

國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文
Master's Thesis

考慮資源運輸路徑之
應急蜂巢式行動通訊網路建置排程
Resource Delivery Path Dependent Deployment
Scheduling for Contingency Cellular Network

研究生：高采衣

指導教授：連耀南

中華民國一百零二年六月

June 2013

考慮資源運輸路徑之應急蜂巢式行動通訊網路建置排程
Resource Delivery Path Dependent Deployment Scheduling
for Contingency Cellular Network

研究生：高采衣 Student：Tsai-I Kao

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

中華民國一百零二年六月

June 2013

考慮資源運輸路徑之應急蜂巢式行動通訊網路建置排程

摘要

當發生大規模的地震或強烈的颱風等重大天然災害時，通訊系統常常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的，然而有效運作的通訊系統卻是災情傳遞、資源調度及救災是否順利的關鍵因素。本文所探討的應急通訊系統利用倖存的連通基地台和斷訊卻沒有損毀的基地台，以無線電連接起來建構一個臨時性的通訊系統，稱為應急蜂巢式行動通訊網路(Contingency Cellular Network, CCN)。由於各地災情狀況不完全相同，CCN 的建構順序必須考慮災區的輕重緩急、時間的急迫等因素依序建構。因此當 CCN 拓樸規劃完成後，根據 CCN 拓樸、各地災情嚴重程度以及拓樸中基地台間的相對距離(運輸時間)進行基地台建構排程規劃，以達到最大的總救災效益。

本文考慮各基地台所能發揮的救災效益、所需建構時間、以及運輸工具從任一基地台到另一基地台所需運輸時間，提出兩個適合 CCN 拓樸樹狀結構的考慮資源運輸路徑之最佳化排程模型 CCNDS-AC 和 CCNDS-UC。CCNDS-AC 限制建構順序必須從連網台往下循序建構，但 CCN-UC 則否。因發生突發性大型天然災害時，可容許的計算時間相當短暫，因此提出了兩個快速的啟發式演算法 DS-ACG 與 DS-UCB，可在短時間內求出一組相當逼近於最佳解的建構排程順序，與 DS-UCB 相互比較。本文以電腦模擬的方式進行小規模實驗與大規模實驗評估，並且用 Genetic Algorithm 來比較啟發式演算法的效能。結果顯示 DS-UCB 明顯優於 DS-ACG 及 Genetic Algorithm。在小規模實驗中 DS-UCB 可求得與最佳解的總救災效益誤差平均約 0.9% 的近似最佳解建構順序。而在大規模實驗下，DS-UCB 與十萬個解中的最佳解—pseudo optimal solution 相較，總救災效益平均高出約 16.7%，而總救災時間平均約少了 19.4%。

Resource Delivery Path Dependent Deployment Scheduling for Contingency Cellular Network

Abstract

When stricken by a large-scale disaster, the efficiency of disaster response operation is very critical to life saving. However, cellular networks were usually crashed in earthquake, typhoons or other natural disasters due to power outage or backhaul breakage. Unfortunately, the efficiency of communication system is a critical factor to the success of disaster response operation. We designed a contingency cellular network (CCN) by connecting physically intact but service-disrupted base stations together with wireless links. Since the transportation capacity may be very limited, scheduling of CCN deployment order according to the demand of disaster operation and traveling time between base stations becomes an important issue. We propose two optimization models: CCN Deployment Scheduling Antecessor Constrained Problem (CCNDS-AC) and CCN Deployment Scheduling Unconstrained Problem (CCNDS-UC), aiming to maximize the efficiency of disaster response operation. Both problems are proven to be NP Hard. We also designed two rapid heuristic algorithms, DS-ASG and DS-UCB to solve the problems respectively when it is needed in urgent.

Finally, we evaluated the proposed algorithms against optimal solutions (in small cases only) as well as genetic algorithm by simulation. The experimental results show that DS-UCB outperforms all other algorithms. In small scale cases, the profit obtained by DS-UCB is only 0.9% smaller than what the optimum solution can get. In large scale cases, as compared to the pseudo optimum solution, which is the best solution among 100,000 solutions, DS-UCB outperforms pseudo optimum solutions in profit by 16.7%, and in traveling time by 19.4%, both in average.

誌謝辭

研究所兩年，真是轉眼間就結束了，非常感謝連耀南教授在研究所期間的教導，老師用問答的方式引導我們研究問題，不僅使教學進行得有趣，也更加訓練我獨立思考的能力。老師亦是我們的生活導師，時常關心我們在課業及生活上的大小事，也讓我們在研究生生活中能倍感溫馨。

這兩年來的生活能夠如此精彩，要感謝一起努力的實驗室夥伴：幽默熱心的山高學長，帶領我認識政大的筱慈學姊，帶給大家歡笑的冠傑學長與郁翔學長，互相加油打氣的誠誠，有你們的陪伴讓我能快速融入研究生活；也感謝資科系夥伴們：一起充當觀光客的璇姊、教我認識攝影的詹詹、照顧我也適時督促我的雅玲、介紹我工讀的英明、常串門子的煜泓，以及標哥、柏硯、偉迪、宇軒、凱柔、筱璇、博為、欣儒，謝謝你們讓我的研究生生活更多采多姿；感謝英文系玉屏助教、Betty 助教、莉華助教、Megan 助教、賴惠玲老師，感謝可愛的你們給我工讀機會，讓我學到很多處事的細節與態度。感謝不顧路途遙遠到政大找我的高中好友優駒駒們，因為有你們讓我從不感到孤單。感謝許多給予我幫助、鼓勵我的人們，因為你們讓我研究更加順利。

最後要感謝我的家人，有你們在背後不斷的鼓勵我、支持我，讓我即使遭遇挫折也能夠一一克服重新出發，謝謝你們。

高采衣 2013

目錄

摘要	i
Abstract.....	ii
誌謝辭	iii
目錄	iv
圖目錄	viii
表目錄	xi
第 1 章、 簡介	1
1.1、 災區應急通訊系統需求分析	3
1.1.1、 大型天然災害發生時救災行動面臨的挑戰	3
1.1.2、 固網與行動通訊系統癱瘓原因	6
1.1.3、 通訊設備修復困難	7
1.1.4、 大型災害的救災時效	8
1.1.5、 應急通訊網建置之挑戰與需求	9
1.1.6、 應急通訊網效能指標	13
1.2、 應急通訊網路簡介	13
1.3、 論文架構	15
第 2 章、 相關研究	16
2.1、 第三代行動通訊架構	16
2.1.1、 通用行動通訊系統陸地無線接入網 (UTRAN).....	17
2.1.2、 核心網路 (Core Network).....	17
2.1.3、 3G 網路通訊協定	18
2.2、 應急通訊系統種類	19
2.2.1、 專用高抗災通信平臺	20
2.2.2、 無線對講機 (Walkie-Talkie).....	21

2.2.3、	業餘無線電 (Amateur radio)	21
2.2.4、	行動衛星通訊 (Satellite mobile phone)	22
2.2.5、	專業用集群通訊系統 (Trunking radio)	23
2.2.6、	移動基地台 (Cell on wheels)	24
2.2.7、	行動隨意式網路 (MANET)	25
2.2.8、	過去應急通訊系統相關研究	26
2.2.9、	應急通訊系統綜合比較	29
第 3 章、	應急蜂巢式行動通訊網路 (CCN)	31
3.1、	系統架構	33
3.1.1、	CCN 連網方式	34
3.1.2、	系統元件	35
3.2、	通訊模式	37
3.2.1、	災區通訊型態分類	37
3.2.2、	CCN 通訊模式	38
3.3、	CCN 功能架構(Functional Architecture)	41
3.4、	CCN 服務流程	44
3.5、	建置與運轉流程	47
3.6、	重要研究議題與未來研究方向	47
3.7、	排程問題之相關演算法	51
3.7.1、	概述	51
3.7.2、	精確解法 (Exact Algorithm)	52
3.7.3、	後啟發式演算法 (Meta-heuristics)	54
3.7.4、	評論	59
第 4 章、	考慮資源運輸路徑之 CCN 建置排程	61

4.1、	設計理念與目標	61
4.2、	CCN 的建構排程	62
4.3、	問題定義	65
4.3.1、	救災效益定義	65
4.3.2、	問題描述	66
4.4、	最佳化問題模型	66
4.4.1、	CCNDS-AC	66
4.4.2、	CCNDS-UC	67
4.5、	複雜度分析	69
4.5.1、	Single Machine Scheduling Problem (SMS).....	69
4.5.2、	CCNDS-AC	69
4.5.3、	CCNDS-AC 計算複雜度	69
4.6、	演算法設計	71
4.6.1、	DS-ACG 演算法	71
4.6.2、	DS-UCB 演算法	77
4.6.3、	Genetic Algorithm	84
4.6.4、	演算法時間複雜度分析(Time Complexity)	89
第 5 章、	效能評估	90
5.1、	實驗目的	90
5.2、	實驗設計	90
5.2.1、	實驗環境	90
5.2.2、	救災效益函數模型	91
5.3、	實驗一：小規模實驗	93
5.3.1、	實驗參數	93

5.3.2、 評估指標	94
5.3.3、 實驗結果	94
5.4、 實驗二：大規模實驗	115
5.4.1、 實驗參數	115
5.4.2、 評估指標	115
5.4.3、 實驗結果	116
5.5、 實驗總結	120
第 6 章、 結論與未來展望方向	123
參考文獻	125

圖目錄

圖 1.1、通訊線路於災害中受損	4
圖 1.2、行動通訊網路受損主要原因	7
圖 1.3、獲救時間與存活率之關係	8
圖 1.4、CCN 網路	15
圖 2.1、System Architecture of 3GPP Release 99	16
圖 2.2、3GPP Release 5 架構	19
圖 2.3、3GPP Release 7 架構	19
圖 2.4、應急通訊系統使用時機	20
圖 3.1、CCN 網路架構	32
圖 3.2、連網台與孤立台	33
圖 3.3、CCN 基地台間之連線	34
圖 3.4、孤立台藉 CCN 連線連上核心網路	35
圖 3.5、CCN 應急修復包(CRP)	35
圖 3.6、孤立台恢復連線方式	37
圖 3.7、CCN 功能架構	41
圖 3.8、一般通話模式通話建立流程	44
圖 3.9、無線電對講機模式通話建立流程	45
圖 3.10、群組通訊模式通話建立流程	46
圖 4.1、不同環境的存活率關係圖	62
圖 4.2、範例：CCN 建構排程相關資訊	63
圖 4.3、範例：CCN 建構排程(排程一)	64
圖 4.4、範例：CCN 建構排程(排程二)	65
圖 4.5、範例：CCNDS-AC 模型可能造成之狀況	68
圖 4.6、DS-ACG 演算法流程圖	72
圖 4.7、演算法 DS-ACG 之 Pseudo Code	73
圖 4.8、DS-ACG 演算法範例	74
圖 4.9、DS-ACG 演算法步驟一	74
圖 4.10、DS-ACG 演算法步驟二	75
圖 4.11、DS-ACG 演算法步驟三	75
圖 4.12、DS-ACG 演算法步驟四	76
圖 4.13、DS-ACG 演算法步驟五	76
圖 4.14、DS-UCB 演算法流程圖	78

圖 4.15、DS-UCB 之 Backtracking 機制流程圖	78
圖 4.16、DS-UCB 演算法 Pseudo Code	79
圖 4.17、DS-UCB 演算法範例	80
圖 4.18、DS-UCB 演算法步驟一	80
圖 4.19、DS-UCB 演算法步驟二	81
圖 4.20、DS-UCB 演算法步驟三	81
圖 4.21、DS-UCB 演算法步驟三(排程範例 1).....	82
圖 4.22、DS-UCB 演算法步驟三(排程範例 2).....	82
圖 4.23、DS-UCB 演算法步驟四(狀況一).....	83
圖 4.24、DS-UCB 演算法步驟四(狀況二).....	83
圖 4.25、Genetic Algorithm 演算法流程圖	84
圖 4.26、演算法 Genetic Algorithm 之 Pseudo Code	85
圖 4.27、Genetic Algorithm 演算法步驟一	86
圖 4.28、Genetic Algorithm 演算法步驟二	86
圖 4.29、Genetic Algorithm 演算法步驟三	87
圖 4.30、Genetic Algorithm 演算法步驟四	87
圖 4.31、Genetic Algorithm 演算法步驟五	88
圖 4.32、Genetic Algorithm 演算法步驟六	88
圖 5.1、救災效益函數模型一 (Profit function 1).....	92
圖 5.2、救災效益函數模型二 (Profit function 2).....	92
圖 5.3、救災效益函數模型三 (Profit function 3).....	93
圖 5.4、實驗一：Total Profit (8 nodes).....	95
圖 5.5、實驗一：Original Deviation of Profit (8nodes).....	96
圖 5.6、實驗一：Normalize Deviation of Profit (8 nodes)	96
圖 5.7、實驗一：Total Time (8 nodes).....	97
圖 5.8、實驗一：Original Deviation of Time (8 nodes)	98
圖 5.9、實驗一：Normalize Deviation of Time (8 nodes).....	98
圖 5.10、實驗一：Total Profit (9 nodes).....	99
圖 5.11、實驗一：Original Deviation of Profit (9 nodes).....	100
圖 5.12、實驗一：Normalize Deviation of Profit (9 nodes)	100
圖 5.13、實驗一：Total Time (9 nodes).....	101
圖 5.14、實驗一：Original Deviation of Time (9 nodes)	102
圖 5.15、實驗一：Normalize Deviation of Time (9 nodes).....	102
圖 5.16、實驗一：Total Profit (10 nodes).....	103
圖 5.17、實驗一：Original Deviation of Profit (10 nodes).....	104

圖 5.18、實驗一：Normalize Deviation of Profit (10 nodes)	104
圖 5.19、實驗一：Total Time (10 nodes).....	105
圖 5.20、實驗一：Original Deviation of Time (10 nodes)	106
圖 5.21、實驗一：Normalize Deviation of Time (10 nodes).....	106
圖 5.22、實驗一：Total Profit (11 nodes).....	107
圖 5.23、實驗一：Original Deviation of Profit (11 nodes).....	108
圖 5.24、實驗一：Normalize Deviation of Profit (11 nodes)	108
圖 5.25、實驗一：Total Time (11 nodes).....	109
圖 5.26、實驗一：Original Deviation of Time (11 nodes).....	110
圖 5.27、實驗一：Normalize Deviation of Time (11 nodes)	110
圖 5.28、實驗一：Total Profit (12 nodes).....	111
圖 5.29、實驗一：Original Deviation of Profit (12 nodes).....	112
圖 5.30、實驗一：Normalize Deviation of Profit (12 nodes)	112
圖 5.31、實驗一：Total Time (12 nodes).....	113
圖 5.32、實驗一：Original Deviation of Time (12 nodes)	114
圖 5.33、實驗一：Normalize Deviation of Time (12 nodes).....	114
圖 5.34、實驗二：Total Profit (50 nodes).....	117
圖 5.35、實驗二：Original Deviation (Error) of Profit (50 nodes).....	117
圖 5.36、實驗二：Normalize Deviation (Error) of Profit (50 nodes)	118
圖 5.37、實驗二：Total Time (50 nodes).....	119
圖 5.38、實驗二：Original Deviation of Time (50 nodes)	119
圖 5.39、實驗二：Normalize Deviation of Time (50 nodes).....	120
圖 5.40、Average Processing Time	122

表目錄

表 1.1、近年大型天然災害傷亡損失記錄	2
表 1.2、應急通訊網建置需求	9
表 2.1、應急通訊系統比較	30
表 3.1、災區通訊需求分類	38
表 4.1、各演算法時間複雜度	89
表 5.1、電腦軟硬體規格	91
表 5.2、實驗一：小規模實驗參數設定	93
表 5.3、實驗一：總救災效益實驗結果與誤差比較 (8 個基地台)	95
表 5.4、實驗一：總救災時間實驗結果與誤差比較 (8 個基地台)	97
表 5.5、實驗一：總救災效益實驗結果與誤差比較 (9 個基地台)	99
表 5.6、實驗一：總救災時間實驗結果與誤差比較 (9 個基地台)	101
表 5.7、實驗一：總救災效益實驗結果與誤差比較 (10 個基地台)	103
表 5.8、實驗一：總救災時間實驗結果與誤差比較 (10 個基地台)	105
表 5.9、實驗一：總救災效益實驗結果與誤差比較 (11 個基地台)	107
表 5.10、實驗一：總救災時間實驗結果與誤差比較 (11 個基地台)	109
表 5.11、實驗一：總救災效益實驗結果與誤差比較 (12 個基地台)	111
表 5.12、實驗一：總救災時間實驗結果與誤差比較 (12 個基地台)	113
表 5.13、實驗二：大規模實驗參數設定	115
表 5.14、實驗二：總救災效益實驗結果與誤差比較 (50 個基地台)	116
表 5.15、實驗二：總救災時間實驗結果與誤差比較 (50 個基地台)	118
表 5.16、各演算法平均執行時間(單位:秒)	122

第1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 1.1 就是近幾年較著名的天然災害統計[9][24][32]，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 公尺高的海嘯及令全世界恐慌的核災，所造成的三種複合式重大災害，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…等天然災害，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使用。

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被洪水淹沒而結構未損，但沿著道路及橋樑鋪設的電力與通訊線路，卻隨道路橋樑坍塌而損毀，造成行動通訊系統也隨之癱瘓，電力與基地台 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衛星通訊能力的強固基地台，但因成本高昂之故，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。

表 1.1、近年大型天然災害傷亡損失記錄

Country	Event	Date	Scale	Damage
Taiwan	921 Chi-Chi Earthquake	1999/09/21	7.3 Richter Scale	Dead : 2,415 Injured : 11,306
USA	Katrina Hurricane	2005/08/23	Category 5 SSHS	Dead : 1,836
China	Sichuan Earthquake	2008/05/12	8.0 Richter Scale	Dead : 69,227 Injured : 374,176
Italy	L'Aquila Earthquake	2009/04/06	6.3 Richter Scale	Dead : 297 Injured : 1,500
Taiwan	88 Flood	2009/08/08	> 2500 mm Rainfall within 2 days	Dead : 681 Injured : 33
Haiti	Haiti Port-au-Prince Earthquake	2010/01/12	7.0 Richter Scale	Dead : 316,000 Injured : 300,000
Chile	2010 Chile Earthquake	2010/02/27	8.8 Richter Scale	Dead : 800
China	Yushu Earthquake	2010/04/14	7.1 Richter Scale	Dead : 2,698 Injured : 12,135
NZ	2011 Christchurch Earthquake	2011/02/22	6.3 Richter Scale	Dead : 185 Injured : 2,000
Japan	2011 Tokyo Earthquake and Tsunami	2011/03/11	9.0 Richter Scale	Dead : 16,079 Missing : 3,499
Turkey	2011 Van Earthquake	2011/10/23	7.3 Richter Scale	Dead : 644 Injured : 4,412
USA	Hurricane Sandy	2012/10/29	Category 3 SSHS	Dead : 117 > \$71 billion USD loss
Iran	2013 Sistan and Baluchestan earthquake	2013/4/16	7.8 Richter Scale	Dead : 35 Injured : 117
China	2013 Lushan Earthquake	2013/4/20	7.0 Richter Scale	Dead : 213 Injured : 11,460

有效運作的通訊系統是災情傳遞、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快速下降，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服務的另一個重點，災區內電力、瓦斯、食物、飲水、禦寒衣物、醫藥等維生系統癱瘓下，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災民，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災民與救災人員使用，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原有行動通訊系統中倖存的連通基地台以及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通訊網路(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。本文旨在探討 CCN 網路的拓樸建置排程規劃以追求最大的救災效益。

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[8][9][16][23][24][28][30][37]。

1.1.1、大型天然災害發生時救災行動面臨的挑戰

■ 建築物倒塌，人員受困，亟待救援

看似高大且堅固的建築物在地震襲擊時應聲倒下，有許多民眾因來不及逃生而被困在瓦礫堆下，等待救援。而受困於災難現場的人員，必須在 72 小時內搶救出來，否則生還機會極為微弱。因此災難發生之初，最重要之救難工作乃在集中所有救難資源搶救受困人員，這段時間稱為「黃金 72 小時」。

■ 通訊網路幾乎全面癱瘓

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。



圖 1.1、通訊線路於災害中受損

■ 災區的交通全面癱瘓，外援進入緩不濟急

道路橋樑柔腸寸斷，甚至被大量志願救災車輛塞爆道路。在莫拉克颱風後，受惡劣天候影響，直昇機亦難以接近災區。而 2010 年海地地震中，太子港機場雖逃過一劫，但因機場無法提供返航油料因而飛機無法降落，關閉數天之久。總而言之，災變初期，尤其是黃金 72 小時之內，主要依賴在地的人力物力投入救援，無法太過依賴外界支援，此外，因無法運送大型機具進入災區，對大型緊急通訊設備，例如具有衛星連線的車載行動通訊基地台，造成進入的障礙。

■ 專業救災人員嚴重不足

有組織有訓練的專業救災人員之數量遠遠不足，尤其是災變初期交通系統癱瘓，災區外及國際的專業救難團隊需克服交通阻隔方能進入災區，因此災變初期必須動員

大量的在地志願人員投入救難救災，但不幸的是，投入救災的在地志願人員並無有效的通訊工具協助彼此通訊連絡。急需一個具有大量廉價通訊設備的緊急通訊系統之協助。

■ 行政指揮系統失靈

各級行政系統可能癱瘓，導致既有通聯組織亦隨之癱瘓，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，完全與外界失聯達數天之久。2010年海地地震中，政府大樓倒塌，所有部長全部失聯，僅有一個光桿的海地總統獨撐大局，所有行政系統形同癱瘓。

