

應急蜂巢式行動網路建構排程

Scheduling of Contingency Cellular Network Deployment

研究生：王彥嵩

Student : Yan-Sung Wang

指導教授：連耀南

Advisor : Yao-Nan Lien

國立政治大學

資訊科學系

碩士論文

A Thesis

submitted to Department of Computer Science

National Chengchi University

in partial fulfillment of the Requirements

for the degree of

Master

in

Computer Science

中華民國一百年十一月

November 2011

## 摘要

大型自然災害會癱瘓通訊系統嚴重影響到救災效率，本論文旨在快速提出一個建構排程供應急通訊系統佈建。無線通訊系統的成熟極大的為使用者帶來便利性，但當發生大規模的地震或強烈颱風等重大天然災害時，通訊系統卻常常因架構原因，隨著電力與交通系統的損毀而癱瘓。由歷年大型災變中多數災區內之行動通訊系統全面中斷即可印證行動通訊系統其實是極為脆弱，而有效運作的通訊系統卻是災情傳遞、資源調度以及互助協調是否順利的關鍵因素。

本篇論文所探討的應急通訊系統是利用倖存的連通基地台和斷訊卻沒有損毀的基地台建構一個臨時性的網路，稱為應急蜂巢式行動網路(contingency cellular network, CCN)。由於災區的交通系統可能癱瘓，因此 CCN 的建構需視各種運輸能力而規劃，而各個地方受災情況不盡相同，CCN 的建構順序也須辨明輕重緩急依序建構，網路拓樸的規劃是本研究團隊的另一研究主題，本文主在探討如何在網路拓樸已知情況下進行 CCN 建構排程以達到最大的救災效益，因此我們提出一適合 CCN 樹狀結構的最佳化排程模型，以追求救災效益的最大化，這些模型可供使用者(救災指揮單位)系統化的解決 CCN 建構排程問題。

模型包含 CCN 樹狀拓樸、基地台數目、基地台建構時間、基地台重要度、拓樸連線集合和建構工作組數。在此模型下提出一個考慮各基地台的時效性以及重要性而進行快速排程的演算法，此演算法透過計算排程總救災效益決定優劣。分三階段實驗。三階段實驗皆可在數秒內得出接近最佳解的排程。

## Abstract

When stricken by a large-scale disaster, the efficiency of disaster response operation is very critical to life saving. We propose to build a contingency cellular network to support emergency communication in large scale natural disasters by connecting disconnected base stations. This paper addresses the deployment scheduling problem. The advance of mobile communication technologies has brought great convenience to users. Cellular phone becomes the first communication tool most people would use in emergency. However, cellular networks were usually crashed due to earthquake, typhoons or other natural disasters due to power outage or backhaul broken. Unfortunately, the efficiency of communication system is a critical factor to the success of disaster response operation such as resource allocation as well as coordination of rescue and relief operations. We designed a contingency cellular network (CCN) by connecting physically intact but service-disrupted base stations together with wireless links. As the disaster area's transport system may be paralyzed, the construction of CCN may have to rely on air transportation such as helicopter or even airdrop. Since the transportation capacity may be very limited, scheduling of CCN deployment order according to the demand of disaster operation becomes an important issue. We model the CCN Deployment Scheduling Problem into a combinatorics optimization problem aiming to maximize disaster operation efficiency. The problem is proven NP Hard. Thus, we design an efficient heuristic algorithm to solve the problem when it is needed in urgent.

## 誌謝辭

在考研究所時沒想過會上，上了之後沒想過會來讀，甚至讀了之後還是覺得格格不入，但在要離校前這一刻，我由衷的慶幸最後選擇了政大資科，這也許是人生的最重大轉折之一，在這裡找到了一位好老師，在老師的課堂上不只學到專業知識，更能聽到老師對於研究方法的獨特見解及堅持，沒有這些鞭策我想即使畢業了也還是懵懵懂懂，少了很多重要的思考。研究過程中不免磕磕撞撞，有失望有絕望，一路堅持下來真的很不容易，希望這些研究成果能給與需要的學弟妹和學長姐省下很多時間。

除了指導老師以外，我也很榮幸的認識很多位背影都會發光的學長姐，小P、怡萱、蟲蟲、卡比、丁丁、玉潔、阿界、阿勇、阿泰、文卿，不只在我剛進實驗室遇到問題時給了很多幫助，即使他們畢業後也還是常受到我的騷擾，在此不只要致上謝意還要加上歉意，感謝你們的包容。同屆的小慈、小M、偉敦、小倩，我會懷念我們一起準備老張考古題的時光，很難得的緣分。學弟妹，冠傑、郁翔、昶瑞、英明、欣諦、昀俊、阿衣、誠誠，我也會懷念你們在我疲累時成為我嘻笑打鬧的犧牲品。

口試完的當天剛好看到了一篇部落格文章：研究所與研究生的意義，裡面所描述的不良研究生幾乎像是以我為範本，完全沒研究生該具備的氣質，整天打球、玩樂，在此我要特別感謝那些陪我一起沒氣質的同窗們：忠義、建良、怡軒、羅賓、建宏、建堡、朱朱、凱賓、漢光、承峰、阿轟、章魚哥、奕愷、明諺、恭良、明龍、紹偉、Wednesday，因為你們研究所都不研究所了，酸甜苦辣都嘗遍，我會珍惜那段日子。最後，我要感謝在教發中心認識的心慧、嘉惠、晨凡、志祥、點高、淑琳，你們讓我體會了很不一樣的生活，少了你們的點綴，應該會很乏味，這兩年多來的收穫清單你們佔了很大一塊，很高興沒錯過你們。謝謝你們陪著我一起度過在政大的這些日子。能畢業的喜悅完全抵不過離開你們的難過。

王彥嵩 November 2011

## 目錄

第一章、簡介.....	1
1.1、災區應急通訊系統需求分析.....	2
1.1.1、救災作業時效性.....	3
1.1.2、大型自然災害發生時救災行動面臨的挑戰.....	4
1.1.3、應急通訊系統需求.....	4
1.2、無線通訊系統損毀原因.....	6
1.3、應急通訊網路簡介.....	9
1.4、排程問題.....	11
1.5、論文架構.....	11
第二章、相關研究.....	12
2.1、第三代行動通訊架構.....	12
2.1.1、通用移動通訊系統無線接入網 (UTRAN).....	12
2.1.2、核心網路 (Core Network).....	13
2.1.3、3G 網路通訊協定.....	14
2.2、緊急通訊系統種類.....	15
2.2.1、無線對講機 (Walkie-Talkie).....	15
2.2.2、業餘無線電 (Amateur Radio).....	16
2.2.3、行動衛星通訊.....	16
2.2.4、集群通訊系統 (Trunking radio).....	17
2.2.5、移動式基地台 (Cell on Wheels).....	18
2.2.6、行動隨意式網路 (MANET).....	18
2.2.7、過去應急通訊系統相關研究.....	19
2.2.8、評論.....	22
2.3、應急蜂巢式通訊系統 (CCN).....	23
2.3.1、系統架構.....	23
2.3.2、系統使用時機.....	24
2.3.3、系統優勢.....	25
2.3.4、系統元件.....	25
2.3.5、重要技術問題探討.....	26
2.4、排程問題.....	30
2.4.1、概述.....	30
2.4.2、精確解法 (Exact Algorithm).....	31
2.4.3、後啟發式演算法 (Meta-heuristics).....	33
2.4.4、評論.....	38
第三章、CCN 建構排程.....	40
3.1、設計理念與目標.....	40

3.2、問題定義.....	41
3.2.1、救災效益定義.....	41
3.2.2、問題描述.....	41
3.3、複雜度分析.....	43
3.4、啟發式排程演算法.....	45
3.4.1、演算法設計.....	45
3.4.2、CCN-DS 演算法.....	46
3.4.3、CCN-DS 演算法範例.....	50
第四章、效益評估.....	53
4.1、實驗設計.....	53
4.1.1、實驗環境.....	54
4.1.2、評估指標.....	54
4.2、救災效益函數.....	54
4.2.1、救災效益函數變數.....	55
4.2.2、救災效益函數模型.....	55
4.2.3、優先權設定.....	56
4.3、實驗參數.....	58
4.4、實驗結果.....	58
4.4.1、實驗一.....	58
4.4.2、實驗二.....	64
4.4.3、實驗三.....	70
4.4.4、實驗四：大規模案例.....	76
4.5、實驗總結.....	77
第五章、結論與未來展望方向.....	80
參考文獻.....	82

## 表目錄

表 1、大型天然災害傷亡損失記錄.....	2
表 2、生還機率.....	3
表 3、緊急通訊系統比較.....	23
表 4、第二階段 CCN-DS SCHEDULE 調整.....	57
表 5、PARAMETERS AND RANGES FOR TEST INSTANCES.....	58
表 6、實驗一結果演算法誤差(10 個基地台).....	59
表 7、實驗一結果演算法誤差(11 個基地台).....	60
表 8、實驗一結果演算法誤差(12 個基地台).....	61
表 9、實驗一結果演算法誤差(13 個基地台).....	62
表 10、實驗一結果演算法誤差(14 個基地台).....	63
表 11、實驗二結果演算法誤差(10 個基地台).....	65
表 12、實驗二結果演算法誤差(11 個基地台).....	66
表 13、實驗二結果演算法誤差(12 個基地台).....	67
表 14、實驗二結果演算法誤差(13 個基地台).....	68
表 15、實驗二結果演算法誤差(14 個基地台).....	69
表 16、實驗三結果演算法誤差(10 個基地台).....	71
表 17、實驗三結果演算法誤差(11 個基地台).....	72
表 18、實驗三結果演算法誤差(12 個基地台).....	73
表 19、實驗三結果演算法誤差(13 個基地台).....	74
表 20、實驗三結果演算法誤差(14 個基地台).....	75
表 21、大規模實驗結果演算法誤差(100 個基地台).....	76
表 22、實驗結果綜合比較初始解誤差.....	78
表 23、實驗結果綜合比較最終解誤差.....	79
表 24、實驗結果綜合比較最終解與初始解差距.....	79

## 圖目錄

圖 1、3G 網路易損毀鏈結 .....	8
圖 2、八八水災中橋樑遭洪水沖毀 .....	8
圖 3、橋梁損毀導致通訊線路損毀 .....	8
圖 4、橋梁斷裂導致線路中斷 .....	9
圖 5、CCN 網路 .....	10
圖 6、新世代行動通訊系統 .....	12
圖 7、3GPP RELEASE 5 架構 .....	14
圖 8、3GPP RELEASE 7 架構 .....	15
圖 9、CCN 使用時機 .....	25
圖 10、不同環境的存活率示意圖 .....	40
圖 11、提出之兩階段建構排程演算法 .....	46
圖 12、加權 PROFIT 計算示意圖 .....	47
圖 13、候選節點示意圖 .....	47
圖 14、第一階段 CCN-DS 演算法中 GREEDY ALGORITHM 設計 .....	48
圖 15、第一階段 CCN-DS 貪婪演算法設計 .....	48
圖 16、第二階段 CCN-DS 演算法設計 .....	49
圖 17、時間點一之候選節點 .....	51
圖 18、時間點一之候選節點選擇 .....	51
圖 19、時間點二之候選節點 .....	51
圖 20、時間點二之候選節點選擇 .....	51
圖 21、時間點三之候選節點 .....	52
圖 22、時間點三之候選節點選擇 .....	52
圖 23、時間點四之候選節點 .....	52
圖 24、時間點四之候選節點選擇 .....	52
圖 25、救災效益函數模型一 (PROFIT FUNCTION 1) .....	55
圖 26、救災效益函數模型二 (PROFIT FUNCTION 2) .....	56
圖 27、救災效益函數模型三 (PROFIT FUNCTION 3) .....	56
圖 28、實驗一結果演算法效能(10 個基地台) .....	59
圖 29、實驗一結果演算法效能(11 個基地台) .....	60
圖 30、實驗一結果演算法效能(12 個基地台) .....	61
圖 31、實驗一結果演算法效能(13 個基地台) .....	62
圖 32、實驗一結果演算法效能(14 個基地台) .....	63
圖 33、實驗一不同基地台個數演算法平均效能比較 .....	64
圖 34、實驗二結果演算法效能(10 個基地台) .....	65
圖 35、實驗二結果演算法效能(11 個基地台) .....	66
圖 36、實驗二結果演算法效能(12 個基地台) .....	67



圖 37、實驗二結果演算法效能(13 個基地台) .....	68
圖 38、實驗二結果演算法效能(14 個基地台) .....	69
圖 39、實驗二不同基地台個數演算法平均效能比較 .....	70
圖 40、實驗三結果演算法效能(10 個基地台) .....	71
圖 41、實驗三結果演算法效能(11 個基地台) .....	72
圖 42、實驗三結果演算法效能(12 個基地台) .....	73
圖 43、實驗三結果演算法效能(13 個基地台) .....	74
圖 44、實驗三結果演算法效能(14 個基地台) .....	75
圖 45、實驗三不同基地台個數演算法平均效能比較 .....	76
圖 46、大規模實驗結果演算法效能(100 個基地台) .....	77

# 第一章、簡介

近年來，全球天災頻傳，極端氣候、大型地震...等，造成大範圍災害，其規模一次比一次嚴重，如表 1 就是近幾年較著名的自然災害統計，以最近 2011 年 3 月 11 日發生的日本東北地震為例，在經歷芮氏 9.0 規模地震之後，緊接著 23 公尺高的海嘯和令全世界恐慌的核災，所照成的三種複合式的重大災害，令多次參與災區救援的救災人員也為之驚訝，而台灣處於環太平洋地震帶以及西太平洋颱風路徑上，四面環海、地形差異大，頻繁的地震、颱風、土石流和水災...等自然災害，更對台灣造成嚴重的損害。所有這些損害當中對人們影響最大的即是基礎設施的毀損，尤其是通訊系統的癱瘓影響救災效率甚鉅。本論文旨在提出一個簡單有效的應急通訊系統提供緊急通訊用。

目前無線通訊已完全融入一般大眾的生活與工作之中，無線通訊系統的成熟為使用者帶來極大的便利性，但當大規模的地震或強烈颱風等重大天然災害發生時，通訊系統卻常常因架構原因隨著電力與交通系統的損毀而癱瘓。以莫拉克風災/八八水災為例，許多基地台因建在高處免於洪水淹沒，結構未損，但隨著道路及橋樑鋪設的電力與通訊線路隨道路橋樑坍塌而損毀，造成行動通訊系統也隨之癱瘓，電力與 backhaul 線路成為行動通訊網路的弱點。由歷年大型災變中，多數災區內之行動通訊系統全面中斷，即可印證行動通訊系統其實是極為脆弱。由於受到諸多外在因素的牽連，建造強固的基地台與交換機房仍是無濟於事，無法保證通訊系統可用度。國家通訊傳播委員會雖然在各地建置具有行動衛星通訊能力的強固基地台，但因成本之故，數量遠遠不足，僅能作為官方救災指揮之用，對於廣大地區的受災與救災人員而言，只是杯水車薪。

而有效運作的通訊系統卻是災情傳遞、資源調度以及互助協調是否順利的關鍵因素，災民的存活率會隨著時間的推移慢慢下降，這也是所謂的救援黃金 72

小時，受困的災民在此段時間內會有較高的存活率，除去災難發生當下的求援，災後組織救援也是通訊的另一個重點，災區內食物、飲水、衣物、醫藥、系統癱瘓下，倖存人員的維生也是救災的重要任務。因此盡速恢復通訊以輔助救災、求援為刻不容緩的事情，越早恢復通訊就能救援越多人出來，因此在災害來臨通訊中斷時，快速的建構一個應急通訊系統供給災區內的受困人員與救災人員使用成為一個關鍵性的問題。應急通訊系統有很多種方法可以建構，本篇論文所探討的應急通訊系統是利用倖存的連通基地台和斷訊卻沒有損毀的基地台建構一個暫時性的網路，稱為應急蜂巢式行動網路(contingency cellular network, CCN)，此種應急通訊系統的主體為兩種基地台：完好維持正常功能可對外連線的稱為**連網台**，功能完整但無法對外進行正常連線的稱**孤立台**。由於災區的交通系統可能癱瘓，且各個地方受災情況不盡相同，因此 CCN 的建構順序必須辨明輕重緩急並視運輸能力依序建構，網路拓樸的規劃是本研究團隊的另一研究主題，而本文旨在探討如何在網路拓樸已知情況下進行孤立台建構排程以達到最大的救災效益。

表 1、大型天然災害傷亡損失記錄[8,18,23]

Event	Date	Dead
Chi-Chi, Taiwan Earthquake	09/21/1999	2,415
Katrina Hurricane	08/23/2005	1,836
SiChuan, China Earthquake	05/12/2008	69,227
Port-au-Prince, Haiti Earthquake	01/13/2010	230,000
Chile Earthquake	02/27/2010	800
QingHai, China Earthquake	04/14/2010	1706
North-East Japan Earthquake	03/11/2011	13,013
Turkey Earthquake	23/10/2011	601

## 1.1、災區應急通訊系統需求分析

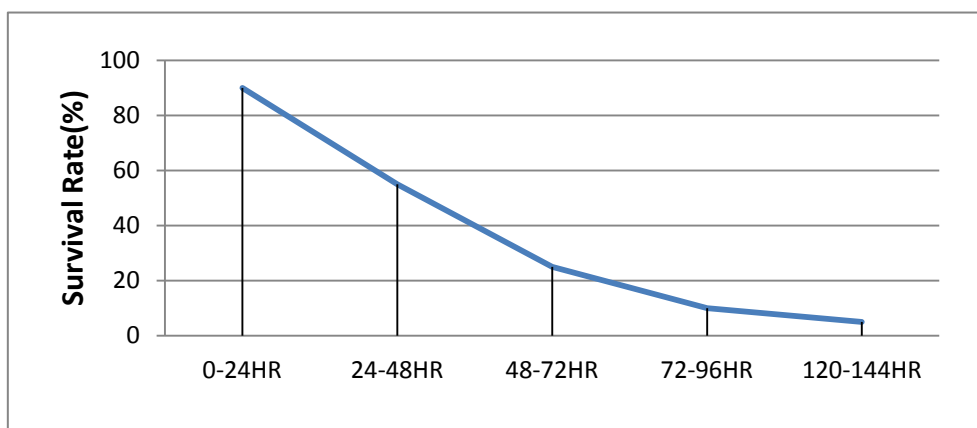
在災害發生時普遍大眾最需要的服務之一是：通訊(包括災情傳遞、受困人員之

求救、救災人員聯繫協調...等)，但在大型天然災害下通訊系統卻非常脆弱，現有很多應急通訊系統尚有很大的改進空間。本節試從 921 地震與歷年來的大型天災中歸納出一些救災行動面臨的挑戰經驗，以供應急通訊系統設計之參考[7,11,17,18]。

### 1.1.1、救災作業時效性

「黃金 72 小時」是很多災害發生後的黃金救援期，這是救援(學)界的共識。救援界認為，災難發生之後存在一個「黃金 72 小時」，在此時間段內，受困災民的存活率極高。任何救援幫助都可能給予傷者存活的機會，72 小時期間，災民的存活率隨時間的消逝呈遞減趨勢。在第一天(即 24 小時內)，被救出的人員存活率在 90%左右；第二天，存活率在 50%-60%；第三天，存活率在 20%-30%。72 小時之後，存活率僅存 5%-10%，如表 2。基於這些因素，救災工作必須與時間賽跑，CCN 由於交通系統的癱瘓而受限於有限的空中運輸能量只能逐步建置，而建構的先後順序應依災區受損程度與受困民眾多寡等因素妥善安排以提高救災效率，因此 CCN 建置排程成為一個關鍵問題。

表 2、生還機率<資料來源: <http://baike.baidu.com/>>



### 1.1.2、大型自然災害發生時救災行動面臨的挑戰

- **交通癱瘓**：大部分災區聯外道路因而中斷；剩下可用的，亦被大量志願救災車輛阻塞，造成災區交通全面癱瘓。
- **外援進入緩慢**：因交通全面中斷的緣故，造成外界支援難以於第一時間進入災區。莫拉克颱風中，因氣候關係，直昇機難以接近災區。而海地地震中，太子港機場雖逃過一劫，但因機場無法提供返航油料以致飛機無法降落，因此災變初期，主要依賴在地的人力物力投入救援，無法依賴外界支援。
- **通訊網路癱瘓**：包括固網、行動電話在內的通訊聯絡網路幾乎全面癱瘓。倖存的通訊網路也因塞滿大量的關懷電話，無法供救災使用。
- **行政指揮系統無法運作**：各級行政指揮系統可能癱瘓，導致既有通聯組織癱瘓。例如八八水災中，小林村長就不幸罹難；2004年七二水災中，台中縣和平鄉松鶴派出所為土石流淹沒，完全外界失聯達數天之久。海地地震中，政府大樓癱塌，政府首長全部失聯，僅剩海地總統獨撐大局，所有行政系統形同癱瘓，難以負起救災規劃和指揮的大責。
- **專業救災人員不足**：專業救災人員之數量遠遠不足，尤其是災變初期。必須動員大量的在地志願人員投入救災，但此些在地志願救難人員卻沒有很適當的無線緊急通訊系統可供使用組織救援。

### 1.1.3、應急通訊系統需求

#### 建置需求：

- **設備取得容易**：所需設備須可就地取材或運送容易，災區內的道路系統可能已經癱瘓，物資大多只能倚賴空中運輸，大型設備如行動基地台，不便利用

空中運輸送至災區。

- 佈建容易：災區現場很難保有夠多的專業人員來建置應急通訊系統。因此，建置的難度越低越好。系統元件取得容易和建置難度低主要目的是希望可以提高建置應急通訊系統的成功率。建置成功率越高，越能廣泛使用於災區。
- 建置速度快：為把握救災的時效性，應急通訊系統建置速度越快越好，隨著時間拉長，死亡率快速增加，而且行動通訊營運商也逐漸修復其通訊網路，應急通訊系統會隨著時間而逐漸被取代，若建置時間緩慢則無建置之必要。

#### **運轉需求：**

- 使用端設備數量多、成本低：若需有特定的話機才可使用，因成本及分配之故很難在短時間讓大量志願救災人員人手一機，這會讓應急通訊系統的服務容量大打折扣。
- 易於使用，不需特別訓練：在緊急狀態下無法從容的提供使用訓練，因此最好是未經訓練就可使用。
- 具備允入控制機制，可抵擋瞬增的話務量：災區話務量常會有瞬增的特性。應急通訊系統因為是臨時搭建，系統能容納的通話數遠不如一般的公眾網路，更容易因這些瞬增的話務量導致整個系統癱瘓。因此，應急通訊網需具備允入控制，當系統負載過高時，需能拒絕新的通話請求。
- 具救災緊急通話先行之功能：應急通訊網需有緊急通話先行的功能，急迫性高的通話可優先使用。如此，可以避免整個應急通訊網被急迫性較低的互道平安通話佔滿，使得急迫性較高的急難救助通話卻無法通話。
- 具行動能力：救災人員執行救災任務需要來回走動，應急通訊系統最好能提供行動能力，讓救災人員可在行動中使用。

分析以上緊急通訊系統需求我們可以歸納出八項評斷指標：

- 使用成本：廣泛使用此緊急通訊系統的成本
- 建構難易度：將此緊急通訊系統建構起來的困難度
- 進入災區難易度：將緊急通訊系統運送進災區的難易度
- 終端設備普及率：災區人員具有此緊急通訊系統通訊設備的程度
- 終端設備操作難易度：災區人員操作此通訊設備的難易度
- 終端設備可移動性：災區人員攜帶通訊設備移動的能力
- 通訊品質：利用此緊急通訊系統的通訊品質
- 系統運轉難易度：緊急通訊系統架構起來後維持運轉的難易度

