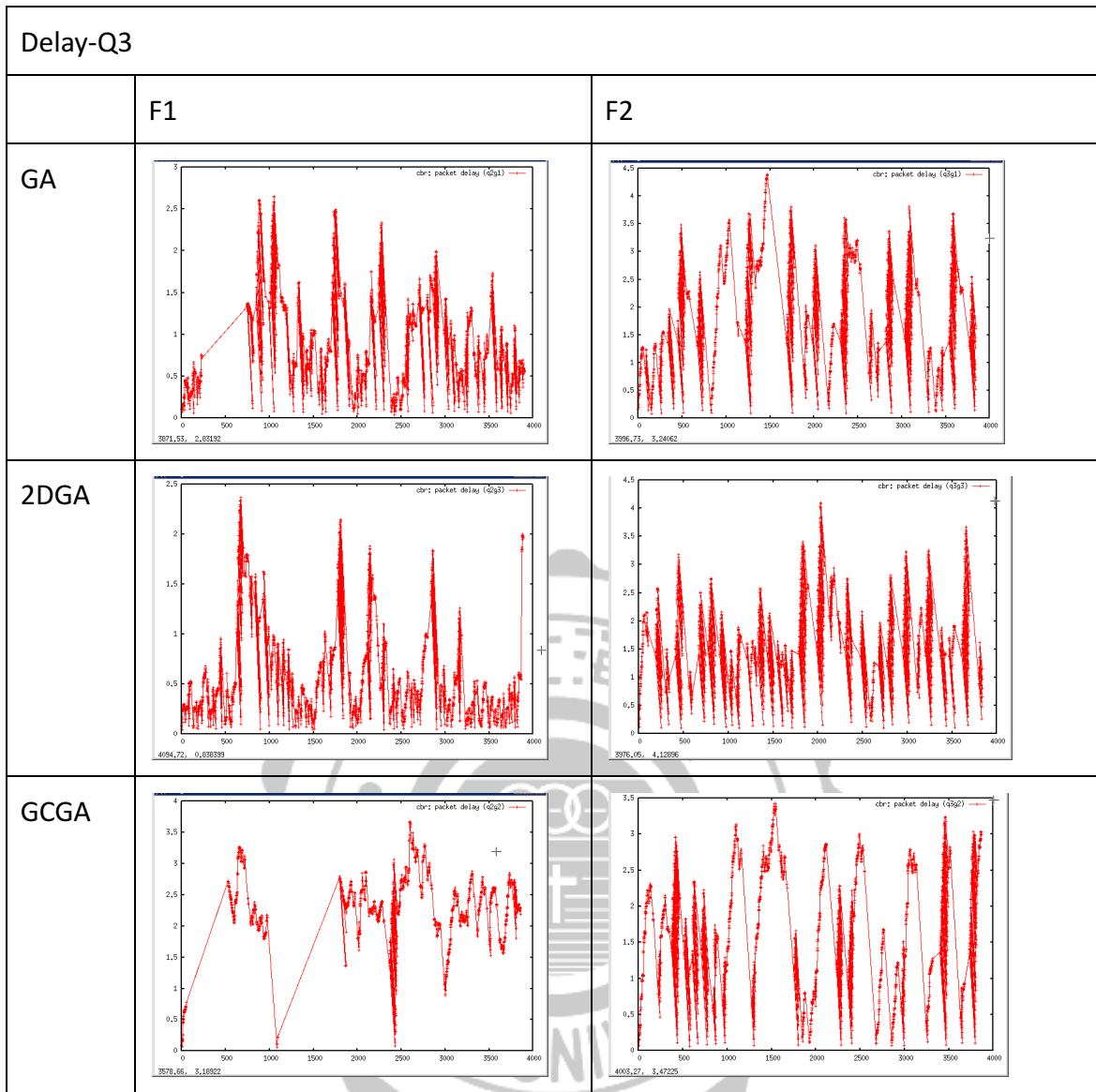
Delay-Q1		
	F1	F2
GA		
2DGA		
GCGA		

三種演算法在題目 Q1 