■ 物資不易協調分配，資源嚴重錯置

因通聯系統癱瘓，災情資訊收集不易，在災情資訊紊亂不全的情況下，難以對救災物資作有效且適當的分配，即使分配得宜，如何送達又是另外一個問題。救災物資的錯置，尤其是醫療用品，使得很多亟待救助的受難者無法即時得到所需資源而喪生(例如：海地地震中，就發覺極度缺乏處理外傷必須的抗生素)。

■ 救災人員彼此溝通困難，不易協調

有組織有訓練的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難救災。而這些志願人員幾乎沒有通訊聯絡工具可協助彼此之間的協調連絡，致使救災效率極低。更有甚者，救災工作因溝通不良而彼此干擾，例如當一個救難團隊在使用高靈敏麥克風探測倒塌建物是否有倖存者時，建物他側卻在乒乒乓乓的進行挖掘工作，而呼嘯而過的救護車聲音也是驚天動地的。

由於通訊聯絡不良，資訊缺乏，資訊無法交流等諸多因素，導致救難工作缺乏效率與救難資源之嚴重錯置，因而喪失了很多可以救人一命的機會，許多生靈因資訊溝通不良而喪失即時獲救之機會，令人扼腕。

1.1.2、固網與行動通訊系統癱瘓原因

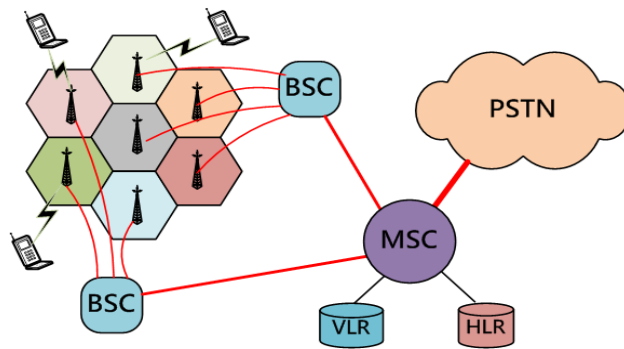
行動電話藉由無線電互相通訊，在一般人之普遍認知中，它不受天災的影響，在災害來臨時可作為緊急通訊之用。但事實上卻非如此，商用行動通訊系統其實必須仰賴固定通訊網路，其基地台之後端大多利用固網幹線連上核心網路，無線的鏈結只存在於終端使用者(手機)與基地台之間，而從基地台到後端機房仍然是利用固網線路連接。從莫拉克風災(八八水災)及 921 集集大地震的經驗中，我們歸納出影響行動電話可用度的主要因素如下：

- 各種機房或因電力中斷且備用發電機因油料告罄，或因冷卻系統遭強震摧毀，而停止運轉。
- 基地台遭強震摧毀或因電力中斷而癱瘓(備用電源僅能支持四至五小時[50]，而八八水災中 3300 座斷訊的基地台中，約 70%是因為電力中斷而中斷服務)。
- 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。

大部分的電力線路與固網線路為了架設與維修方便，經常是沿著道路橋樑鋪設。而道路橋樑的損毀必將導致電力與通訊線路中斷，如圖 1.1 所示。而行動通訊系統也常隨之中斷，如圖 1.2 所示，(A)是行動通訊系統的基本架構，基地台後端必須有固網連線(Backhaul)連到控制器或交換機，(B)是八八水災中受損的一座橋，從(C)可以很清楚的看出即使基地台本身完好無缺，但因後端連線斷掉，而被迫停止運轉。以莫拉克風災/八八水災為例，基地台多建在高處可避免被洪水淹沒，但道路柔腸寸斷，二十餘座大型橋樑遭洪水沖毀，而隨著道路及橋樑鋪設的通訊線路也隨之柔腸寸斷，造成行動通訊系統也隨之癱瘓，電力與基地台 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連累，建造強固的基地台與交換機房仍是無濟於事，無法大幅提升系統可用度。

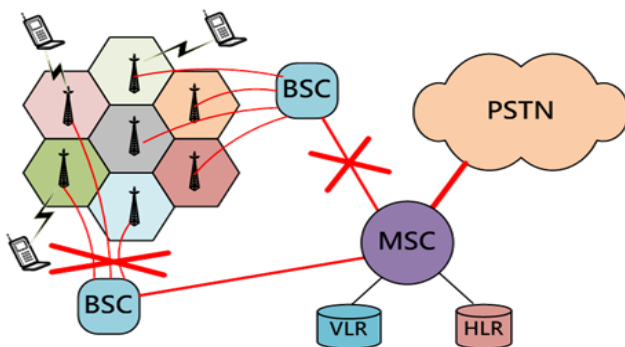
在沒有行動通訊系統的支援下，救難工作只能靠原始的面對面方式進行溝通，無奈因地形阻隔，交通不便，效率極差，甚至在一棟倒塌大樓兩側之團隊都無法面對面溝通，因而互相干擾救災行動。



(A) 行動通訊網路架構



(B) 固網隨橋斷而斷



(C) 行動通訊因後端連線中斷而癱瘓

圖 1.2、行動通訊網路受損主要原因

1.1.3、通訊設備修復困難

在災害中，由於交通系統癱瘓，大型修復機具無法進入災區，進行第一時間的搶修，加上技術人力不足，所需資材調度不及等諸多因素，搶修實際毀損的基地台通訊設備，並使災區通訊全面恢復，向來是一項艱鉅的任務。以 921 地震為例，中華電信耗費 15 天，

才搶通災區電信網路。在八八水災中，斷訊基地台總數達 3300 餘座，中華電信斷訊基地台達 1800 座，其中 550 座在兩天之後仍無法恢復運轉。換言之，在關鍵的黃金 72 小時內，大量的行動電話將陷於癱瘓，無法及時修復。再觀察近年來各處天然災害的搶救效率，即使如美國、日本這種先進國家也無法在黃金 72 小時內修復大部分的電力及通訊系統。且不談卡翠納颱風摧毀新奧爾良讓美國的緊急救災機關 FEMA (Federal Emergency Management Agency) 備受責難，就連 2011 年一個輕度熱帶氣旋橫掃美東就讓數百萬戶居民斷水斷電達數星期之久。2011 年的日本地震更讓世人大吃一驚，因地震頻仍而對地震之準備號稱世界之最好的日本，竟然不堪一擊。世人在面對大型天然災害時，其實仍然是脆弱不堪的，通訊設備因為受制於橋樑道路的脆弱，不但容易受損，也不容易快速修復。

1.1.4、大型災害的救災時效

「黃金 72 小時搶救時間(Golden 72 Hours)」，指的是在災難發生後，搶救倖存生命的關鍵救難時機。災後受困的人員會因外傷、失溫及缺乏食物飲水等因素，使得存活機率隨著時間流逝而急速下降。根據統計，在災後 24 小時內獲救的存活率可高達 90%；在災後 25~36 小時間獲救，存活率銳減為 50%~60%；在災後 36~72 小時間獲救，存活率僅剩 20%~30%；在超過 72 小時後獲救，存活率則剩下 5%~10%，受困人員能倖存的機率就極低了，如圖 1.3 所示。

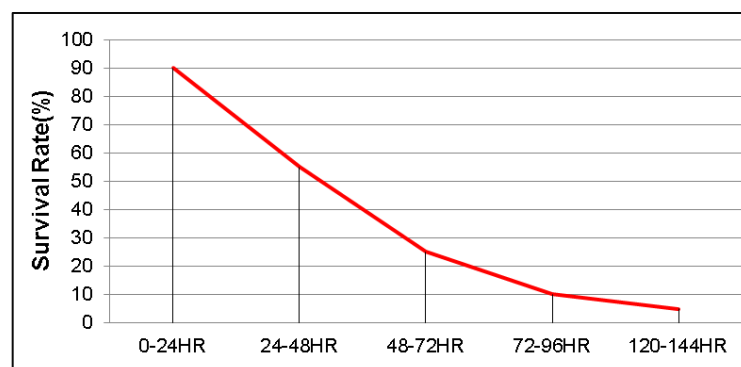


圖 1.3、獲救時間與存活率之關係

1.1.5、應急通訊網建置之挑戰與需求

由於時間與資源之限制，災區應急通訊網的建置面對了許多挑戰與特殊需求。在嚴格的時間限制和極端的環境條件之下，應急通訊系統的建置與一般的通訊系統完全不同。

根據我們的親身經歷和觀察，在災區的應急通訊系統應該要能夠迅速被建立和提供緊急的通訊服務。在嚴格的時間限制和極端的環境條件之下，建置應急通訊系統的需求可歸納成七個面向如表 1.2 所示。普及性和可用性是終端使用者需求。而實際可行性、負載能力、持續性、可調性和維運性則是用來滿足網路管理者的需求。

表 1.2、應急通訊網建置需求

7-ability	Popularity	<ul style="list-style-type: none"> ● User friendly ● Sufficient amount of terminals
	Usability	<ul style="list-style-type: none"> ● Task original communication services ● Adequate quality of service ● Long standing time of terminals ● Mobility
	Practicability	<ul style="list-style-type: none"> ● Low development cost ● Easy acquisition of equipment ● Construct rapidly and easily
	Capacity	<ul style="list-style-type: none"> ● Sufficient number of concurrency users ● Resist the burst of call request
	Sustainability	<ul style="list-style-type: none"> ● Reliability ● Fast recovery
	Adaptability	<ul style="list-style-type: none"> ● Self-adjustment
	Operability	<ul style="list-style-type: none"> ■ OAM functions

I. 終端使用者需求

1. **普及性(Popularity)**：由於缺少終端設備，許多常見的應急通訊系統，例如衛星通訊系統，群集無線電系統和業餘無線電系統只能用在特殊的小群組。大多數的受災者和志願救災團隊通常無法利用這些通訊系統進行通話。使用者在使用群集無線電系統和業餘無線電系統時，需要經過特殊訓練才能使用，群集無線電系統和業餘無線電系統的普遍性是有限的。易於使用的終端設備的普及性成為一個應急通訊系統的重要需求。其要件有二：(a)易於使用。(b)可方便及低價的普及於災區使用者。

2. **可用性(Usability)**：由於在災區的極端條件之下，應急通訊系統必須妥善處理所有使用者的各種通訊需求，為了滿足可用性，應急通訊系統應該提供任務導向的通訊服務，並且可以支援移動性的通訊服務和擁有良好的通訊品質。此外，終端設備的待機時間最好能長於一天，避免頻繁的充電。

- **任務導向的通訊服務(Task Oriented Communication Services)**：包含了傳統電話服務模式(POTS)，對講機模式(walkie-talkie)和群組通訊模式(Agency)的服務。因為災難應變工作者需要互相進行通訊，只提供普通的通訊服務是不夠的。在救災團隊之間需要進行通訊時，往往不知通話對方的身份及電話號碼，應急通訊系統應能提供對講機通訊模式以增加救災團隊之間的通訊效率。此外，在進行救災任務時，常需連絡特定任務群組而非對特定個人，例如某一傷患需要外科醫生治療時，救援人員未必知道特定醫生，此時如有個代表外科醫生的群組通訊代表號，就可讓救援人員直接撥打代表號，即可連絡上任一個外科醫生。
- **適當的通訊品質服務(Adequate Quality of Service)**：救災工作常陷於兵荒馬亂，吵雜無比的環境中，良好的通訊品質可減少通訊連絡的失誤，降低救災任務忙中有錯的機會，以提升救災的效率。反之亦然。
- **長效的終端設備(Long Standing Time of Terminals)**：災區中的電力供應常常中斷，縱使有應急的發電設備，也是小規模居多。而隨身攜帶充電器的使用者為

數不多，因而在災區中為終端設備充電極為不易，終端設備待機時間的長短變成為一項重要需求。多年來我們一直致力於應急通訊網路的研究，我們發現在沒有備用電池的情況之下，筆記型電腦的待機時間只有大約 2 至 3 個小時。即使能夠找到電源進行充電，在使用 2 至 3 個小時之後，必須再進行充電。因此，如果能使用長時間的終端設備，提供至少 10-12 小時的使用時間，每天只須充電一次即可應付一整天的救災需求。

- **移動性(Mobility)**：災區中的使用者，一則多在戶外，二則常需移動，因此終端設備必須具備高度移動性。為了支援移動性，應急通訊系統使用無線網路將優於有線網路。由於移動性對大多數的通訊系統來說是不可或缺的，因此，所使用的終端設備需要為可攜帶性的設備。

II. 網路管理者需求

3. 實際可行性(Practicability)：實際可行性是應急通訊系統中的最基本也最首要的需求。

首先要考慮的是建置與操作應急通訊網路的可行性。需要符合以下三大特點：低開發及建置成本，建構速度快，設備容易取得（重型設備可能因道路中斷或地形阻隔而無法運入災區）。

- **低開發成本(Low Development Cost)**：比起正常的通訊系統應急通訊網路不從事營利，而使用應急通訊系統的機會也是備而罕用，導致市場狹小。通訊設備廠商缺乏誘因投資於應急通訊網路的研究發展。因此系統開發成本必須嚴格控制，而盡量使用災區倖存的通訊設備正是降低開發成本的一個重要手段。
- **易於獲取網路設備(Easy Acquisition of Equipment)**：由於災區之交通可能癱瘓，地形可能複雜，使外部援助難以運送進入災區。直升機或空投成為最終運輸工具。因此，設備的大小和重量應盡量縮小。相較於完全由災區外部運入的應急通訊系統，我們設計的應急蜂巢式通訊網路，利用既有的網路設備所建構，其優越性是顯而易見的。

- **建構速度快速(Easy and Rapid Construction)**：受困者的存活率取決於搶救時間。受困者越早獲救，存活率越高。存活率在 24 小時內是 90%，在 25 至 48 小時之間降至 50%，49 至 72 小時只剩 20%，超過 72 小時後，存活率不到 5%。為了挽救更多的生命，應急通訊系統應盡快建構。此外，專業技術人力可能不足，應急通訊系統可能需要由非專業人員協助建造，因此應急通訊系統之建構，應盡量減低對專業技術人力的需求。
4. **負載能力(Capacity)**：災區內的通訊需求量與實際承載能力可能存有極大差異，必須有適當的允入控制機制。以汶川地震為例，災區內部通話量為平時的 10 倍；外界至災區為平時的 5 至 6 倍；北京至災區則為平時的 80 倍，由此可知災區的通話量遠較平常高上數倍。應急通訊網路能負荷的通話量遠較正常時期之公眾網路小，難以容納如此大量的通話量，再者應急通訊網路不應支援與救災無關的通話。因此，應急通訊網路應具有適當的拒絕服務請求的能力，以免爆量的通訊要求造成網路擁塞。
5. **持續性(Sustainability)**：既有的通訊網路之搶修通常需時數天至數星期之久，以 921 地震為例，中華電信耗費 15 天，才全面恢復電信網路，因此應急通訊網路在一般的公眾網路恢復之前應穩定的運轉一段時間。以下是兩個主要性能因素：
- **可靠性 (Reliability)**：在資源許可之下，盡可能提高系統的可靠性，避免頻繁的中斷。
 - **快速恢復(Fast Recovery)**：應急通訊網路未必有超高的可靠度，在服務中斷後，應能輕易及快速的修復完成並且持續提供服務。
6. **可調性(Adaptability)**：災區中受災情況往往變化莫測，應急通訊系統應具備可調整的能力，在有限的資源下，盡力提供災區通訊服務，避免通訊資源的不當配置影響救災效率。調適方式可為手動或自動，視開發資源而定。
7. **維運性(Operability)**:運轉、營運、管理和維護功能稱為 OAM 功能。
- **OAM Functions**：當應急通訊網路因為不明原因損壞時，OAM 必須協助儘快找

出原因、排除故障，恢復運轉。OAM 的功能主要是指以下幾個方面：系統維護、數據管理、警告處理、測試管理、話務統計、環境與電源監控、操作權限管理、軟體更新管理。

1.1.6、應急通訊網效能指標

分析以上應急通訊系統需求，我們可以歸納出八項評斷指標：

- I. 使用成本：廣泛使用此應急通訊系統的成本
- II. 建置難易度：將此應急通訊系統建置起來的困難度
- III. 設備取得難易度：將應急通訊系統運送進入災區的難易度
- IV. 終端設備普及率：災區人員具有此應急通訊系統通訊設備的程度
- V. 終端設備操作難易度：災區人員操作此通訊設備的難易度
- VI. 終端設備可移動性：災區人員攜帶通訊設備移動的能力
- VII. 通訊品質：利用此應急通訊系統的通訊品質
- VIII. 系統運轉難易度：應急通訊系統建置起來後維持運轉的難易度

1.2、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)、移動式基地台等。而近年來有許多研究倡議使用 MANET (Mobile Ad Hoc Network) [4][11][38]建構應急通訊系統。這些系統或多或少存在一些缺陷，例如：

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，但它的缺點很明顯在於價格太高因而普及率低，一般人不會持有此種設備。

集群通訊系統具有充沛的調度功能(群組呼叫、優先分級、快速接續...等)，通常為

軍、警及專業救難隊等專業單位所擁有，需要專業團隊臨時建構才可使用，因係專用設備，價格昂貴無法供應大量的終端設備。

移動式基地台為國內最常見的應急通訊系統，常被當成是現場緊急通訊的首選方式，但因造價昂貴而數量稀少。

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，但其數量稀少，且通常是固定式，無法任意移動，而使用者須經由專業訓練才得以使用。

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，也不及行動電話之普及與方便。

Walkie-Talkie 則與上述所提幾種應急通訊系統不一樣，其優點眾多，包括體積小、重量輕、可供長時間使用、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，因此我們提出了兩種方法解決此問題，第一種為利用 MANET 應急通訊系統，稱為 P2PNet，在大型天然災害發生之初期，可迅速的讓受災人員與救難人員以自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，其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。第二種是應急蜂巢式行動通訊網路(Contingency Cellular Network, CCN)，利用現有的行動通訊系統中未損毀但失去電力或連網能力的基地台，以 Wi-Fi 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 1.4，本論文即是在此基礎(CCN)上進行研究。

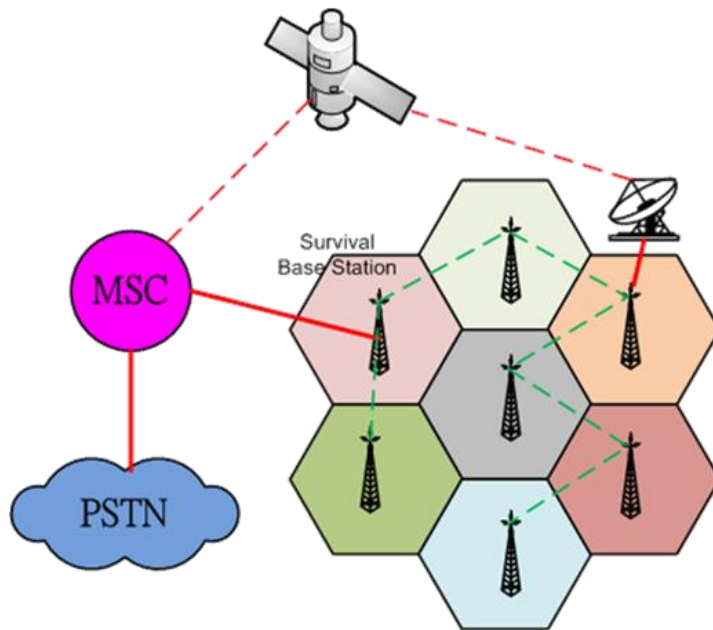


圖 1.4、CCN 網路

1.3、論文架構

本文共分成六章，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。第三章說明如何用現行的 3G 行動通訊網路來建構一個應急通訊系統。第四章介紹本論文提出的考慮資源運輸路徑之排程方法，並針對此應急通訊系統的建構排程問題，提出合適之模型與演算法。第五章則藉由實驗數據驗證本論文提出之方法可行性。第六章則為結論與未來發展。

第2章、相關研究

2.1、第三代行動通訊架構

第三代行動通訊簡稱 3G (3rd-generation)，是指高速數據傳輸的蜂巢式行動通訊技術。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(1G)類比式行動電話系統與第二代(2G)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之 GSM、CDMA 等數位調變式手機，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[42]，如圖 2.1，其中，UTRAN 用於處理所有與 Radio 相關的功能，而 CN 則處理行動通訊系統內的所有語音呼叫和資料傳輸與內外網路間的交換與繞送。