## 1.2、無線通訊系統損毀原因

因為行動電話使用無線通訊，在一般人普遍認知中它的抗災應付緊急情況的能力應該是相當高，在災害來臨時可作為緊急通訊之用，但事實上卻非如此，商用行動通訊系統其實必須仰賴固定通訊網路，其基地台之後端多利用固網幹線連上核心網路，無線的部分只存在於終端使用者(手機)到基地台那一段，從基地台到機房仍然是利用固網線路，從 921 地震以來各次大型災害的經驗中我們歸納出無線通訊系統主要失效原因成三部分：1.電力中斷、2.線路中斷、3.基地台與機房毀損，而無線通訊系統中斷的原因也發現並不全都是因為基地台本身的損毀，其原因是由很多外在因素交織在一起導致的結果，而整體無線通訊系統架構的脆弱並無法單純從加固基地台或提升基地台密度來彌補，以莫拉克風災/八八水災為例，許多基地台因建在高處免於洪水淹沒，結構完整並未損毀。但因隨著道路及橋樑鋪設的電力與通訊線路隨道路橋樑坍塌而損毀，造成行動通訊系統也隨之癱瘓。電力與 backhaul 線路成為行動通訊網路的弱點，以下是毀損原因以及目前應對方法詳細分析：

- **基地台機房電力中斷**：電力中斷導致機器停止運行(備用電源僅能支持一至二小時)，在八八水災中 3300 座斷訊的基地台中約 70% 是因為電力中斷而斷訊。目前多數基地台機房不具備有發電機，只有部分設備內部會自備電源(直流蓄電池)，另外就是利用 SMR(switch module rectifier)把 UPS(Uninterruptible Power Supply)提供的交流電轉換成-48V 的直流電供機房短暫使用。
- **Core network 機房電力中斷**：機房會有發電機的配置，在設備停電後啟動電池儲備電力的同時，立即啟動發電機供電，但一旦油料告罄，無法獲得補充，機房即會因為電力中斷而停止運作。
- **基地台與機房損毀**：基地台或機房遭自然災害摧毀，有可能是因土石流的沖擊、大樓倒塌壓毀或是於地震中震毀，也可能是因機房所需冷卻水供應中斷，交換機因過熱而停機。(冷卻水可能因各種原因而供應中斷，例如管線遭強震撕扯而破裂或循環冷卻用水蒸發殆盡但自來水供應卻中斷而無法補充)。
- **線路中斷**：基地台連接基地台控制器 (Base Station Controller)或行動交換機 (Mobile Switching Center)的後端固定網路線路 (backhaul)損毀，3G 網路連線常出現損毀的地方有三處，如圖 1。
  - Node B 與 RNC 之間連線中斷
  - MSC/VLR、SGSN 與 HLR、GGSN 之間連線中斷
  - GMSC、GGSN 與外部網路連線中斷

如圖 2、3、4，在 921 地震及莫拉克風災/八八水災中，道路柔腸寸斷，大型橋樑遭洪水沖毀，如圖 2，而隨著道路及橋樑鋪設的電力及通訊線路也隨之柔腸寸斷，莫拉克風災/八八水災中 3300 餘座斷訊的基地台中，大部分本身並未受損，而是因電力與 backhaul 線路損毀而停止運轉，這些線路成為行動通訊網路的弱點。



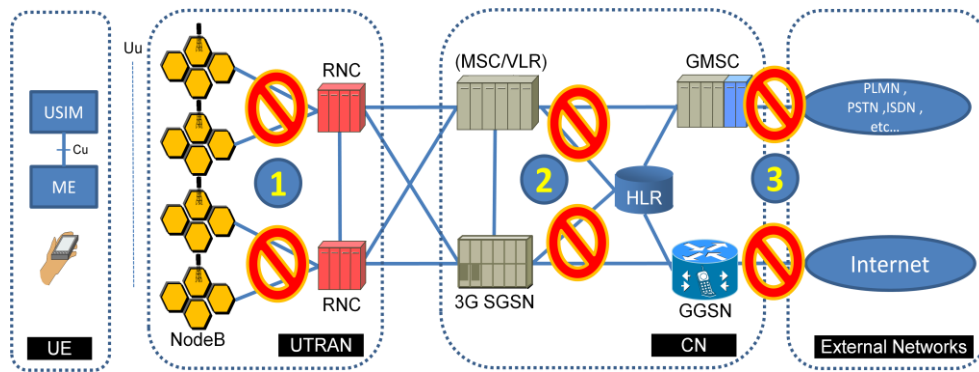


圖 1、3G 網路易損毀鏈結



圖 2、八八水災中橋樑遭洪水沖毀



圖 3、橋梁損毀導致通訊線路損毀



圖 4、橋梁斷裂導致線路中斷

### 1.3、應急通訊網路簡介

目前常見的應急通訊系統有無線對講機(Walkie-Talkie)、業餘無線電(Amateur Radio)、行動衛星通訊、集群通訊系統(Trunking Radio)、移動式基地台，而近年來有許多研究倡議使用 MANET(mobile ad hoc network) [4,5,26]建構應急通訊系統；行動衛星通訊是利用人造衛星作為中繼站轉發無線電信號，在使用者之間進行的通訊，可不受任何環境限制也不受天災之影響，但它的缺點同樣明顯，價格太高因而普及率低，一般人不會持有此種設備；集群通訊系統則是由早期的專用無線電系統逐漸發展形成的，具有充沛的調度功能(群組呼叫、優先分級、快速接續...等)，此系統通常為警察等專用單位所擁有，需要建構才可使用；移動式基地台為國內最常見的應急系統方式，移動式基地台常被當成是現場緊急通訊的首選方式但因造價昂貴數量稀少；Amateur Radio 則是俗稱的業餘無線電，只要頻率相同，電波所及範圍內即使不知道對方身分也可通訊，具有很好的廣播功能，缺點為數量稀少；MANET 則是具有多跳、自組織、自癒的寬頻無線網路，並不需要有線基礎設施支持，在 MANET 網路中所有節點可隨時加入或離開，只需任意與其中一個節點相連即可，建構快速但穩定度低；Walkie-Talkie 則與上述所提

幾種應急通訊系統不一樣，其優點眾多，有著體積小重量輕、長時間使用、不需事先佈建...等，但也有著現實的問題，在很多國家除專業救難隊以外幾乎沒有普遍擁有。

需額外設備的應急通訊系統在災害發生後往往因為道路支離破碎不能直接進入災區，即使進入了也只有少數人可以使用，無法普及到一般災民。因此我們目前提出了兩種方法解決此問題，第一種為利用 MANET 發展出一個隨意型網路應急通訊系統，此系統透過筆記型電腦建構成行動隨意網路並利用 P2P Based VoIP 通訊技術以支援緊急通訊與資訊服務，稱為 P2PNet，在大型自然災害發生之初期，可迅速的讓受災人員與救難人員以現有的電腦設備建構簡單的 MANET 模擬 Walkie-Talkie 進行短距離群組通訊，系統優勢為筆電等設備可就地取材，只需具備基礎電腦知識即可架設使用。第二種即是應急蜂巢式行動網路 (Contingency Cellular Network)，利用現有的無線通訊系統中未損毀但失去電力或連網能力的基地台以 WiFi 等無線設備互相連線，建立一個臨時網路供災區的手機用戶使用，其優勢為大量的手機用戶可立即投入救災使用，如圖 5，本論文即是在此基礎(CCN)上進行研究。

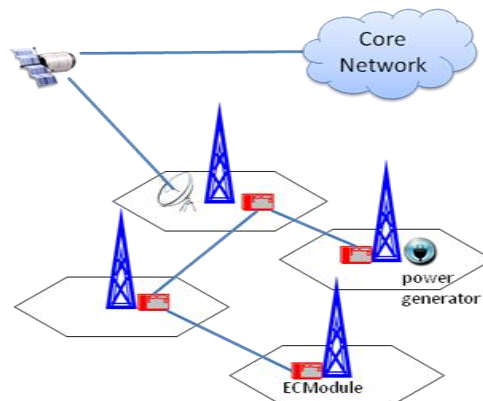


圖 5、CCN 網路

## 1.4、排程問題

排程問題在數學規劃中屬於組合性最佳化問題(combinatorial problem)。然而，大多數組合性最佳化問題是 NP-Hard 問題，意即求解該類問題的解題時間會隨著問題的大小增加而呈現指數成長，為了要在設定的時間內滿足需求，必須尋求更有效率的求解方法，實務應用上往往選用較快速的次佳解法。

排程問題的核心是對有限資源的選擇應用，有限的資源對不同的程序執行進度有重要影響，而由一系列相互關聯的程序才能組成一個工作，此類問題稱之為 RCPSP (resource constrained project scheduling problem)，其調度決策需同時滿足各程序之間的時序約束和資源約束，滿足此兩項條件下尋找符合目標之排程問題解。

## 1.5、論文架構

本文共分成五章，第二章介紹現行的 3G 行動網路架構和目前常見的應急通訊系統，並說明如何用現行的 3G 行動網路架構建構一個應急通訊系統，以及幾個關於排程問題的相關研究，分析他們在此排程問題的適用性。第三章介紹我們提出之排程方法。第四章則藉由實驗數據驗證我們的方法可行性。第五章則為結論與未來發展。

## 第二章、相關研究

### 2.1、第三代行動通訊架構

第三代行動通訊架構簡稱 3G (3rd-generation)，是指高速數據傳輸的蜂巢行動通訊技術。3G 技術能夠同時傳送聲音(通話)及數據(電子郵件、即時通訊等)。代表特徵是提供高速數據服務。相對於第一代類比式行動電話系統(1G)與第二代只具有通話和一些諸如時間、日期等固定格式數據的手機通訊技術規格的 GSM、CDMA 等數位調變式手機 (2G)，3G 手機是將無線通訊與 Internet 等多媒體傳輸結合的新一代行動通訊系統。主要由 UTRAN(UMTS Terrestrial Radio Access Network)與 CN(Core Network)兩部分組成[29]，如圖 6。

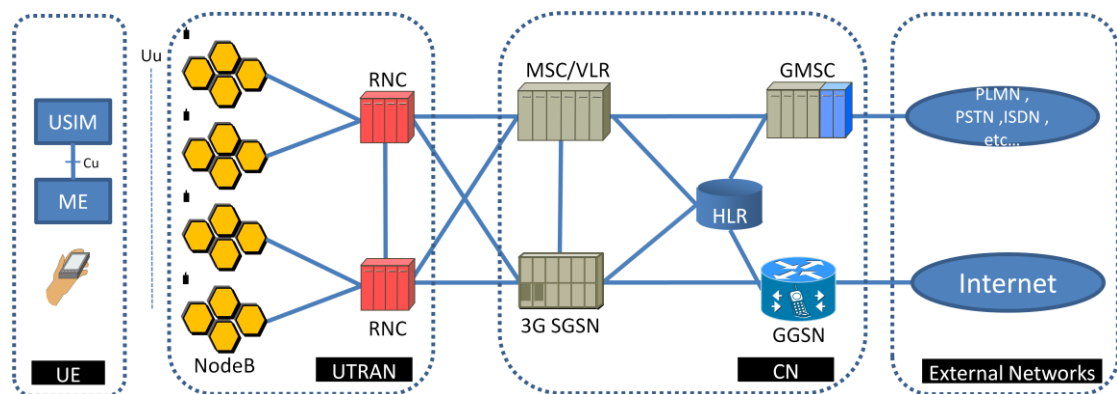


圖 6、新世代行動通訊系統

#### 2.1.1、通用移動通訊系統無線接入網 (UTRAN)

UTRAN(UMTS Terrestrial Radio Access Network) 由多個 RNS (Radio Network Sub-system)所組成，每個 RNS 包括一個 RNC 與其數個相連的 Node B，RNC 與 Node B 之間使用 Iub interface 相連，每一個 RNC 透過 Iu-PS interface 與一個 SGSN 之間相連，並透過 Iu-CS interface 與一個 MSC 相連。

- **RNC(Radio Network Controller)**：無線網路控制器是 3G 網路的一個關鍵部分。它提供 Mobility Management、呼叫處理、鏈接管理和切換機制，具體工作為管理用於傳輸用戶數據的無線接入、管理和優化無線網路資源和無線連結維護，RNC 擁有和控制管轄範圍內所有 Node B 的無線電資源，作為 service access point 提供服務給 Core Network，為連到 Core Network 之前的最後部分，以台灣而言一台 RNC 大約控制 50~300 台 Base Station。
- **Node B**：即是 Base Station，負責 WCDMA 系統的無線收發，通過 Iub 接口和 RNC 互連，主要處理 Uu 接口物理層協議。功能有展頻、調變、通道訊號編碼及通道訊號解碼，還包括基頻信號和射頻信號的相互轉換等功能。

## 2.1.2、核心網路 (Core Network)

CN 由 CS 交換機、PS 路由器、資料庫及長途幹線組成，主要設備存放於機房中，包含了 MSC/VLR、GMSC、HLR、SGSN、GGSN 五個部分：

- **HLR(Home Location Register)**：是一種用戶資料庫，保存用戶的基本資料，如 SIM 的卡號、手機號碼和用戶狀況，例如當前的位置、是否已經關機等。
- **VLR(Visitor Location Register)**：保存用戶的狀態資料。
- **MSC(Mobile Switching Center)**：負責路由的語音通話和短信，以及其他服務(如電話會議，傳真和電路交換數據)。
- **SGSN(Serving GPRS Support Node)**：負責數據包的 Mobility Management、路由轉發、會話管理、邏輯鏈結管理、加密和輸出等功能。
- **GMSC(Gateway MSC)**：提供 CS domain 連接到外界網路的交換機。
- **GGSN (Gateway GPRS Support Node)**：提供 PS domain 連接到外界網路的交換機。

### 2.1.3、3G 網路通訊協定

**Before 3GPP release 5**：在 3GPP Release 5[29]之前的 3G 架構下，用戶手機本身會具備自己的 IP 位置，透過 Node B 連線到 RNC 後，RNC 會用自己的 IP 將用戶的封包封裝起來，並且與 SGSN 透過 Iu-ps 協定連線，然後 SGSN 會再用 Gn/Gp 協定與 GGSN 連線，中間其實就是 Tunnel，一直到 GGSN 以後，才會將原本用戶手機的封包解封裝後送到 Internet，如圖 7。

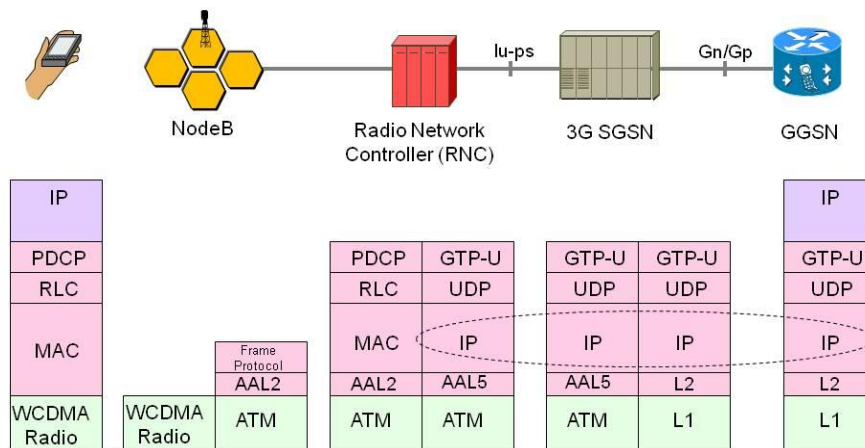


圖 7、3GPP Release 5 架構

**3GPP R5 to R7**：新的 3GPP R5 到 R7 版本[29]，精簡了網路的架構，首先是 HSDPA 移除了 Drift RNC 元件，並且將 Drift RNC 的功能移到 Node B 上面，再來是修改 SGSN，讓 RNC 可以直接與 GGSN 使用 Tunnel 連線，如圖 8，在 HSPA+的版本，又將 RNC 的部分功能移到 Node B 上面。



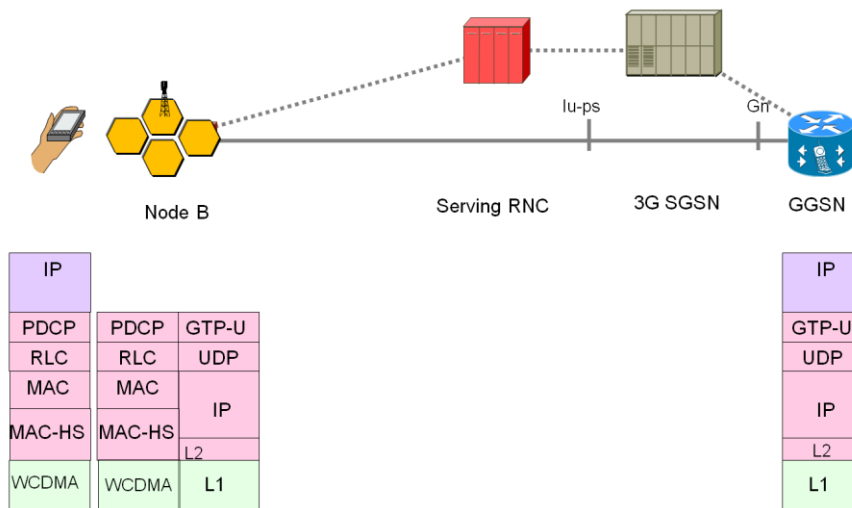


圖 8、3GPP Release 7 架構

## 2.2、緊急通訊系統種類

### 2.2.1、無線對講機 (Walkie-Talkie)

一種手持的雙向無線電收發器，其原理很像 2.2.2 節所介紹之 Amateur Radio，使用免執照的 ISM 頻道，同時間只有一位使用者可以廣播語音訊息(半雙工)[17]。

優點：

- 不需佈建任何線路即可使用
- 體積小重量輕，可隨身攜帶
- 電池充電後可長時間使用
- 只要頻率相同，電波所及範圍內即使不知道對方身份地點即可通訊

缺點：

- 在世界很多地方普及率低 (例如：台灣)
- 沒受過訓練很難直接使用(緊急時無經驗之使用者必須在短時間內讀懂說明書自行學習使用)



- 沒有優先分級能力

### 2.2.2、業餘無線電 (Amateur Radio)

類似於廣播電台，差別只在於電波所及的範圍較小，原理為通過無線電進行訊號傳輸，早期使用長波段，因為長波段能量損失小且能繞過障礙物，但由於長波的天線設備龐大、昂貴、通訊量小，後期使用能藉電離層反射的短波，使得無線電設備價錢大幅降低，一般使用者也有能力使用，因此短波成為業餘愛好者的使用主流，將 Amateur Radio 當作緊急通訊系統會具有的優點為：電波所及範圍內即使不知道對方身份地點亦可大範圍廣播通訊，適合做訊息發佈功能，缺點為普及率和移動性非常低，且使用上困難，沒受過訓練很難直接使用，緊急時架設的架站很難找到操作者[13]。

### 2.2.3、行動衛星通訊

行動衛星通訊系統是由通訊衛星和通訊地面站兩部分所構成。行動衛星通訊是指利用人造衛星作為中繼站轉發無線電訊號，在兩個或多個使用者之間進行數據傳輸。在此系統中，通訊衛星只是一個掛在空中的通訊中繼站。由於高度關係它視野開闊，只要在覆蓋範圍內，不論距離遠近都可以通訊，通過它轉發和反射電報、電視、廣播和數據等無線信號[16,30]。

衛星通訊系統可以由衛星所在高度分為同步衛星(Geostationary, GEO)、中軌道衛星(Medium Earth orbit, MEO)和低軌道衛星(Low Earth orbit, LEO)。同步衛星所在高度為三萬五千七百多公里，中軌道衛星所在高度為一萬至兩萬公里，低軌道衛星所在高度為七百五至兩千公里之間。高軌道的同步衛星因高度太高傳至地面的訊號太微弱，導致接收設備造價高昂，體積龐大(常見的碟型天線之直徑至少數英尺)，且延遲時間太長不適合作為語音通訊使用，更不適合作為行動衛星

通訊之用。目前常用於行動衛星通訊系統的為中低軌道衛星，中地球軌道(MEO: Medium-Earth Orbit)距地 5000~15000km，美國的奧德塞計劃(Odyssey)即是使用中軌道衛星；低地球軌道(LEO: Low-Earth Orbit)距地 500~1500km 的高度，Motorola 主導的銜計劃(Iridium)是使用低地球軌道衛星之計劃。

行動衛星通訊優點有：覆蓋面廣、通訊距離遠、不受地理條件限制、通訊頻帶寬、傳輸容量大、建構靈活、易於實現多點通訊、具有優良的廣播特性、性能穩定可靠、傳輸距離遠等優點。缺點有：通訊時間延遲較長、容易受外部條件影響、價格高昂、普及率低。

#### 2.2.4、集群通訊系統 (Trunking radio)

由早期的專用無線電調度系統逐漸發展形成的，系統中每一個無線設備都會透過一個或多個中繼站來把訊息發散出去，這種通訊系統主要用於對戶外作業的移動用戶提供調度與指揮控制等服務，具有普通無線電通訊的語音、數據等功能外，還具備群組呼叫、優先分級、快速接續等能力[31]，作法是由中央控制器集中控制和管理系統中的每一個頻段，以動態方式迅速的把空閒頻段分配出去，用戶群會呈現樹狀結構，常用於指揮調度通訊，缺點為必須事先建構，以下介紹美規的 Project 25[1]以及歐規的 TETRA[2]應急通訊系統，兩者皆採用 TDMA 時分技術。

- **Project 25** [1]：APCO(Association of Public-safety Communications Officials-international)於 1989 年推動的計劃(簡稱 P25)其制定了相關標準來提供服務和各廠商互連相容能力(Multi-Vendor Interoperability)，以求找到符合公共安全與關鍵性任務需求之解決方案，P25 具備支援傳統/數位(Analog/Digital)與中繼集群模式，在小於 200 個使用者時或小規模地方政府受限於預算時才採用傳統模式，否則通常採用中繼集群模式。
- **TETRA (Terrestrial Trunking Radio)** [2] 又稱 Trans-European Trunking Radio，

為專業移動無線電(Professional Mobile Radio, PMR)和雙向收發器(walkie-talkie)規範，類似於 P25 為專門設計用於公共安全與關鍵性任務需求之無線電通訊規範，除了以公共安全(Public Safety)與關鍵性任務需求之解決方案為考量設計外，也提供給鐵路運輸列車服務和捷運系統無線電通訊服務等大眾交通系統。與 P25 不一樣地方為 TETRA 只提供數位式中繼集群模式。

### 2.2.5、移動式基地台 (Cell on Wheels)

移動式基地台實際上就是一個可移動的通訊系統，通過開到緊急現場的車載平台，搭建通訊網路，實際處理現場傳輸來的語音、影像、圖片等數據，實現現場各種不同規格、不同頻段的通訊網路的交換，構成統一的應急指揮平台。

由於移動式基地台具有架設速度快、運用靈活、調度方便、自帶電源設備等特點，因此，在大多數自然災害、突發事件和重大事件的發生的情況下，應急通訊車通常是現場緊急通訊的首選方式之一，但專業設備需專人操作，成本高昂無法大量佈署導致接通手機數量有限，並且需要完好交通系統才能進入災區，在大型自然災害中所能負擔的通訊比例不大，並且因為交通可能斷絕，這些設備無法在第一時間送進災區，延誤救災效率，目前中華電信在台灣北部有 18 台、中部有 11 台、南部有 8 台移動式基地台[3]。