的封包延遲都獲得相同的結果，顯示在題目規模

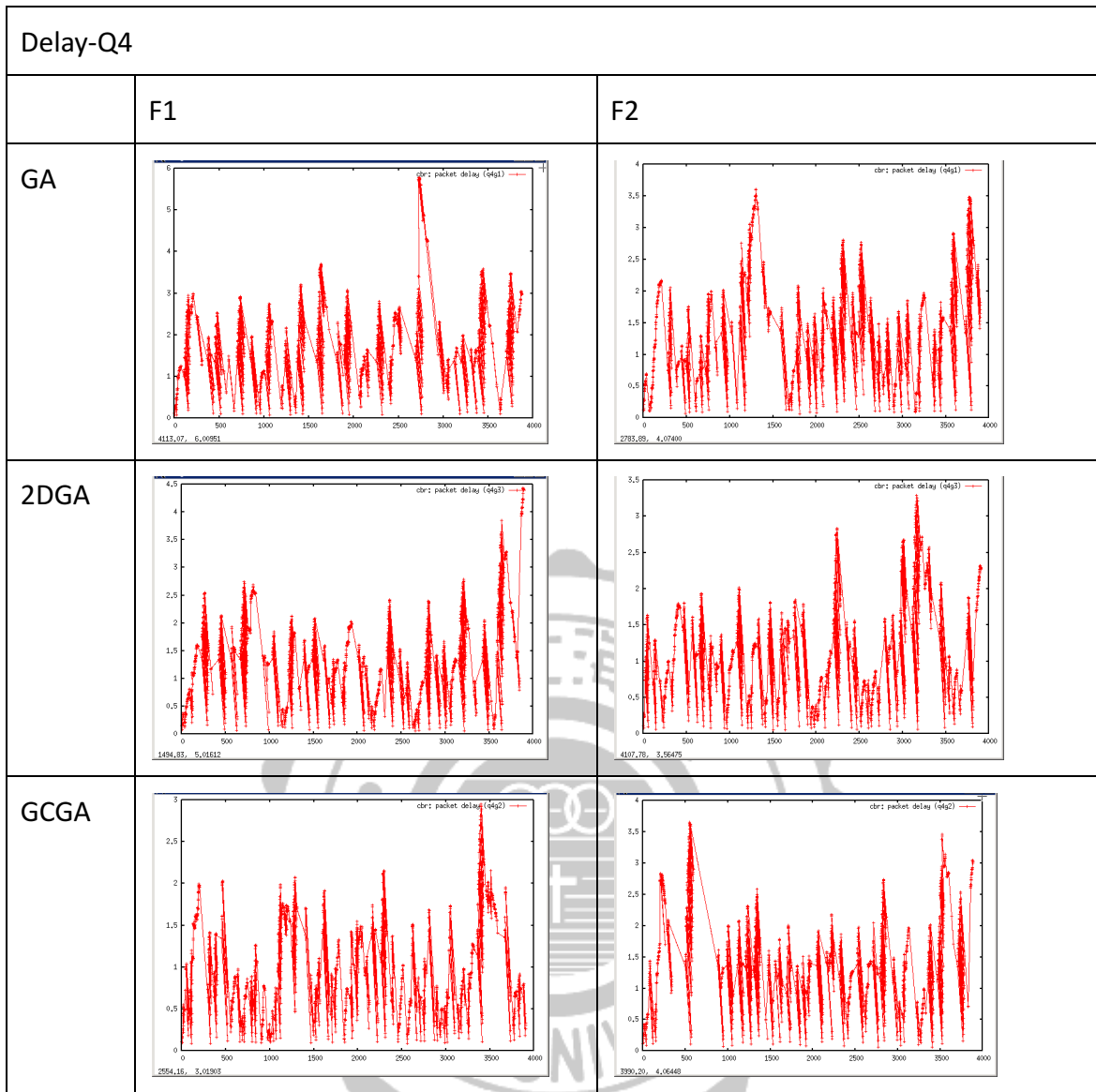
最小時，演算法皆能找到最佳解答。