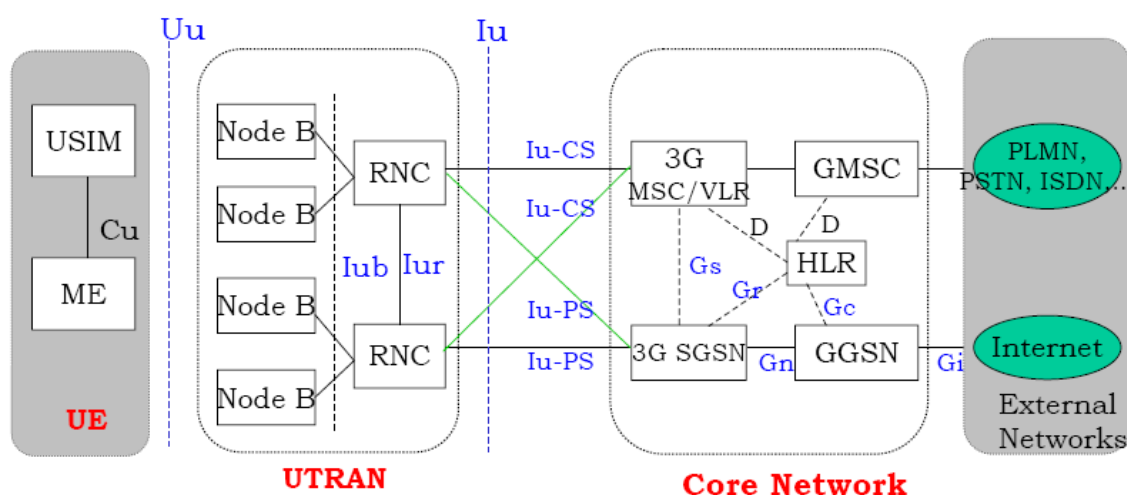


圖 2.1、System Architecture of 3GPP Release 99

2.1.1、通用行動通訊系統陸地無線接入網 (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 介面相連，每一個 RNC 透過 Iu-PS 介面與一個 SGSN 相連，並透過 Iu-CS 介面與一個 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以及無線連結維護，意即 RNC 控制管轄範圍內所有 Node B 的無線電資源，包括無線電頻道的指配、回收與管理，作為 Service access point 提供服務給 Core Network。以台灣而言，一台 RNC 大約控制 50~300 座基地台。
- **Node B**：即是基地台(Base station)，配備收發天線及無線電頻道，提供無線電通道資源，通過 Iub 介面和 RNC 互連，主要處理與 UE (User Equipment)間 Uu 介面實體層協議。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[29]。

2.1.2、核心網路 (Core Network)

核心網路(Core Network)分為 CS-CN (Circuit Switched Core Network)和 PS-CN (Packet Switched Core Network)，由 CS 交換機、PS 路由器、資料庫及長途幹線組成，主要設備存放於電信機房中，包含了 HLR、MSC/VLR、GMSC、SGSN、GGSN 五個部分：

- **HLR (Home Location Register)**：本籍位置記錄器，是一永久性用戶資料庫，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、是否開機等)。行動業者所有客戶的 Service profile 都儲存於 HLR，直到客戶退租為止。
- **MSC (Mobile Switching Center)**：行動電話交換機，負責所管轄服務區內行動客戶

的移動管理及呼叫處理。

- **VLR (Visitor Location Register)**：訪客位置記錄器，通常每個 MSC 都有自己專屬的 VLR，以記錄當時正漫遊在其服務區內的行動客戶相關資料，如客戶目前所在位置區、Service profile...等。
- **GMSC (Gateway MSC)**：閘口行動電話交換機，提供 CS domain 連接到外界 PSTN (Public Switched Telephone Network)或其他 PLMN (Public Land Mobile Network)的交換機。
- **SGSN (Serving GPRS Support Node)**：負責數據封包的 Mobility management、路由轉發、會話管理、邏輯鏈結管理、加密和輸出等功能。
- **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 [42]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 連線，然後 SGSN 會再用 Gn/Gp 介面與 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除封裝送到 Internet，如圖 2.2。

3GPP R5 to R7：新的 3GPP R5 到 R7 版本[42]，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，再來是修改 SGSN，讓 RNC 可以直接與 GGSN 使用 Tunnel 連線，如圖 2.3，在 HSPA+的版本，又將 RNC 的部分功能移到 Node B 上面。

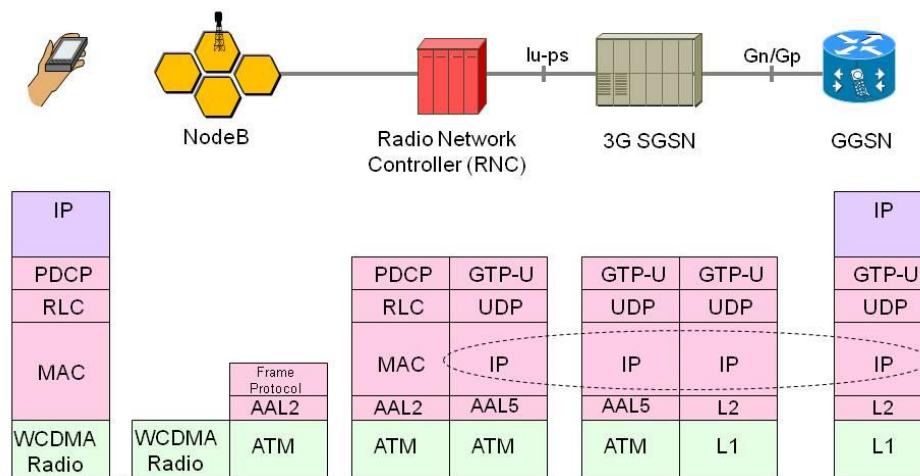


圖 2.2、3GPP Release 5 架構

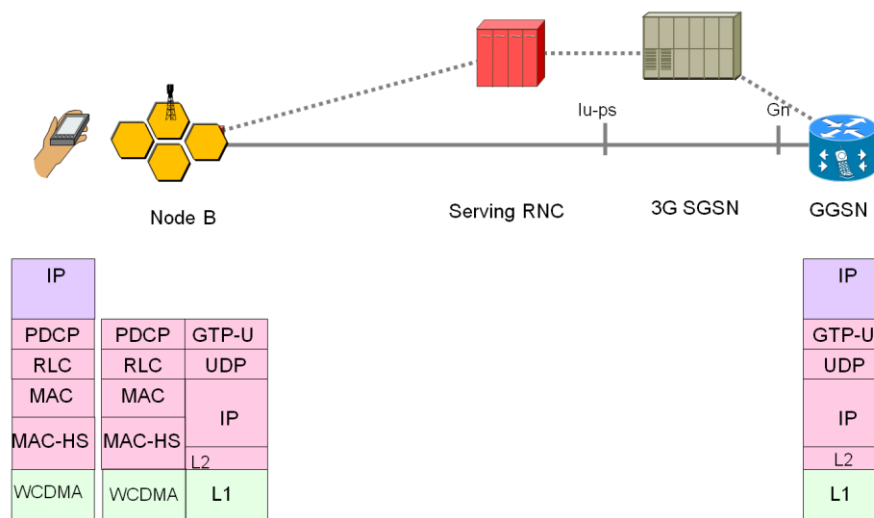


圖 2.3、3GPP Release 7 架構

2.2、應急通訊系統種類

■ 應急通訊系統使用時機

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，快速建置一個應急通訊系統可在此時提供倖存者及救災人員所需的通訊服務，隨著時間的推移，電信公司將會逐步修復原有的行動通訊系統，而應急通訊系統的作用

也會慢慢下降直到所有基地台修復完成，如圖 2.4 所示。

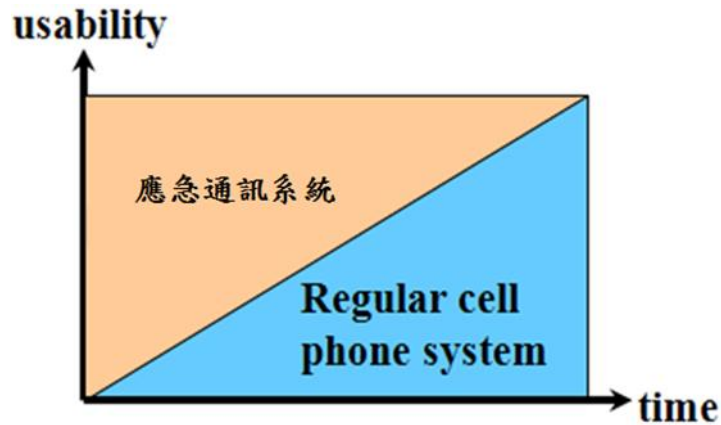


圖 2.4、應急通訊系統使用時機

我們以 1.1.5 所歸納的需求，檢視分析現有應急通訊系統的適用性。

2.2.1、專用高抗災通信平臺

專用高抗災通信平臺[51][52]為國家通訊傳播委員會於莫拉克風災後，協調地方政府與電信業者共同出資，就偏鄉通訊設施進行改善，於高雄那瑪夏、茂林、桃源、六龜、杉林、鳳山等 6 處，以「消防救災體系與行動通信系統結合」、「整合光纖、微波、衛星鏈路形成多重中繼傳輸備援路由」及「加強電力備援系統」等設計理念完成之應急通訊平臺。其佈建的方法為在災前預先佈建強固機房並於特定基地台佈建衛星、微波等無線通訊設備，以確保政府救災體系緊急通訊順暢。

■ 優點：

- 災前即已佈建完成，災難發生時，馬上就可以使用。
- 系統可靠性高。
- 結合行動通信系統與消防救災體系。

■ 缺點：

- 由於成本過高，佈建數量極為有限，無法全面佈建，僅能佈建於少數具高潛在危險的特定區域。

2.2.2、無線對講機 (Walkie-Talkie)

無線對講機(俗稱 Walkie-Talkie)是一種手持的雙向無線電收發器，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) [23]。無線對講機不需佈建通訊網路，只要雙方擁有無線對講機即可進行通話。

■ 優點：

- 不需佈建通訊網路即可使用。
- 體積小、重量輕，可隨身攜帶。
- 電池充電後可長時間使用。
- 電波所及範圍內即使不知道對方身份、地點也可通訊。

■ 缺點：

- 在世界很多地方普及率低，例如：台灣在八八水災中，政府花了 7/14 天從廠商借得 240/1052 支無線對講機，太少也太慢。
- 需要簡單學習才能使用，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，尤其是普及率低的地方。
- 沒有優先分級能力。

2.2.3、業餘無線電 (Amateur radio)

業餘無線電[9][19]，俗稱火腿(Ham radio)，與無線對講機相似，但通訊的距離較遠。其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。

■ 優點：

- 不需佈建通訊網路即可使用。

- 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。

■ 限制：

- 普及率低，擁有業餘無線電設備的民眾非常稀少。
- 使用困難，需要執照方能操作，但擁有執照的人員數量稀少，例如：在八八水災中，7 天之後第一座業餘無線電台才架設完成。
- 行動力低。

2.2.4、行動衛星通訊 (Satellite mobile phone)

1990 年代數個使用非同步衛星的行動衛星通訊系統被提出，例如 GSM 的衛星版—Motorola 的鉞計劃(Iridium)、IS-95 的衛星版—Qualcomm 的全球通(Globalstar)，主要提供語音以及低速率資料傳輸服務[44]。

為了降低延遲時間，避免通話受到干擾，行動衛星通訊大多使用軌道高度 10,000~20,000km 的中軌道衛星搭配 750~2,000km 的低軌道衛星通訊系統。衛星就像不斷移動的基地台，一般而言，中軌道衛星繞行地球一周約為 6 個小時，而低軌道衛星繞行地球一周則約為 100 分鐘。由於衛星高掛於太空中，不受地震等天災影響，在災害來臨時，成為一個可靠的應急通訊系統，但因其使用頻段較高，易受惡劣天候影響。衛星造價高昂且使用者數量稀少，導致通話費用極高，雖然行動衛星通訊手持設備可以僅操作在衛星通訊模式，但因通話費用因素，一般皆為衛星/地面蜂巢式雙模手持設備，在地面蜂巢式行動通訊系統的服務範圍內，優先使用地面蜂巢式行動通訊系統，否則使用含蓋範圍廣的行動衛星通訊系統，如鉞計劃和全球通[27][44]。

鉞計劃 (Iridium)：鉞計劃為包含 11 個衛星軌道平面，66 顆衛星的行動衛星通訊系統。每顆衛星重 689 公斤，衛星間可以彼此通訊。運作於 2001 年 3 月，提供語音、傳真、資料和 GPS (Global Positioning System)服務，採用分頻多工結合分時多工技術及

QPSK 調變技術。

全球通 (Globalstar):全球通開始營運於 2000 年春天，為一包含 6 個衛星軌道平面，48 顆衛星的行動衛星通訊系統。全球通每顆衛星重 450 公斤，衛星間彼此無法通訊。提供語音、傳真、資料、GPS 和 paging 服務，使用分碼多工技術和 QPSK 調變技術。

■ 優點：

- 覆蓋面廣，通訊距離遠。
- 不受地震等天災、地理條件影響限制。
- 可隨身攜帶。
- 易於實現多點通訊、具有優良的廣播特性。

■ 限制：

- 價格高昂。
- 普及率非常低。
- 易受氣候影響。

2.2.5、專業用集群通訊系統 (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[46]。其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，例如：美規的 Project 25[1]以及歐規的 TETRA[2]應急通訊系統。由於需專業人員架設，一般未經訓練的民眾不會使用。因此，主要使用者為軍、警或專業救難團隊。

Project 25: APCO (Association of Public-safety Communications Officials-international) 於 1989 年推動的計劃(簡稱 P25)，制定了相關標準來提供服務以及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 具備支援類比/數位(analog/digital)之中繼集群模式，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類比模式，否則通常採用數位中繼集群模式。

TETRA (Terrestrial Trunking Radio)：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。

■ 優點：

- 通訊網路架設快。
- 涵蓋範圍廣。
- 可靠性高。

■ 限制：

- 話機數量有限。
- 需經專業訓練才會使用。
- 適用於特定使用者，主要為軍、警或專業救難團隊。
- 因體積、重量過大無法空投，如果交通系統癱瘓，不易運送至災區。

2.2.6、移動基地台 (Cell on wheels)

移動基地台實際上就是一個可移動的通訊系統，透過開到現場的車載平台，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，在大多數天然災害、突發事件和重大事件發生的情況下，應急通訊車通常是現場應急通訊的首選方式之一，但專業設備需專人操作，且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，在大型天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，相較於數千座癱瘓的基地台，數量遠不敷所需。

■ 優點：

- 佈建速度快。
- 擁有行動電話之一般民眾皆可使用。

■ 限制：

- 造價高昂數量不足，無法大量部署。
- 因體積、重量過大無法空投，如果交通系統癱瘓，不易運送至災區。

2.2.7、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，並由這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，如兩個相距較遠的設備要進行通訊時，須藉由中間節點的轉發才能實現[11][38]。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。

在我們的先前研究中，提出一個 P2Pnet，利用志願救災人員的筆記型電腦等建構成 MANET 網路平台，再利用 VoIP 技術實現應急通訊系統，稱為 P2Pnet [18][24][25][26]，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[23]。

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。[3][18][21][24][25][26][35][36]

這個系統可以提供部分的救災人員使用，但是仍有未足之處。一是可以支援的人數僅限於擁有筆記型電腦的人，遠不敷所需，而擁有行動電話手機的人數遠高於此，幾乎人人攜帶行動電話，二是 P2Pnet 系統仍須一定的專業知識才能安裝並操作本系統，三是筆電內建的 Wi-Fi 的通訊距離較短，雖經過 Multi-hop 轉接，能支援順暢通話的 VoIP 仍然不能及遠。

■ 優點：

- 可以使用災區內志願救災人員的筆電等設備就地取材來建構，節省大量經費。
- 不受交通系統癱瘓之影響，就地取材，立即建構，在第一時間投入救災。

■ 限制：

- 使用者必須具備建置系統的技術知識，並非一般使用者可以使用。
- 具通話品質的 VoIP 之有效距離極短，有待克服。
- 尚在實驗階段，並無成熟產品，尚須繁複的設定方能使用。

2.2.8、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 [38]：本篇主要探討災害發生後，如何利用自主機器人在災區內建立 Wireless ad hoc network，使受困的災民能和外界通訊。作者將事先預估的人群分布資訊搭配分散式演算法運行於機器人上，並藉由最小生成

樹(Minimum spanning tree)演算法來改進服務範圍重疊之缺點，當機器人之數量較少時使用第一種演算法較為合適，若該區的頻寬需求較大時，則使用第二種演算法較為合適。

- Autonomous Community Construction Technology for Timely Transmitting Emergency Information [41]：本篇主要探討 Wireless sensor network 在災害發生時的相關應用，目前 Wireless sensor network 已被廣泛應用於應急管理體系 (EMS)。由於一般即時發送緊急訊息的系統均使用集中式的管理，並不適合救災情況會隨時變動的災區使用，因此，作者提出利用 Autonomous Decentralized System (ADS)的方式在災區傳送即時訊息，稱為 Autonomous Community Construction Technology。最後經由模擬可知，作者所提出的方法非常適合在災區傳送即時訊息。
- An Integrated Communication-Computing Solution in Emergency Management [6]：本篇作者在多年來致力於緊急情況和危機管理系統之研究，在本文中，作者藉由 Multiple Parallel 的模型來整合網路層(Network layer)和網路應用層(Application layer)，在不同層的節點可以依據其他節點所給予的資訊來選擇最適當的網路，作者期望能在災害發生時，建立一個較穩定的異質通訊網路(Heterogeneous meshed communication system)。
- Computational Public Safety in Emergency Management Communications [33]：本篇主要探討應急管理系統中的各種不同類型之網路，作者討論了各種無線網路的通訊方案，並分析其使用的可行性。例如，傳播延遲，封包傳遞率和傳輸率。針對 WiMAX、Xbee、藍芽和 Wi-Fi 等環境進行測試，作者發現 Wi-Fi 是目前應急管理最適合的網路環境，其網路的傳輸速率和涵蓋範圍都優於其他環境。
- Ad Hoc Communications for Emergency Conditions [11]：本篇主要探討在災區的通訊方式，作者提出利用智慧型手機作為節點，並透過 Wi-Fi 的方式建立成 Ad hoc 網路進行通訊。整體架構包括廣播和路由協議，其主要的方式為，將訊號向間隔 120 度

角的節點進行廣播並建立起網路拓樸，在緊急情況之下，即可架設短距離的 Ad hoc 網路供通訊使用。

- Taiwan Earthquake Event Report, Risk Management Solutions [9]：本篇主要探討在 1999 年 9 月 21 日發生於台灣的一個大地震，集集(chi-chi)大地震，作者將此次地震所造成的相關災害透過照片和圖表等數據來說明，其中包括許多交通建設的損壞、房屋倒塌、電力系統中斷和基礎設施損毀...等，並於本文各章節中探究其原因，提出適當的建議，以供相關單位可針對目前缺失的部分進行補強，減少人員的傷亡和財務的損失。
- Improving Disaster Management [39]：本篇主要探討在災區的訊息傳遞方式，作者認為有效的訊息傳遞在災害發生時是非常重要的，無論是災情的傳遞或是受困的災民都急需訊息傳遞的需求。因此，學者們制定了一套災害管理系統，利用手機當作節點來傳遞訊息，如此一來，當災害發生時，即可迅速的將訊息透過節點和節點之間的 relay 傳送至目的地。
- A Disaster Information System by Ballooned Wireless Ad Hoc Network [36]：本篇主要在探討如何在通訊網路全面中斷的情況下，建構一個可以使用的網路環境。作者將無線傳輸設備綁在多個氫氣球上，在距離地面約 40m~80m 的空中建立起 Ad hoc 網路，恢復某一特定地區的網路。作者於某一校園進行實際測試，並透過一中央伺服器(WIDIS)將該區的資訊與外界相連。
- Construction of Wireless Network for Information Communication for a Disaster-affected Island [40]：本篇作者主要探討災害發生後，海底線路損毀，造成大陸本島與其相鄰小島之間的通訊中斷，並提出相關的應對方法。研究團隊將無線發射器與大範圍指向性天線架設在船隻上，利用海上的船隻當作訊號傳送的中繼節點，將大陸本島的網路訊號透過無線傳輸的方式傳送到相鄰的小島上，藉此恢復陸地與小島的通訊。

- Network Topology Planning for Contingency Cellular Network [14]: 本篇主要探討災害發生後，災區通訊網路基礎設施常因災害遭受嚴重損毀，無法正常運作，在缺乏通訊系統的支援下，大大提高救援的困難度。作者提出一個可快速恢復特定區域通訊服務的網路，並為其設計通訊的拓樸結構，不但將通訊品質納入考量，還考慮拓樸結構中的流量負載，以避免建立通訊流量過度集中及訊號衰減過大的拓樸。文末，提供數個演算法，用以建立應急蜂巢式行動網路拓樸。

2.2.9、應急通訊系統綜合比較

表 2.1 以應急通訊系統八項評斷指標進行優劣分析，沒有任何一種應急通訊系統是完美無缺的，以簡單易行的 Walkie-Talkie 而言，在很多地區有極高的普及率(例如：美國)，但在台灣其普及率極低，就無法在災害發生時投入應用，畢竟一般民眾不願為了機率極低的大型天災而隨時隨地攜帶手機以外的通訊設備。其他幾種通訊系統或多或少都有缺陷，無法支援大量志願救災人員及災民的通訊需求。