### 2.2.6、行動隨意式網路 (MANET)

Ad hoc 網路是一種沒有有線基礎設施支持的移動網路，由具有無線區域網路能力的筆電或平板電腦構成，每個節點皆可移動，並由此些節點構成一個網路，在 Ad Hoc 網路中，當兩個移動設備在彼此的連線覆蓋範圍內時，它們可以直接通訊，但是由於移動設備的通訊覆蓋範圍有限，如要兩個相距較遠的設備要進行通

訊，需要通過它們之同共同的移動設備來轉發才能實現[5,26]。它有以下特點

- **無控制節點**：所有節點皆可隨時加入或離開
- **容易組織**：不需要固定網路設施支持，能在任何時間、地點快速建構
- **平衡性差**：容易形成不平衡網路對個別節點有很高的負擔

我們先前的研究提出了一個利用志願救災人員的行動電腦建構隨意型網路並運用 P2P 技術建構成緊急通訊系統的設計，稱為 P2PNet，可以在沒有連接 Internet、沒有伺服器的情況下支援緊急的通訊與資訊運用[17]。

P2Pnet 依照傳統網路分層的概念，在網路層與傳輸層之間加入一個名為「網路服務層」的中介層，以完成 P2Pnet 所需功能。在實體層與網路層中，P2Pnet 將會試圖利用所有可運用的資源，包含 WiMAX，Mesh Network 與 VANET 等，機會網路(opportunistic network)的技術將會把各個獨立的 P2Pnet 整合成為一個互通的網路，在此情況下，網路節點之間可以將封包廣播到其鄰近的節點，在短時間內建立可用的通訊管道，以提供災區作為初期的緊急通訊之用。

### 2.2.7、過去應急通訊系統相關研究

- Autonomous Networked Robots for the Establishment of Wireless Communication in Uncertain Emergency Response Scenarios [26]：利用自主移動機器人在災區和救難中心建構一個 wireless ad hoc network 達到通訊目的，作者透過預測人群聚集的地方提出一個在每個機器人上運行的分散式演算法使他們共同最大化災民可能出現的地點，目標是最大限度地連接到網絡上的災民。同時為了要有效的分配時間和資源根據機器人的重疊範圍透過最小生成樹(minimum spanning tree)制定一個改進演算法。實驗結果顯示第一種演算法適合機器人數量少的時候，第二種演算法適合機器人能提供大頻寬的環境。

- Autonomous Community Construction Technology for Timely Transmitting Emergency Information[28]：本篇討論一個 Autonomous Community Construction Technology 應用在傳送有時效限制的緊急訊息。Wireless Sensor Network 在災難應急網路中已有廣泛的應用，傳統的及時緊急訊息傳送是建立在靜態以及集中式管理的條件下，可是實際的及時緊急訊息傳送是在隨時會發生變動的災區。本論文討論的 autonomous community construction technology 利用 Autonomous Decentralized System (ADS)每個子系統能自動收集、判斷和運作，改善傳統方法的限制，來達成在動態災區的及時緊急訊息傳送。本論文的結果也經由 Omnet 模擬傳統方法與本論文提出的 autonomous community construction technology 在災區的及時緊急訊息傳送，證實有助於及時訊息傳送。
- An Integrated Communication-Computing Solution in Emergency Management [3]：在本篇中作者透過設計一個有效的模型整合各種網路以建立一個可靠且迅速的異質通訊網路(heterogeneous meshed communication system)支持不同情況下的緊急危機管理系統，為了驗證有效性作者提出了一個整合網路應用層(application level)和網路層(network level)的實例，透過 multiple parallel modules (ParMods) 各節點(node)互相交換彼此的資訊(e.g. network latencies)，網路應用層(application level)即可在考慮到及他節點時選擇最符合需要的網路。
- Computational Public Safety in Emergency Management Communications [22]：在本篇中，作者分析了各種無線網路通訊選項在應急通訊背景下的可靠度。除了 propagation delay, packet delivery ratio 和 transmission rates 等傳輸品質以外還加上在有妨礙通訊的條件進行實驗。作者的實驗環境為完全擬真(使用實際的設備在現實的環境下)。實驗的對象有 Xbee、Bluetooth、WiMax 和 Wi-Fi，經過實驗作者認為在考量到經濟效益之下 Wi-Fi 是目前應急管理通

訊在災害現場的最佳選擇，其傳輸速率和範圍都優於其他選項。

- Ad Hoc Communications for Emergency Conditions [5]：作者提出了利用普及率很高的手機為節點建構一個 ad hoc 網路來進行通訊，其架構包含了可以自行適應網絡條件的廣播和路由協議。其做法為由一節點開始，向各間隔 120 度角的節點廣播，透過此廣播方式來組織拓樸，並由此拓樸節點知道路由傳輸路徑。經過作者實驗在節點密度高的時候傳輸延遲會有顯著的改善。
- Taiwan Earthquake Event Report, Risk Management Solutions [8]：本篇主要探討在 1999 年 9 月 21 號台灣中部發生的集集(chi-chi)大地震，作者利用詳細的數據及照片說明了此次地震的相關影響，包含了建築物的倒塌、基礎設施的損毀、交通的中斷、經濟影響以及大面積的地型變化和景觀改變等，並在每章節細部的分析損毀原因，其中特別值得注意的是對於電力系統中斷的原因及影響進行了結構性的說明，台灣北部為高科技工業密集地，如果此次震央再往北一點，可能會影響到全球高科技產業的供應，因此早些年已有電力中斷的應對措施，同時為了預防台灣南北兩地電力不平衡，建立了南北電力傳輸線，提供災難發生時，電力可以南電北送或北電南送。最後作者認為經過集集大地震後收集的資料，成為台灣地震歷史中最完整的紀錄，經過此次地震，台灣在五年內建置了超過 1000 個地震記錄儀器，資料也比以往震度七級以上的資料多五倍，這些資料對以後建築防震的標準有相當大的幫助。
- Improving Disaster Management [27]：作者認為通訊是災害發生時很重要的一環，但當基礎建通信設損毀或電話網路超過負載時，必須制定一套有效的災害管理機制。但本篇研究學者的研究方向與傳統現場工作人員認為的救災工具存在很大的差異，研究學者認為資訊系統雖然可以處理大量的數據，評估災難現象以輔助救災，但目前的問題在於資訊的傳達而不是在技術上，學者們開發出一套利用日常個人通訊設備當作節點傳遞災情訊息的系統，此系

統可以有效的在災害發生時迅速的將災情透過語音或文字廣播出去減少其他傳達方式的傳遞耽擱。

- A Disaster Information System by Ballooned Wireless Adhoc Network [25]：本篇主要在探討當災難發生時通訊網路、交通中斷時，如何建構一個可靠、穩定的網路環境，研究團隊利用輕氣球(距地面 40m~80m)在空中建立一個即時的隨意式網路(adhoc network)以供訊息傳遞，連線拓樸方式為最小生成樹，並透過一固定伺服器(WIDIS)與外部網路相連，與許多研究不同的是此系統(Wide area Disaster information Network, WDN)已被實做出來，而不是一假設系統。

### 2.2.8、評論

從表 3 中可以發現，不管是哪種緊急通訊系統都會有些缺陷，以最符合要求的 Wakie-Tailie 來說，在很多地區(例如：美國)各項要求都符合了災區的通訊要求，但對於台灣來說普及率低就不符合災區通訊所要求的條件，其他幾種通訊系統都有類似一樣的問題，無法支援大量志願救災人員及災民的通訊需求。而近年來的研究主要偏向於透過隨意網路或者異質網路來建構一個新的網路達到訊息傳輸的目的，此些研究都面臨一些客觀環境上的問題無法解決，最明顯例子為需事先建構，包含了需安裝軟體到終端設備上或需要伺服器來儲存和運算資料，對一個無法預知何時會發生的突發狀況而言這些條件在災害來臨時是否能順利克服大有疑問，而以系統面來討論要建構出一個網路必須要有足夠多的節點來構成拓樸，在災區很難達到足以進行傳輸的節點數量或密度，我們曾實際透過 W-Fi 進行 VoIP 實驗，當距離超過 25m 時封包將會大量遺失，無法進行有效的通訊，因此這種系統所能發揮的能效還是非常有限，我們所提出的 CCN 應急蜂巢式行動網路可以更有效的支援大規模的災區內行動電話用戶。

表 3、緊急通訊系統比較

	使用成本	建構難易度	進入災區難易度	終端設備普及率	終端設備操作難易度	終端設備可移動性	通訊品質	系統運轉難易度
Walkie-Talkie	低	不需建構	易	視地區而定	需簡單學習	高	中	無
Amateur Radio	中	需專業人士架構	易	低	需專業職照	低	中	無
行動衛星通訊	高	既存	易	低	易	高	中	無
集群通訊系統	高	簡單	難(需地面交通系統支援)	低	需簡單學習	高	高	中
移動式基地台	高	簡單	難(需地面交通系統支援)	高	易	高	高	中
MANET	低	需專業安裝設定軟體	就地取材	中	易	中	低	中
CCN 應急通訊系統	低	中	重量輕可空投	高	易	高	高	中

## 2.3、應急蜂巢式通訊系統 (CCN)

### 2.3.1、系統架構

我們先前的研究[17,18,32]發覺大部分斷訊基地台之結構完整，但因停斷電或後端線路毀損使其無法提供正常服務。因此提出利用空投或直昇機等方式提供緊急修復包(可儲備於國家防救災中心，或行動電話公司)，修復包內含發電機、燃油、無線通訊...等設備，借由此些基本設備，使基地台能維持基本運轉，基地台再利用無線通訊設備以跳接方式互連，回復與核心網路之間的連線，使其能連上交換



機等設備，恢復部分通訊功能。

應急蜂巢式行動網路(Contingency Cellular Network)即是我們提出之利用公眾行動通訊網路建構的應急蜂巢式通訊系統，此系統依附於原有的行動通訊網路，其重要性隨著原有基地台的修復而慢慢下降，基地台依對外連線能力分為兩類：

- **連網台(Connected station)**：功能完整未受損，與後端核心網路(機房)連線正常，可持續提供服務之基地台稱為連網台。
- **孤立台(Isolated station)**：未受損，但因為電源或者是與後端核心網路連線中斷，無法提供服務之基地台。

我們又定義**鄰台**為相鄰兩基地台之間彼此可以透過無線方式相連，互相交換資料，滿足這條件的基地台互為彼此的鄰台。孤立台利用鄰台連結到連網台，由於連網台的功能沒有受損，即可透過連網台連線到後端核心網路，在 2G 為 BSC 及 MSC(Mobile Switching Center) 3G 為 RNC(Radio Network Controller)及 MSC、SGSN，基地台必須與他們建立連線才可交換信令(signaling)與資料(data)提供電信服務，因此 CCN 需要利用各種無線連線方式[22]取代已經斷掉的固網連線與鄰台相連，鄰台之間將會不斷相連擴展，形成一個全新的網路。

### 2.3.2、系統使用時機

在災害初期原有通訊系統由於災害發生而無法使用，但災害初期是受困人員存活率最高的時候，而具有專業通訊設備的外援卻無法在第一時間進入，要突破救援瓶頸就必須依靠當地有行動能力的倖存者互相幫忙，這些倖存者並沒有接受過專業的訓練也沒合適的通訊設備將災情傳達出去尋求幫助或者互相協調組織救援，CCN 應急通訊系統在此時能提供給這些倖存者通訊的功能，使用原有手機不需要訓練即可快速上手使用，但由於此系統是依附原有通訊設備，隨著時間的推移電信公司將會逐步修復原有基地台，此系統的作用也會慢慢下降直到所有基地台

修復完成，如圖 9 所示。

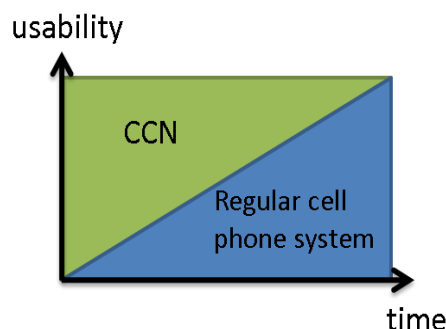


圖 9、CCN 使用時機

### 2.3.3、系統優勢

- 重覆使用原有行動通訊基地台，大幅降低成本且涵蓋範圍廣。且因原有之地理拓樸均經精心設計可直接使用，不需耗時費力在緊急中進行複雜的地理拓樸設計，且都位於高處極易以無線電互相連接，避免因高樓大廈阻擋無線電的连接。
- 手機普及率非常高，不需改裝即可投入使用，因此不需投入大量金錢購置使用端設備，且使用者不需訓練，只要會使用原有手機即可使用。
- 這些基地台都已在災區內，不會因為交通因素導致設備因道路、橋梁的中斷無法進入災區，延誤建構緊急通訊系統的時間。
- 額外設備(緊急修復包)重量極輕，可以用直升機空運或空投。

### 2.3.4、系統元件

CCN 之建構需要在各孤立台附加額外設備，平時包裝成「緊急修復包」儲存於國家救災單位，於災害發生時運送至災區建構 CCN。

**緊急修復包內含：**

- **Power Module**：災區因為災害的發生，基礎建設遭到毀損，無法提供災區基地台運轉所需的電源，雖然基地台備有備用電源，但一般的基地台備用電源僅能維持一至數小時，不足以支撐到災區電力系統復原，台灣在 2009 年發生的八八水災中 3300 座斷訊的基地台中約 70% 就是因為電力中斷而斷訊，因此提供能讓基地台持續運轉的小型輕便發電機為很重要的元件之一。
- **Inter-Cell Communication Module (ICC Module)**：孤立台為了要與連網台相連，必須透過與鄰台的多重跳接方式相連，其中與鄰台的連線即是利用 ICC Module，它必須具有無線連通能力(例如：WiMax、WiFi)，借此與鄰台相連進而恢復聯外能力，最後與後端核心網路相連。
- **Satellite Communications Module**：衛星通訊設備，如果災區內沒有連網台則須利用衛星通訊設備架設至少一個連網台，此外如果災區面積太大，離連網台太遠的孤立台所傳送的資料須經多次轉送，為了避免多次轉送佔用太多頻寬，可以考慮挑選某些孤立台加裝衛星通訊設備直接把孤立台建構成連網台，將一個龐大負擔過重的 CCN 網路，分解成數個負擔較輕的 CCN 網路。由於衛星通訊設備價格昂貴，無法普及所有基地台，只能用於少數基地台。
- **Emulated Controller Module (EC Module)**：負責處理、運算、轉送資料，當孤立台透過 ICC Module 與鄰台相連時，由此模組來控制包含資料轉送路徑、基地台頻寬分配控制、通話允入控制、Admission control ... 等應急通訊系統功能；同時也是應急通訊系統與原先受損公眾通訊系統之間的介面，藉由此模組互相傳輸控制信令及語音。

### 2.3.5、重要技術問題探討

為提供應急通訊所需之功能，如允入控制、Intranet 建構、自動化建構等功能，需改變現有基地台運作流程才可實現。而改變方法可分為二種：嵌入式與外接式。

現有行動電話系統，如 GSM 2G 和 3G，並未將應急通訊納入系統設計考量，如採用嵌入式的作法，成本和技術門檻高，難以說服行動通訊業者採納。反觀外接式的成本和實作技術則遠低於嵌入式，行動通訊業者不須大幅變更系統即可配合，因此配合意願較高，較符合應急通訊系統的需求，避免業者之抵制。

但新一代的系統，如 LTE，已將應急通訊納入系統設計的考量，大幅降低使用嵌入式所需的成本和技術。因此若能先以外接式的方式提供救災通訊所需基本的功能；針對新一代的基地台，部分進階功能則可用嵌入式的方式來實作。二種方式互相搭配運用，不但可滿足救災通訊所需的功能，又可有效控制實作的成本和技術門檻，增加 CCN 網路實現之可能性。

- **嵌入式：**將應急通訊所需之功能嵌入基地台中，此種方式需修改基地台的軟硬體。好處是災害來臨時，可立刻啟動運轉，在災害發生之初期立即發揮功能支援搶救工作。而且系統運轉較具效率，可提供的功能多且較具彈性，適合用來實作複雜度較高的功能。缺點為需大量修改現存基地台，大幅增加建構成本，且新功能嵌入的實作方法隨基地台製造廠商或型號不同而異，實現難度頗高。
- **外接式：**推持現有基地台運作機制，藉由外接的設備(例如，EC Module)來提供新功能。好處是設備數量大幅減少，不需更改現有基地台，同一外接設備可適用於不同的系統。缺點是必須在緊急時運送至災區建構，耽誤災區的搶救工作，且僅能提供較簡單的基本功能。

#### **建構 CCN 需克服的相關議題如下：**

- **整體網路拓樸規劃：**此為首要議題。因多數的基地台無法直接連上後端網路，需透過多重跳接的方式連上。網路拓樸的規劃將決定整個網路的效能、救災效益和穩定度。救災效益需考量的因素有災區緊急或受災輕重程度、投入救災人員的數量等。而設計策略需將救災效益最大化並維持流量分配增加其穩

定度。分流的設計是依照各區受災程度[10,28]估算可能的通話量並加以分流，避免網路擁塞以提高整體網路使用率。如此便可在有限的資源下提供最大的通話量並可避免網路擁塞和維持網路的穩定。

- **建構排程**：由於災區的交通系統可能癱瘓，因此 CCN 的建構需視各種運輸能力而規劃，且各個地方受災情況不盡相同，CCN 的建構順序也須辨明輕重緩急依序建構，如何在網路拓樸已知情況下決定 CCN 構排程以達到最大的救災效益，即為排程意義所在，於第三節詳細敘述。
- **網路頻寬分配**：在孤立台提供服務時，因資料需透過鄰台轉送，各節點所需頻寬包含發自本基地台的流量以及為鄰台轉送的流量。每個基地台可接受的使用者數量需作合理分配，讓每個基地台分配到的通訊頻道數量符合救災通訊需求，避免分配失衡。
- **Intranet 建構**：在現有的行動通訊架構下，兩部手機通話的語音封包傳送路徑均需繞送至後端核心網路的交換機進行連接，即使這兩隻手機位於同一基地台的服務範圍內也是如此。災區內所有的通話，包含災區內部通話，均需佔用到連外的頻道，而 CCN 網路對外的頻道數極為有限。每個基地台分配到的頻道數遠低於原先所能支援的頻道，因此會剩下大量的閒置無線電頻道無法使用，殊為可惜。而救災通訊裡通話數需求最大且最為重要的是災區內部通訊。救災人員之間常會有多而頻繁的通話行為。因此若建立 Intranet 的通道，讓同屬同一基地台的手機之間可以利用閒置無線電頻道彼此互通，而不需佔用聯外頻寬將可大幅提升救災效率。例如，讓手機撥 118 即可轉換成無線對講機 (Walkie-Talkie)。對救災人員之間的溝通協調將有極大的幫助。
- **基地台介面整合**：CCN 網路運用原有的基地台，使用 EC Module 和基地台介接與手機連線。因此這些外來設備必須無縫的與基地台介接，包括各種通訊協定，以確保基地台可恢復運作。現有行動網路基地台來自不同廠商且軟

硬體更新頻繁。EC Module 在基地台更新之後，若沒有跟著更新可能變得無法使用。因屬應急備用器材，很難像基地台一樣，擁有豐富的經費和資源一再更版。若能於現有協定加入緊急介接介面，以確保相關應急通訊器材的可用性，是極重要的。

- **允入控制政策制定：**允入控制的作用有二：一是避免系統因接受過多的服務請求，造成資源耗盡，以確保系統穩定性；二是安排優先次序，確保重要性高的請求可以優先使用。災區通話量為平時的數倍且應急通訊系統容量不及於平時的公眾網路，允入控制可保護系統免於被大量的通話請求癱瘓，以維持系統的穩定度，在 2005 年時倫敦地鐵車站爆炸事件，原是屬於小區域局部傷亡事件，原不應該造成通訊系統癱瘓，但爆炸發生後，公眾電信網路話務量急遽增加，大量使用者同時撥著電話確認親友平安，且佔線時間過長，影響到倫敦地區的通訊網路運作和政府搶救工作執行。此外，災區通話種類繁多且重要性不等，從緊急救助到互道關懷電話都有。允入控制可確保緊急救助電話優先使用系統，避免頻寬被重要程度較低的通話佔滿，更有利於災區救援的進行，因此具有優先等級控管能力的允入政策是一個主要研究議題。
- **自動化建構：**救災分秒必爭，若能事先規劃並架設好 EC Module 和無線連線設備，減少通訊服務空窗期，這樣就可以爭取更多寶貴的時間。由於基地台的位置及可供連線的方向都是預知的，基地台之間的互連可以預先規劃並架設好 EC Module 和無線連線設備，位於受災風險高之區域可考慮使用嵌入式設計，當災害發生時孤立台即可啟動自動化建構程序，自動建構一個初期的 CCN，協助第一時間的救災工作及收集各種資料，直到電源耗盡，所收集到的資訊將可加速第二階段 CCN 的建構。

## 2.4、排程問題

### 2.4.1、概述

排程問題常見於傳統製造業或是高科技產業，用於對生產排程的規劃，期望可以使總加權生產完工時間最小，由於排程問題過於複雜，在現實環境下要求得最佳解須花費大量的時間，因此目前相關研究排程逐漸傾向使用啟發式演算法 (Heuristic Algorithm)，以期能快速求出與最佳解相差不遠的次佳解。

排程問題有多種不同的分類方式，若以工件處理為特質來分類，可分單操作 (Single-Operation) 排程以及多重操作 (Multiple-Operation) 排程問題，單操作排程指的是每一工件僅需經一個工作站操作便可完成，多重操作排程則是每一工件必須經數個工作站操作才可完成，而以機器的數量以及工件的路徑型態來做為依據，每種操作又可以分別再做分類：

**單操作排程**問題依工作站中處理機器的數量不同分為：

- 單一機器排程 (Single-Machine Scheduling)：僅一個工作站，而工作站中只有一部機器。
- 平行機器排程 (Parallel-Machine Scheduling)：僅一個工作站，但工作站中有超過一部以上的機器。這些機器可以同時處理且不互相影響，且每一個工件可以在任何一部機器上處理。

**多重操作排程**問題依工件處理的路徑型態不同分為：

- 流線型排程問題 (Flow-Shop Scheduling Problem)：所有工件的操作順序都是相同的，即每一個工件所需的操作機器順序是一樣的。
- 零工式排程問題 (Job-Shop Scheduling Problem)：零工式排程中，每一個工件

的操作順序都是不同的。

- 開放式排程問題(Open-Shop Scheduling Problem)：開放式排程問題與零工式排程問題的相異點在於開放式排程中的所有工件經過機器的順序是任意的。

## 2.4.2、精確解法 (Exact Algorithm)

通過嚴謹數學模型，利用數學或數據結構搜尋方式求得問題解，先將所有的資料數據化，並且轉換成符合結構的數學因子供運算式使用，再經由反覆的運算來獲得最佳解，其最大的缺點是其反覆的過程，運算因子累積越來越大，需要許多時間來運算，在運算因子基數龐大的問題中並不適用。此種解法通常在解決非 NP 問題上。