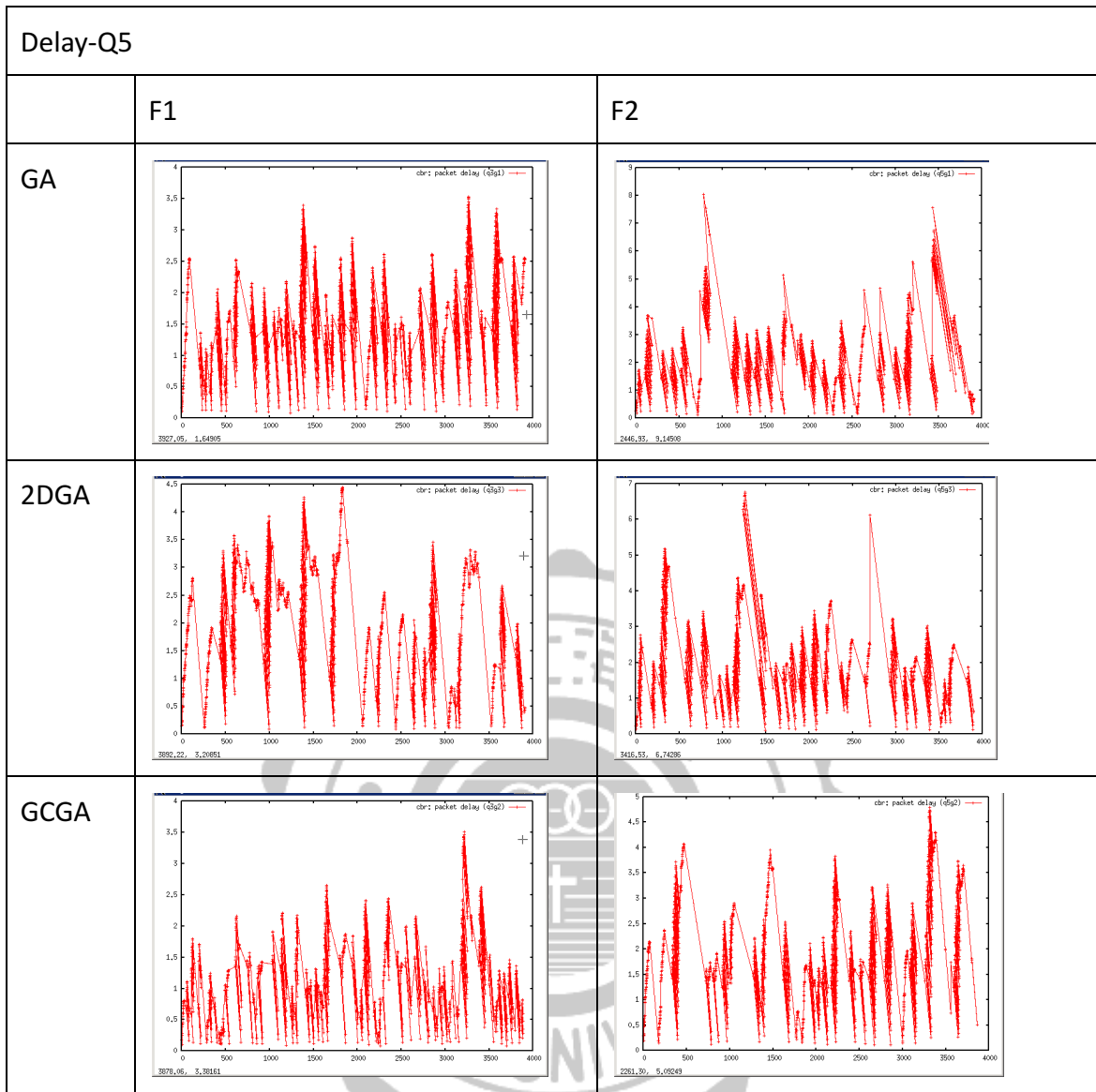
GA 擁有較大的封包延遲數據，而 2DGA 與 GCGA 的結果相差無幾。



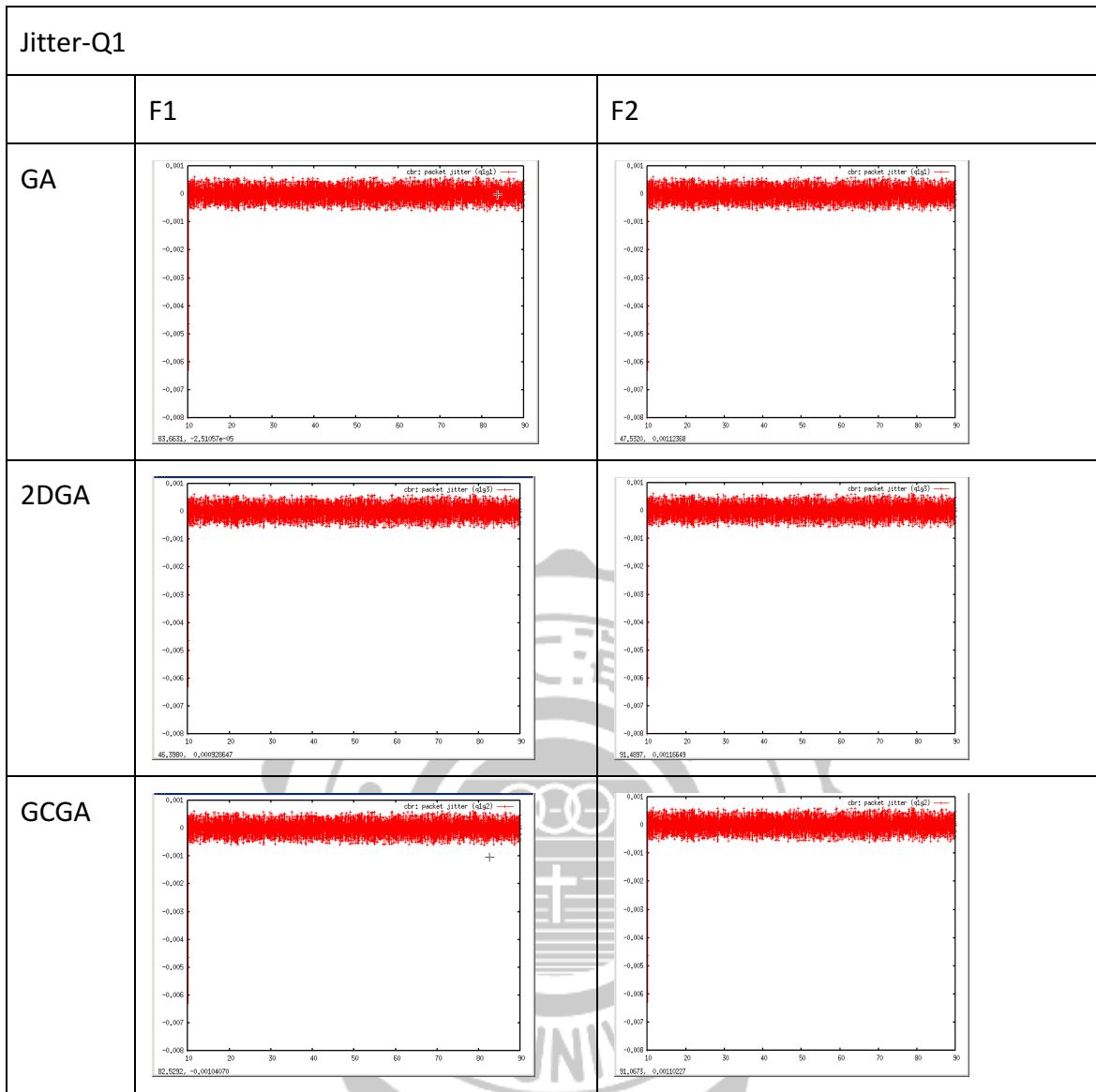