而近年來的很多研究主要在利用隨意網路或者異質網路來建構一個 MANET 網路以支援應急通訊，這些研究也都面臨一些客觀環境上的問題，我們設計實驗過的 P2Pnet，當距離超過 25m 時，封包將會大量遺失，VoIP 的語音品質大受影響，這種系統所能發揮的功效還是非常有限，如果有一個系統能快速方便的讓民眾的手機恢復部分通訊功能，例如在基地台的範圍內網內互打，或當成無線對講機使用，將可以對救災工作提供更大的方便。再者，在資通傳資源極為有限的情況之下，我們應該盡可能利用所有可利用的資源投入救災工作。因此，設法讓斷訊手機恢復部分通訊功能成為一個值得研究的問題。我們所提出的應急蜂巢式行動通訊網路可以更有效的支援大規模的災區內行動電話用戶。

表 2.1、應急通訊系統比較

	使用成本	建構難易度	設備取得難易度	終端設備普及率	終端設備操作難易度	終端設備移動性	通訊品質	運轉難易度
專用高抗災通信平臺	高	不需建構	易	低	易	低	高	中
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	需專業執照	低	中	無
行動衛星通訊	極高	既存	易	低	易	高	中	無
集群通訊系統	高 (量少)	簡單	難(需道路運送)	低	需簡單學習	高	高	中
移動式基地台	高 (量少)	簡單	難(需道路運送)	高	易	高	高	中
MANET	低	需專業安裝設定	就地取材	中	易	中	低	中
CCN 應急通訊系統	低	中	重量輕 可空運	高	易	高	高	中

第3章、應急蜂巢式行動通訊網路 (CCN)

應急蜂巢式行動通訊網路(Contingency Cellular Network, CCN)，以下簡稱 CCN 網路。CCN 網路的建置目的是在大型災害發生後的交通及通訊尚未修復之緊急黃金 72 小時救援時期，提供即刻臨時性的通訊。CCN 網路建置於原 2G、3G 蜂巢式行動通訊網路之上，係利用在災害中服務中斷但結構完好的行動通訊網路基地台，藉由臨時供應的電力，恢復其運轉，並藉由無線通訊設備[24]連接鄰近的基地台，再藉由點對點(Hop-by-hop)方式遞送訊號至有對外連線能力之基地台或通訊節點(如衛星通訊裝置、移動基地台等)，形成一臨時性的通訊網路，使災區人員能藉由 CCN 網路利用手機進行通訊，降低災區的通訊阻礙，進而提升救災效率。

CCN 網路利用現有的 2G、3G 蜂巢式行動通訊網路建構，由於手機、平板電腦等行動裝置在現今社會中十分普及，使用者不需要持有額外的設備即可加入 CCN 網路進行通訊，且在大型災害中，基地台運作停擺多肇因於停斷電及後端線路(Backhaul)的毀損，大部分的斷訊基地台，其實本身並無損壞，若可直接利用這些既有的基地台設備，則 CCN 網路的建置成本可大幅降低。再者，蜂巢式行動通訊網路的基地台分佈拓樸，在建置時期即經過完善的設計，不但設置的地理位置經過測量規劃，而且多設置於高處，訊號良好、無 Line-of-sight 問題，故在緊急時期，若可直接使用蜂巢式行動通訊網路的基地台，則不須花費額外的人力及時間選擇 CCN 網路的無線電站台架設地點，可以快速佈建。另外，建置 CCN 網路所需的額外設備(應急修復包)具有可事前儲備、輕量等優點，當災害來臨時，可直接由災區內部供應或由直升機空投取得，使得系統建置更為容易。

圖 3.1 所示，CCN 利用無線通訊設備連結斷訊的基地台群，並將訊號遞送至具備

與後端核心網路通訊能力之基地台，藉此使斷訊基地台回復通訊。

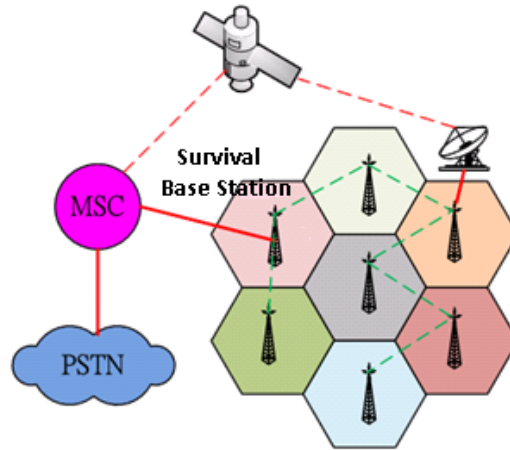


圖 3.1、CCN 網路架構

■ CCN 可行性分析：

我們利用原有基地台建構應急通訊系統的方式具有下列幾項優勢：

- I. 一般民眾逃難時，多半攜帶手機，如能救活基地台，讓受困與救災人員立即恢復通訊能力，效益極大。
- II. 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。
- III. 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，延誤建構應急通訊系統的時間。
- IV. 額外設備(應急修復包)重量極輕，可以用直升機空運或空投。
- V. 基地台的拓樸，多半經過精心設計，地點絕佳，不需耗時費力選擇無線電基站地點。
- VI. 基地台之間距離不遠，且都在高處，不虞視線受阻，容易用簡單的無線電設備互連。

手機普及率非常高，使用者可直接用既有手機不需改裝即可通訊，且使用者不需訓練，只要會使用原有手機即可使用(但通訊功能僅限於救災使用)。

3.1、系統架構

我們先前的研究[23][24][47]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應急修復包(Contingency Recovery Package, CRP, 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，藉由這些基本設備，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，使其能連上後端核心網路，恢復部分通訊功能。應急蜂巢式行動通訊網路(Contingency Cellular Network)架構如圖 3.1 所示。

我們將基地台依其對外通訊的連線能力，定義如下：

- **連網台(Survival base station)**：與後端核心網路正常通訊，傳遞資料並持續提供服務之基地台，稱為連網台，如圖 3.2。連網台的構成要件為其基地台本身完好或配有衛星等通訊設備，使其具有與後端核心網路通訊之能力。

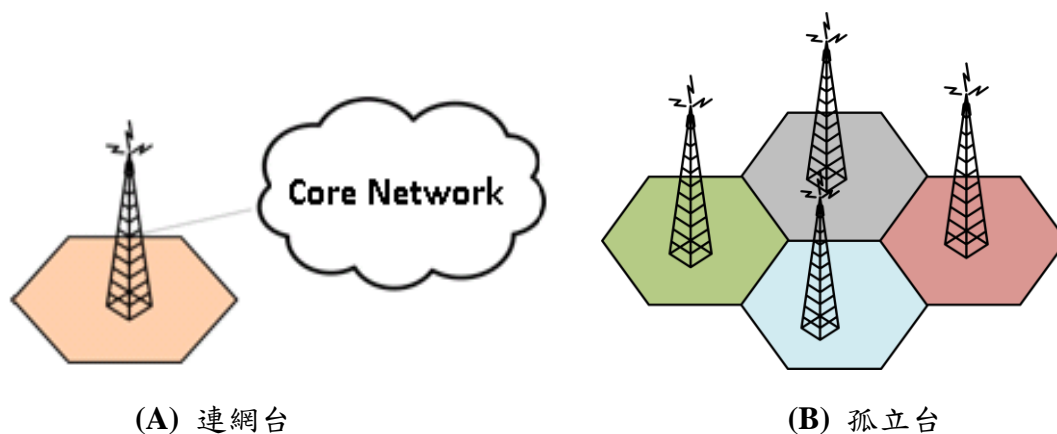


圖 3.2、連網台與孤立台

- **孤立台(Isolated base station)**：無法與後端核心網路通訊，造成通訊服務提供中斷，但基地台設備本身並無毀損之基地台，稱為孤立台，如圖 3.2。其形成孤立的原因可能為電力中斷、實體對外線路損毀所導致。

3.1.1、CCN 連網方式

當兩基地台彼此相鄰，可用無線通訊方式連線互相交換資料時，這兩基地台互為彼此的鄰台。

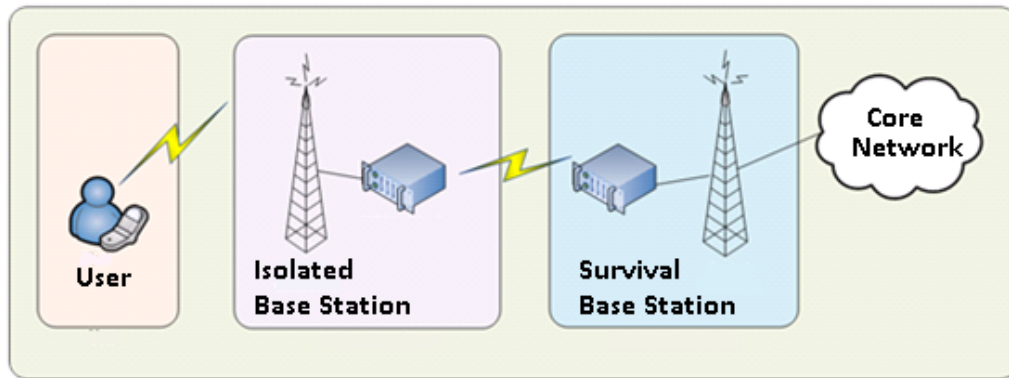


圖 3.3、CCN 基地台間之連線

CCN 網路即是在孤立台上佈建簡易的輔助設備「應急修復包(CRP)」，使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 系統為 BSC (Base Station Controller)及 MSC (Mobile Switching Center)，而在 3G 系統則為 RNC (Radio Network Controller)及 MSC、SGSN (Serving GPRS Support Node) 等，基地台必須與它們建立連線才可交換信令(Signaling)與資料(Data)提供電信服務，如圖 3.3。CCN 將利用各種無線連線方式[33]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。CCN 通訊模式如圖 3.4。

■ 轉送拓樸 (Forwarding Topology) :

CCN 的大架構是一個轉送拓樸，轉送拓樸擁有數個連網台，其餘皆為孤立台。孤立台使用多重跳接的方式連接到鄰近的連網台，孤立台才得以對外連線溝通。每一個連網台與數個孤立台的連線間接形成了一個的 CCN 樹狀轉送拓樸(Forwarding Tree)，因此可將轉送拓樸視為是由多個樹狀拓樸所組成。但由於連網台對外的頻寬極為有限，每一個連網台能負荷的孤立台有一定之數量，若某一樹狀拓樸中的孤立台數量太多，可能造成話務須經多次轉送、頻道壅塞、浪費太多頻道資源與通訊品質不良等問題，由轉送拓樸切

割成多個樹狀轉送拓樸時應審慎評估。

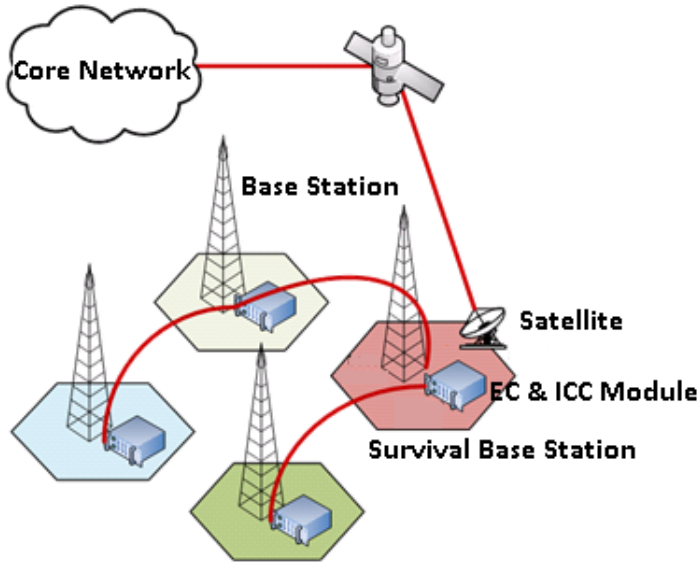


圖 3.4、孤立台藉 CCN 連線連上核心網路

3.1.2、系統元件

CCN 之建構需要在各孤立台附加額外設備，所需的元件為發電機、燃油、無線通訊設備，部分則需衛星通訊設備。平時包裝成「應急修復包(Contingency Recovery Package)」，如圖 3.5，儲存於國家救災單位，於災害發生時運送至災區建構 CCN。相關元件介紹如下：

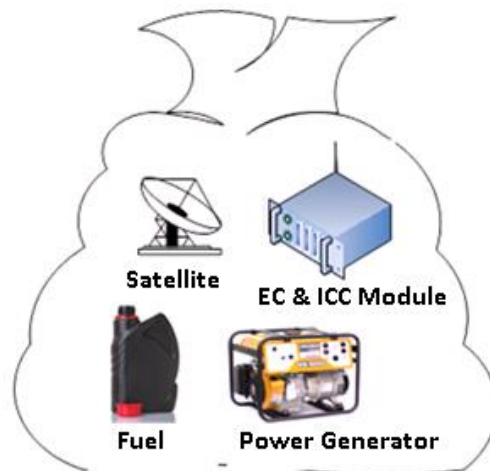


圖 3.5、CCN 應急修復包(CRP)

- **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(如前所述，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)，因此能讓基地台持續運轉的小型輕便發電機及燃油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達成，包括「孤立台—孤立台」以及「孤立台—連網台」間的連接，均須仰賴 ICC Module 的無線通訊能力(例如：WiMAX、Wi-Fi)。孤立台藉由 ICC Module 與鄰台相連進而恢復連外能力，最後與後端核心網路相連，達成資料遞送。
- **Emulated Controller Module (EC Module)**：負責處理、運算、轉送資料，為 CRP 的核心運算元件，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、通話允入控制等，並提供虛擬鏈結讓基地台與後端連線，也須提供虛擬 BSC 功能提供網內互打服務[29]。EC Module 也是 CCN 與原先受損的行動通訊網路系統間之通訊介面，CCN 藉由此模組與既有的行動通訊網路介接，交換控制信令與音訊。
- **Satellite Communications Module (Optional)**：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且增加了斷訊的機會，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。
- **CCN 孤立台恢復連線方式**：如圖 3.6 所示。手機信號透過孤立台 EC Module 處理並經 ICC Module 轉送至鄰台。此時孤立台資料便可透過鄰台，以多重跳接的方式，

連上後端核心網路。

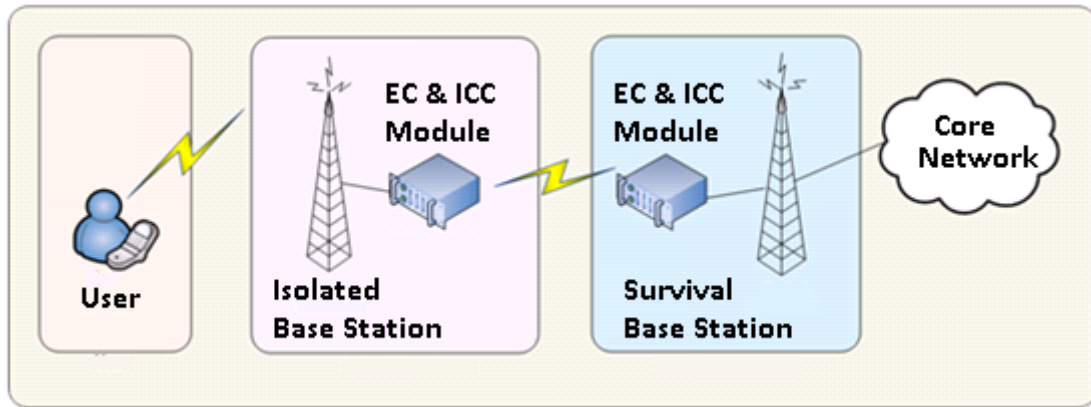


圖 3.6、孤立台恢復連線方式

除了衛星通訊模組之外，其他硬體模組造價均很便宜，一般電腦搭配通訊模組即可，即使經費有限，也可大量佈建，而且不需購置終端設備，使用者使用自己原有的手機即可，每個使用者之單位建置成本可大幅下降。

3.2、通訊模式

3.2.1、災區通訊型態分類

我們將災區通訊依通話功能大概分為四類，分別為急難救助、災情回報、救災相關和互道平安。詳細分類如表 3.1 所示。依急迫性由高而低排序，依序為急難救助、災情回報、救災相關和互道平安。「通話分類」欄位中的數字代表該通話類別的急迫性，數字越小急迫性越高。

表 3.1、災區通訊需求分類

通話分類	通話功能	發/受話端	說明
1	急難救助	受困人員 → 災區(外)民眾 緊急救難人員 緊急救難中心	受困民眾對外求援之用
2	災情回報	受困人員 ↔ 緊急救難中心 災區居民 ↔ 緊急救難中心 緊急救難人員 ↔ 緊急救難中心	各地災情狀況之回報、傳達
3	救災相關	緊急救難人員 ↔ 緊急救難中心 緊急救難人員 ↔ 緊急救難人員 緊急救難中心 → 受困民眾/災區居民 災區(外)民眾 → 緊急救難中心	救災資源分配、指揮調度、救災人員追蹤以及二次災害之預警等
			A 緊急醫療救護及協助
			B 醫療設備、器材、人員之調度
			C 緊急救難機具及設備調度
			D 二次災難預警廣播
			E 受災民眾協尋作業
			F 救援物資之協調、調度、補給、發放
			G 緊急救難人員調度、狀況回報
			H 後續就醫治療及照護
4	互道平安	受困民眾/居民 ↔ 災區居民 受困民眾/居民 ↔ 災區外民眾 緊急救難人員 ↔ 緊急救難中心 緊急救難人員 ↔ 緊急救難人員親屬	脫困人員報平安、災區外人員詢問 災區內民眾狀況

3.2.2、CCN 通訊模式

經分析災區通訊需求，我們規劃了數種通訊模式：一般模式，無線對講機模式，以及群組通訊模式。

■ 一般模式(Ordinary Mode)

此即原有的行動通訊模式(POTS)，在 CCN 服務範圍內的手機可與 CCN 內外任何電話通

訊，使用者不須進行任何設定，即可如常使用這個模式。但此種通訊模式必須佔用寶貴的連外資源，因為此類通訊必須經過交換機進行交換，必須使用原有之 HLR/VLR 等資料庫，所佔用的 CCN 資源非常可觀。必須配備具有差異化優先等級的允入控制，方能有效運用寶貴的資源。

■ 無線對講機模式(Walkie-Talkie Mode)

在正常運轉下，基地台都可以擁有上百個無線電頻道供使用者使用，而 CCN 內的基地台所獲配的連外頻寬通常將遠低於所獲配的無線電頻道，造成無線電頻道之閒置。我們設計的無線對講機模式利用閒置的頻道讓 CCN 服務範圍內的手機在同一基地台涵蓋範圍內進行廣播通話(一對多)，而不必使用連外頻寬。

無線對講機模式對災區內的使用者特別有用，因為志願救災人員以及受困災民彼此之間並不熟識，無由得知欲通話對象之電話號碼，故沒有限定接收者的廣播模式非常適合於災區使用，在緊急中也免去撥號碼的不方便，受困民眾更可能是在黑暗中，或困在狹小空間中，無法方便撥號。因此，這個模式相當有用。以下是幾種可能的方式，讓不熟悉此操作模式的使用者在災變來臨時，迅速瞭解並啟動此模式：

● 由 119 轉告

一般民眾在緊急時，最可能撥的號碼是 119，但在災區內，連外的 119 也可能不通。當 CCN 建置完成後，可以在災區內設立 119 專線（稱為 CCN-119，以與正常 119 區別），就近處理求援電話。可由 CCN-119 轉告使用者，撥一個特定的號碼，(例如 118)，即可將手機轉變成無線對講機。

● 國家指定特殊緊急號碼

為了減輕 CCN-119 的負擔，以及節省使用者的時間，可由國家指定一個特殊緊急號碼(例如 118)並廣為散佈，不過，因為此種號碼很少有使用的機會，民眾不易熟記，也浪費了一個國家編碼，因此國家願意為救災而採用此方式的意願不得而知。

- **由 CCN 基地台傳送簡訊到服務範圍內所有手機通知撥號方式**

此法很容易實行，可是受困的使用者如果不方便讀取簡訊，將會錯失與救援人員聯繫的機會。

- **群組通訊模式(Agency Mode)**

災區中，大量的救災通訊都具備幾個共同特色：

- **彼此不知電話號碼，也沒有時間記憶或記錄對方號碼**

災害發生初期，大部分依靠在地的志願性救災人員，臨時組成的團隊，彼此不知對方的電話號碼，而且在兵荒馬亂之中，亦無時間記憶，也很可能不方便將電話號碼存入手機之中。

- **大部分的通訊是對特定角色，而非特定個人**

例如：向任一個骨科醫生求救、需要某型血液進行輸血急救、需要重機械支援、需要油壓剪切斷鋼筋、向指揮中心回報狀況...等。這些通話對象，均非特定個人，而是一個特定角色。

- **緊急程度較高，且各有不同**

上舉數例都是緊急任務，比一般電話較為緊急，但彼此緊急程度則有所不同。

由於以上的特點，我們規劃了**群組通訊模式**，遵循以下數個原則：

- 群組通訊模式內不同的群組各自負責不同的任務，這些不同的群組稱為 agency group
- 每一個群組，指定一個代表號，各群組有優先順序，代表號則於災區廣為宣傳
- 每一個群組的成員註冊其電話號碼及所屬群體
- 呼叫某一群組時，該群組所有成員的手機都會接到來電訊號
- 受話群組的任一個成員可以承接呼叫