### 分支界限法 (Branch and Bound)

分枝界限法把問題的可行解展開成類似樹分枝的樣子，將原來的大問題分解為小問題(子集)，每次分支都相當於一次限制，限制解的範圍，再經由此些分枝尋找最佳解，這種方法不是盲目的窮舉搜索，而是在搜索過程中通過限制，停止對某些不可能得到最優解的子空間進一步搜索，一旦確定得到的子集中所有的可能解都不是所求問題的解，則停止對這個子集的可能解作進一步的限制，即不必產生這個節點的子節點了，這一過程一直進行到找出可行解為止，故它比窮舉法效率更高。其演算法可以分成三部份，一、子集合生成規則：如何分割所有可行解，讓其子集合會有相同界限。二、限界或剪枝規則：凡是界限超出已知可行解值那些子集不再做進一步分支。三、界限計算：對每個節點要用統一方法計算出解集合約束函數值的界限作為控制搜索方向和是否進一步生成和搜索該節點子節點的判斷根據。此種方法常用來找組合問題中的解，不同的地方是在於對整個問題可能解空間(solution space)搜索，所設計出來的演算法其時間複雜度比貪婪算法



高，但能保證求出問題的最佳解。步驟：

1. 先設定問題解為目標解的相反極值。
2. 根據分枝法則(Branching rule)，從尚未被限制的節點中選擇一個節點，並在此節點的下一階層中分為幾個新的節點。
3. 計算每一個新分枝出來的節點的上/下限值。
4. 對每一節點進行限制條件測試，判斷此解是否有可能優於目前最佳解，如果沒有則停止此枝幹的生成。
5. 判斷是否仍有尚未被限制的節點，如果有，則進行步驟二，如果已無尚未被限制的節點，則演算停止，並得到最優解。

### **動態規劃法 (Dynamic Programming)**

動態規劃法要是將一個大問題分解成多個小問題來求解，以反向工作的方式，求解路徑中連接兩點的值，動態規劃法缺乏效率，比較適合小問題和批次問題，此類方法雖然可以求得最佳解，但其求解範圍小，與分支界限法不一樣的地方在於是從小到大，而不是從大到小。

求解關於  $n$  個元素的某個最佳解，將  $i$  個要素組成子集的最佳解求出來放在一張表裡。當元素個數增加一個的時候，可以根據此表來計算最佳解是否發生變化，並建立一張新的表。重覆這一過程，直到由  $n$  個元素組成的全集為止，最終將求出關於  $n$  個元素的最佳解。亦即以空集合的最佳解為初值開始，每次增加一個元素，求得一個最佳解，直到全集為止。適用動態規劃的問題必須滿足最優化原理和無後效性。

- **最優化原理**：不論過去狀態和決策如何，對前面的決策所形成的狀態而言，餘下的決策必須構成最優策略的性質，也就是一個最優化策略的子策略總是最優的。

- **無後向性**：將各階段按照一定的次序排列好之後，對於某個給定的階段狀態，它以前各階段的狀態無法直接影響它未來的決策，而只能通過當前的這個狀態。換句話說，每個狀態都是過去歷史的一個完整總結。這就是無後向性，又稱為無後效性。

### 2.4.3、後啟發式演算法 (Meta-heuristics)

新興的最佳化演算法，這類演算法的概念經常是由觀察自然界所獲得的靈感，常能發現接近最佳解的次佳解，但沒辦法證明不會得到較壞的解。可以應用在非常廣泛的問題上，但由於通常使用亂數搜尋技巧所以並不能保證效率，一般可在合理的時間內解出答案(每次求解時間可能會不一樣)。目前常見的隨機型演算法可分為兩類，一是單粒子演算法，由一個初始解進行優化，優化完的解將成為新解再以此解進行改良，Hill-Climbing Algorithm、Taboo Search、Simulated Annealing...等，都是屬於這一類。另一類是多粒子型的演算法，由多個初始解開始進行優化，解與解之間會有一定的連繫，透過這些解的連繫，使各個新解朝某一優化方向偏移(指引)，藉此得到較優解，Genetic Algorithm、Particle Swarm Optimization 與 Ant Colony Optimization...等是屬於這一類。單粒子演算有速度快、簡單...等優點，但是對於較佳解分布廣散的情況下，很容易陷入局部最佳解，無法找到全域最佳解，而多粒子演算法，由於粒子隨機分散，相對而言容易全域最佳解，但其計算時間長，實作也較為複雜。

#### 單粒子的隨機演算法：

- **Hill-Climbing Algorithm**：只對鄰近點進行比較，且不允許向結果較差的方向移動，使得爬山演算法會落入山谷區(在最小化問題上)或山峰區(在最大化問題上)而無法跳出。
- **Taboo Search Algorithm**：選擇所有鄰近點解其中最好的一個進行進一步的搜

索，為避免搜索返回剛剛離開的局部最優點而形成循環，通過建立一個禁忌列表來限制某些移動，並透過解禁規則使其有機會跳出局部最佳。

- Simulated Annealing：模擬退火法改進爬山演算法的缺點，採用溫度調控特性，讓退火演算法有機會跳脫較差的區域找到更好的解。但當較好的區域距離目前所在地區較遠時，模擬退火演算法通常難以逆向爬升脫離較大的山谷。

#### 多粒子的隨機演算法：

- Genetic Algorithm：在初始群體的基礎上，透過交叉和變異操作來獲得新解，淘汰其餘的解，進而使解的質量不斷得到改善。
- Particle Swarm Optimization Algorithm：利用模擬鳥群覓食的社會行為所衍生的，鳥群覓食時除了根據自己本身的認知，也藉由群體間的合作溝通，知道其他更佳的覓食地點，最後漸漸所有的鳥群都會飛往最佳的覓食地點。
- Ant Colony Optimization Algorithm：模仿螞蟻覓食行為的演算法，螞蟻發現食物時會在移動中會分泌一種特殊的費洛蒙，借此通知其他螞蟻循該路徑去搬運食物，而其他螞蟻會依循著費洛蒙的多寡來決定要走的路徑為何，並遵循費洛蒙強度較高的路徑去走，此些費洛蒙會隨時間而消失，此時若有其他螞蟻也循該路徑回來則會繼續分泌費洛蒙補強，最後即會找到一最佳路徑。

#### Hill-Climbing Algorithm (HC)

單粒子隨機搜尋，單粒子型算法中最簡單的一種，執行速度很快。缺點為由於演算法只對鄰近點進行比較，且不允許向結果較差的方向移動，使得爬山演算法很容易陷入局部最佳解，因而喪失找到全域最佳解的可能。其原理為對解目前所在的位置上，進行判斷，看朝哪個方向更動會有較佳解，朝較佳解的方向移動一小步；如此反覆進行，直到任一方向都不比現在解優為止。

## Taboo Search (TS)

單粒子隨機搜尋，對於所有鄰點解進行評價並選擇其中最好的一個進行進一步的變化的變化。為了避免變化返回剛剛離開的局部最優解而形成循環，通過建立一個禁忌列表來限制某些變化。這種禁忌狀態在特定的條件(解禁規則)下也可重新激活。禁忌搜尋法就如同多數的啟發式演算法一樣，沒有所謂的最佳參數設定，必須依問題的大小和複雜性決定。禁忌搜尋法可以細分成四個組成要素[12]：

- 鄰近搜尋(neighborhood search)：鄰近搜尋是藉由目前解的組成元素彼此交換，將具有先後順序的粒子彼此進行位置交換，使得一個解改變到另一個解，在每次的改變中，搜尋最優的鄰近解作為新的解。假如新的解優於目前的最佳解，則將新解當成目前最佳解；不斷反覆此步驟直到滿足停止條件為止。
- 禁忌名單(taboo list)：禁忌名單是用來記錄過去搜尋中每次發生解改變時的變化過程，為一個限制變化的記憶架構。通常禁忌名單越大則愈有機會跳出區域最佳解，缺點為所需的記憶體空間也越大，電腦每次所需偵測時間越長。
- 解禁規則(aspiration rule)：用來刪除一個被列在禁忌名單中的變化，此方法可以排除之前為了增加求解上的效率將該解列入禁忌名單中的動作。
- 停止條件(stopping criterion)：用來終止禁忌搜尋的條件，一旦搜尋達到這些條件則停止搜尋，取搜尋終止時點之最佳排程解作為最終解。

## Simulated Annealing (SA)

單粒子隨機搜尋，為一種模擬物理學上物質結晶退火程序的一種隨機性最佳化演算法，先經過快速運算找出一個初始解，新解將從初始解的變化來生成。如果新點好於當前解則被接受；否則，它以一定的機會被接受，接受機會依賴於該解變壞的程度以及當前的溫度參數，透過溫度參數此機制使解有機會跳出局部最優解，

隨著演算法的進行，溫度被逐步降低以減小接受壞的新點機會，當達到規定的溫度後算法終止，最後固定下來的解即為最終解，模擬退火算法與初始值無關，具有漸近收斂性。新解的產生和接受可分為如下三個步驟[24]：

1. 隨機產生一個新解，再經過簡單的變化產生一個位於解空間的新解，此變化可能為部分或全部的粒子交換，並會對溫度函數有一定的影響。
2. 判斷新解是否被接受，判斷的依據是一個接受準則，如果新解好於原解則被接受，否則由溫度函數決定是否被接受。
3. 當新解被接受時，用新解代替當前解，同時修正目標函數值。在此基礎上開始下一輪變化。當新解為捨棄時，則在原解的基礎上繼續下一輪變化。

### **Genetic Algorithm (GA)**

多粒子隨機搜尋，模擬達爾文生物進化論的物競天擇的演算法，在已生成的初始群體解上，透過對多個初始解交叉和變異操作來獲得新解。在新解生成後，留下所求解問題的目標函數(所求目標)高的解來構成下一代，而其餘的解透過選擇機制被淘汰，因此解的質量將不斷改善。優點在於採用啟發式群體隨機搜索的方法，在搜索的過程中不易陷入局部最優，缺陷為局部搜索能力差容易過早收斂[6,21]。

遺傳操作是模擬生物基因遺傳的做法，在通過編碼組成初始群體解後，對兩個以上的解進行操作得到一群新的解，常見的操作方法分為交叉以及重組兩大類，並從中實做優勝劣汰的進化過程，遺傳操作可使過程中的解，一代又一代的優化，進而靠近全域最佳解，遺傳操作包括以下三個基本遺傳算子(genetic operator)：選擇(selection)、交叉(crossover)、變異(mutation)。遺傳算子操作皆為隨機變化，因此群體中個體向最優解遷移的規則是隨機的。

- 選擇(selection)：從群體解中選擇優勝的解，此判定依據是由一開始設定的

評估方式決定，不同問題會有不同評估方式。而其目的是為了把優化的解遺傳到下一代。

- 交叉(crossover)：交叉是指把兩個父代個體的部分結構加以替換重組而生成新個體的操作。
- 變異(mutation)：將解中的某些運算子隨意變動，其目的有兩個：一、當遺傳算法通過交叉算子已接近全域最佳解時，利用變異算子此種操作加速向最優解收斂。二、使遺傳算法可跳出區域最佳解，防止出現過早收斂現象。

### Particle Swarm Optimization (PSO)

多粒子隨機搜尋，以粒子的移動來模擬鳥群覓食的社會行為所衍生的，鳥群覓食時除了根據自己本身的認知，飛向曾去過的較佳覓食地點，也藉由群體間的合作溝通，知道其他鳥所發現的更佳的覓食地點，最後漸漸所有的鳥群都會飛往最佳的覓食地點。

將每隻鳥看成一個粒子，粒子群則為鳥群，鳥從一個地點飛到下一個地點則看成一次粒子的演化，鳥群中的最佳覓食地點相當於粒子群演化過程中出現的最佳解，而每隻鳥自己飛過的較佳覓食地點則為每個粒子曾走過的區域最佳解，最後鳥群要找到的最佳覓食地點即為粒子群要找到的全域最佳解[9,14]。

每顆粒子與每顆粒子之間形成一個網路，粒子之間訊息相互溝通，彼此分享資料，綜合粒子本身過去探索過最佳解的位置而做出移動的判斷，使每一個粒子皆往最佳解的位置移動。使用公式 1 改變個體的速率與粒子位置：

$$V_{id} = w * V_{id} + c_1 * Rand * (p_{id} - x_{id}) + c_2 * Rand * (p_{gd} - x_{id}) \dots \dots \dots (1)$$

- $V_{id}$ ：每一 Particle 在第 d 維之速度，i：Particle 之編號，d：維度
- w：Inertia Weight
- $c_1$ 、 $c_2$ ：比例常數，決定受整體族群影響比例
- Rand：介於 0 至 1 的亂數

- $p_{id}$ ：每一 Particle 到目前為止，所出現的最佳位置
- $p_{gd}$ ：所有 Particle 到目前為止，所出現的最佳位置
- $x_{id}$ ：每一 Particle 目前之所在

### Ant Colony Optimization (ACO)

多粒子隨機搜尋，演算法是模仿螞蟻覓食行為的演算法，由於螞蟻沒有視覺，因此自然中螞蟻覓食的時候，若發現食物會在移動中會分泌一種特殊的費洛蒙，借此通知其他螞蟻循該路徑去搬運食物。若遇到障礙物，螞蟻會任意去嘗試各種方向的路徑，在走過的路徑上殘留一定的費洛蒙量，而螞蟻會依循著費洛蒙的多寡來決定要走的路徑為何，並遵循費洛蒙強度較高的路徑去走，此些費洛蒙會隨時間而消失，此時若有其他螞蟻也循該路徑回來則會繼續分泌費洛蒙補強之。此些費洛蒙的強弱會成為食物的良好指標，讓螞蟻得以搬回最多的食物。但如果搜尋的螞蟻數量少，會產生兩種可能，一是要花很多時間才能找到目標，或者因費洛蒙消失的影響，使得此路徑並未被採用到，因此較多的螞蟻比較有機會探索到較佳的值。

#### 2.4.4、評論

精確解法與後啟發式演算法對於本研究的問題有一個共同的缺陷，即對於最終解皆是利用的搜尋方式尋找出來，儘管利用了各種方法減少搜尋個數和避免區域最佳解，仍然無法滿足本研究要求能快速得出相近於最佳解的需求，特別是在當問題中判斷是否為最佳解的衡量值會隨時間改變時，計算量將迅速成長且可能會無法滿足演算法使用的基本條件。而在我們研究中當原先建構 CCN 排程改變後各基地台的建構時間也會隨之改變( profit 會隨著時間改變)，相當於各節點的衡量值一直在變化，在此情況下精確解中原先子問題的最佳解在經過排程的改變(問題中衡量值的改變)後，將不能保證是子問題中的最佳解，因此必須重新運算保

證子問題為最佳解。對於後啟發式演算法也會有相同問題，原先的解在經過排程的移動過後，將不能保證對生成出來的新解一定有參考價值(取決於救災效益函數)，因為排程的改變會影響到原先解的參考價值，進而影響到收斂後的最終解。後啟發式演算法能給與本問題找解的一個指引，但我們卻無法驗證此指引的存在以及正確性，在無法大量模擬的情況下更難證明此種解法的正確性，因此我們捨棄利用這幾種方法，而是利用速度快的貪婪式演算法，並配合後啟發式演算法中的精華，對於初始解進行優化。



## 第三章、CCN 建構排程

### 3.1、設計理念與目標

在建構 CCN 時由於空投或直昇機運送能力以及建置工程人員數量有限，所需資源(緊急修復包)無法一次到位，需多架次輪流運送，才可完成 CCN 通訊網路的建置，因而衍生修復排程(deployment scheduling)的議題。即如何依基地台服務地區的重要性，例如受災嚴重程度，以及傷患的人數和緊急程度(圖 10)...等等，來排定配送次序和配送路徑(基地台的建構次序)，以使效益最大化，本論文將「盡可能提高災後存活人數」定為救災效益。使用者(通常是國家救災單位)可跟據各自的考量以及收集到的災情資訊定義各基地台的效益。而修復一基地台所帶來的效益是一個時間函數，隨時間推移而改變，基地台越早修復投入救災，效益就越高。我們假設 CCN 的拓樸在排程之前已經確定，而且是一個樹狀拓樸，稱為轉送拓樸(Forwarding Tree)，因此修復排程問題可簡單描述為：在有限時間內針對給定的轉送拓樸，找出一組基地台修復順序，使所有基地台效益總和最大化。

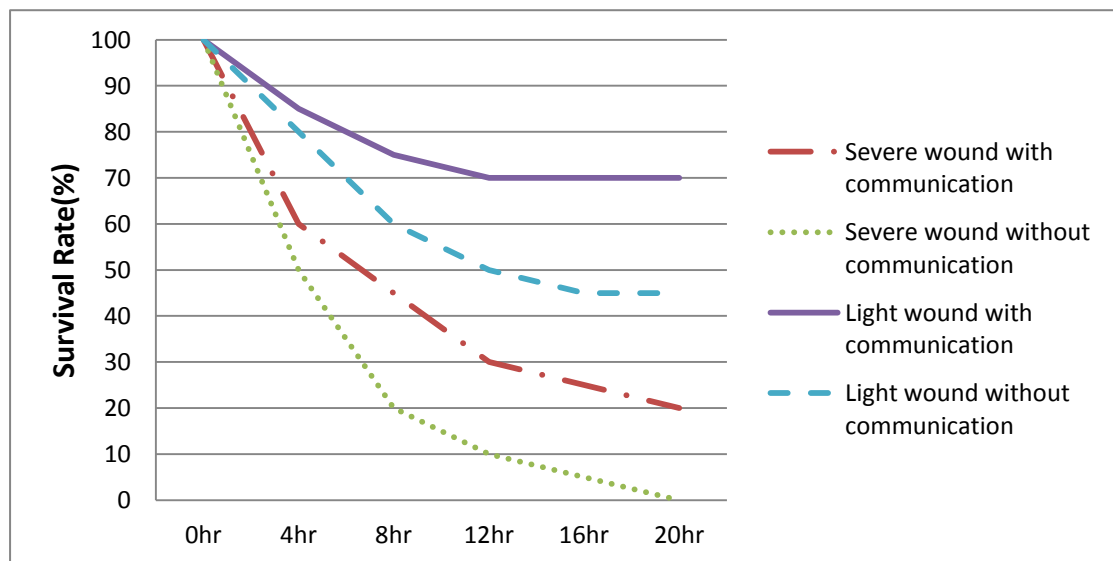


圖 10、不同環境的存活率示意圖

## 3.2、問題定義

### 3.2.1、救災效益定義

當一個孤立台建構完成加入 CCN 應急通訊系統時所能帶給其服務範圍內人員的救災助益稱為救災效益(profit)，一個簡單的定義是「倖存者的存活率」。隨著時間增加，倖存者的存活率會跟著下降，救災效益也同步下降，但由於各個地方災情不一，倖存者所需要的協助也不大一樣，因此各基地台會有不同的救災效益函數(Profit function)。由於災區的情況通常由國家救災單位收集整合，工程人員並無能力定義救災效益函數，因此雖然救災效益函數的模型是一個重要的研究議題，但已超出我們的研究範圍，必須由國家救災單位定義。

### 3.2.2、問題描述

給予一確定 CCN 應急通訊系統網路樹狀轉送拓樸  $T$ ，此樹狀轉送拓樸內有  $i$  個節點( $v_i$ )， $i-1$  條鏈結( $e_{i,j}$ )， $e_{i,j}$ 代表節點 $v_i, v_j$ 的鏈結， $s_k$ 則是代表建構排程，由於此問題模型中所有節點所需的建構時間都不一樣，因此用 $r_i$ 代表建構此節點所需花費的時間，問題模型一時同時間只會有一組工作組進行建構 CCN 應急通訊系統，問題模型二時同時間會有多組工作組進行建構 CCN 應急通訊系統，借由增加工作組組數來模擬現實情況。

在問題描述中所使用到的名詞定義如下：

- Schedule：CCN 應急通訊系統建構順序。
- 節點：基地台在本章節的數學代號。
- 轉送拓樸 (Forwarding Tree):CCN 應急通訊系統中串聯各節點之拓樸連線。
- 候選節點:所有尚未排入建構 CCN 應急通訊系統排程孤立台中的合法(可行)

選擇，合法選擇是指此候選節點的父節點已被排入排程中，一但此節點(孤立台)建構完成即能提供服務產生救災效益。

- 問題模型一：

- Given a forwarding tree,  $T(V,E)$ , where
  - $V = \{v_i | i=1, \dots, n\}$  is the set of survival base stations.
  - $v_0$  is the root node that has an external link(連網台)
  - $E = \{e_{i,j} | v_i, v_j \in V, e_{i,j}$  is the link of  $(v_i, v_j)\}$
  - $R = \{r_i | i=1, \dots, n, r_i \in Z^+\}$ ,  $t_i$  is the construction time of isolated station  $v_i$ .
  - $\pi = \{\pi^i | (\pi(1), \dots, \pi(i)), i = 1, \dots, n\}$ ,  $\pi$  is a CCN construction schedule.
  - $t = \{\sum_{k \in \pi^i} r_{\pi(k)} | i = 1, \dots, n, t \in Z^+\}$
  - $P = \{p_i(t) | i=1, \dots, n, t \in Z^+\}$ ,  $p_i(t)$  is the profit of  $v_i$ , if node  $v_i$  is constructed at time  $t$ .
  - $S = \{s_k | s_k$  is a permutation of  $v_i, i=1, \dots, n, k=1, \dots, n!\}$  is the set of CCN construction schedules.
  - $q(s_k)$  is the total profit of schedule  $s_k$
  - The CCN deployment scheduling problem is to find  $s_k \in S$ , such that  $q(s_k)$  is maximized

- 問題模型二：

- Given forwarding a tree,  $T(V,E)$ , where
  - $V = \{v_i | i=1, \dots, n\}$  is the set of survival base stations
  - $v_0$  is the root node that has an external link(連網台)
  - $E = \{e_{i,j} | v_i, v_j \in V, e_{i,j}$  is the link of  $(v_i, v_j)\}$
  - $R = \{r_i | i=1, \dots, n, r_i \in Z^+\}$ ,  $t_i$  is the construction time of isolated station  $v_i$ .
  - $\pi = \{\pi^i | (\pi(1), \dots, \pi(i)), i = 1, \dots, n\}$ ,  $\pi$  a CCN construction schedule.
  - $t = \{\sum_{k \in \pi^i} r_{\pi(k)} | i = 1, \dots, n, t \in Z^+\}$

- $P = \{p_i(t) \mid i=1, \dots, n, t \in Z^+\}$ ,  $p_i(t)$  is the profit of  $v_i$ , if node  $v_i$  is constructed at time  $t$ .
- $n \in Z^+$ : the number of CCN construction working team
- $S = \{s_k \mid s_k \text{ is a permutation of } v_i, i=1, \dots, n, k=1, \dots, n!\}$  is the set of CCN construction schedules.
- $q(s_k)$  is the total profit of schedule  $s_k$
- The CCN deployment scheduling problem is to find  $s_k \in S$ , such that  $q(s_k)$  is maximized

### 3.3、複雜度分析

在這一節中，我們將對單機排程問題(Single machine scheduling problem)和應急通訊網路排程問題(CCN deployment scheduling problem)做問題分析，並證明應急通訊網路排程問題為一 NP-Hard 問題。

#### Single machine scheduling problem (SMS) :

The non-preemptive single machine scheduling problems is as follows. A set  $J$  of  $n$  jobs has to be scheduled on a single machine. Each job  $i \in J$  requires uninterrupted processing for a period of length  $p_i$  and weight  $w_i$ , where  $p_i, w_i$  is positive integer. Let  $(P, W)$  denote the set of  $(p_i, w_i)$ . And the machine can handle no more than one job at a time. The goal is to find a schedule (i.e. permutation  $\pi: N \rightarrow \{1, \dots, n\}$ ) such that the sum  $\sum_{i \in J} (w_i * C_i(\pi))$ , where  $C_i(\pi)$  is the time at which job  $i$  completes in the given schedule  $\pi$ , is minimized.