2DGA 的平均延遲較低，與 GCGA 較為接近，至此 GA 較差，是本部分實驗的分水嶺。



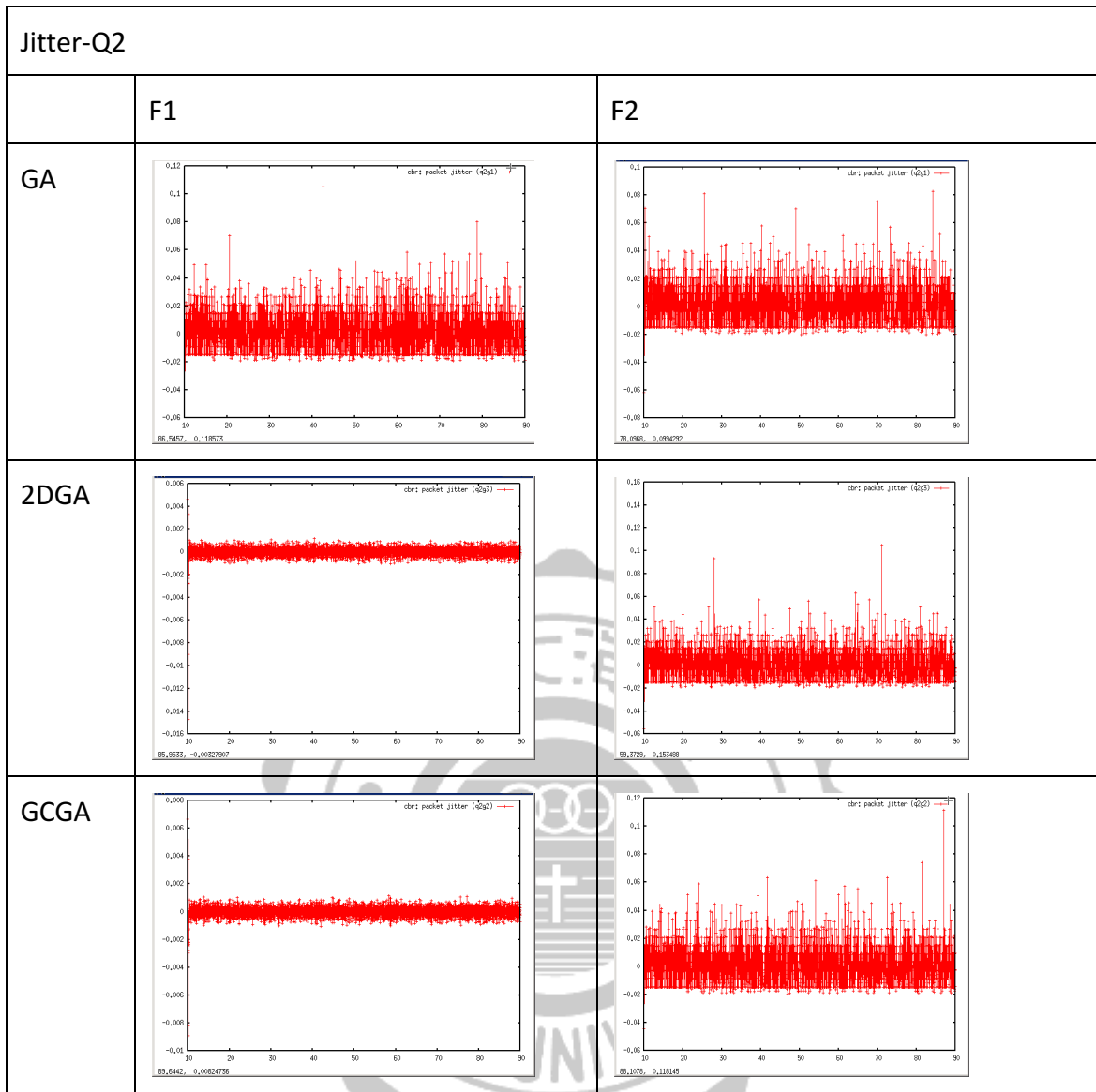