註冊成員的電話號碼，將記錄於 CCN 資料庫中，CCN 的管理者可根據各個群體的優先次序進行允入控制，此外，在允入一般模式的電話時，也可根據這個資料庫決定各

個電話號碼的優先度。群組通訊模式利用閒置的頻道讓 CCN 服務範圍內的手機可與 CCN 內任何受話群體內的任一個成員通話(非廣播通話)，無論呼叫者和受話者是否在同一基地台的涵蓋範圍之下。呼叫某一群體時，該群體所有成員的手機都會接到來電訊號，受話群組的任一成員可以承接呼叫，當群組內任一成員承接呼叫後，呼叫者即可和該成員進行通話，而此時群組內其他成員的手機的來電訊號即會消失(一對一)。

此外，如果 CCN 無法取得 HLR 的資訊時，原先的電話號碼在內部無法被 Terminate，換言之，原有電話號碼在 CCN 內部是無效的。為解決此問題，CCN 管理人員可以臨時指派號碼給使用者。而號碼的指派，可參考國家編碼計畫，採用 12 或其他 1 字頭的三位或四位號碼，簡化撥號。CCN 臨時指派號碼可透過現場公告、簡訊或 CCN-104 查號等機制提供給 CCN 服務範圍內的使用者查詢使用。

3.3、CCN 功能架構(Functional Architecture)

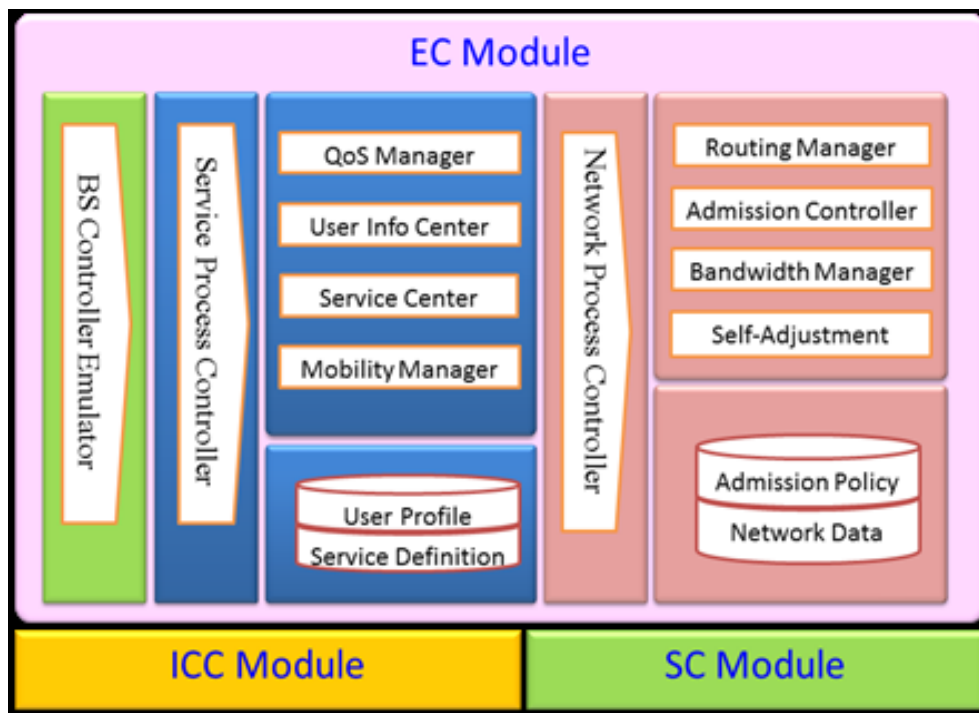


圖 3.7、CCN 功能架構

系統架構如圖 3.7 所示，EC Module 內各元件的功能詳述如下：

- **服務流程控制器(Service Process Controller)**：服務流程控制器和相關的功能元件相互合作，提供使用者服務。
- **用戶個人資料(User Profile)**：記錄一些用戶資訊，例如手機號碼、IMSI、agency group、service level agreement (SLA) 等。
- **服務定義(Service Definition)**：定義 CCN 通訊模式提供服務的流程及其功能。通訊模式可以被新增或修改藉由新增或修改它的服務定義。
- **虛擬 BSC(BS Controller Emulator)**：處理基地台之間的通訊協定和將信號及資料轉換到 VoIP 封包之中。現今的行動網路是由多家電信公司的基地台組成，電信公司的軟體和硬體更新時，若 EC Module 沒有立即更新，可能無法正常的運作。EC Module 是一種應急設備，並無商業利益可圖，不易獲得充裕的資金及資源供 EC Module 不斷的進行更新。為了確保 CCN 能夠有效發揮功能，我們使用虛擬 BSC 來處理基地台間的連線問題。當基地台更新後，只需更新虛擬 BSC，其他的功能元件則不需改變。
- **服務中心(Service Center)**：由於 CCN 擁有三種不同的通訊模式。服務中心會視來電請求為何種通訊模式，提供相對應功能以滿足使用者的需求。
- **用戶資訊中心(User Info Center)**：負責管理用戶的身份。用戶資訊中心將使用者分成兩類，一種是擁有戶資訊的 agency group 成員以及沒有用戶資訊的匿名使用者。當匿名使用者連接到 CCN 時，用戶資訊中心會預設他的 agency group 及 SLV 並且將這些用戶資訊記錄下來。
- **QoS 管理員(QoS manager)**：負責決定通話品質。由於不同的通話會有不同的緊急程度，因此我們根據緊急程度提供不同的通話品質，越急迫的通話獲得越大的頻寬以提供更好的服務品質，反之亦然。通話的緊急程度由發、受話者事先註冊的 agency group 決定。若發、收話者雙方事先並未註冊，則將會優先考慮那些已經註冊的成

員。根據通話的迫切性給予不同的頻寬，期望能在不影響救災效率的前提之下，盡可能的提供更多的通話服務。

- **行動管理員(Mobility Manager)**：負責找出受話者的位置。
- **網路流程控制器(Network Process Controller)**：網路服務流程控制器和相關的功能元件共同合作提供服務。
- **網路資料(Network Data)**：記錄網路狀態資訊，例如網路拓樸、頻寬分配的狀況以及頻寬的利用率等。
- **允入政策(Admission policy)**：記錄網路的允入政策，如鄰台間的頻寬分配。
- **允入控制器(Admission Controller)**：災區中的通話量遠超過平時的通話量，應急蜂巢式行動網路的負載量遠小於一般的公共網路，難以處理大量湧入的通話，因此並非所有的通話都能夠被接通。允入控制器根據允入政策進行允入管控。
- **路由管理員(Routing Manager)**：負責規劃管理網路拓樸以及轉送路徑。在緊急維修規劃期以 off-line 方式進行網路拓樸規劃，其後系統在緊急服務運轉期會根據災區通訊需求進行動態調整網路拓樸，以達到最大救災效益。
- **頻寬管理員(Bandwidth Manager)**：負責分配及管理無線頻寬。由於資訊需透過鄰台間轉送，因此每個基地台能使用的頻寬數量需要合理的分配，以滿足救災需求，避免基地台的頻道數量分配失衡。系統透過頻寬管理員，根據 CCN 相關人員在緊急維修規劃期所做的規劃進行分配。若頻寬管理不當，某些基地台的頻寬可能會被轉送的話務給壅塞，無法提供通訊服務。最壞的情況是所有的頻寬都被一些受災輕微的區域所佔據，使得受災較嚴重的區域獲得不到任何的頻寬。為了達到最佳救災效益，必須合理的分配每個基地台所能使用的頻寬。
- **自我調整(Self-Adjustment)**：當網路環境變動時，負責重新規劃網路拓樸及分配頻寬資源。

3.4、CCN 服務流程

根據元件的功能，我們將 CCN 提供服務的流程分為三個層面，如圖 3.8 所示，分別為存取用戶、服務程序、網路服務。用戶存取負責與終端用戶之間的溝通；服務程序負責處理用戶的請求；網路服務負責網路管理和資料數據的傳輸，本節介紹三種通訊模式的運作流程。

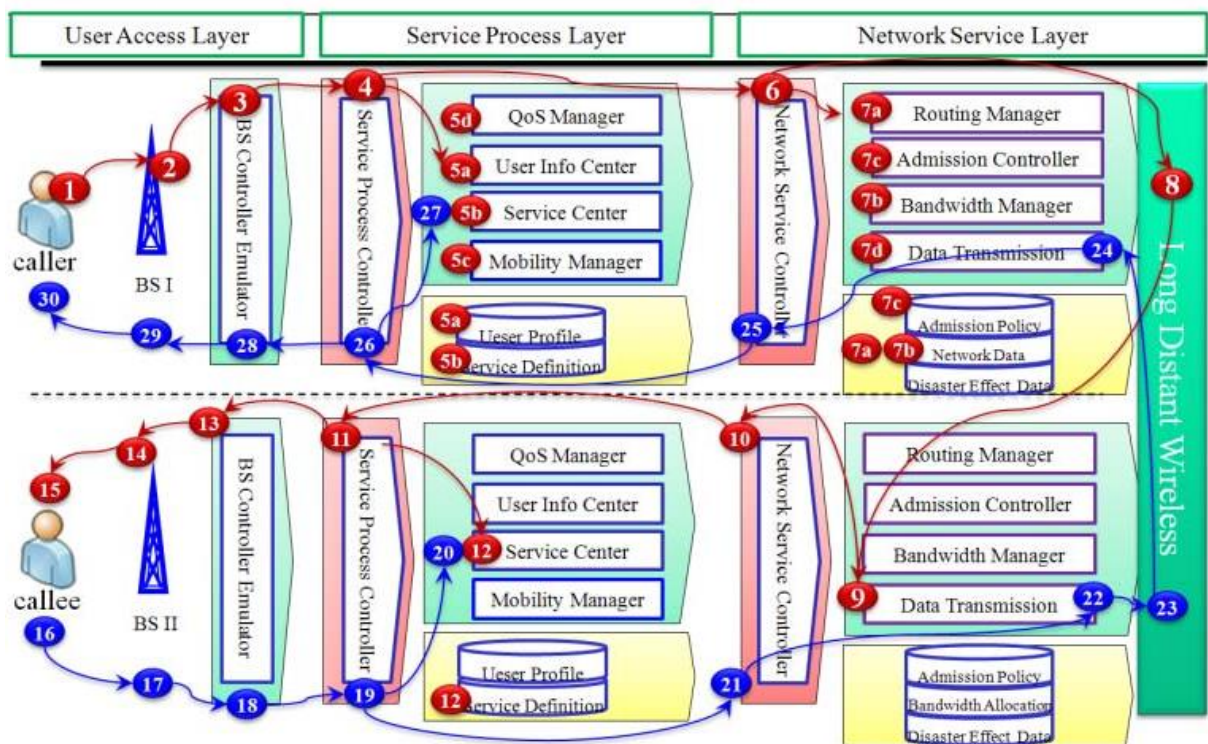


圖 3.8、一般通話模式通話建立流程

■ 一般模式建立服務流程：

STEP 1~3 存取用戶：當發話者撥打電話時，虛擬基地台控制器(BSC)會將發話者的請求連同發話者、受話者雙方的電話號碼一起傳送至服務程序層。

STEP 4~5 服務程序：在這個階段服務流程控制器和相關的 EC Module 功能元件會負責處理使用者的呼叫請求。首先，系統從用戶資訊中心查詢到使用者的個人資訊。然後，系統利用服務中心針對不同的呼叫請求提供相對應的服務程序，接著，系統利用行動管

理員找到受話者的位置，若是群組通訊模式，會找到 K 個相鄰的群組成員。再來，系統會透過 QoS 管理員根據發、受話者的用戶資訊來決定通訊品質，最後，系統將通話請求連同發、受話者的用戶資訊、通訊的品質、以及受話者的位置一起傳送到網路服務層。

STEP 6~8 網路服務：在這階段網路服務控制器以及 ICC Module 相關的功能元件共同合作提供網路服務。首先系統透過路由管理員查尋出轉送路徑，接著頻寬管理員會透過允入控制器根據頻寬使用率以及通話的緊急程度評估這個通話請求，如果這個通話請求被准許了，頻寬管理員會分配這個通話所需要的頻寬，最後通話請求會透過長距離的無線設備以多重跳接的方式轉送到目的地的基地台。

STEP 9~10 網路服務：系統透過多重跳接的方式將通話請求傳送到目的地。而網路服務控制器則會將通話請求轉送到服務程序層。

STEP 11~12 服務程序：服務流程控制器透過服務中心查詢出服務的項目為何，並且根據查詢的結果提供服務。

STEP 13~15 存取用戶：服務流程控制器透過虛擬基地台控制器送出一個信號給受話者，此時受話者的手機會響起。

STEP 16~30 建立通話：當受話者接起電話後，系統會送出一個認證給發話者並且在發話者和受話者之間建立一條虛擬的通道。

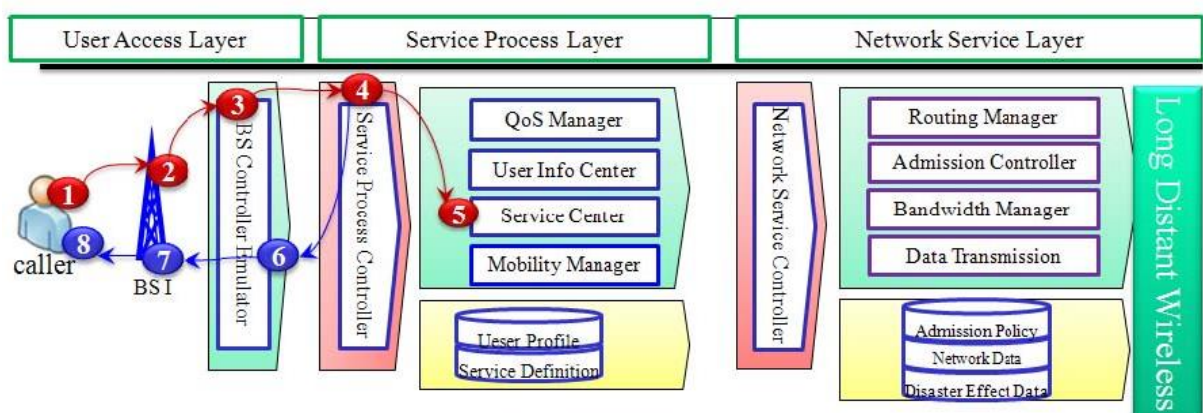


圖 3.9、無線電對講機模式通話建立流程

■ 無線電對講機模式建立服務流程：

無線電對講機模式建立服務流程如圖 3.9 所示。STEP 4~5，服務流程控制器接收到通話請求後，系統會透過服務中心辨識通話請求的種類。若通話請求為無線電對講機模式，此時系統會透過服務流程控制器會將發話者加入無線電對講機群組。由於無線電對講機模式是在相同的基地台之下使用，因此發話者的聲音會透過虛擬基地台控制器被廣播到無線電群組內的其他成員，不須透過長距離無線電設備轉送。

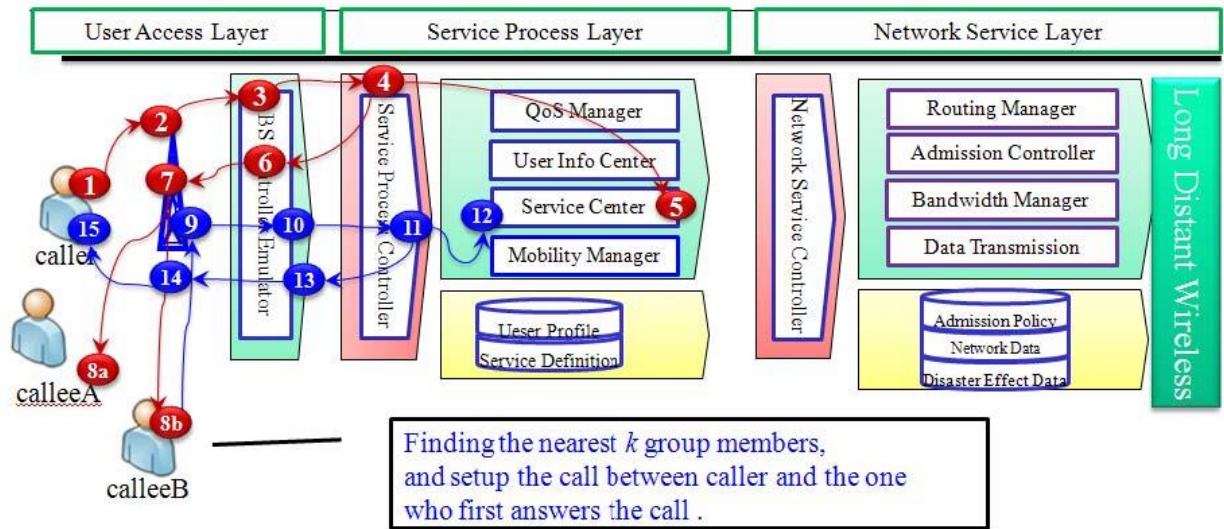


圖 3.10、群組通訊模式通話建立流程

■ 群組通訊模式建立服務流程：

群組通訊模式建立服務流程如圖 3.10 所示。STEP 4~5，服務流程控制器透過服務中心辨識通話請求的種類並且找出受話群的組成員。若群組內成員都在同一個基地台下時，系統會將通話請求廣播至所有的群組成員。接著當群組內某個成員接起電話時，系統就會在發話者和這位成員之間建立通話。若群組內成員並不在同一個基地台下時，此時系統送出通話請求的程序則如圖 3.8 所示。

3.5、建置與運轉流程

建置災區電信網之步驟分四個階段，各階段需完成的工作如下：

- **第一階段：災情評估 (Damage Assessment Phase)**

由國家救災單位蒐集災情並評估災損。而基地台若事先佈建有 CCN 元件，則倖存之孤立基地台需嘗試自主建構與鄰台之間的無線電鏈路、找出訊息轉送路徑、重建與後端核心網路之間的連線，回覆損害原因至控制中心並繼續提供通話服務，直到備用電力耗盡為止(約可維持 1 至數小時不等)。

- **第二階段：緊急維修規劃期 (CCN Planning Phase)**

選定欲建構 CCN 之災區選定修復區域與基地台。依基地台回覆之資訊及從各方面搜集之災情擬訂維修計劃，包含規劃最佳之無線連線網路拓樸、排定基地台修復次序、規劃修復路徑、妥善分配頻寬、以及依輕重緩急訂定允入政策等。

- **第三階段：緊急維修建置期 (CCN Deployment Phase)**

依據第二階段規劃之結果，進行 CCN 網路佈建與設定。

- **第四階段：緊急服務運轉期 (CCN Operation Phase)**

依所分配頻寬及訂定之允入政策，提供允許開放的服務、決定是否接受新的服務請求、並確保緊急通話可優先取得服務。

3.6、重要研究議題與未來研究方向

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，需整合現有基地台運作流程才可實現。而整合方法可分為二種：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門檻則遠低於嵌入式，行動通訊業者不需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。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。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。

■ **建構 CCN 網路需克服之挑戰如下：**

- **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，因而其中跳接點的選擇、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。在設計 CCN 拓樸時，我們將依據 CRP 的數量、各災區災情嚴重性、顧及通訊品質及通訊需求人數等種種因素，選擇數個孤立台佈建 CRP，並藉由多重跳接的方式連接後端網路，以恢復通訊。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適當地分散流量，避免建立負載量不均衡、跳接點過多的網路拓樸，增加其穩定度。分流的設計是依照各區受災程度估算可能的通話量並加以分流，避免話務集中在某些網段，造成網路提前壅塞。如此便可在有限的資源下提供最大的通話量，

並可避免網路壅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與建置人員數量而規劃，以派送建構 CCN 所需的 CRP 資源。由於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 建構排程，以達到最大的救災效益，即為建構排程意義所在。
- **頻寬分配**：CCN 網路中基地台分為連網台以及孤立台，當孤立台提供服務時，因資料需透過鄰台轉送，各節點所需頻寬包含發自本地基地台的流量以及為鄰台轉送的流量。考量 CCN 頻寬資源有限，且各地災情嚴重程度不同、緊急程度不同、使用者話務功能重要性不同、通訊品質需求不同，在最大化救災效益的前提下快速妥適的分配有限之頻寬，讓每個基地台分配到的頻寬資源符合救災通訊需求，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，即使這兩部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，如前所述，連網台對外的頻道數極為有限，在 CCN 多重跳接且連網台連外頻道極為有限的情況下，各孤立台能分配到的實際連外頻道數，必遠低於其裝備所能支援的頻道數量，因此，各孤立台將會有大量的剩餘閒置頻道。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。救災人員之間常有頻繁的通話需求。因此，建立 Intranet 的通道，讓同屬同一基地台信號範圍內的手機之間，可以利用閒置無線電頻道彼此互通，在不佔用連外頻道的情況下，支援無線對講機模式與群組通訊模式，提升整體通訊系統的利用率與可用性，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 利用蜂巢式行動通訊網路原有的基地台設備，另外介接具有運算處理能力的 EC Module，以達到提供行動裝置通訊服務的目地。因此這些外