#### CCN Deployment Scheduling problem (CCN-DS) :

The non-preemptive CCN deployment scheduling problem is as follows. A set  $N$  of  $n$  nodes organized in a tree structure has to be fixed by work teams. A parent node must

be fixed before its child nodes. Each node  $i \in N$  requires uninterrupted deployment processing for a period of length  $r_i \in R$  and weight  $w_i \in W$ , where  $r_i \cdot w_i$  is positive integer. And the work team can handle no more than one job at a time. The goal is to find a schedule such that the sum  $\sum_{i \in J}(w_i / C_i(\pi))$ , where  $C_i(\pi)$  is the time at which node  $i$  completes in the given schedule  $\pi$ , is maximized.

### **CCN-DS is in NP :**

We first show that  $CCN-DS \in NP$ . Assuming that we are given a forwarding tree  $T(V,E)$ , as well as a schedule, we can use a double loop to verify that a parent node must be fixed before its child nodes in  $T$ . The verification algorithm can affirm the schedule is a valid CCN-DS schedule within  $O(n^2)$  time.

### **CCN-DS is NP-Hard :**

We now prove that CCN-DS problem can be reduced to SMS problem straightforwardly. Given an instance  $A:[J,P,W]$  in SMS, we can find an instance  $B:[N,L,W,E,n]$  in CCN-DS such that an optimal solution  $\pi_b$  for  $B$  is also an optimal solution for  $A$ . Let  $N=J$ ,  $L=P$ ,  $W'=W$ ,  $E=\{e_{root,i} | v_{root}, v_i \in V\}$ ,  $n(\text{work teams})=1$ . The verification can be performed in polynomial time. Let total weighted completion time of SMS for  $\pi$  is  $T.W.C(\pi)$ , total weighted profit of CCN-DS for  $\pi$  is  $T.W.P(\pi)$ . We prove 3 Lemmas first :

**Lemma 1:** Any valid schedule  $\pi_b$  for  $B(CCN-DS)$  is a valid solution for  $A(SMS)$ .

Proof: Any permutation of  $J$  is valid schedule for  $A$ , and  $\pi_b$  is a permutation of  $N$ , which is  $J$ . Therefore  $\pi_b$  is a valid solution for  $A$ . Q.E.D.

**Lemma 2:** Any valid schedule  $\pi_a$  for  $A$  is also a valid schedule for  $B$ .

Proof: A valid schedule  $\pi_a$  for  $A$  is a permutation of  $J$ . Since each node in  $B$  can directly connect to the root of  $B$ , therefore any permutation of  $N$ , which is  $J$ , is a

valid schedule of B. Thus  $\pi_a$  is also a valid schedule for B. Q.E.D.

**Lemma 3:** If  $T.W.C(\pi_a) < T.W.C(\pi_b)$  then  $T.W.P(\pi_a) > T.W.P(\pi_b)$

Proof: If  $\sum_{i \in J}(w_i * C_i(\pi_a)) < \sum_{i \in J}(w_i * C_i(\pi_b))$ , by Equal Division Theorem, we can get  $\sum_{i \in N}(w_i/C_i(\pi_a)) > \sum_{i \in N}(w_i/C_i(\pi_b))$ . Q.E.D.

Next, We prove by contradiction that an optimal solution  $\pi_b$  to B must be an optimal solution to A. By Lemma 1, we know  $\pi_b$  is also a valid schedule for A, whose total weight completion time is  $T.W.C(\pi_b)$ . Assume  $\pi_b$  is not an optimal schedule for A, there must be another schedule  $\pi_a$ , whose total weight completion time  $T.W.C(\pi_a)$  is smaller than  $T.W.C(\pi_b)$ . By Lemma 2,  $\pi_a$  is also a valid schedule for B, whose total profit is  $T.W.P(\pi_a)$ . By Lemma 3, we can obtain  $T.W.P(\pi_a)$  is bigger than  $T.W.P(\pi_b)$ . This contradicts to the fact that  $\pi_b$  is an optimal solution for B. Therefore,  $\pi_b$  must be an optimal solution for A. Q.E.D.

## 3.4、啟發式排程演算法

### 3.4.1、演算法設計

CCN 建構排程為一 NP Hard 問題，礙於時間急迫，當災區範圍大而須建構基地台數量多的時候，無法在限定時間內找出一組最佳解。因此，本論文提出了一個兩階段啟發式演算法可在短時間內計算出一個接近最佳解的 CCN 建構排程：(1) 使用貪婪式演算法計算得到一個初始解；(2) 給予一個調整指引，根據此指引反覆運算調整初始解的基地台建構順序逐步找出更佳的解，如圖 11 所示。一個規劃的限制條件為父節點(Parent node)修復順序需早於子節點(Child node)，以確保基地台修復後，可立即恢復正常運作，否則須等父節點基地台修復才能運作。

在使用貪婪式演算法時，考量周圍孤立台個數以及包含自己本身與周圍孤立

台的救災效益下，每一步皆選取當時總效益最高的基地台，快速找出一組解作為初始的修復排程，且由於各災區環境狀況不一，演算法裡並沒有固定的最佳考量子樹階層數和階層權重，因此在此步驟將會對不同子樹階層數和階層權重都進行計算，不同的考量都會得出一組排程，再從中選擇一組總救災效益最高的排程當做初始解。

第二階段演算法以初始的修復排程為基礎，考慮救災效益變化的速度，給予一個變化指引，根據此指引反覆運算逐步找出更佳解。如此，在限定時間內即可提供一組不錯的修復排程，並且每當災區情況發生改變時能快速的重新運算修改建構排程。

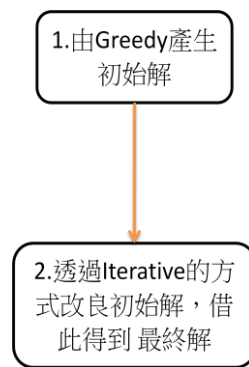


圖 11、提出之兩階段建構排程演算法

### 3.4.2、CCN-DS 演算法

各基地台 Profit 會隨著救災效益函數變化，而父節點必須在子節點建構前被建構，因此在選擇下一個被修復節點時，同時要考慮到更後面的子節點，以此為基礎 CCN-DS 第一階段貪婪演算法在選擇候選節點時，候選節點之預估 profit 為其本身以及子樹(子節點)在此時間的 profit 總和，被加總的子樹 profit 會隨著與此候選節點距離之增加而減輕比重，意即考慮周邊孤立台對此候選節點的重要性影響，子樹階層越多的候選結點其重要性越高，而子樹階層權重越小代表子樹對此候選節點影響越小，以圖 12 為例，假設距離 root 每增加 1 長度，權重即減 0.1，此

設定下節點 A 的預估 Profit： $\text{Profit}(A)=1+(7+2)*0.9+(10+3+20+15)*0.8$ 。

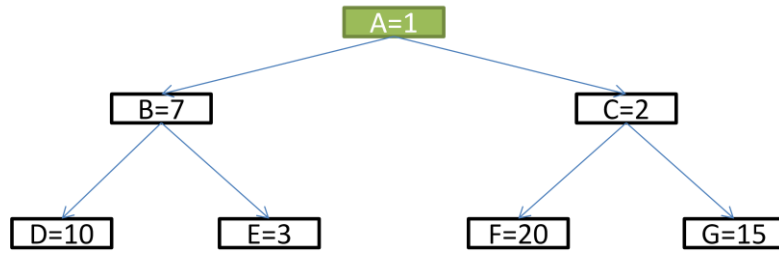


圖 12、加權 profit 計算示意圖

尋找周圍可連通的孤立台之中 profit 最高的節點當作下一修復基地台，以圖 13 為例，目前 schedule 為[A, B]，候選節點：C=20, D=5, E=12，我們將選擇候選解中加總預估 profit 最大的 C 加入 schedule 中得到新 schedule [A, B, C]。當所有孤立台排入 schedule 後，第一階段貪婪演算法結束。

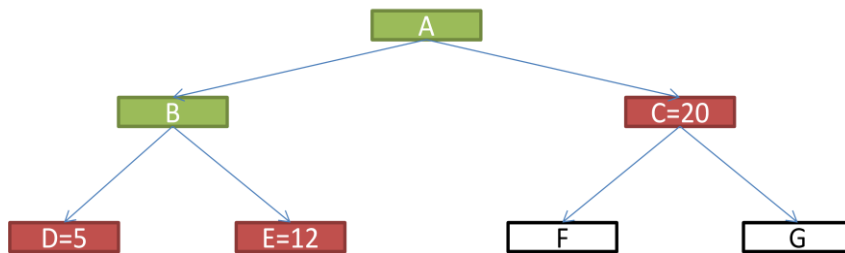


圖 13、候選節點示意圖

第二階段演算法是對每個救災效益函數給與一個優先權，考慮各孤立台對應的救災效益函數優先權並與初始 profit 兩者進行計算得出一個孤立台優先權值，以此優先權值為依據對初始 schedule 進行調整，期望得到更佳建構 CCN 排程，將於 4.2.3 節詳細介紹。

### CCN-DS 演算法流程

- 第一階 CCN-DS 段演算法中之 Greedy Algorithm 設計



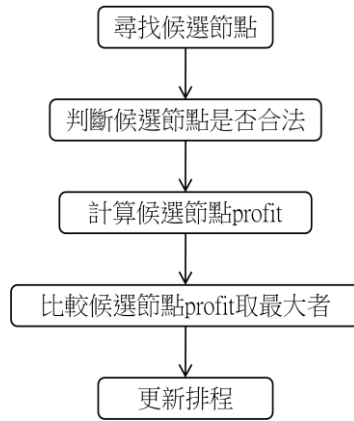


圖 14、第一階段 CCN-DS 演算法中 Greedy Algorithm 設計

。 第一階段 CCN-DS 貪婪演算法

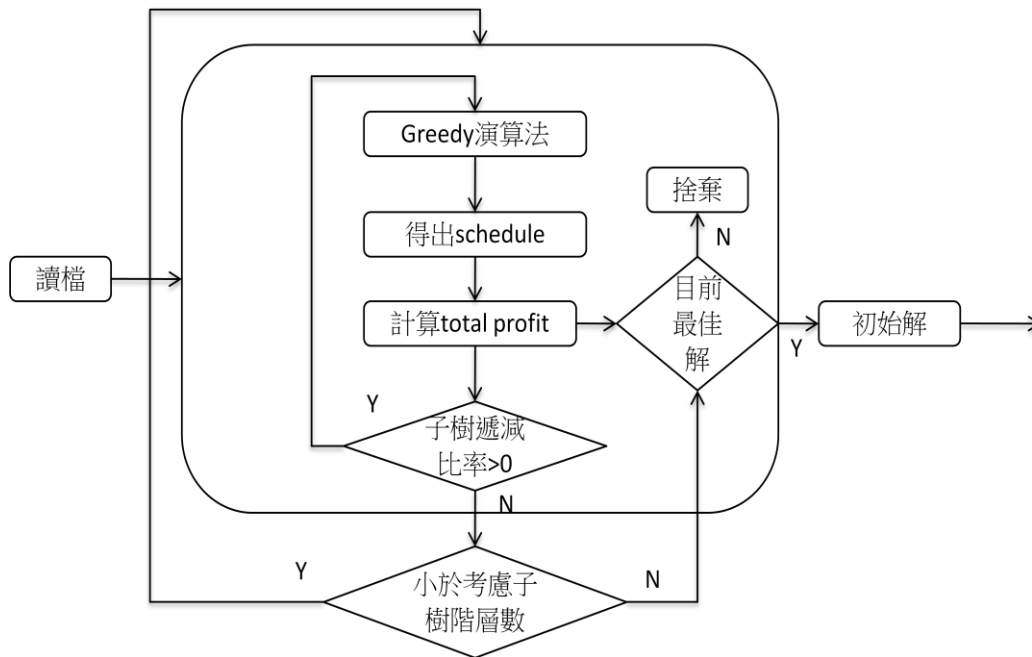


圖 15、第一階段 CCN-DS 貪婪演算法設計

。 第二階 CCN-DS 段演算法

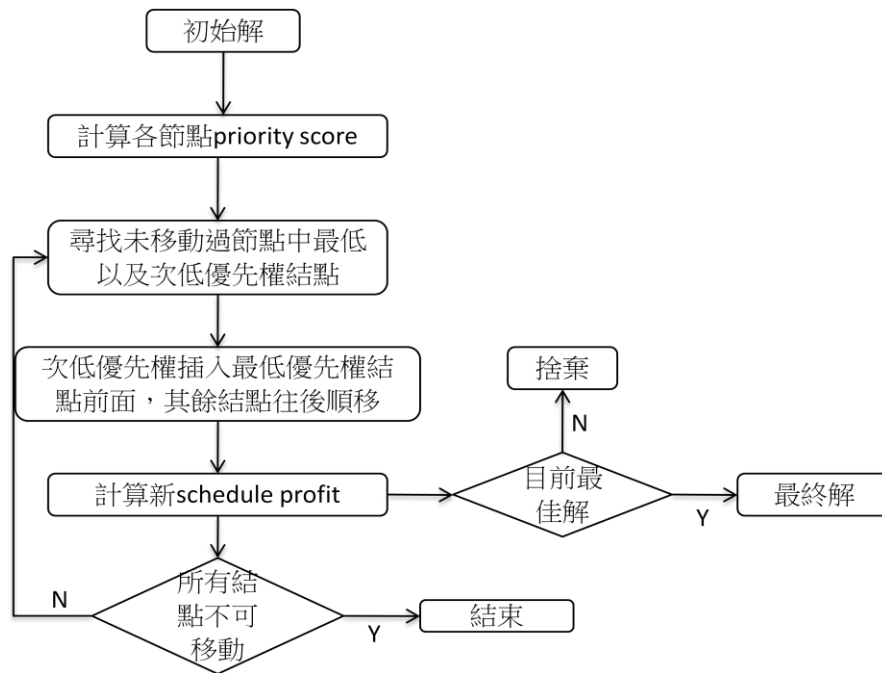


圖 16、第二階段 CCN-DS 演算法設計

CCN-DS 演算法虛擬碼 (Pseudo Code)

```

Algorithm(CCN_TreeData){
    if(countTreeLevel < considnerTreeLevel){
        if(LevelDecreasesWeight < 1){
            initialSchedule = FirstPhase(TreeData, countTreeLevel,
            LevelDecreasesWeight);
        }
    }
    finalSchedule = SecondPhase(initialSchedule);
}

FirstPhase (TreeData, countTreeLevel, LevelDecreases){
    while(NodeIsLegal && AllLegalNode!=null){
        maxProfitNode = the maximum ProfitNode;
        initialSchedule.add(maxProfitNode);
    }
}
  
```

```

countProfit(AllLegalNode){
    if(nodeLevel < countTreeLevel && nodeLevelWeight > 0){
        nodeProfit = profit(node,time) + profit(childrenNode,
            time)*childrenNodeLevelWeight;
    }
}

profit(node, time){
    profit = nodeMathFunction(time);
}

SecondPhase (initialSchedule){
    start with the lowest priority node;
    while(the current node is not updated){
        if(exists lower priority node preceding current node in the current
            Schedule){
            update current schedule(move current node forward to the place
                preceding the highest among lower priority nodes) ;
        }
    }
}

```

### 3.4.3、CCN-DS 演算法範例

我們以圖 17 到圖 24 解釋 CCN-DS 演算法，每兩張圖為一時間狀態，共有四個時間點，圖 17 為一開始狀態，假設連網台為 A，孤立台為 B、C、D、E、F、G 各孤立台值為此時間點透過救災效益函數得出之 profit，候選節點為 B、C，預估 profit 分別為 21.9、15.6，使用 CCN-DS 演算法直到時間點四(圖 24)時，Schedule 為[A, B, E, C]，候選節點：D、F、G，預估 profit 分別為 8、2、4，選 D 當下一建構節點。

1. 時間點一，目前 Schedule : [A]，候選節點 : [B,C]，如圖 17

$\text{Profit}(B)=10+(5+8)*0.9=21.7$ ， $\text{Profit}(C)=3+(10+4)*0.9=15.6$ ，選[B]，如圖

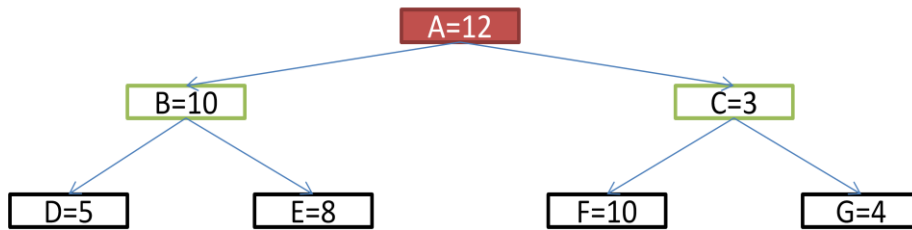


圖 17、時間點一之候選節點

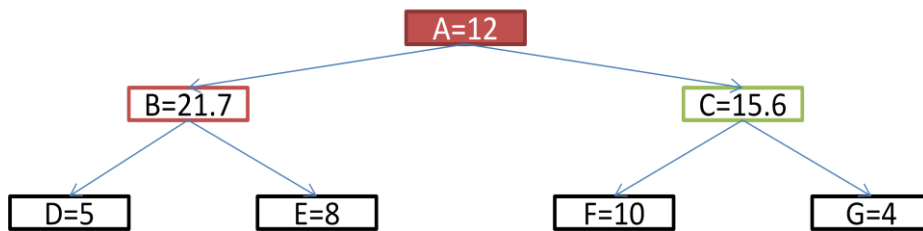


圖 18、時間點一之候選節點選擇

2. 時間點二，目前 Schedule : [A, B]，候選節點 : C、D、E，如圖 19

$\text{Profit}(C)=4+(6+4)*0.9=13$ ， $\text{Profit}(D)=7$ ， $\text{Profit}(E)=15$ ，選[E]，如圖 20

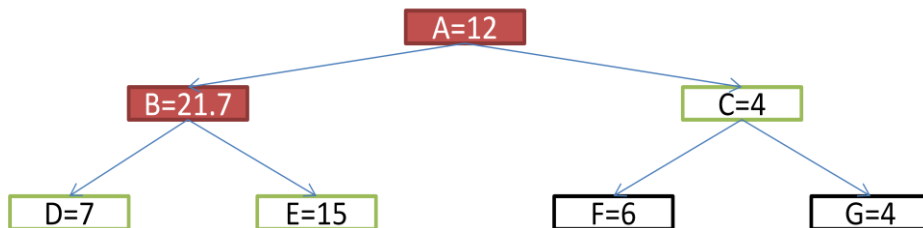


圖 19、時間點二之候選節點

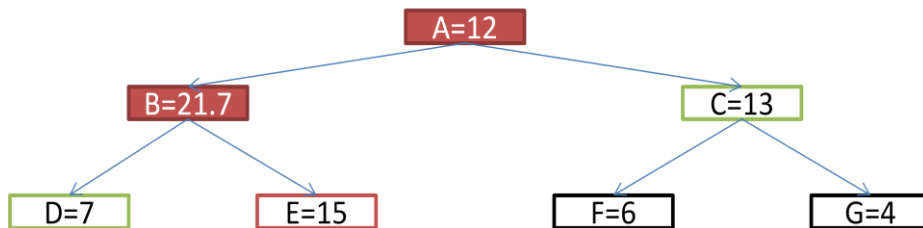


圖 20、時間點二之候選節點選擇

3. 時間點三，目前 Scheduling : [A, B, E]，候選節點 : [C、D]，如圖 21

$\text{Profit}(C)=3+(4+4)*0.9=11.2$ ， $\text{Profit}(D)=9$ ，選[C]，如圖 22

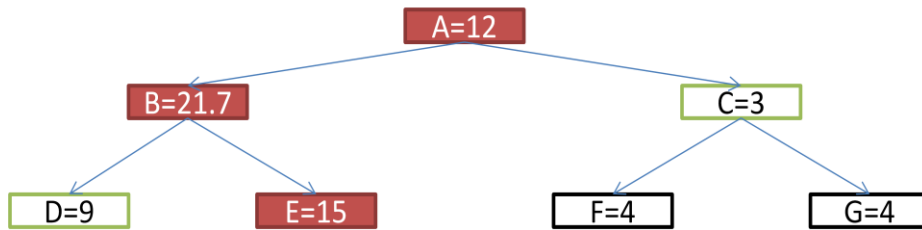


圖 21、時間點三之候選節點

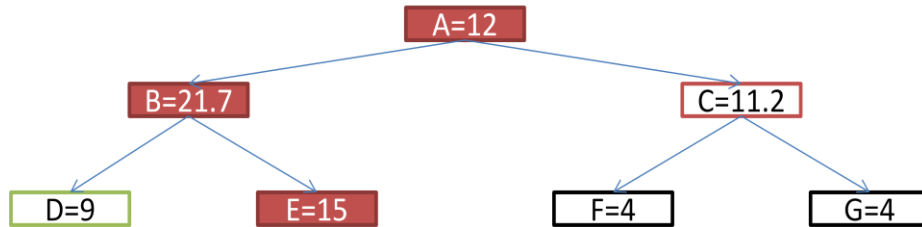


圖 22、時間點三之候選節點選擇

4. 時間點四，目前 Scheduling : [A, B, E, C]，候選節點 : [D、F、G]，如圖 23

Profit(D)=8，Profit(F)=2，Profit(G)=4，選[D]，如圖 24

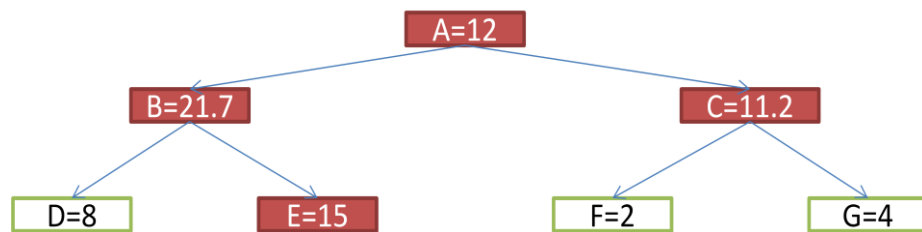


圖 23、時間點四之候選節點



圖 24、時間點四之候選節點選擇

## 第四章、效益評估

我們對所提出之 CCN-DS 演算法進行效能評估，判斷演算法解跟最佳解的差距，因本問題之複雜度，難以利用數學的方式證明與最佳解的差距，因此我們使用電腦隨機產生大量模擬案例，評估 CCN-DS 演算法的效能，觀察在不同規模之下其與最佳解之差距。

### 4.1、實驗設計

利用三階段實驗逐步逼近現實環境，在小規模案例時(基地台個數 10 - 14)我們可以找出最佳解以相比較，但基地台個數在 14 以上即無法快速找到最佳解，而以 10 萬個解的最佳解與最差解作為比較基準。

實驗一中假設建構每個基地台時所需花費的時間一樣，並且只有一組團隊在建構 CCN 應急通訊系統。實驗二則假設各基地台的建構時間不同，為實驗一的延伸。實驗三則是符合現實情況增加了建構團隊組數，有多組同時進行建構。

- 實驗一：

  - 孤立台所需要建構時間一樣

  - 建構工作組個數為 1

- 實驗二：

  - 孤立台所需要建構時間不一樣

  - 建構工作組個數為 1

- 實驗三：

  - 孤立台所需要建構時間不一樣

  - 建構工作組個數為  $n$ ， $n > 1$

- 實驗四：

實驗一到實驗三環境的大規模案例測試集中在實驗四，在實驗四中各個實驗環境的基地台個數為 10 到 14 以及 100 個，除了基地台個數為 14 時做四個獨立測試之外，其他五種則各做十個獨立測試。