2DGA 的延遲較高，GA 其次，GCGA 延遲最低，GCGA 的結果最好。



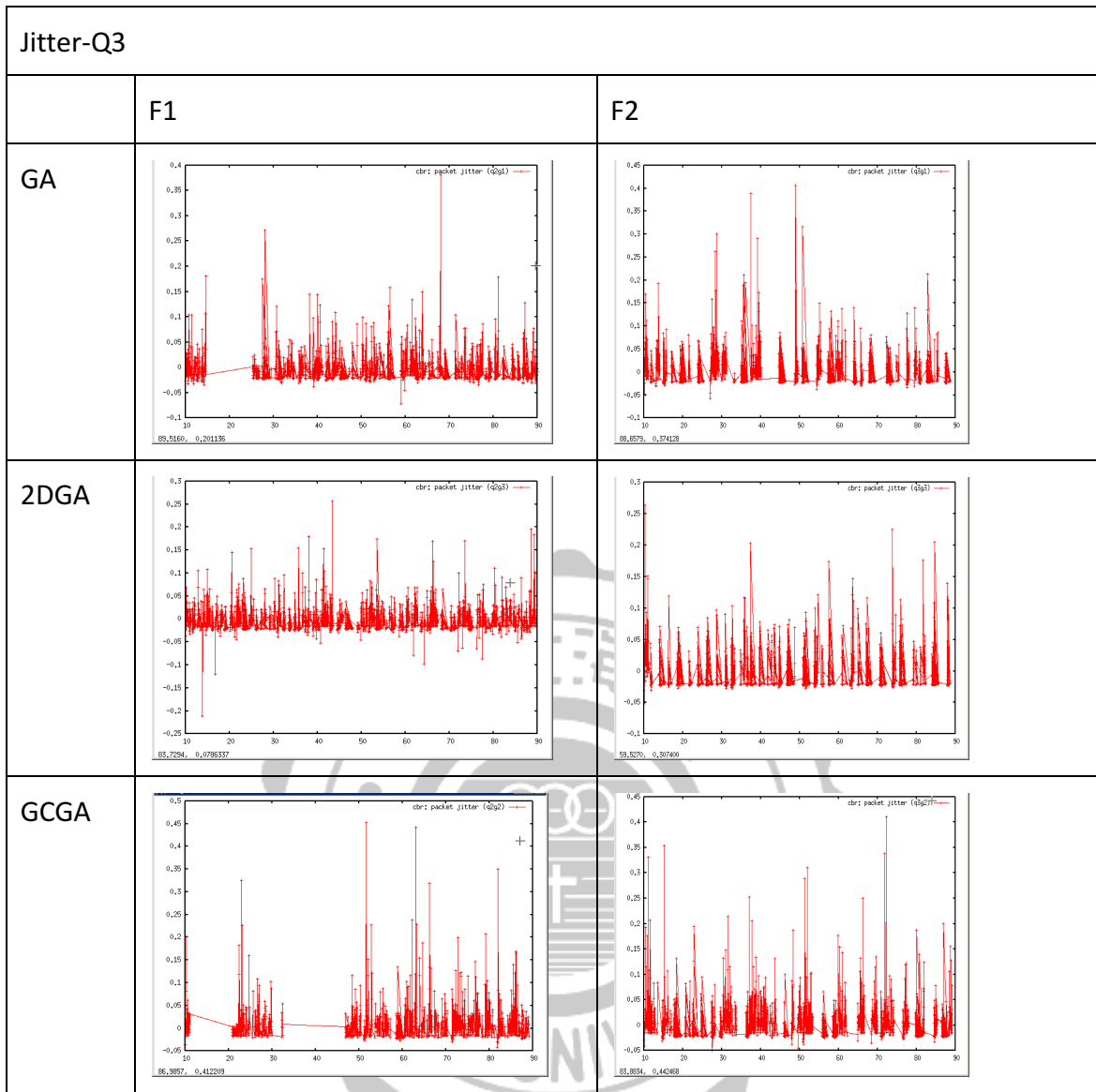