來設備必須無縫地與基地台介接，包括各種通訊協定，以確保基地台可恢復運作。

- **允入政策制定：**允入控制讓系統依據特定條件，選擇性地接受使用者請求，其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 年倫敦地鐵車站爆炸事件，原是屬於小區域局部傷亡事件，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，將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 (ICC Module)，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 ICC Module，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，孤立台在失去備用電力之前即可啟動自動化建構程序，利用分散式演算法的方式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。為提高自動建構的成功率和可用度，網路拓樸規劃需確保各基地台至連網台均有替代路徑；除此之外，若鄰近有設置強固基地台，則可優先選定其作為連網台。因為強固基地台平時即備有衛星通訊，具有聯外能力且抗災系數高，較一般基地台更適於擔任連網台。
- **跨網路 CCN：**(使用不同電信公司基地台共建 CCN)：災害來臨時，各行動通信公司的網路可能支離破碎，同一家電信公司倖存的孤立基地台可能無法順利連接成一

個可用的 CCN，如果聯合各家公司的基地台，順利建成 CCN 的機會大幅增加，但是跨網路 CCN 的技術挑戰更艱鉅，尤其是控制信令將需要大修改，更是加倍的困難。

3.7、排程問題之相關演算法

3.7.1、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)，以期望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。

3.7.2、精確解法 (Exact Algorithm)

又稱為最佳解法，透過嚴謹的數學模型或電腦數據結構規劃，利用數學法則或數據結構搜尋的方式求得問題的解。使用此演算法時，須要先將問題中的資料數據化，並透過相關轉換將其轉換成符合數學表達式子的因子來供運算式使用，當條件都滿足時，即可經由反覆的計算來獲得最佳解。其最大的缺點是反覆的運算過程中，所需要的運算時間會因為數值的累積而越來越多，因此，在運算因子基數龐大的問題之中並不適用。此種解法通常用來解決非 NP 難度的問題。常見的精確解法類型如下：

■ 分枝界限法 (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，一但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。

3.7.3、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。

■ 單粒子型的隨機演算法：

- 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解的變化。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[17]：

- 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[34]：

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[10] [31]。

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[12][20]。

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，這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。這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。

3.7.4、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點也會隨之改變救災效益也會隨著時間改變，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，並且配合後啟發式演算法，在短時間內求得較佳解。

第4章、考慮資源運輸路徑之 CCN 建置排程

4.1、設計理念與目標

當大型天然災難發生後，造成各地區不同的災害程度，進而啟動 CCN 應急蜂巢式行動網路系統。當 CCN 拓樸規劃完善後，緊接著就是快速建構 CCN 拓樸。建構 CCN 拓樸時，由於運送交通工具以及建置工程人員數量相當有限，所需資源無法一次到位，需多次直升機、車輛、人力輪流運送，才可完成 CCN 應急蜂巢式行動網路拓樸的建置。如何根據各個基地台服務地區的受災嚴重程度，例如圖 4.1 所示不同環境下的時間與生還率關係圖，訂定一個修復排程的順序，即基地台的建構次序，使得救災總效益為最大，是相當複雜且重要之議題。由於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，這使得排程問題更加複雜。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，稱為樹狀轉送拓樸 (Forwarding Tree)，修復排程問題是：在有限時間內針對給定的樹狀轉送拓樸，找出一組基地台修復順序，使所有基地台之效益總和得以最大化。

我們在[15]中已有初步的建構排程研究，該論文單純考慮孤立台之建構時間，並未將運輸工具的路線也納入考慮，若考慮更貼切實際之災區狀況時，運輸路徑之選擇也應納入排程考慮，因此，本論文將考慮基地台間相對距離，提出考慮資源運輸路徑之 CCN 建置排程問題模型，目標在短時間內找出一組建構排程順序，使整個 CCN 拓樸能夠快速完成並且得到較高的救災效益。

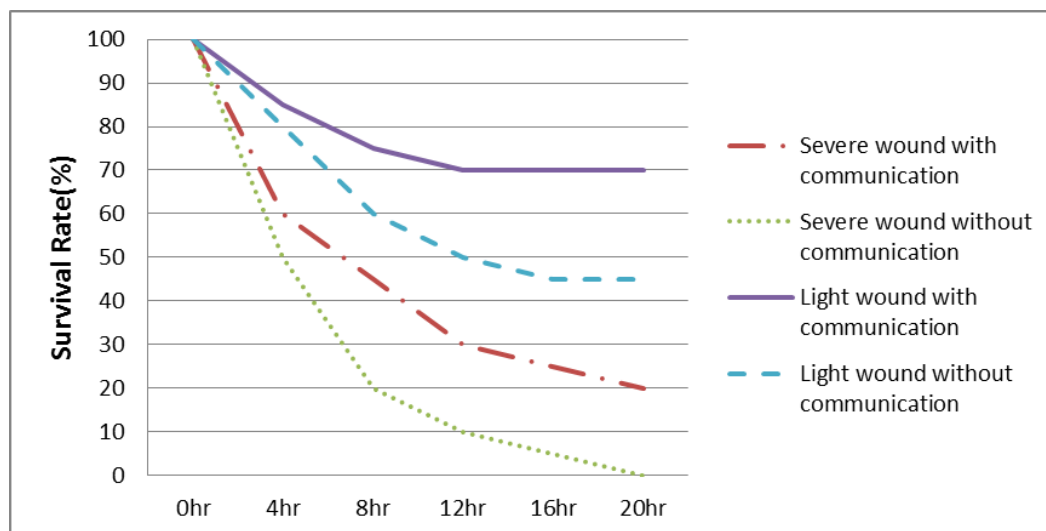


圖 4.1、不同環境的存活率關係圖

4.2、CCN 的建構排程

給定一個確定的 CCN 應急通訊系統網路樹狀拓樸，若建構之運輸工具可一次運送多個應急修復包 (CRP)，則可節省許多建構時間及運送資源。例如：假設共有 10 個基地台需建構，但運輸工具之容量只能運送 5 個應急修復包 (CRP)，本問題則又細分為兩大問題：

- (1) 分割問題：那些群組基地台先建構？另外那些群組基地台後建構？
- (2) 路徑問題：建構路徑順序可花費最小時間成本且達到最高救災效益？

在此簡化問題，先不考慮分割問題，假設運輸工具一次可運送所有的應急修復包 (CRP)，單一考慮基地台間在建構排程問題。

如圖 4.2 所示，給定一個 CCN 拓樸，其中包含的資訊有：

1. 轉送拓樸 (Forwarding Topology)：基地台之間的互連狀況，假設是樹狀拓樸，內含父子關係。
2. 運送時間成本 (Traveling time)：將距離轉換為時間，直接以時間代表任兩基地

台間的運送時間成本。

3. 救災效益 (Profit)：即各基地台在某時間點被建構時，所能發揮的救災助益。

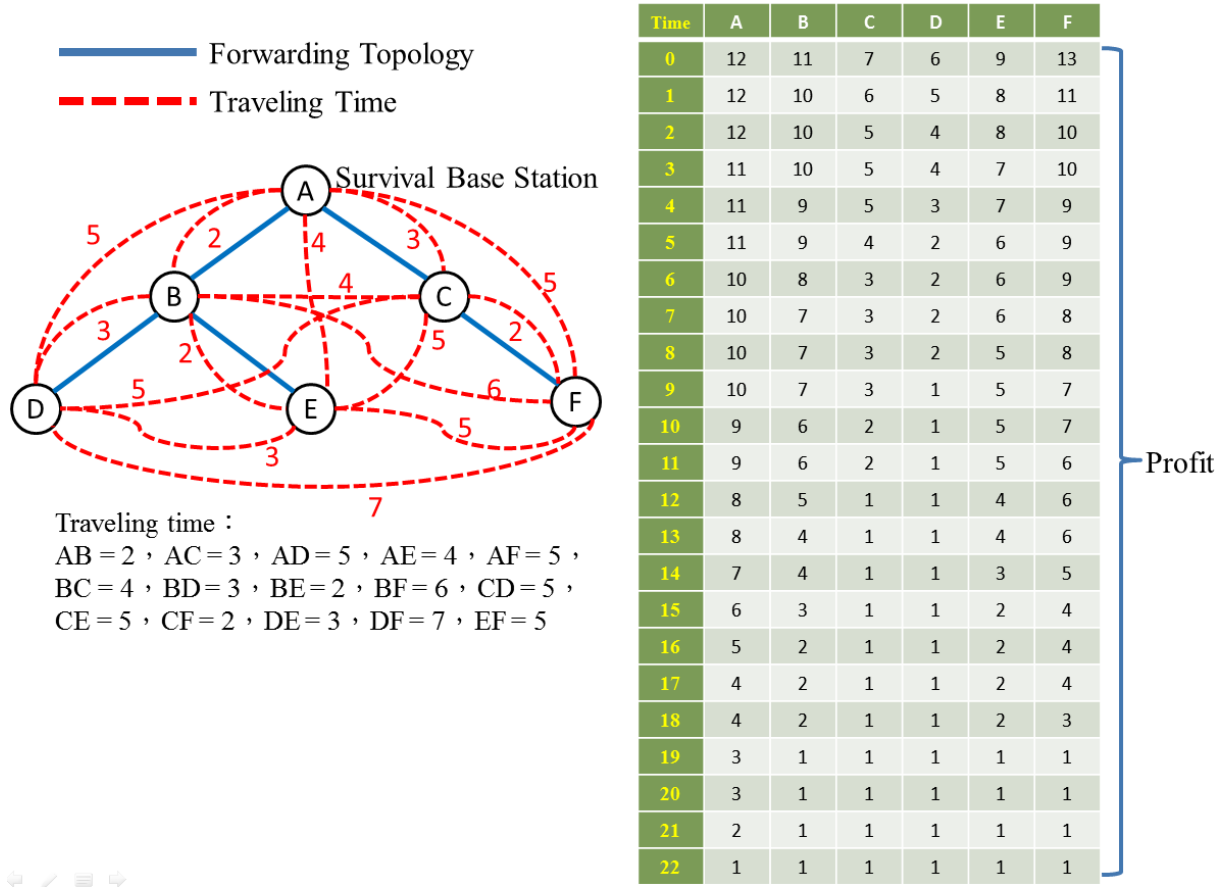
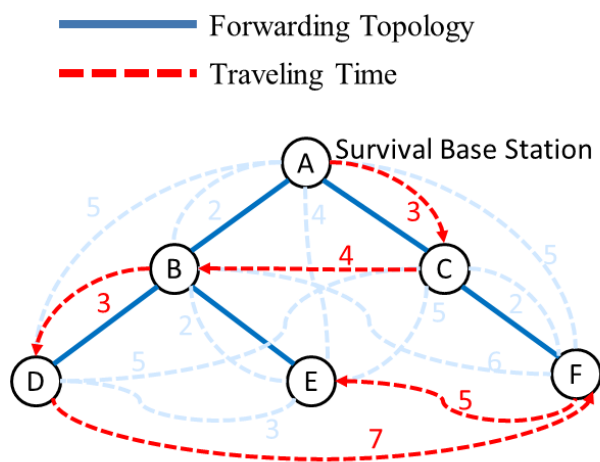


圖 4.2、範例：CCN 建構排程相關資訊

根據上述之 CCN 樹狀拓撲、運送時間成本及救災效益等資訊，不同的建構順序會得到不同的總救災效益，在此舉例兩個建構排程，如圖 4.3、圖 4.4 所示。圖 4.3 中基地台建構排程順序是 $A \rightarrow C \rightarrow B \rightarrow D \rightarrow F \rightarrow E$ ，對照救災效益表格可知，基地台 A 在時間點 0 被建構因而獲得了 12 單位的救災效益，經過 3 單位運送時間成本來到了基地台 C，建構基地台 C 因而獲得了 5 單位的救災效益，再經過 4 單位運送時間成本來到了基地台 B，建構基地台 B 獲得了 7 單位的救災效益，依序往下對照運送時間成本，即可獲得基地台被建構時的救災效益，而總救災效益的計算方式是根據各基地台所被建構之時間點，所獲得的救災效益進行加總。由圖 4.3、圖 4.4 範例可得知，圖 4.4 所得之總

救災效益 37 單位高於圖 4.3 的總救災效益 30 單位，且圖 4.4 總花費時間 14 單位亦少於圖 4.3 所花費之總時間 22 單位。因此，根據建構路線的不同而得到地不同的建構排程順序，其所得到的總救災效益以及總花費時間，有很大的不同，若花費過多時間建構基地台造成救災效益低落，將嚴重影響救災的進行，應審慎評估各種因素的影響，找出擁有最大總救災效益的 CCN 建構排程。



Schedule 1 :
 $A \rightarrow C \rightarrow B \rightarrow D \rightarrow F \rightarrow E$
 Total Time :
 $3+4+3+7+5=22$
 Total Profit :
 $12+5+7+1+4+1=30$

Time	A	B	C	D	E	F
0	12	11	7	6	9	13
1	12	10	6	5	8	11
2	12	10	5	4	8	10
3	11	10	5	4	7	10
4	11	9	5	3	7	9
5	11	9	4	2	6	9
6	10	8	3	2	6	9
7	10	7	3	2	6	8
8	10	7	3	2	5	8
9	10	7	3	1	5	7
10	9	6	2	1	5	7
11	9	6	2	1	5	6
12	8	5	1	1	4	6
13	8	4	1	1	4	6
14	7	4	1	1	3	5
15	6	3	1	1	2	4
16	5	2	1	1	2	4
17	4	2	1	1	2	4
18	4	2	1	1	2	3
19	3	1	1	1	1	1
20	3	1	1	1	1	1
21	2	1	1	1	1	1
22	1	1	1	1	1	1

Profit

圖 4.3、範例：CCN 建構排程(排程一)

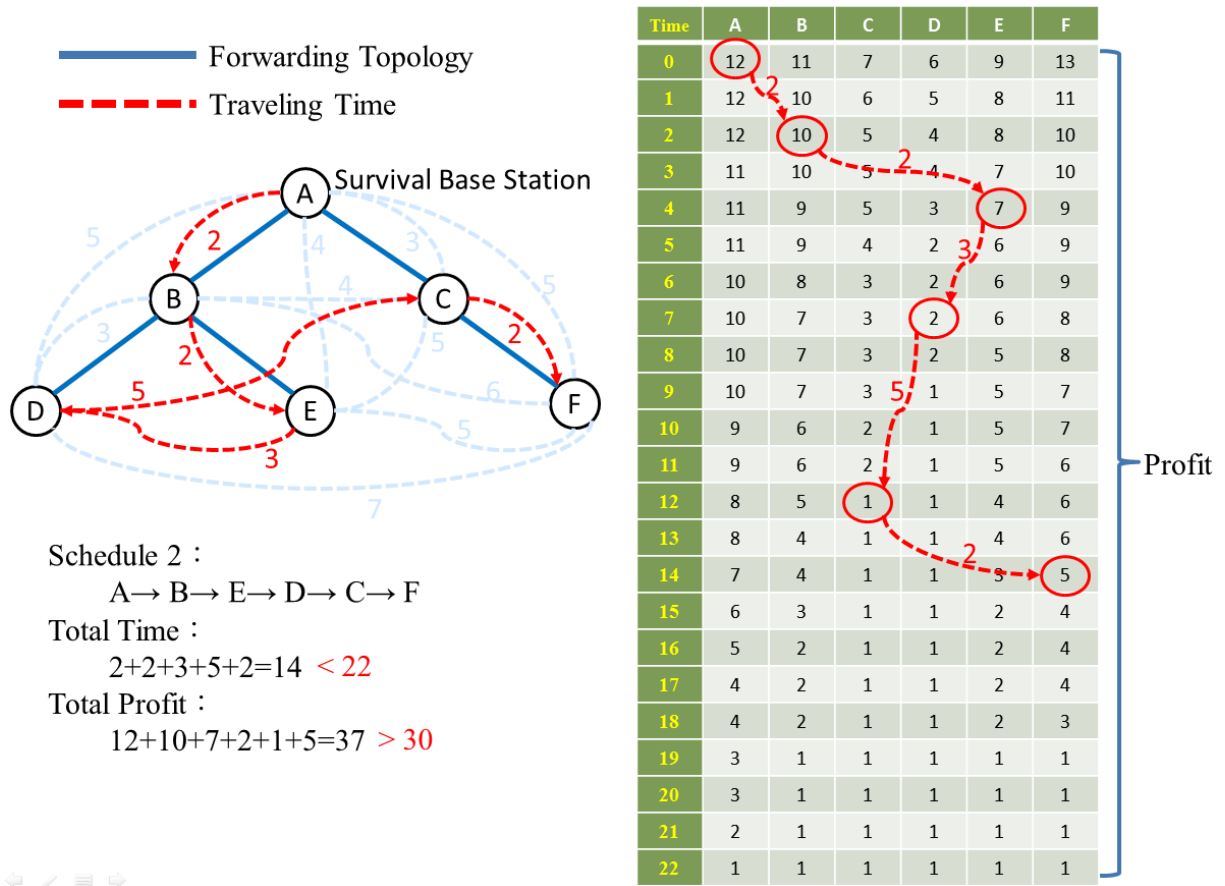


圖 4.4、範例：CCN 建構排程(排程二)

4.3、問題定義

4.3.1、救災效益定義

當一個孤立台建構完或加入 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」。隨著時間推移，待救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。

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。

4.3.2、問題描述

我們將建構排程問題以數學模型表達，除了 CCN 拓樸本身的樹狀結構以 edge 表示以外，另外訂定任兩點基地台間的路網關係為時間成本之 edge。

■ 名詞定義：

- 救災效益(Profit)：恢復該孤立台的通訊對於該地區能帶來的救災助益。
- 樹狀轉送拓樸 (Forwarding Tree)：CCN 中串聯各基地台之拓樸連線。
- 節點：CCN 拓樸中之基地台。
- Schedule：CCN 應急通訊網路建構順序。

- #### ■ 最佳化目標：在有限資源及有限時間下，找出一組建構 CCN 排程順序，使 CCN 服務救災總效益達最大。

4.4、最佳化問題模型

本研究提出了兩個問題模型，第一個問題模型為 CCN Deployment Scheduling Antecessor Constrained (CCNDS-AC)，由連網台出發，依據 CCN 樹狀轉送拓樸結構，優先建構父節點，單一基地台建構完成的瞬間，即可立刻服務基地台涵蓋災區。第二個問題模型為 CCN Deployment Scheduling Unconstrained (CCNDS-UC)，撤除樹狀轉送拓樸結構之限制條件進行基地台建構排程，由於不限定由父節點開始往下建構的因素，因而可考慮到更多更廣的建構排程可能性。

4.4.1、CCNDS-AC

CCNDS-AC 的要旨是從連網台出發，依據 CCN 拓樸樹狀結構、路網等相關資訊，遵循優先建構父節點之規則建構，若節點的父節點尚未建構，則不在考慮的候選節點清單範圍內，單一基地台建構完成的瞬間，即可立刻服務基地台涵蓋災區，意即救災效益在建構完成瞬間獲得。CCNDS-AC 問題模型描述如下：

Given a forwarding tree $T(V, E)$, D , P , S where

- $V = \{v_i | i = 1, 2, \dots, n\}$ is the set of survival base stations and isolated base stations.
 - v_1 is the root node that has an external link (survival base station).
- $E = \{e_{ij} | e_{ij} \text{ is the link of } (v_i, v_j) \text{ and } v_i, v_j \in V\}$
- $D = \{d_{ij} | d_{ij} \text{ is the traveling time of } (v_i, v_j) \text{ and } v_i, v_j \in V\}$
- $S = \{s_k | s_k \text{ is a permutation of } V, k = 1, \dots, n!\}$ is the set of CCN construction schedules.
- $C = \left\{ c_i(s_k) \left| \begin{array}{l} c_i(s_k) \text{ is the time at which node } v_i \text{ has be repaired in the schedule } s_k \\ \text{and } v_i \in V, s_k \in S \end{array} \right. \right\}$
- $P = \{p_i(t) | i = 1, \dots, n, t \in Z^+\}$, $p_i(t)$ is the profit of v_i , if node v_i is constructed at time t , $t = c_i(s_k)$ and $s_k \in S$.
- The CCN deployment scheduling antecessor constrained problem is to find $s_k \in S$, such that $\sum_{v_i} p_i(t) = \sum_{v_i} p_i(c_i(s_k))$ is maximized.
- Subject to v_i must be constructed before v_j , if v_i is the antecessor of v_j .