#### 4.1.1、實驗環境

我們利用 JAVA 針對提出的 CCN-DS 演算法實做，將亂數產生的 Instance 進行計算，得出最佳解、最差解、第一階段貪婪演算法解及第二階段演算法解，憑以計算 CCN-DS 演算法在不同環境條件下之效能，以下是電腦軟硬體規格：

- 處理器：Intel(R) Xeon(R) CPU X3440 @ 2.53GHz 四核
- 記憶體：4.00 GB
- 系統類型：64 位元 MS Win 7
- IDE(Integrated Development Environment)：Eclipse

#### 4.1.2、評估指標

由於實驗案例是由亂數隨機產生，每次實驗皆獨立不相關，因此計算所得效能必須經過正規化(Normalization)，如公式 2 所示，稱為誤差(Deviations from optimum solution，簡稱 Deviation)。

- 誤差 = (最佳解 - 演算法解) / (最佳解 - 最差解).....(2)

### 4.2、救災效益函數

各基地台之救災效益是當該基地台恢復服務時，對所服務災區所提供的效益，而救災效益函數為隨著時間變化的救災效益(一個簡單的救災效益定義是災民的存活率)，隨著時間的推移救災效益將會越來越低，根據經驗我們知道此下降曲線並非一線性函數，但由於救災單位在緊急時無法在短時間內精確的定義救災效益函數，因此我們設計了三個簡單的兩段式線性模型，使用於我們的實驗中，這些

救災效益函數模型可以提供救災單位參考。

#### 4.2.1、救災效益函數變數

我們設計的兩階段線性模型包含四個變數：

- X：Initial Profit，初始效益
- P：Turning point，第一與第二階段效益惡化率的分類點
- S1：First deterioration rate (slope 1)，第一階段效益惡化率
- S2：Second deterioration rate (slope 2)，第二階段效益惡化率

#### 4.2.2、救災效益函數模型

- 救災效益函數模型一 (Profit Function 1)，如圖 25，模型特點是當孤立台恢復時間超過 Turning point 時不會有任何救災效益產生，反之如果在 Turning point 之前都是維持相同救災效益，這種模型常發生在功能導向的孤立台中，例如為了跟某個救援單位保持通訊，但當此單位已獲得通訊功能時搶修此孤立台不會有任何效益(超過時間將會被移除修復排程名單外)。

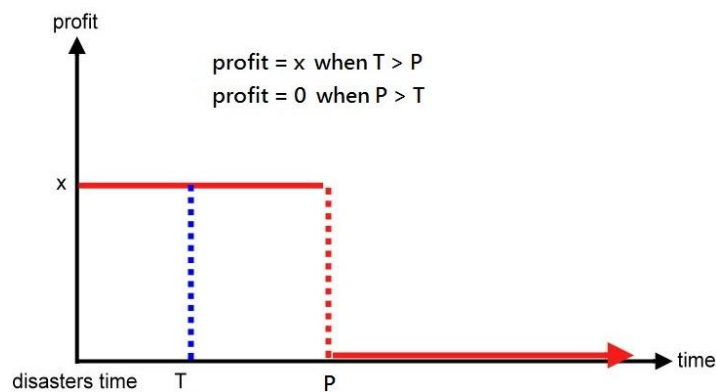


圖 25、救災效益函數模型一 (Profit function 1)

- 救災效益函數模型二 (Profit Function 2)，如圖 26，這個模型特點是救災效益從一開始就隨著時間線性下降，超過 Turning point (例如黃金 72 小時)時



即變成沒有效益可以套用的情況有：在預計時間後搜救行動可完成、或無生還機會、或有其他通訊方式可取代...等。

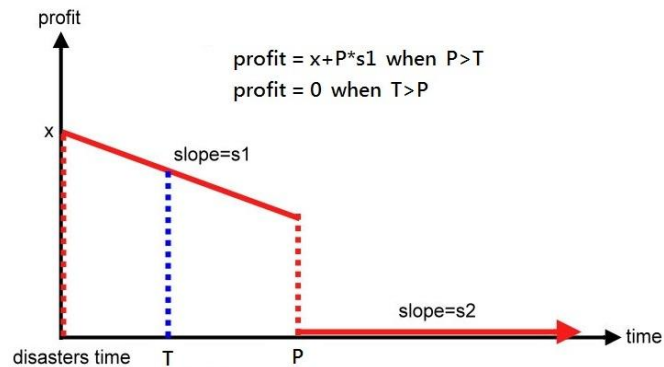


圖 26、救災效益函數模型二 (Profit function 2)

- 救災效益函數模型三 (Profit Function 3), 如圖 27, 本模型類似 Profit Function 2, 差別在於生還者在過了 Turning point 之後還有存活機會, 但會隨著時間快速下降。

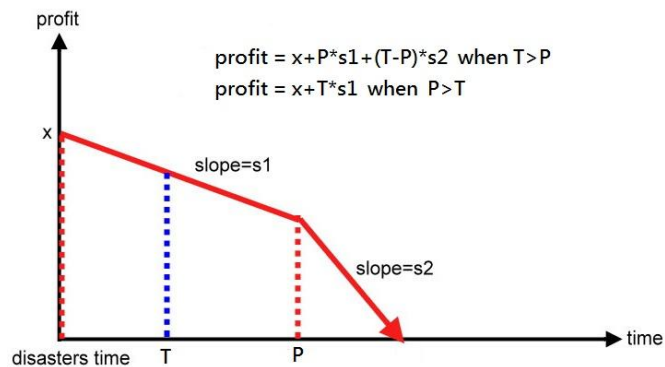


圖 27、救災效益函數模型三 (Profit function 3)

### 4.2.3、優先權設定

第二階段演算法會使用到優先權值, 其計算必須仰賴救災效益函數, 如公式 3 所示:

$$\text{Priority score} = \text{Initial Profit} * \text{Turning point score} * \text{Slope1} * \text{Slope2} * \text{Model score} \dots (3)$$

- Initial Profit: 初始孤立台救災效益。
- Turning point score: 如經過 Turning point 後的線性變化是歸零, 則 Turning

point 時間越前的孤立台必須越早被救援，因此給予較高的優先權。

- Slope1、Slope2：救災效益惡化率。
- Model score：不同的救災效益函數代表了各孤立台不同的救災效益變化方式，有些變化較迅速有些則較緩慢，因此對不同的救災效益函數模型也列入優先權考量。

範例：由優先權次低的節點 C 開始往前挪，如表 4 步驟 2 所示，高優先權節點 C 插入低優先權節點 B 前面，餘下節點 D、E 往後順移，以此類推，直到最後 priority schedule 為高到低(D,E,A,C,B)。類似於 select sort，差別在於由於已知 priority 大小，所以可以直接找出位置，插入節點前面，剩下節點往後順移，如表 4 步驟 7 所示，最小的不會一次換到最後。此設計是因為如此重組(往後順移)對順序影響小，大部分的順序不會更改，只要往前插入之節點是合法(可行解)即可得到一新的 Schedule。

表 4、第二階段 CCN-DS schedule 調整

Step	Schedule	Priority
1	(A,B,C,D,E)	(3,1,2,5,4)
2	(A,C,B,D,E)	(3,2,1,5,4)
3	(A,C,B,D,E)	(3,2,1,5,4)
4	(A,C,E,B,D)	(3,2,4,1,5)
5	(E A,C,B,D)	(4,3,2,1,5)
6	.....	.....
7	(D,E,A,C,B)	(5,4,3,2,1)

### 4.3、實驗參數

表 5、Parameters and Ranges for Test Instances

Parameters	Range of values
Simulation Time	200 hr
Initial Profit	Uniform(30, 100)
Turning point	Uniform (0, 168)
Slope1	Uniform (0, 0.5)
Slope2	Uniform (0, 0.5)
Restoration Time	Uniform (5, 15) hr
Forwarding Tree Size	10、11、12、13、14、100

### 4.4、實驗結果

本節呈現並分析實驗結果，4.4.1 節至 4.4.4 節為實驗一至實驗四的結果，而 4.5 節為綜合分析。Initial Solution 是 CCN-DS 第一階段貪婪演算法中的解，而 Final Solution 則是第二階段的解，Performance Range 為最佳解與最差解之差距。

#### 4.4.1、實驗一

表 6、7、8、9、10 為實驗一(建構各孤立台時間一樣，建構 CCN 工作組個數為 1)的實驗結果，由這些結果可以發現，在部分案例中第二階段演算法並無法改進第一階段貪婪演算法的結果，且隨著基地台數量的增加，第二階段演算法的改良會呈現微幅下降的趨勢，其原因可能為：當基地台數量增加時，受限於父節點要優先於子節點前修復條件，排程能夠變化的選擇有限，因此即使子節點的有較高優先權也無法提前建置，整體表現而言隨著基地台數量的，增加誤差值並沒有明

顯增長或趨減。第一階段 CCN-DS 貪婪演算法平均誤差值為 31.2%，而第二階段演算法可以改良到 20.2% 平均誤差值。

表 6、實驗一結果演算法誤差(10 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	628.72	602.02	26.7	615.92	618.02	40
2	550.98	507.48	43.5	530.08	538.48	28.7
3	559.49	494.69	64.8	544.45	544.45	23.2
4	639.28	570.18	69.1	607.28	617.88	30.9
5	719.69	630.65	89.04	704.79	712.89	7.6
6	595.439	562.14	33.299	578.24	592.64	8.4
7	632.73	569.25	63.48	598.63	602.13	48.2
8	568.56	510.36	58.2	549.16	549.16	33.3
9	751.219	690.53	60.689	740.73	746.93	7
10	505.91	422.02	83.89	476.83	484.13	25.9
Average						25.3

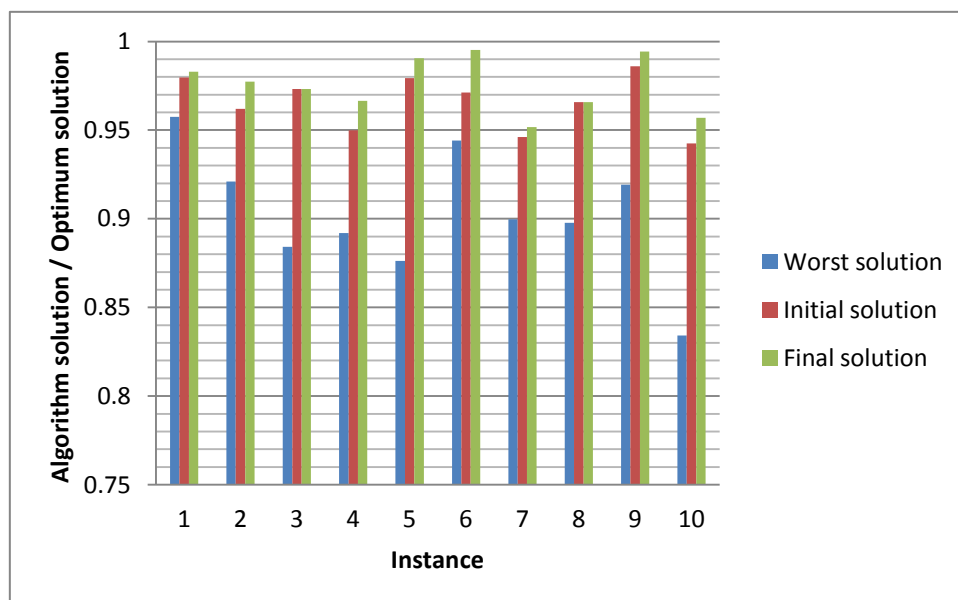


圖 28、實驗一結果演算法效能(10 個基地台)

表 7、實驗一結果演算法誤差(11 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	623.22	592.82	30.4	609.42	613.02	33.5
2	693.32	642.92	50.4	688.92	691.02	4.5
3	729.57	621.67	107.9	707.77	709.27	18.8
4	723.519	651.069	72.45	687.62	692.42	42.9
5	634.1	583.65	50.45	606.5	619.3	29.3
6	536.91	486.61	50.3	513.11	526.39	20.9
7	609.199	498.58	110.619	584.5	599.4	8.8
8	715.16	643.92	71.24	699.56	709.56	7.8
9	537.55	488.25	49.3	515.45	518.05	39.5
10	504.39	437.389	67.001	481.29	484.59	29.5
Average						23.5

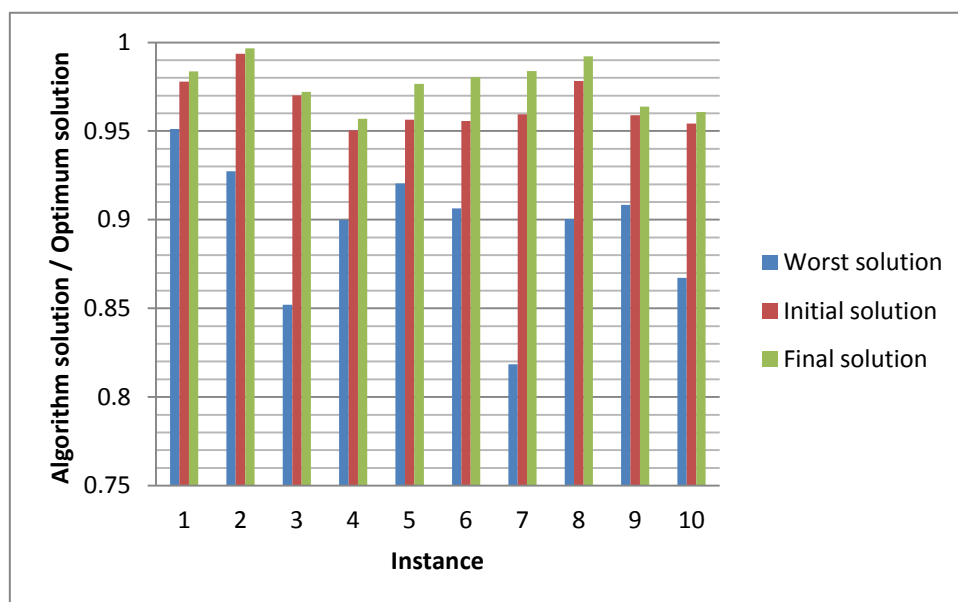


圖 29、實驗一結果演算法效能(11 個基地台)

表 8、實驗一結果演算法誤差(12 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	681.5	568.3	113.2	658	665.8	13.8
2	676.17	577.51	98.66	626.11	665.07	11.2
3	788.07	668.72	119.35	751.87	777.07	9.2
4	734.209	606.659	127.55	719.06	719.06	11.8
5	727.18	624.959	102.221	698.62	703.62	23
6	659.52	481.389	178.131	633.64	633.64	14.5
7	510.19	433.82	76.37	485.7	499.2	14.3
8	666.99	574.18	92.81	642.89	651.29	16.9
9	735.129	535.48	199.649	687.13	688.93	23.1
10	654.51	553.74	100.77	611.06	635.31	19.0
Average						15.7

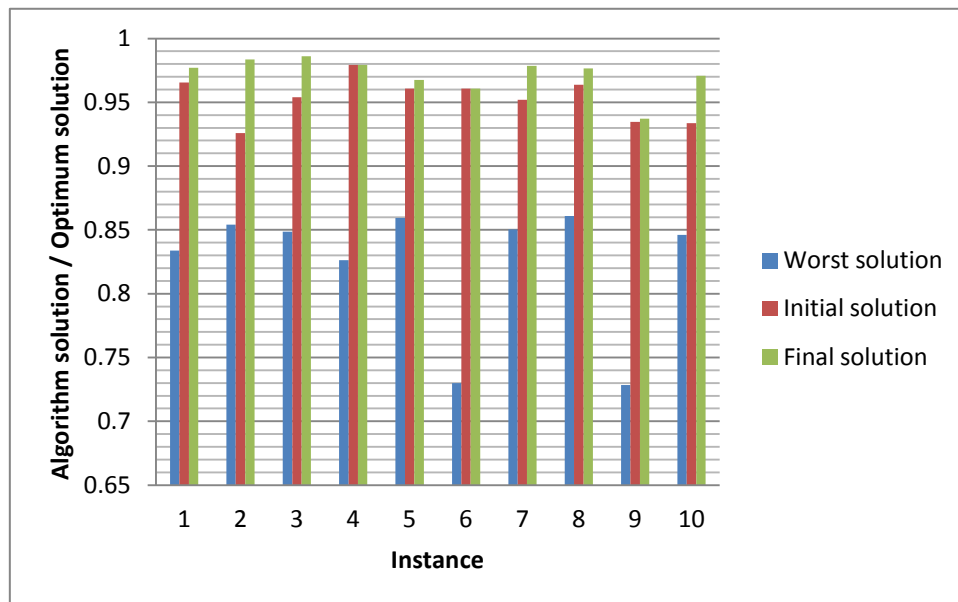


圖 30、實驗一結果演算法效能(12 個基地台)

表 9、實驗一結果演算法誤差(13 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	820.019	650.56	169.459	799.22	799.22	12.2
2	713.92	474.67	239.25	593.6	621.32	38.7
3	848.26	678.35	169.91	811.95	811.95	21.3
4	746.58	640.34	106.24	699.88	726.28	19.1
5	867.34	677.869	189.471	818.05	839.34	14.7
6	725.699	539.89	185.809	686.2	704.2	11.5
7	666.66	526.57	140.09	639.02	643.12	16.8
8	750.29	681.889	68.401	724.79	724.79	37.2
9	770.739	701.68	69.059	716.68	756.03	21.2
10	851.79	656.79	195	818.19	818.19	17.2
Average						21

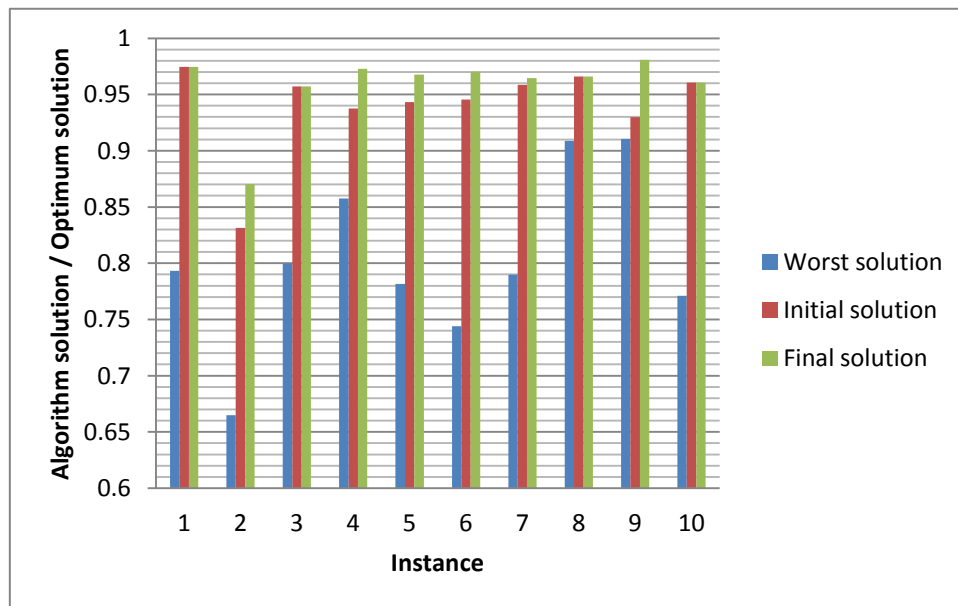


圖 31、實驗一結果演算法效能(13 個基地台)

表 10、實驗一結果演算法誤差(14 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	831.67	588.54	243.13	797.57	811.57	8.2
2	794.94	606.04	188.9	738.04	745.44	26.2
3	743.04	570.489	172.551	715.34	715.34	16
4	821.569	693.77	127.799	780.96	805.87	12.2
Average						15.7

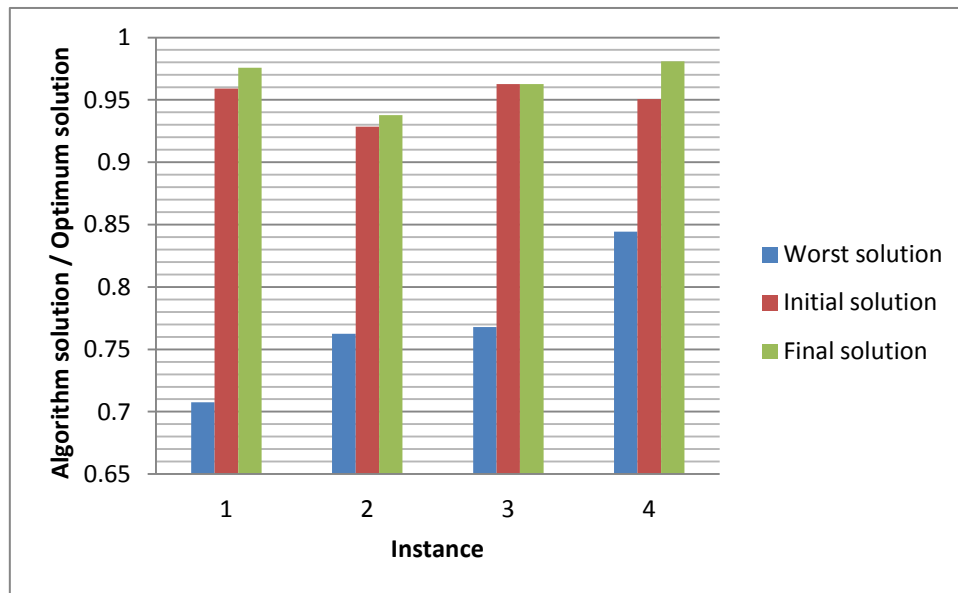


圖 32、實驗一結果演算法效能(14 個基地台)



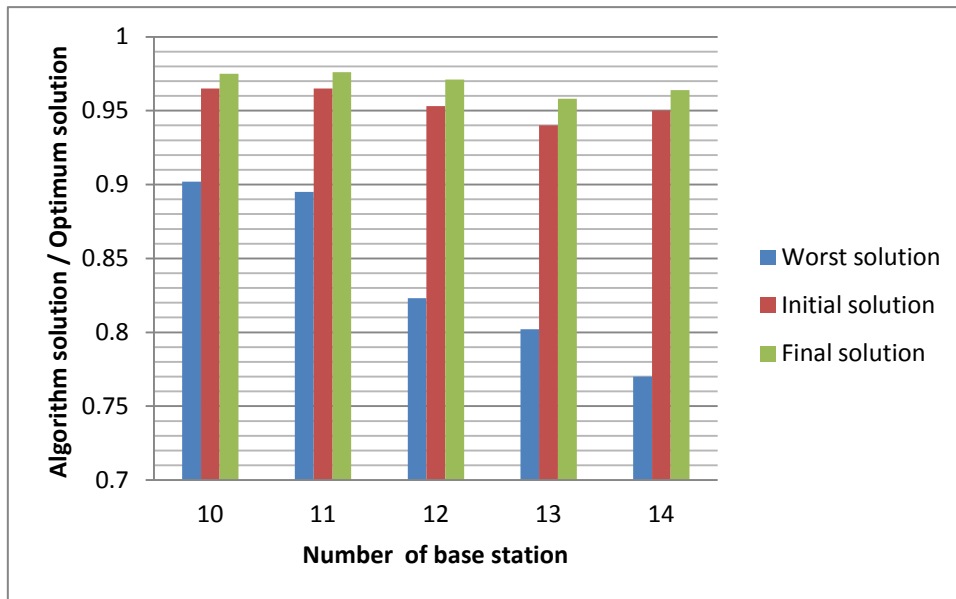


圖 33、實驗一不同基地台個數演算法平均效能比較

#### 4.4.2、實驗二

表 11、12、13、14、15 為實驗二(建構各孤立台時間不同，建構 CCN 工作組個數為 1)的實驗結果，由這些結果可以發現與實驗一的差別在於不同基地台數量之間的誤差較平均，不會有某一數量基地台表現很好或很差的情形，且相較於實驗一的環境，第二階段 CCN-DS 演算法能提供較多改善，但整體平均誤差仍高於實驗一，第二階段演算法對於第一階段貪婪演算法的改進與實驗一相同，隨著基地台數量的增加其效能也呈現下降趨勢。第一階段 CCN-DS 貪婪演算法之平均誤差值為 34.5%，第二階段演算法則改良到 23.1%，相較於實驗一，實驗二的平均誤差高約 3%。