2DGA 有較高的延遲，GA 其次，GCGA 有最低延遲，為最佳的結果。



三種演算法的抖動率都相同，是接受的抖動率數據。

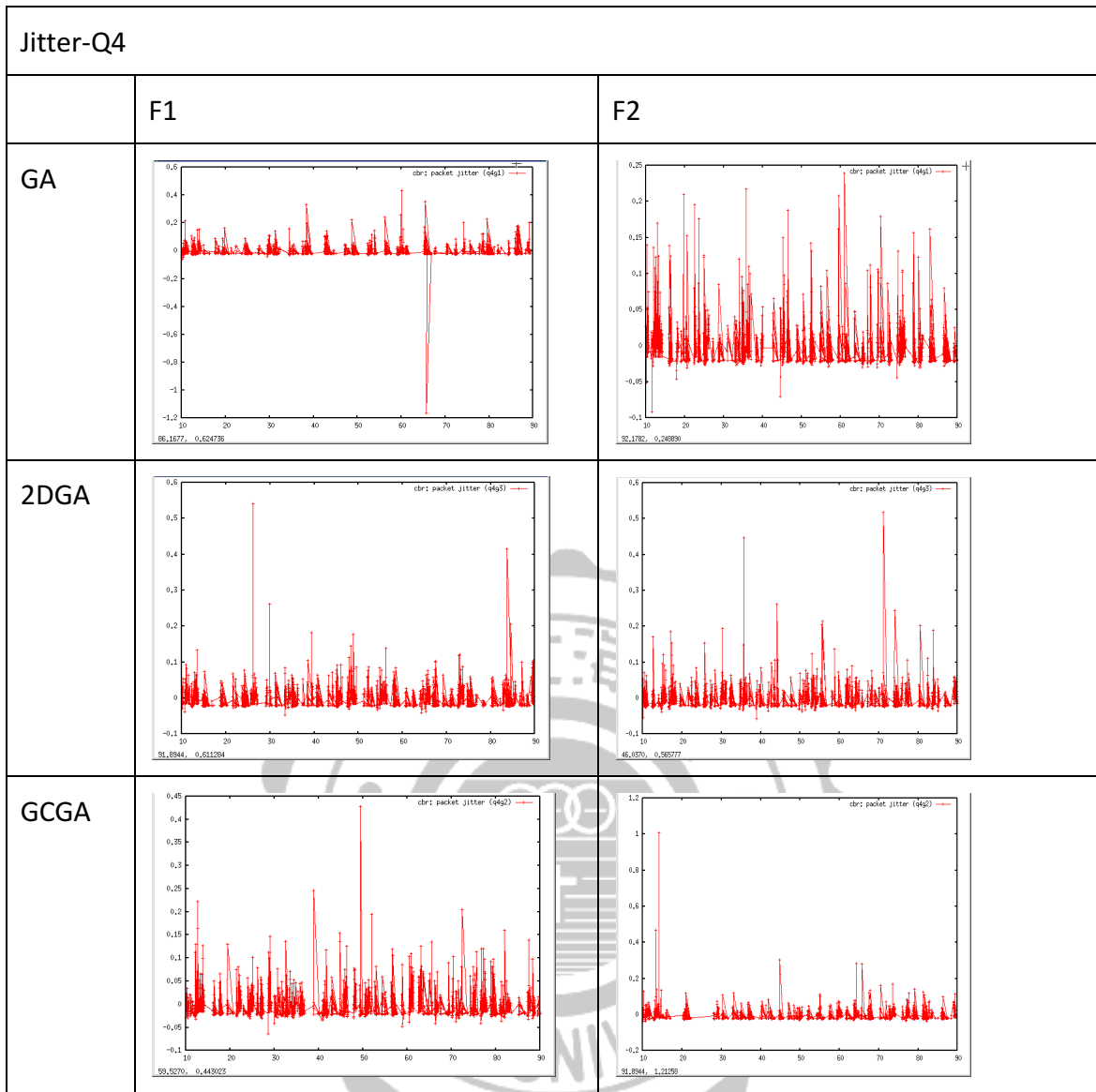