4.4.2、CCNDS-UC

CCNDS-AC 為了從連網台出發，並且遵循父節點優先建構之限制條件，可能會導致建構排程過程中，動線太過於重複，花費更多時間導致救災效益的損失。如圖 4.5 所示，Schedule 1 為了優先建構父節點，由救災總部 (Headquarter) 往連網台出發，而花費了 5 個時間成本，在依序建構了 B 與 E，總花費時間為 9 個時間成本，而 Schedule 2 則由靠近救災總部(Headquarter) 的 E 點出發，依序往連網台建構 B 與 A，總花費時間為 4 個時間成本，可輕易地看出，同樣建構了 A、B 與 E，CCNDS-AC 的限制卻花費了更多的建構時間，也浪費了運載工具資源，亦可能影響總救災效益。因此我們提出 CCNDS-UC，希望能改善上述之問題而獲得更大的救災效益。問題模型不從連網台優先建構，撇除樹狀結構之限制條件，任意一點皆可納入建構。

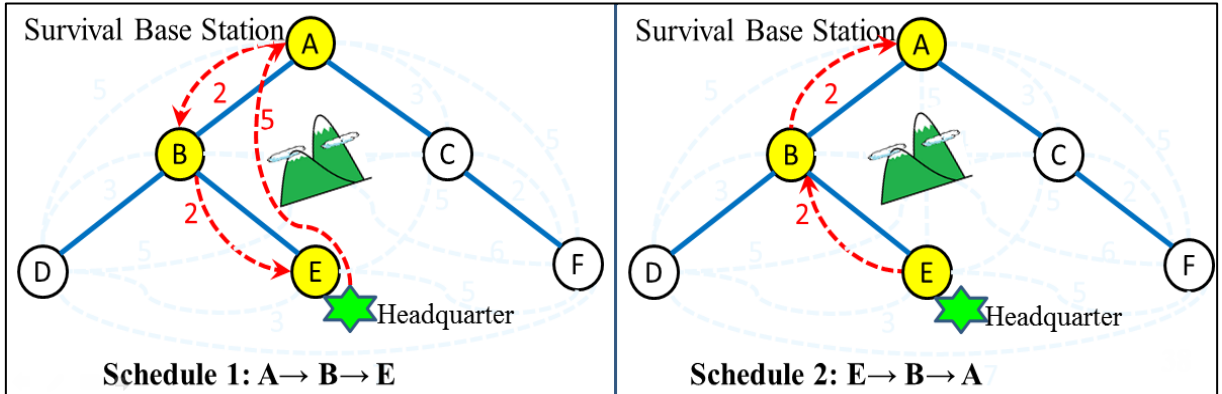


圖 4.5、範例：CCNDS-AC 模型可能造成之狀況

CCNDS-UC 問題模型描述如下：

Given a forwarding tree $T(V, E)$, D , P , S where

- $V = \{v_i | i = 0, 1, 2, \dots, n\}$ is the set of survival base stations and isolated base stations.
 - v_1 is the root node that has an external link (survival base station).
 - v_0 is the CCN headquarter
- $E = \{e_{ij} | e_{ij} \text{ is the link of } (v_i, v_j) \text{ and } v_i, v_j \in V\}$
- $D = \{d_{ij} | d_{ij} \text{ is the traveling time of } (v_i, v_j) \text{ and } v_i, v_j \in V\}$
- $S = \{s_k | s_k \text{ is a permutation of } V, k = 1, \dots, n!\}$ is the set of CCN construction schedules.
- $C = \left\{ c_i(s_k) \left| \begin{array}{l} c_i(s_k) \text{ is the time at which node } v_i \text{ has be repaired in the schedule } s_k \\ \text{and } v_i \in V, s_k \in S \end{array} \right. \right\}$
- $P = \{p_i(t) | i = 1, \dots, n, t \in Z^+\}$, $p_i(t)$ is the profit of v_i , if node v_i is constructed at time t , $t = c_i(s_k)$ and $s_k \in S$.
- The CCN deployment scheduling antecessor constrained problem is to find $s_k \in S$, such that $\sum_{v_i} p_i(t) = \sum_{v_i} p_i(c_i(s_k))$ is maximized.
- ~~Subject to v_i must be constructed before v_j , if v_i is the antecessor of v_j .~~

4.5、複雜度分析

在這一節中，我們將對 Single machine scheduling problem (SMS) 和 CCN Deployment Scheduling Antecessor Constrained (CCNDS-AC) 做問題分析，並證明 CCNDS-AC 為一 NP-Hard 問題。

4.5.1、Single Machine Scheduling Problem (SMS)

The non-preemptive single machine scheduling problem is as follows. Given a set J of n independent jobs that has to be scheduled on a single machine. Each job $j_i \in J$ contains uninterrupted processing time $u_i \in U$ and weight $w_i \in W$, where u_i and w_i are positive integers. The single machine can handle only one job at a time.

To find a schedule s_k such that $\sum_{i=1}^n (w_i * C_i(s_k))$ is minimized, where s_k is a permutation of all jobs ($k = 1, 2, 3, \dots, n!$), $C_i(s_k)$ is the time at which job j_i completes in the given schedule s_k .

4.5.2、CCNDS-AC

The non-preemptive CCNDS-AC problem is as follow. Given a forwarding tree $T(V, E)$ that has n nodes, the tree structure has to be repaired by a work team. Each antecessor node should be repaired before it's child node. Each node $v_j \in V$ requires uninterrupted delivery time $d_{ij} \in D$ and profit $p_j(t) \in P$, where t , d_{ij} and $p_j(t)$ are positive integers. The work team can handle only one job at a time.

To find a schedule $s_k \in S$ such that the sum $\sum_{v_i} p_i(t) = \sum_{v_i} p_i(c_i(s_k))$ is maximized, $c_i(s_k)$ is the time at which node v_i has be repaired in the given schedule s_k .

4.5.3、CCNDS-AC 計算複雜度

CCNDS-AC is in NP :

We first show that CCNDS-AC \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 CCNDS-AC schedule within $O(n^2)$ time.

CCNDS-AC is NP-Hard :

We now prove that CCNDS-AC problem can be reduced to SMS problem straightforwardly. Given an instance $A:[J, U, W]$ in SMS, we can find an instance $B:[V, E, D, P]$ in CCNDS-AC such that an optimal solution s_k for B is also an optimal solution for A . Let $V=J$, $E=\{e_{root,i}/v_{root}, v_i \in V\}$, $D=U$, $P=W$. The verification can be performed in polynomial time. Let total weighted completion time of SMS for s_k is $T.W.C(s_k)$, total weighted profit of CCNDS-AC for s_k is $T.W.P(s_k)$. We prove 3 Lemmas first :

Lemma 1: Any valid schedule s_k for B (CCNDS-AC) is a valid solution for A (SMS).

Proof: Any permutation of J is valid schedule for A , and s_k is a permutation of V , which is J . Therefore s_k is a valid solution for A . Q.E.D.

Lemma 2: Any valid schedule s_k for A is also a valid schedule for B .

Proof: A valid schedule s_k for A is a permutation of J . Since each node in B can directly connect to the root of B , therefore any permutation of V , which is J , is a valid schedule of B . Thus s_k is also a valid schedule for B . Q.E.D.

Lemma 3: If $T.W.C(s_{k1}) < T.W.C(s_{k2})$ then $T.W.P(s_{k1}) > T.W.P(s_{k2})$

Proof: If $\sum_{i \in J}(w_i * C_i(s_{k1})) < \sum_{i \in J}(w_i * C_i(s_{k2}))$, by Equal Division Theorem, we can get $\sum_{v_i} p_i(C_i(s_{k1})) > \sum_{v_i} p_i(C_i(s_{k2}))$. Q.E.D.

Next, we prove by contradiction that an optimal solution s_{k2} to B must be an optimal solution to A . By Lemma 1, we know s_{k2} is also a valid schedule for A , whose total weight completion time is $T.W.C(s_{k2})$. Assume s_{k2} is not an optimal schedule for A , there must be

another schedule s_{k1} , whose total weight completion time $T.W.C(s_{k1})$ is smaller than $T.W.C(s_{k2})$. By Lemma 2, s_{k1} is also a valid schedule for B, whose total profit is $T.W.P(s_{k1})$. By Lemma 3, we can obtain $T.W.P(s_{k1})$ is bigger than $T.W.P(s_{k2})$. This contradicts to the fact that s_{k2} is an optimal solution for B. Therefore, s_{k2} must be an optimal solution for A. Q.E.D.

4.6、演算法設計

CCNDS-AC 與 CCNDS-UC 皆為 NP-Hard 問題，但又因為救災的進行急迫，時間相當寶貴，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。因此我們為這兩個問題模型設計了快速的啟發式演算法。可在短時間內計算出一個接近最佳解的 CCN 建構排程。

4.6.1、DS-ACG 演算法

Deployment Scheduling Antecessor Constrained Greedy Algorithm (DS-ACG)是針對第一個問題模型 CCNDS-AC 所設計的演算法，CCNDS-AC 的有一個相當明確的限制條件，要求建構 CCN 拓樸須由父節點依序建構而下，由此方式為確保 CCN 建置人員在建置某一基地台時，能立刻啟用 CCN 服務該災區的用戶，也確保往後子節點被建置時亦能立即服務該區用戶。在父節點優先建構的限制條件下，DS-ACG 演算法由連網台開始排入建置排程，每一次的建置排程，將選擇節點之子節點加入即將建置排程的候選點序列中，再由 DS-ACG 演算法根據此候選序列尋找單位時間獲得最大救災效益的候選節點做為下一個修復節點，加入排程，不斷重複此步驟，直到 CCN 拓樸中的所有節點皆被修復為止，以下用演算法流程圖、Pseudo code 及演算法範例來進一步說明 DS-ACG 演算法。

■ DS-ACG 演算法流程圖

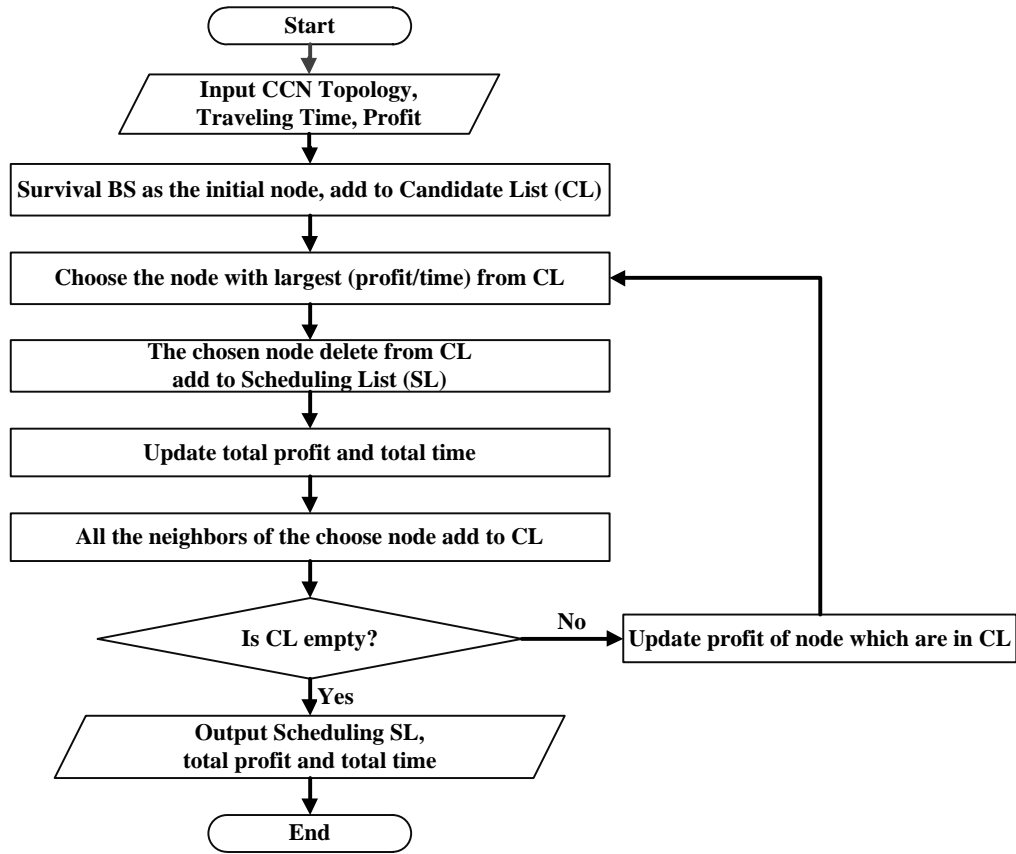


圖 4.6、DS-ACG 演算法流程圖

```

DS-ACG Pseudo code:
Input CCN Topology, Traveling Time, Profit
SL=null      /*the Scheduling List*/
CL={}        /*the Set of Candidate nodes*/
CL.add(Survival BS);
while (size of CL !=0) do
    for each node in CL do
        calculate the node's profit and time
        if (profit/time) >(maxProfit/maxTime)
            maxProfit = profit
            maxTime = time
            maxPtofitNode =node
        end if
    end for
    SL=SL+ maxPtofitNode          /*update Scheduling List*/
    totalProfit=totalProfit+maxProfit /*update total Profit*/
    totalTime=totalTime+maxTime    /*update total Time*/
    CL.remove(maxPtofitNode)
    for each node in maxPtofitNode's childList do
        CL.add(node)
    end for
end while

```

圖 4.7、演算法 DS-ACG 之 Pseudo Code

■ DS-ACG 演算法範例

在此我們以一個簡單的範例說明 DS-ACG 演算法，如圖 4.8 所示，圖中包含三大部分：CCN 樹狀拓樸、路網成本及救災效益時間總表。CCN 樹狀拓樸，共有 A、B、C、D、E、F 此六個節點需要修復，實線代表 CCN 中基地台與基地台間的無線鏈結；虛線代表節點與節點之間的相對時間，亦即路網成本，在此以下三角的表格 Traveling Time 來代表；救災效益時間總表為圖中右方的大表格，直接的表示出隨著時間的消耗，各基地台的救災效益也隨之而變。

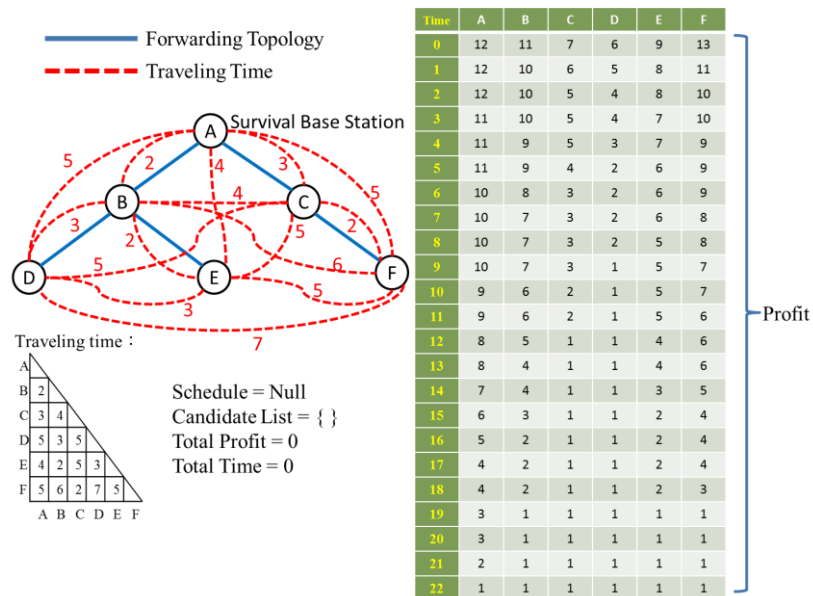


圖 4.8、DS-ACG 演算法範例

如圖 4.9 所示，CCNDS-AC 的限制條件是必須由父節點優先建構，因此演算法由連網台節點 A 開始修復建構，此時所獲得的排程是：A，救災效益為 12。

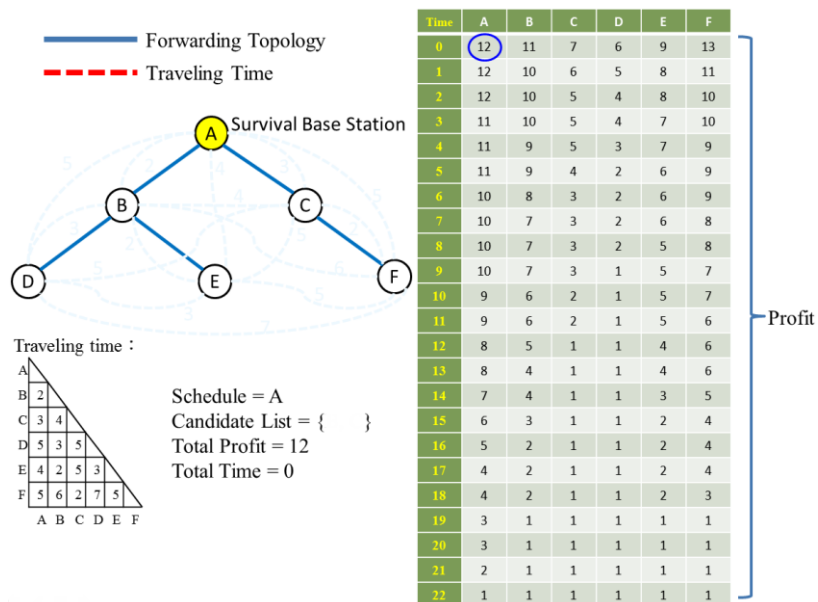


圖 4.9、DS-ACG 演算法步驟一

如圖 4.10，將剛修復節點 A 的子節點 B、C 加入候選節點清單中，對照救災效益時間總表，發現經過 2 個時間單位到達節點 B，可獲得救災效益 10；經過 3 個時間單位到

達節點 C，可獲得救災效益 5。

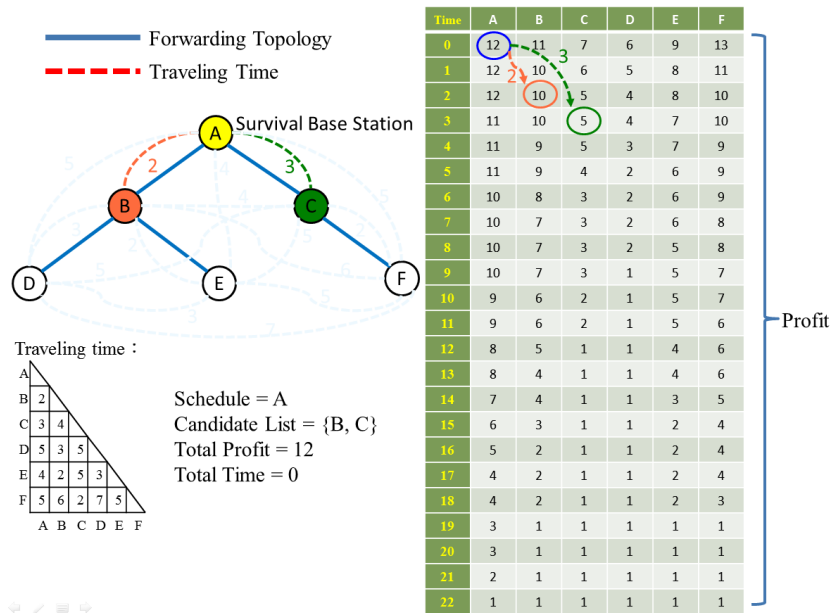


圖 4.10、DS-ACG 演算法步驟二

DS-ACG 由候選節點清單中，挑選單位時間獲得最大救災效益的節點為下一節點。

如圖 4.11 所示選擇 B 點，放入排程當中，候選節點清單中刪去節點 B。

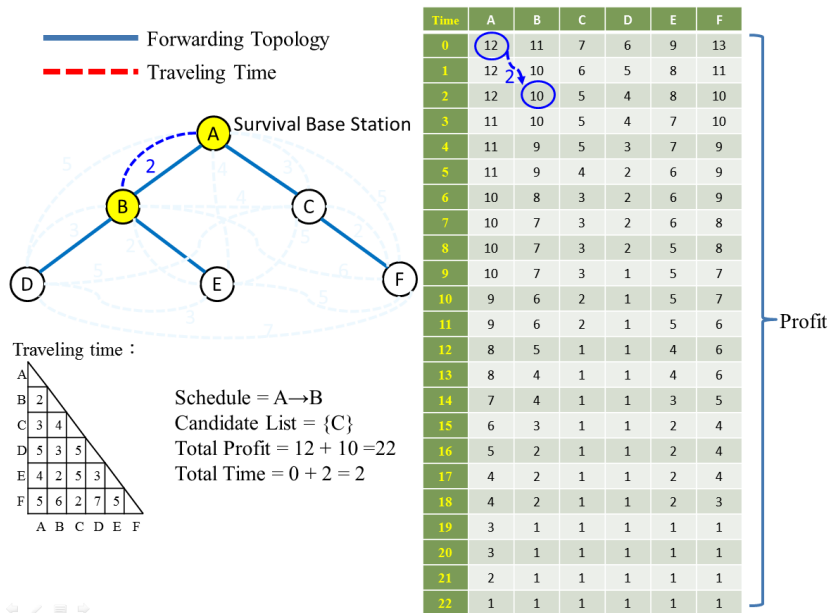


圖 4.11、DS-ACG 演算法步驟三

如圖 4.12 所示，同步驟二，將剛修復節點 B 的子節點 D、E 加入候選節點清單中，

此時的候選節點共有 C、D、E 這三點。

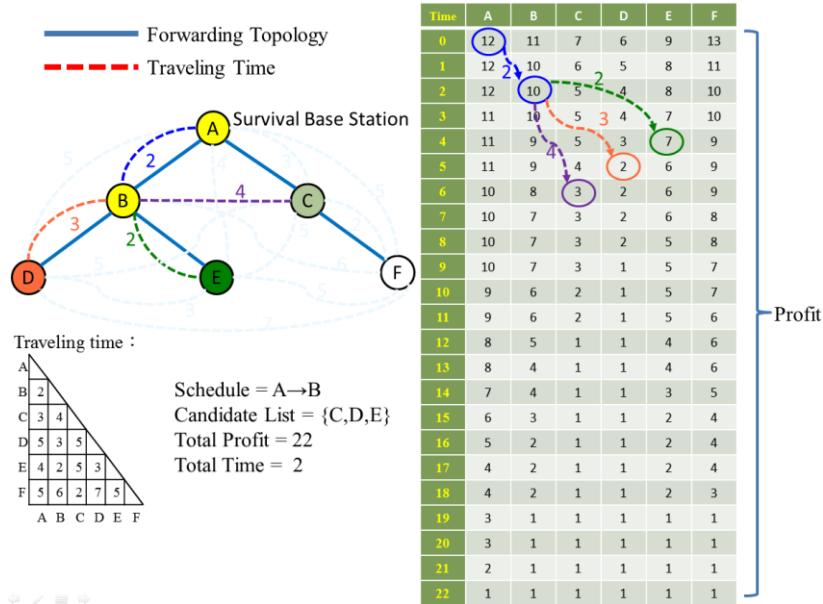


圖 4.12、DS-ACG 演算法步驟四

如圖 4.13 所示，同步驟三，DS-ACG 演算法選擇 E 點，放入排程當中，候選節點清單中刪去節點 E。

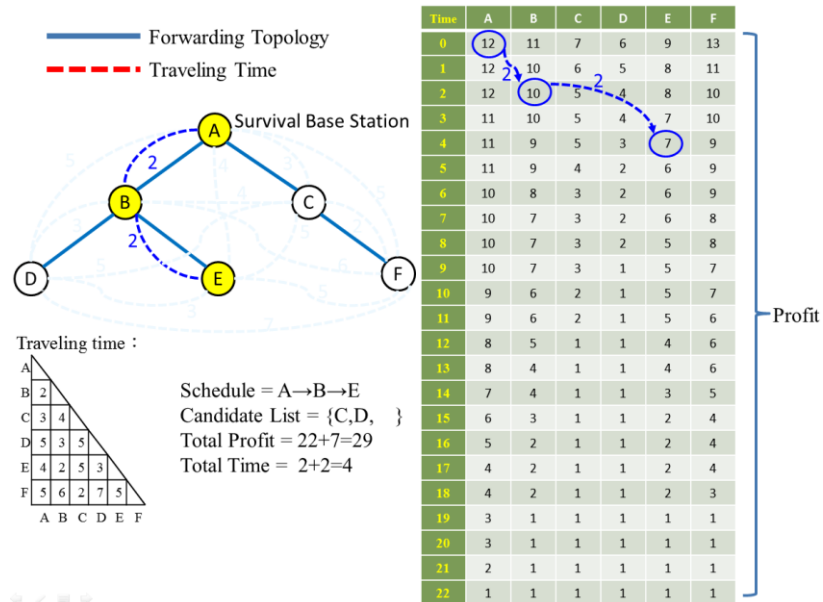


圖 4.13、DS-ACG 演算法步驟五

重複步驟二及步驟三，直到六個節點皆被加入建構修復排程後，才完成 DS-ACG 演算

法。

4.6.2、DS-UCB 演算法

Deployment Scheduling Unconstrained Backtrack Algorithm (DS-UCB) 是針對第二個問題模型 CCNDS-UC 所設計的演算法，在 4.4.2 已說明捨棄父節點優先建構的限制條件，更進一步考慮其他排程是有其必要性的，在此不再贅述。由於在此問題模型相較於 CCNDS-AC 又更為複雜，若單純使用貪婪演算法，可能遭遇無法獲得救災效益導致救災效益無法直接計算而不選擇鄰近節點之狀況，因此設計出不同於貪婪演算法的啟發式演算法並且搭配 Backtracking 機制。