表 11、實驗二結果演算法誤差(10 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	633.3	609.38	23.92	621.69	623.15	42.4
2	556.26	515.819	40.441	527.84	546.64	23.7
3	565.75	511.489	54.261	537.74	552.03	25.2
4	630.37	558.91	71.46	580.05	616.94	18.7
5	719.02	642.62	76.4	706.46	711.76	9.5
6	595.18	570.93	24.25	583.44	588	29.6
7	619.47	470.78	148.69	582.76	586.29	22.3
8	580.4	527.61	52.79	561.71	561.71	35.4
9	767.42	705.029	62.391	757.9	763.79	5.8
10	513.16	440.92	72.24	493.69	497.59	21.5
Average						23.4

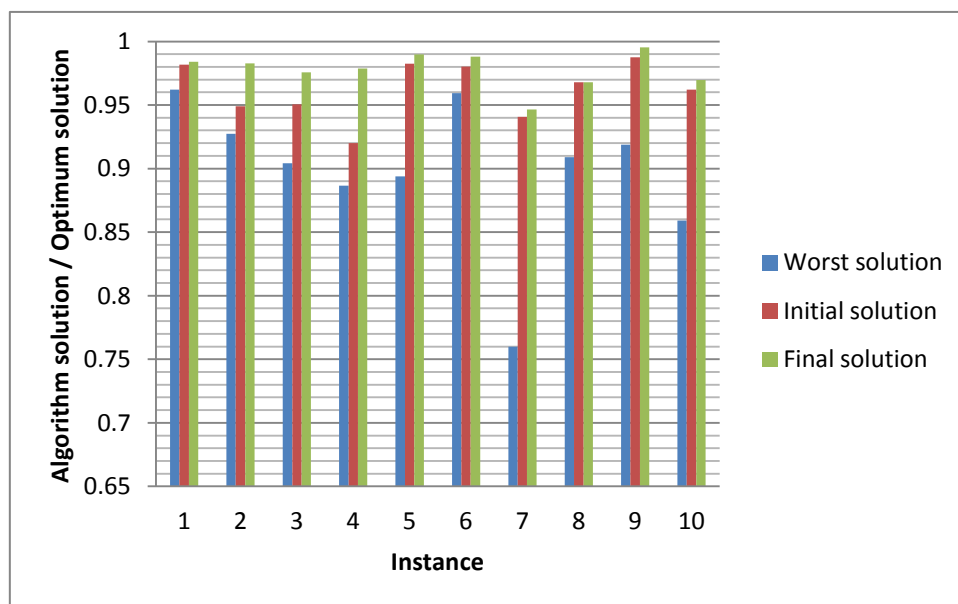


圖 34、實驗二結果演算法效能(10 個基地台)

表 12、實驗二結果演算法誤差(11 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	623.079	588.21	34.869	610.27	614.68	24
2	702.189	649	53.189	697.58	699.23	5.5
3	734.73	597.76	136.97	703.78	716.24	13.4
4	719.229	659.12	60.109	685.37	694.8	40.6
5	638.16	575.5	62.66	612.03	619.53	29.7
6	537.66	483.539	54.121	512.6	523.15	26.8
7	599.62	489.12	110.5	558.07	583.24	14.8
8	716.3	643.82	72.48	696.85	709.91	8.8
9	538.35	493.63	44.72	517.26	519.94	41.1
10	503.17	454.63	48.54	494.17	495.5	15.8
Average						22

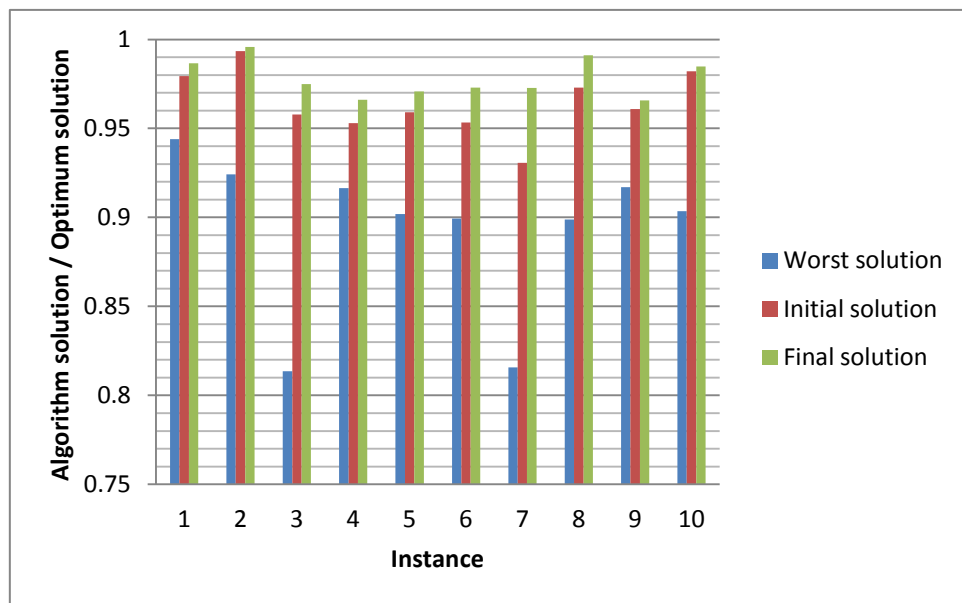


圖 35、實驗二結果演算法效能(11 個基地台)

表 13、實驗二結果演算法誤差(12 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	681.92	504.93	176.99	611.6	660.14	12.3
2	669.93	577.84	92.09	621.7	628.53	44.9
3	800.98	677.749	123.231	762.87	792.65	6.7
4	759.369	696.79	62.579	736.94	737.43	35
5	738.52	651.809	86.711	709.49	712.89	29.5
6	647.39	469.92	177.47	619.71	640.35	3.9
7	524.219	470.339	53.88	498.09	507.67	30.7
8	664.089	572.73	91.359	637.79	644.6	21.3
9	729.64	571.46	158.18	706.38	706.38	14.7
10	682.26	575.04	107.22	622.8	630.59	48.1
Average						24.7

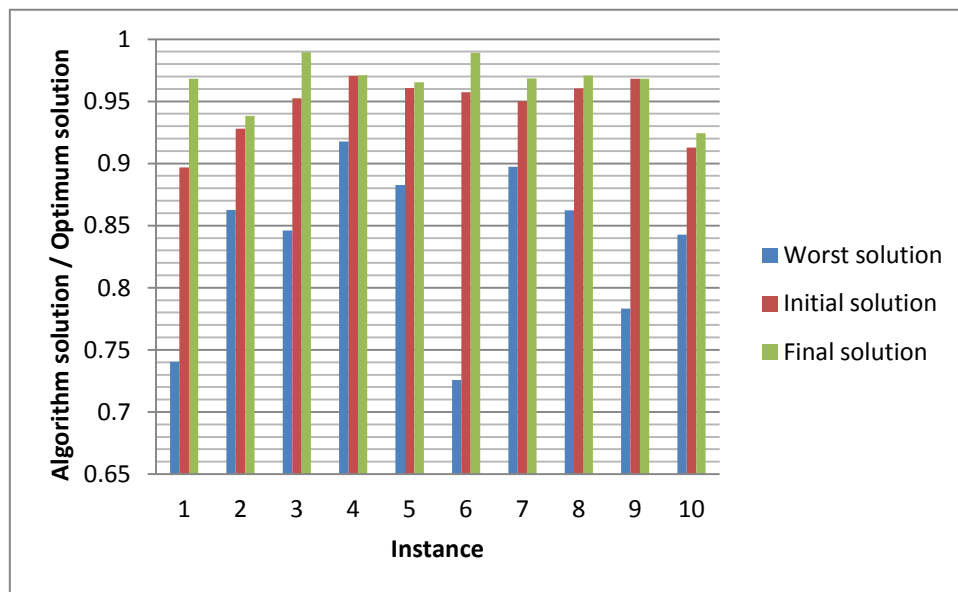


圖 36、實驗二結果演算法效能(12 個基地台)

表 14、實驗二結果演算法誤差(13 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	805.21	636.27	168.94	753.21	765.72	23.3
2	730.51	475.04	255.47	647.57	647.57	32.4
3	842.91	678.25	164.66	816.73	820.16	13.8
4	756.48	689.9	66.58	731.23	741.98	21.7
5	879.19	678.84	200.35	824.82	862.58	8.2
6	738.93	613.5	125.43	714.33	725.39	10.7
7	675.599	536.61	138.989	638.06	644.37	22.4
8	756.13	699.24	56.89	735.78	735.78	35.7
9	774.3	707.63	66.67	719.53	761.07	19.8
10	870.699	746.969	123.73	832.97	840.27	24.5
Average						21.3

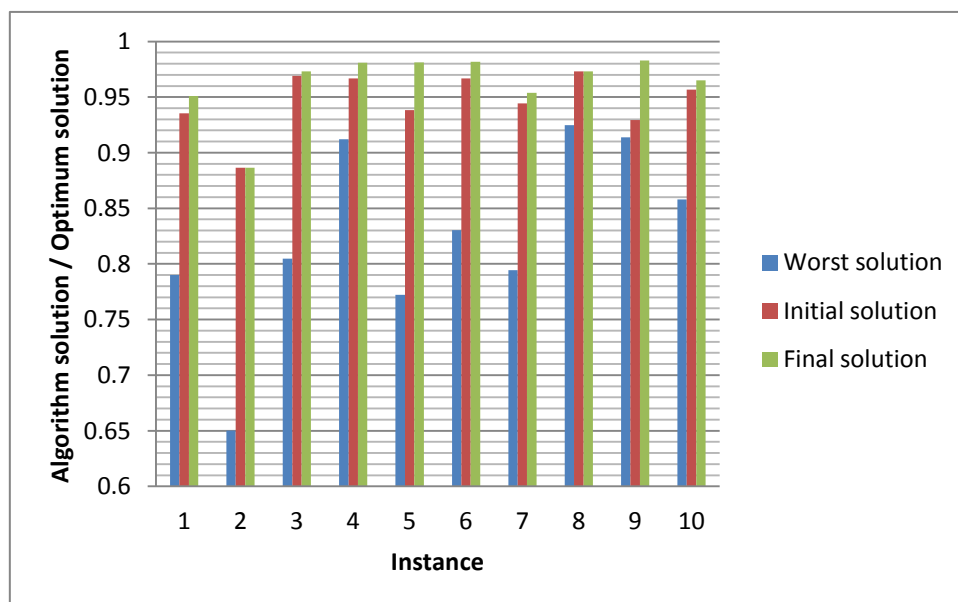


圖 37、實驗二結果演算法效能(13 個基地台)

表 15、實驗二結果演算法誤差(14 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	852.26	621.97	230.29	784.81	785.29	29
2	793.68	581.2	212.48	726.49	726.49	31.6
3	751.38	606.419	144.961	724.96	724.96	18.2
4	826.109	680.07	146.039	776.87	799.97	17.8
Average						24.2

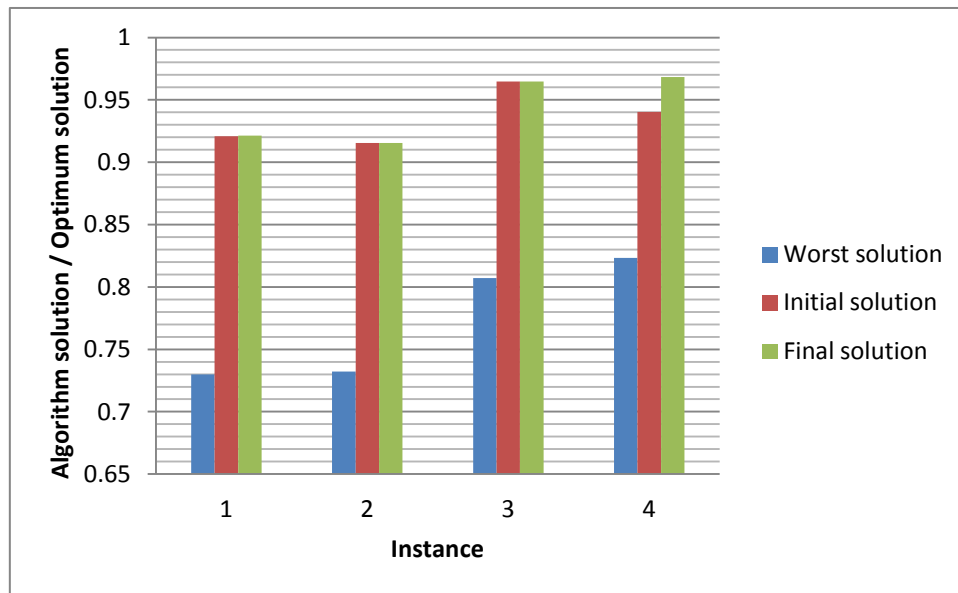


圖 38、實驗二結果演算法效能(14 個基地台)

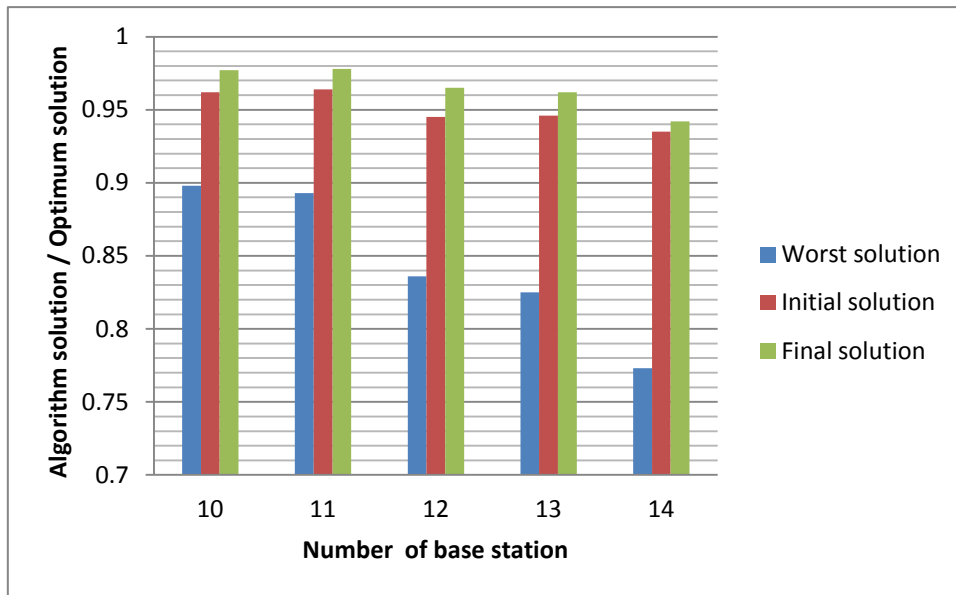


圖 39、實驗二不同基地台個數演算法平均效能比較

#### 4.4.3、實驗三

表 16、17、18、19、20 為實驗三(建構各孤立台時間不同，建構 CCN 工作組個數為 2)的實驗結果，由這些表可以發現 CCN-DS 第一階段貪婪演算法表現並不太好，誤差都在 33% 以上，需經過第二階段演算法才能改進。第一階段貪婪演算法隨著基地台個數增加，其誤差有緩慢下降的趨勢，平均值為 38.2%。第二階段演算法之解在基地台個數增加時，並無明顯的增減趨勢，誤差平均值為 24.8%，兩階段演算法之平均誤差都超過實驗一及實驗二，其可能原因為：同時有多組工作組在進行修復，計算複雜度較高，簡單的啟發式邏輯(heuristic)難以奏效。

表 16、實驗三結果演算法誤差(10 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	645.48	590.79	54.69	636.97	637.92	13.8
2	570.64	553.93	16.71	560.82	569.68	5.7
3	594.209	564.189	30.02	573.15	591.44	9.2
4	698.639	652.6	46.039	670.8	670.8	60.4
5	715.17	664.9	50.27	669.52	692.34	45.4
6	616.42	603.5	12.92	613.02	613.94	19.1
7	651.09	591.01	60.08	630.87	631.11	33.2
8	596.48	569.68	26.8	588.67	588.67	29.1
9	797.7	747.24	50.46	759.35	759.35	76
10	531.85	504.01	27.84	530.78	530.78	3.8
Average						29.6

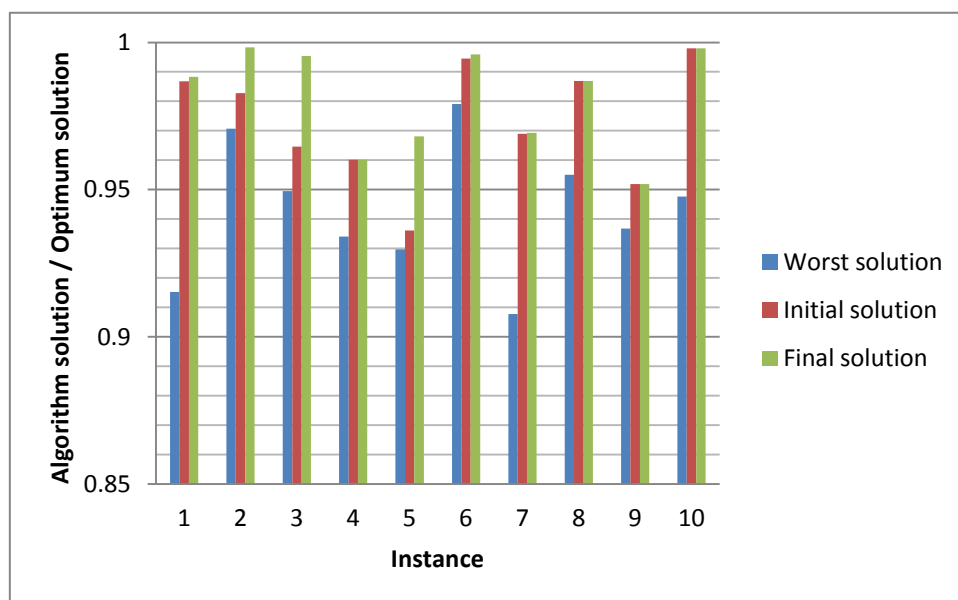


圖 40、實驗三結果演算法效能(10 個基地台)



表 17、實驗三結果演算法誤差(11 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	653.03	631.12	21.91	644.72	648.44	20.9
2	684.069	672.67	11.399	682.96	682.96	9.7
3	699.89	649.76	50.13	679.91	687.21	25.2
4	747.97	728.11	19.86	741.2	747.09	4.4
5	678.28	649.35	28.93	667.78	671.93	21.9
6	572.73	548.36	24.37	564.11	567.73	20.5
7	627.64	561.64	66	570.73	613.06	22
8	744.219	678.86	65.359	693.02	740.84	5.1
9	568.8	545.11	23.69	559.68	560.25	36
10	554.57	528.06	26.51	552.29	552.29	8.6
Average						17.4

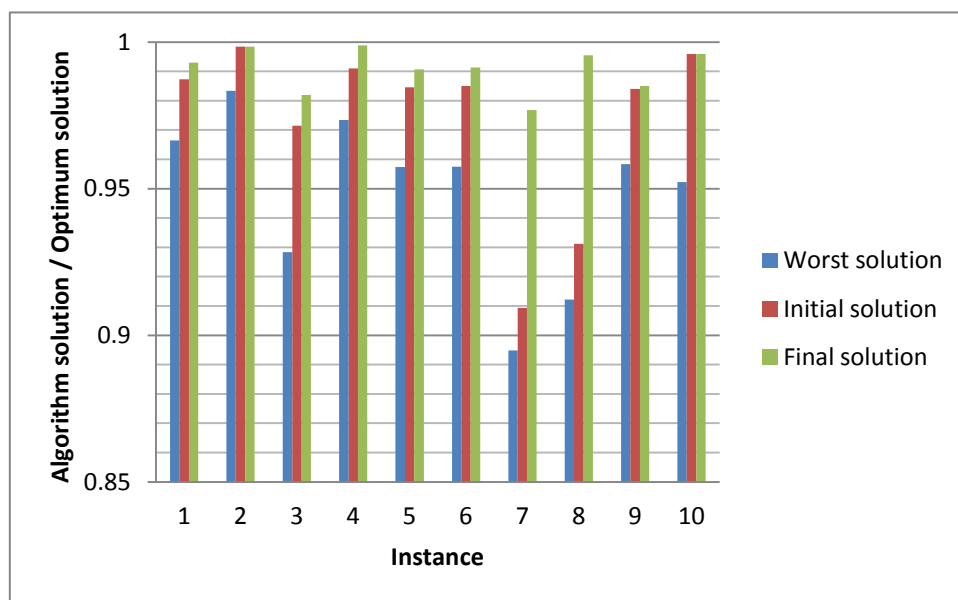


圖 41、實驗三結果演算法效能(11 個基地台)

表 18、實驗三結果演算法誤差(12 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	727.349	675.05	52.299	696.27	710.32	32.5
2	703.36	660.69	42.67	694.29	696.18	16.8
3	774.95	726.109	48.841	763.58	765.06	20.2
4	793.18	758.07	35.11	779.34	781.14	34.2
5	778.06	741.35	36.71	760.11	764.65	36.5
6	689.19	661.199	27.991	679.02	683.18	21.4
7	579.68	555.02	24.66	573.3	574.29	21.8
8	676.62	639.25	37.37	664.79	664.85	31.4
9	775.11	737.97	37.14	764.08	764.08	29.6
10	685.329	651.219	34.11	667.33	677.44	23.1
Average						26.8

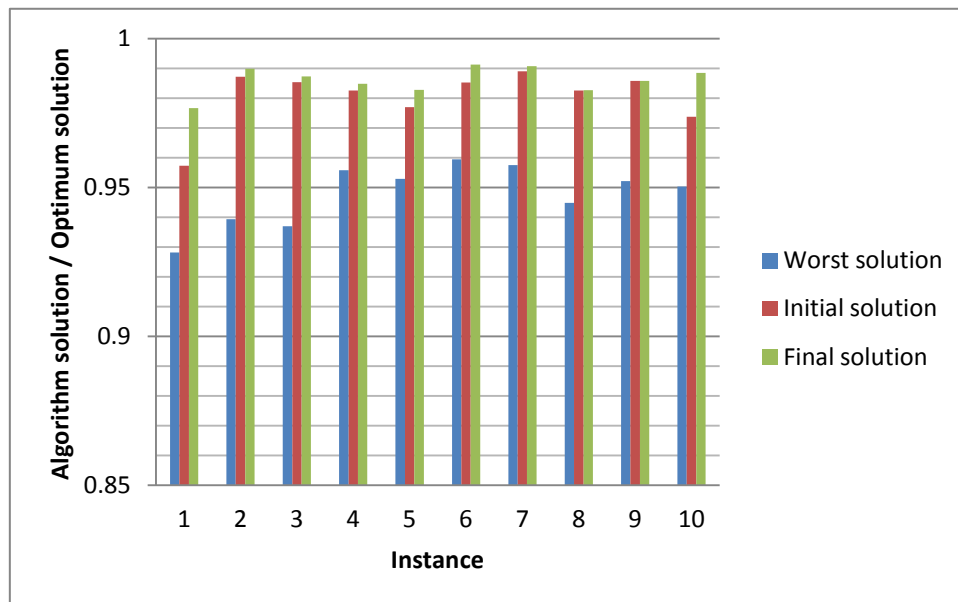


圖 42、實驗三結果演算法效能(12 個基地台)