相較於 Q1 抖動率有上升趨勢，小規模題目時 2DGA 與 GA 略佳。

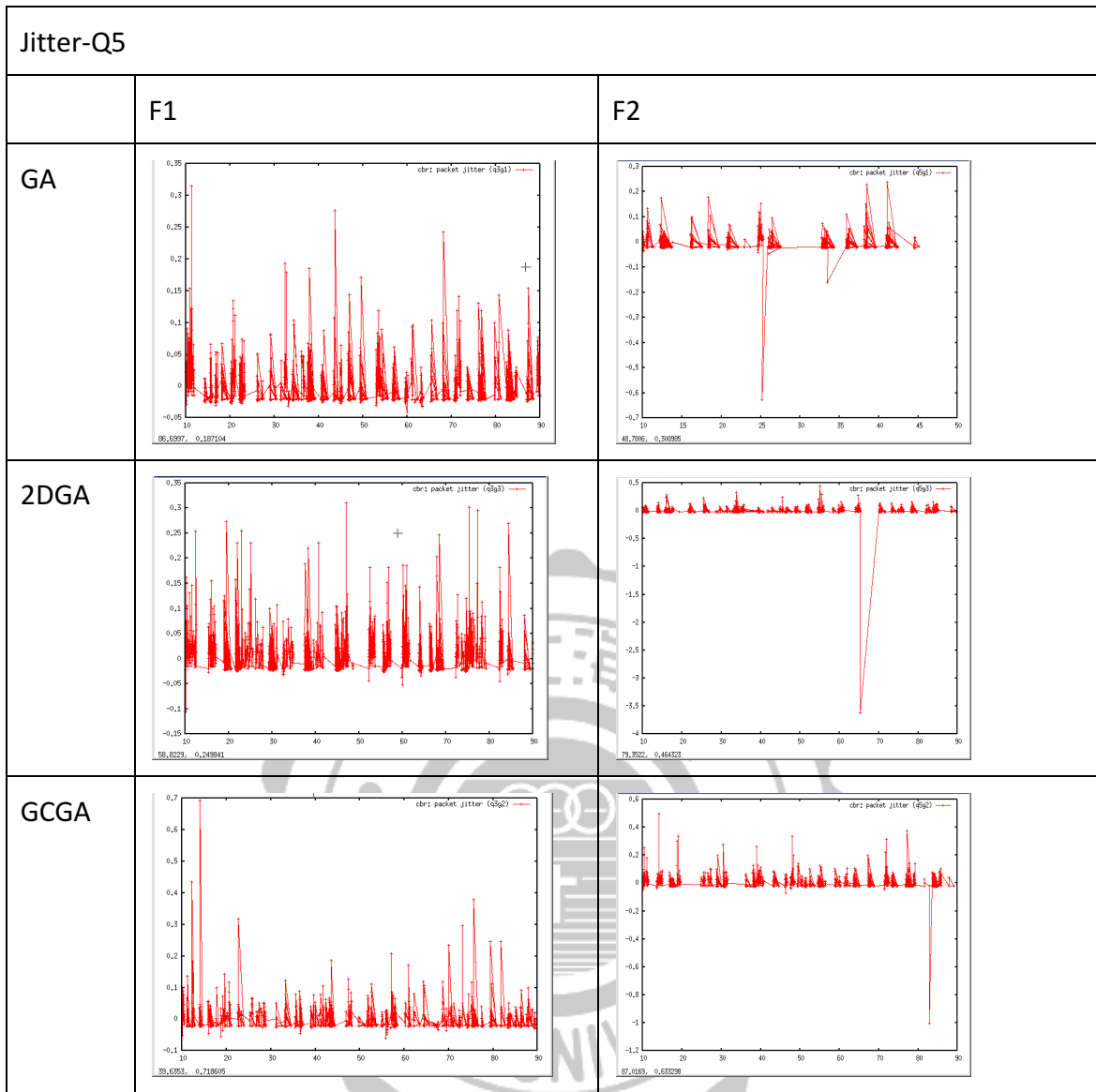


因題目規模增加，至此 GCGA 名次攀升為抖動率最低，為最佳。





隨題目規模增加，GA 與 2DGA 之抖動率數據顯助攀升。



至此題目，可看出抖動率為 GCGA 最佳，2DGA 其次，GA 最差。