表 19、實驗三結果演算法誤差(13 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	878.079	828.99	49.089	864.58	864.58	27.4
2	725.569	677.15	48.419	713.76	713.76	24.3
3	858.13	825.62	32.51	840.02	844.42	42.1
4	804.979	770.21	34.769	792.66	798.96	17.3
5	889.869	832.449	57.42	886.16	886.16	6.4
6	805.68	767.349	38.331	793.21	797.77	20.6
7	737.75	692.71	45.04	710.03	714.02	52.6
8	780.319	751.43	28.889	777.3	777.3	10.4
9	796.17	753.42	42.75	773.42	785.58	24.7
10	921.78	871.77	50.01	910.24	912.16	19.2
Average						24.5

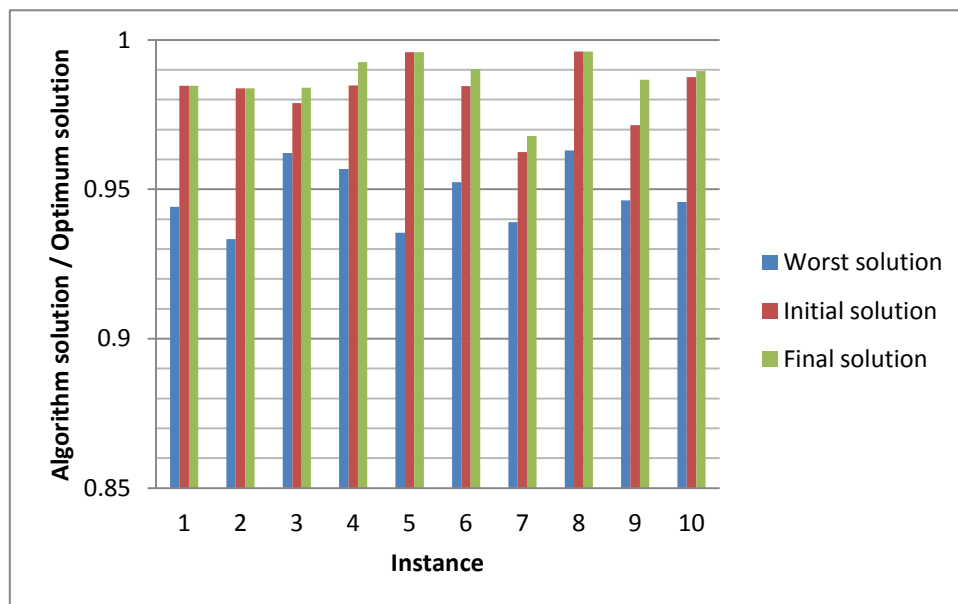


圖 43、實驗三結果演算法效能(13 個基地台)

表 20、實驗三結果演算法誤差(14 個基地台)

Instance	Optimum solution	Worst solution	Performance range	Initial solution	Final solution	Deviation (%)
1	868.17	806.45	61.72	856.12	857.29	17.6
2	865.41	804.439	60.971	836.96	842.86	36.9
3	801.15	723.399	77.751	790.25	790.25	14
4	908.82	858.13	50.69	879.38	891.58	34
Average						25.6

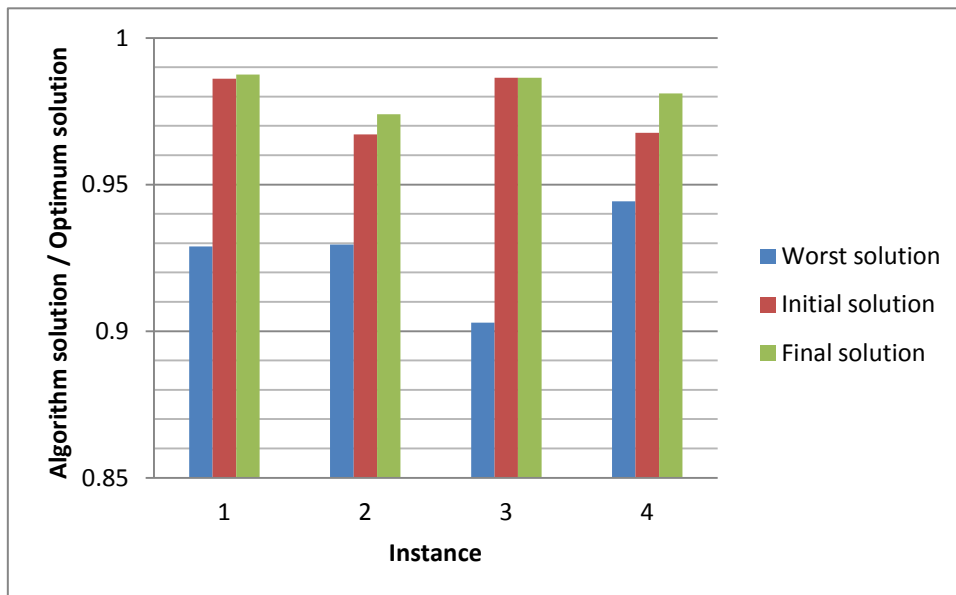


圖 44、實驗三結果演算法效能(14 個基地台)

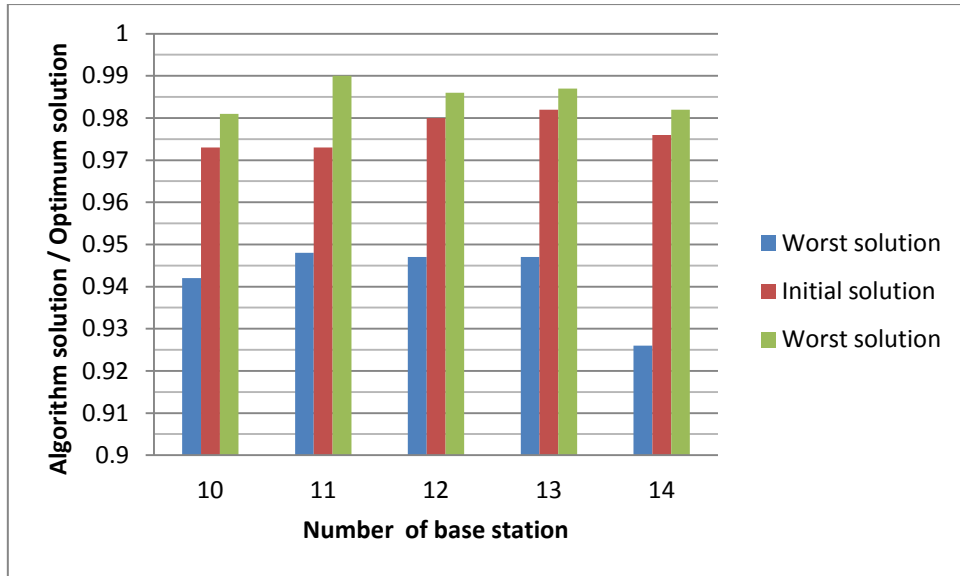


圖 45、實驗三不同基地台個數演算法平均效能比較

#### 4.4.4、實驗四：大規模案例

實驗四進行實驗一到實驗三環境的大規模案例實驗，基地台個數設定為 100，在我們的實驗環境中，當基地台數量高於 14 時，計算最佳解的時間太長難以忍受，因此放棄計算最佳解，而是採用 10 萬個解中的最佳解與最差解，稱為準最佳解 (Pseudo optimum solution) 與準最差解 (Pseudo worst solution)，我們用準誤差來評估效益(公式 4)，以此來分析 CCN-DS 演算法在大規模案例時的效能。實驗一到實驗三每個環境進行十次獨立實驗，並計算其平均效能，由表 21 可發現 CCN-DS 演算法的效能在三階段實驗都能超過準最佳解。

。 準誤差 = (演算法解 - 準最佳解) / (max(準最佳解, 演算法解) - 準最差解)...(4)

表 21、大規模實驗結果演算法誤差(100 個基地台)

	Pseudo optimum solution	Pseudo worst solution	Performance range	Initial solution	Final solution	Average improvement (%)
Experiment 1	4126.2	2797.8	1328.4	4611.7	4681.2	29.47
Experiment 2	4097.7	2680.6	1417.1	4521	4569.2	23.07
Experiment 3	5502.6	4850.9	651.7	5667.9	5708.8	21.82

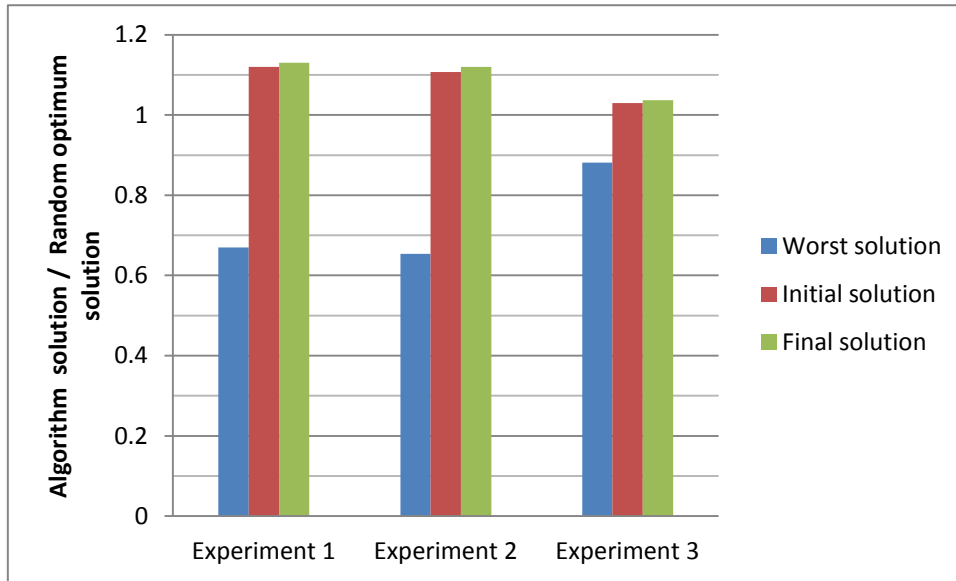


圖 46、大規模實驗結果演算法效能(100 個基地台)

#### 4.5、實驗總結

在實驗一環境下，CCN-DS 演算法的第一階段貪婪演算法平均誤差為 31%，第二階段演算法可減少 11%的誤差；在實驗二環境下，第一階段貪婪演算法之平均誤差為 34.5%，而第二階段演算法可減少 11.3%的誤差；在實驗三環境下，第一階段貪婪演算法的平均誤差為 38%，而第二階段演算法可減少 13.2%的誤差，如表 22、23 所示。很明顯的可以發現隨著環境複雜度的增加，第一階段貪婪演算法與最佳解誤差會逐步上升，而第二階段演算法所能提供的改善會增加，最終解的誤差隨著環境複雜度之增加而呈微幅上升的趨勢，如表 24 所示。

我們分析各個單獨案例的實驗結果，會造成此種原因是因為，在實驗三中基地台個數少時的情況下，會出現救災效益偏低的案例而影響到整體平均誤差，這些效能偏低的案例在實驗一與實驗二時能借由第二階段演算法改良，從而提高救災效益。分以下兩部分來討論這種現象，(1)第二階段演算法對案例的改進；(2)第二階段演算法對實驗三某些案例無法改進。

◦ **第二階段演算法對案例的改進：**

我們分析最終解的結果可以發現，能大幅度改進建構排程的案例多半是由於能提前建構距離 root 較遠的節點。在第一階段貪婪式演算法時，離 root 近而有較多子節點的候選節點，因為其子節點的建構效益會計入父節點的效益，導致其加權效益遠高於子節點少的候選節點，較會獲得演算法的青睞而被優先建構，如此第一階段貪婪演算法在排程時會優先挑選接近於 root(連網台)的節點，而離 root 遠的節點則會被安排在 schedule 後面，而在第二階段演算法時獲得補救得到提前建構的機會，進而有較高的救災效益。

◦ **第二階段演算法對實驗三某些案例無法改良原因：**

實驗三是由多組建構團隊一起建構 CCN 應急通訊系統，因此距離 root 遠的節點有機會與父節點同時建構，造成在第一階段貪婪演算法就有較高機會被挑選到。

其次以基地台個數多寡來分析其對演算法的影響。當基地台個數越多時第一階段貪婪式演算法表現得越好，此原因可能是因為當節點越多時，越接近 root 的節點由於後面所可能影響到的子節點越多，因此需要優先建構，這符合第一階段演算法的想法：將子節點的救災效益考量到權重後加到父候選節點上，使得第二階段演算法效能改善的幅度越來越少。

表 22、實驗結果綜合比較初始解誤差

Number of base station	Experiment 1 (%)	Experiment 2 (%)	Experiment 3 (%)	Average Deviation (%)
10	37.28	40.8	46.47	41.51
11	34.94	34.24	40.49	36.55
12	28.13	35.53	36.86	33.50
13	32.76	33.92	33.03	33.23
14	22.99	28.21	34.57	28.59
Average Deviation (%)	31.22	34.54	38.28	

表 23、實驗結果綜合比較最終解誤差

Number of base station	Experiment 1 (%)	Experiment 2 (%)	Experiment 3 (%)	Average Deviation (%)
10	25.36	23.45	29.61	26.14
11	23.59	22.09	17.48	21.05
12	15.72	24.75	26.81	22.42
13	21.04	21.31	24.56	22.30
14	15.7	24.2	25.66	21.85
Average Deviation (%)	20.282	23.16	24.824	

表 24、實驗結果綜合比較最終解與初始解差距

Number of base station	Experiment 1 (%)	Experiment 2 (%)	Experiment 3 (%)	Average Deviation (%)
10	11.92	17.35	16.86	15.37
11	11.35	12.15	23.01	15.50
12	12.41	10.78	10.05	11.08
13	11.72	12.61	8.47	10.93
14	7.29	4.01	8.91	6.73
Average Deviation (%)	10.93	11.38	13.46	



## 第五章、結論與未來展望方向

大型自然災害發生時通訊系統常會癱瘓，而有效運作的通訊系統卻是災情傳遞、資源調度以及互助協調是否順利的關鍵因素。我們提出利用無線電將功能完整但無法對外進行正常連線的基地台連接起來，建構一個臨時性的行動通訊網路，稱為應急蜂巢式行動通訊網路(Contingency Cellular Network, CCN)，由於各個地方受災情況不盡相同，CCN 的建構順序必須辨明輕重緩急(例如完成一基地台建構可提升之災民存活率)依序建構，因此在網路拓樸規劃完成後，必須根據網路拓樸及各地災情進行基地台建構排程規劃以達到最大的救災效益。

本研究提出一個適合 CCN 樹狀結構的最佳化排程模型，以追求救災效益的最大化。這些模型可供使用者(救災指揮單位)系統化的解決 CCN 建構排程問題。各基地台建構完成時可獲得之效益由救災單位根據所收集到的各地災情自行定義，本模型及排程演算法可決定建構基地台之優先順序。

我們證明 CCN 之排程問題屬於 NP-Hard 問題，但因為突發性大型自然災害時，可容許之計算時間極為短暫，因此我們提出一個啟發式演算法以快速得到一個相近於最佳排程的建構排程。所提出的演算法分為二階段，首先利用貪婪式演算法獲得初始解，再反覆調動建構次序逐步提升救災效益得到更佳解。如此，在限定時間內即可提供一組不錯的修復排程，並且每當災區情況發生改變時能快速的重新運算修改建構排程。

我們利用電腦模擬評估演算法在三種情境下之效能，實驗一假設建構每個基地台時所需花費的時間相同，並且只有一組建構團隊。實驗二為假設各基地台有不同的建構時間。實驗三則假設有很多個建構團隊。實驗案例由亂數隨機產生，以正規化的誤差值來評估排程之優劣。結果顯示，我們提出之演算法在實驗三可在數秒內得出 24.8% 的最佳解誤差，實驗一跟二與最佳解誤差分別是 20.2%、

23.1%。

本研究提出的模型仍有改進之空間，例如多元化運輸建構設備的工具，各基地台的建構時間與直升機的飛行路線由無關改成相關。此外，如能聯合不同電信公司之基地台建構 CCN 時，各基地台相對位置不再是規則的蜂巢架構，各地區的基地台不再是單一選擇，而必須在服務範圍互相重疊的數個基地台內擇一加入 CCN，避免服務範圍之重疊浪費寶貴的資源。

## 參考文獻

- [1] Association of Public-Safety Communications Officials International, Project 25, <http://www.apointl.org/frequency/project25.php>, retrieved May 2010.
- [2] Alfayez Adel, Assiri Majid, Clerk Rutvij, and Alsaadan Usamah, "Evaluating the Viability of TETRA for US Public Safety Communication," *University of Colorado at Boulder Interdisciplinary Telecommunications Program Capstone Project*, Boulder, USA, Nov. 2009.
- [3] Carlo Bertolli, Daniele Tarchi, Romano Fantacci, Marco Vanneschi, and Andrea Tassi, "An Integrated Communication-Computing Solution in Emergency Management," *ACM International Wireless Communications and Mobile Computing Conference 6th*, Caen, France, June 2010.
- [4] Melanie Basich, "Wireless When You Need It," <http://www.policemag.com/Channel/Technology/Articles/Print/Story/2009/01/Wireless-When-You-Need-It.aspx>, retrieved Jan. 2009.
- [5] Arjan Durrezi, Mimoza Durrezi, Vamsi Paruchuri, and Leonard Barolli, "Ad Hoc Communications for Emergency Conditions," *IEEE International Conference on Advanced Information Networking and Applications*, Biopolis, Singapore, Mar. 2011.
- [6] Jinling Du, and Dalian Liu, "Hybrid Genetic Algorithm for the Multi-objective Flexible Scheduling Problem," *IEEE International Conference on Computational Intelligence and Security*, Nanning, China, Dec. 2010.
- [7] Raheleh Dilmaghani, and Ramesh Rao, "A Systematic Approach to Improve Communication for Emergency Response," *Proc. of 42nd Hawaii Int'l Conference on System Sciences*, Waikoloa, Big Island, Hawaii, Jan. 2009.
- [8] Weimin Dong, et al., Chi-Chi, "Taiwan Earthquake Event Report, Risk Management Solutions," Inc., [https://www.rms.com/Publications/Taiwan\\_Event.pdf](https://www.rms.com/Publications/Taiwan_Event.pdf), retrieved Mar. 2010.
- [9] R. C. Eberhart, and J. Kennedy, "New Optimizer Using Particle Swarm Theory," *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Oct. 1995.
- [10] Takahiro Fujiwara, and Takashi Watanabe, "Bidirectional Communications for Damage Monitoring Using Sensor Networks in Emergency Conditions," *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Newport Beach, California, June 2010.
- [11] ITR-RESCUE, Robust Networking and Information Collection Project, <http://www.itr-rescue.org/research/networking.php>, retrieved Feb. 2010.
- [12] Bahareh Jalili, and Mehrdad Dianati, "Application of Taboo Search and Genetic Algorithm in planning and optimization of UMTS radio networks," *ACM International Wireless Communications and Mobile Computing Conference 6th*, New York, USA, June 2010.
- [13] Hong Kong Amateur Radio Transmitting Society, *Proposal on Amateur Radio Emergency Service in Hong Kong-Mobileizing Radio Amateur's community Resources for Disaster and Emergency Communications*, Document No. 06/XIII/018, Aug. 2005.
- [14] J. Kennedy, and R. C. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Network*, vol. 4, Perth, Australia, Nov. 1995, pp. 1942-1948.
- [15] Richard E. Krock, "Lack of Emergency Recovery Planning Is a Disaster Waiting to Happen," *IEEE Communications Magazine*, vol. 49, no.1, Jan. 2011, pp. 48-51.
- [16] Jaeao Lim, Richard Klein, and Jason Thatcher, "Good Technology, Bad Management: A Case Study of the Satellite Phone Industry," *Journal of Information Technology Management*, vol. XVI, no.2, 2005, pp. 48-55.
- [17] Yao-Nan Lien, Li-Cheng Chi and Yuh-Sheng Shaw, "A Walkie-Talkie-Like Emergency

- Communication System for Catastrophic Natural Disasters," Kaohsiung, Taiwan, *Proc. of ISPAN09*, Dec. 2009.
- [18] Yao-Nan Lien, Hung-Chin Jang, and Tzu-Chieh Tsai, "A MANET Based Emergency Communication and Information System for Catastrophic Natural Disasters," *IEEE Workshop on Specialized Ad Hoc Networks and Systems*, Montreal, Canada, June. 2009.
- [19] Kelly T. Morrison, AT&T, "Rapidly Recovering from the Catastrophic Loss of a Major Telecommunications Office," *IEEE Communications Magazine*, vol. 49, no.1, Jan. 2011. pp. 28-35.
- [20] J. Chris Oberg, Andrew G. Whitt, Robert M. Mills, "Disasters Will Happen - Are You Ready? ," *IEEE Communications Magazine*, vol. 49, no.1, Jan. 2011. pp. 36-42.
- [21] Ren Qing-dao-er-ji, and Yuping Wang, Xiaojing Si, "An Improved Genetic Algorithm For Job Shop Scheduling Problem," *IEEE International Conference on Computational Intelligence and Security*, Nanning, China, Dec. 2010.
- [22] Cristina Ribeiro, and Alexander Ferworn, "Computational Public Safety in Emergency Management Communications," *ACM International Wireless Communications and Mobile Computing Conference 6th*, New York, USA, Oct. 2010.
- [23] Yang Ran, "Considerations and Suggestions on Improvement of Communication Network Disaster Countermeasures after the Wenchuan Earthquake," *IEEE Communications Magazine*, vol. 49, no.1, Jan. 2011. pp. 44-47.
- [24] María Luisa Santamaría, and Sebastià Galmé, "Multi-objective Simulated Annealing Approach for Optimal Routing in Time-Driven Sensor Networks," *IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, Singapore, July 2011.
- [25] Yoshitaka Shibata, Yosuke Sato, Naoki Ogasawara, Go Chiba, "A Disaster Information System by Ballooned Wireless Adhoc Network," *IEEE International Conference on Complex, Intelligent and Software Intensive Systems*, Fukuoka, Japan Mar. 2009.
- [26] Stelios Timotheou, and Georgios Loukas, "Autonomous Networked Robots for the Establishment of Wireless Communication in Uncertain Emergency Response Scenarios," *ACM symposium on Applied Computing*, New York, USA, Mar. 2009.
- [27] Search Underwood, "Improving Disaster Management," *Comm. of ACM*, vol. 53, no. 2, Feb. 2010, pp. 18-20.
- [28] Fan Wei, Md.Emadatul Haque, Yukihiro Fukunaga, Takehiro Gouda, Xiaodong Lu, and Kinji Mori "Autonomous Community Construction Technology for Timely Transmitting Emergency Information," *IEEE Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, Shaanxi, China, Oct. 2010.
- [29] 3GPP, Specifications, <http://www.3gpp.org/Specifications>, retrieved Nov. 2011.
- [30] 姚國章, 應急管理信息化建設, 北京大學出版社, ISBN : 9787301155806,2009.09
- [31] 孫玉, 應急通信技術總體框架討論, 人民郵電出版社, ISBN : 7115208328,2009
- [32] 連耀南, 黃智賢, 大型自然災害下大規模救災緊急通訊系統方案, *Proc. of 2010 National Symposium On Telecommunications*, TaoYuan, Taiwan, Dec. 2010.