當 CCN 樹狀拓樸、時間成本及救災效益等初始條件輸入後，DS-UCB 根據初始條件尋找距離救災總部最靠近的點當作目前建構點，加入暫定排程中，緊接著判斷目前建構點是否擁有救災效益得以計算；若救災效益可計算，則加入正式排程中，清空暫定排程，再次尋找距離目前建構點最短時間成本的節點，進行救災效益可否計算之判斷；若目前建構點的救災效益無法計算，則將父節點加入暫定排程中，再次判斷目前節點可否計算救災效益，若否，則再將上一層的父節點加入暫定排程中，直到目前節點的救災效益可以計算才停止，根據此暫定排程進行 Backtracking 的檢查機制，考慮此暫定排程中其周遭 3-hop 以內的孤立點加入暫定排程後能否獲得更大的救災效益，互相比較得出最大救災效益的暫定排程後加入正式排程中，將目前節點指向暫定排程的最後一個節點，完成 Backtracking 機制，清空暫定排程，再次尋找距離目前建構點最短時間成本的節點，進行救災效益可否計算之判斷；直到 CCN 拓樸中的所有節點皆被修復為止。DS-UCB 進一步考慮鄰近節點之路途與救災效益狀況，具有將鄰近的節點一併修復建構以節省更多的時間與運載資源的優點，以下用演算法流程圖、Pseudo code 及演算法範例來進一步說明 DS-UCB 演算法。

■ DS-UCB 演算法流程圖

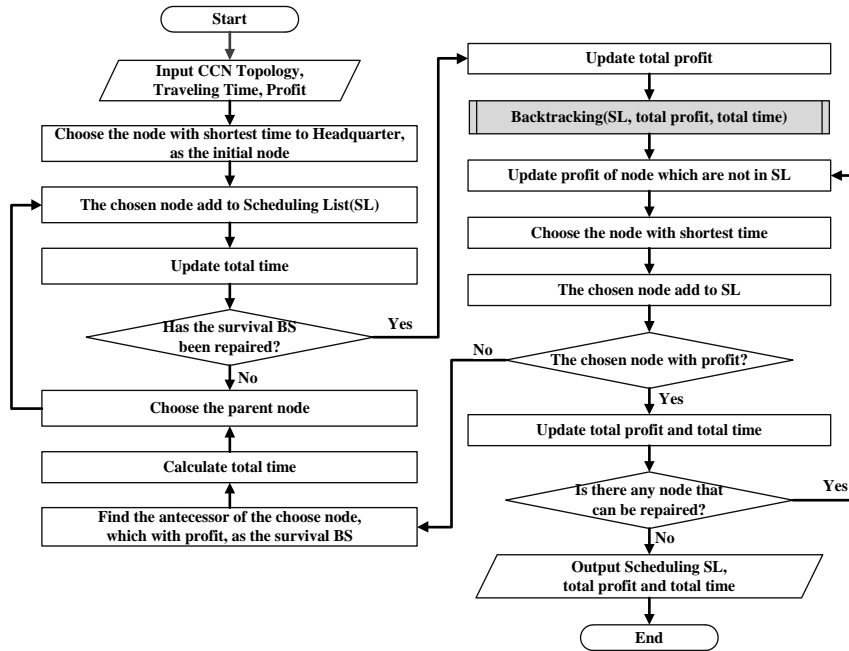


圖 4.14、DS-UCB 演算法流程圖

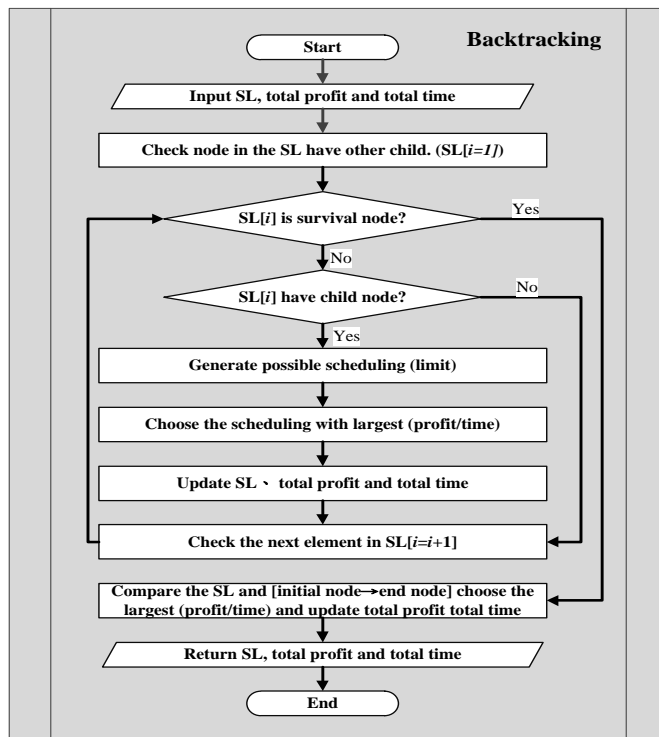


圖 4.15、DS-UCB 之 Backtracking 機制流程圖

```

DS-UCB Pseudo code:
Input CCN Topology, Traveling Time, Profit
SL=null      /*the Scheduling List*/
CL={}        /*the Set of Candidate nodes*/
startNode=the closest node of Headquarter
while(SL.length!=node Number of CCN topology) do
    SL.add(startNode)
    if(startNode don't have profit) do
        /*initial path from start node to SBS*/
        while(startNode don't have profit) do
            temp_Sch.add(parentNode)
            startNode=parentNode
        end while
        backtrack_Sch=Backtrack(temp_Sch);
        max_Sch=temp_Sch
        if profit(temp_Sch) < profit(backtrack_Sch) do
            max_Sch=backtrack_Sch
        end if
        totalProfit=totalProfit+ profit(max_Sch) /*update total Profit*/
        totalTime=totalTime+ time(max_Sch) /*update total Time*/
        SL.add(max_Sch)
        startNode =last node of SL
    else
        totalProfit=totalProfit+ profit(startNode) /*update total Profit*/
        totalTime=totalTime+ time(startNode) /*update total Time*/
    end if
    startNode=the closest node of startNode
end while

Backtrack(temp_Sch) do
    Check temp_Sch neighbor and generate finite schedules
    Compare schedules' total profit
    Choose the schedule with largest total profit as the backtrack_Sch
    return backtrack_Sch
end Backtrack

```

圖 4.16、DS-UCB 演算法 Pseudo Code

■ DS-UCB 演算法範例

我們以一個簡單的範例說明 DS-UCB 演算法，由於此演算法較為複雜，在此簡化範例，路網成本及救災效益時間總表省略，僅顯示 CCN 樹狀拓樸以表達演算法之進行。

實際演算法依然會參照路網成本及救災效益時間總表進行演算法。如圖 4.17 所示，圖中 CCN 樹狀拓樸，共有 A、B、C、D、E、F、G、H 此八個節點需要修復，星號代表 CCN 救災總部(CCN Headquarter)所在位置，實線為 CCN 的無線鏈結。

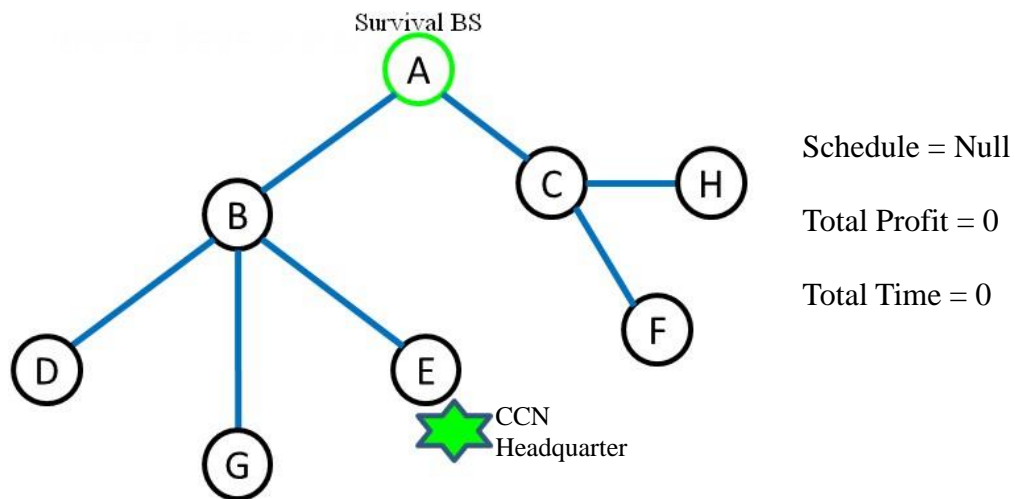


圖 4.17、DS-UCB 演算法範例

如圖 4.18 所示，選擇靠近 CCN 救災總部的節點 E 作為出發點。E 點為孤立點，當下並無提供服務，因此無 profit 可言。

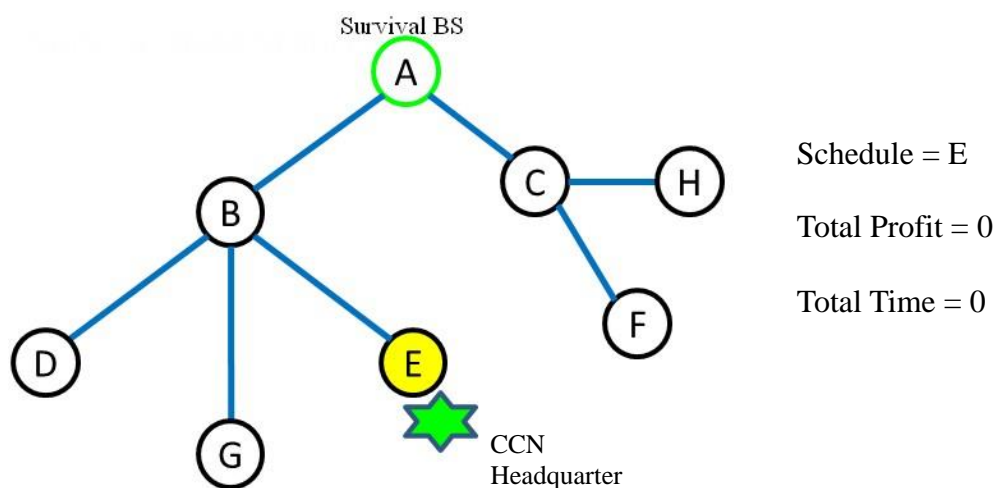


圖 4.18、DS-UCB 演算法步驟一

如圖 4.19 所示，由於節點 E 沒有 profit，因此往父節點 B 排程，但 B 點依然是孤立點無救災效益，再繼續往父節點 A 排程，由於節點 A 是連網台具有 profit，因此得到一個初步的排程 E→B→A。排程直到父節點有 profit 為止。

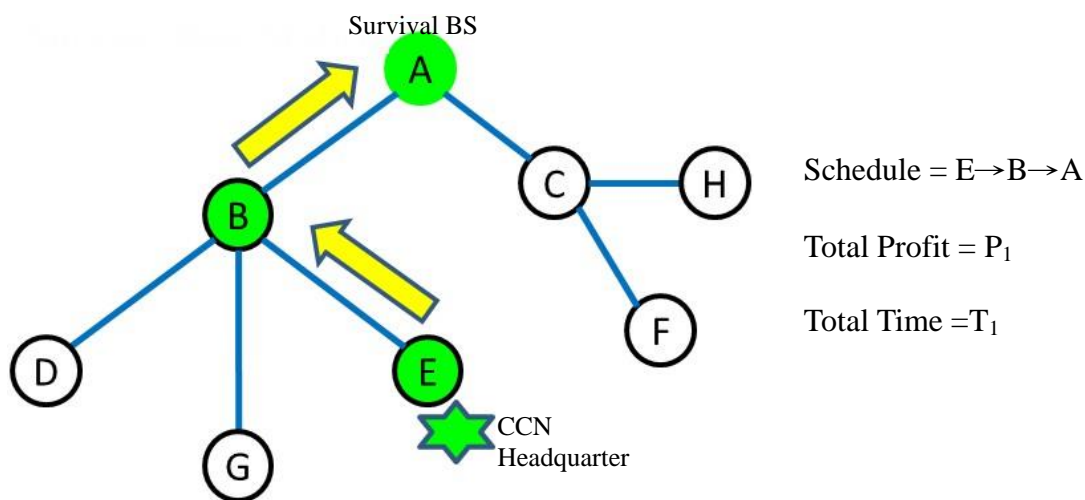


圖 4.19、DS-UCB 演算法步驟二

如圖 4.20 所示，此步驟為 Backtrack，根據上一步驟所得的初始排程 E→B→A，將鄰近節點也加入初始排程，產生有限個的排程如圖 4.21、圖 4.22，亦考慮由出發點 E 直接到連網台節點 A 的排程，比較這些排程的總救災效益以及總花費時間。

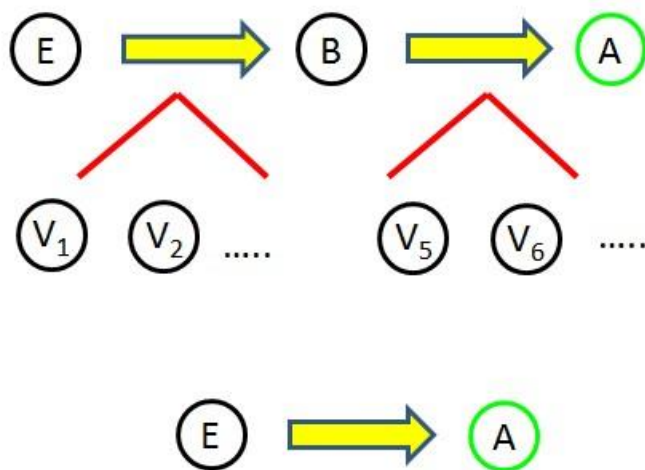


圖 4.20、DS-UCB 演算法步驟三

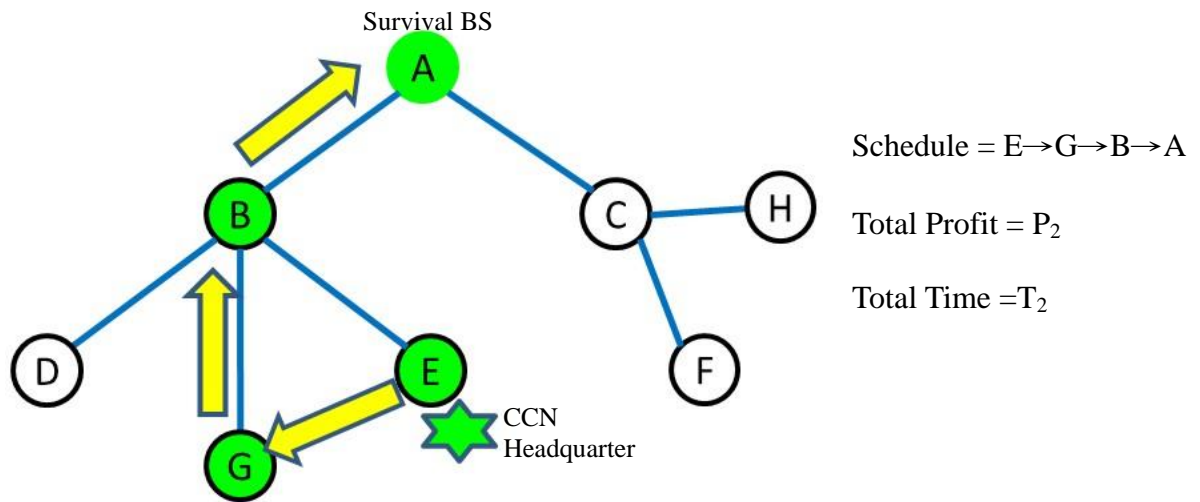


圖 4.21、DS-UCB 演算法步驟三(排程範例 1)

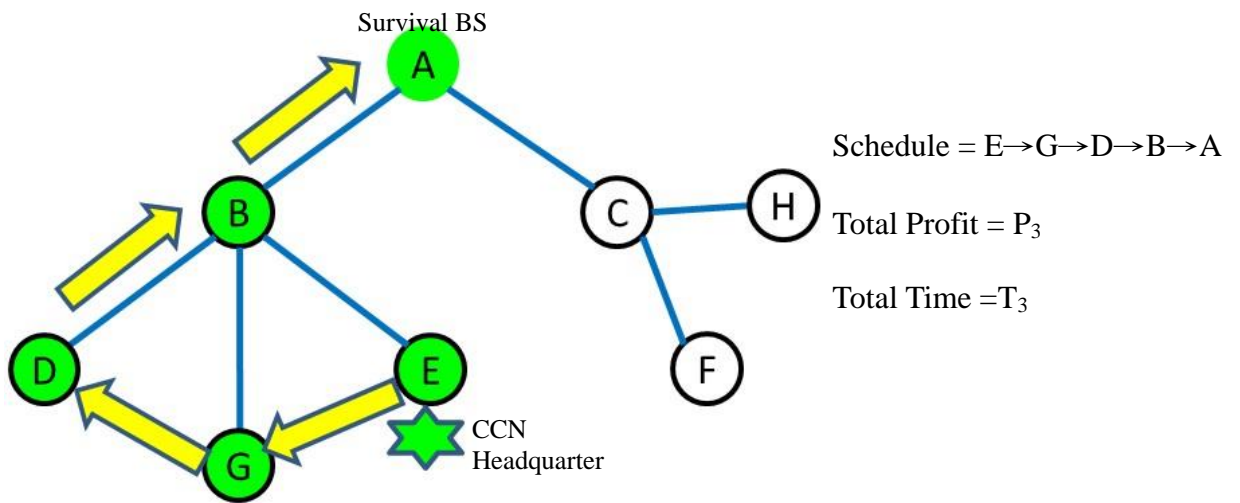


圖 4.22、DS-UCB 演算法步驟三(排程範例 2)

由這些 Backtrack 產生的有限組排程與初始排程中，互相比較彼此的總救災效益及總時間，挑選出單位時間獲得最大救災效益的排程，定為此階段的修復排程。

經過 Backtrack 後，救災團隊所在位置是有 profit 的節點 A，下一步地挑選方式，則是選擇最靠近的點加入排程，因此可能遭遇兩種狀況。狀況一：下一節點亦可獲得 profit

服務災區，以此例而言，修復 C、D、G 皆可服務災區，此時直接更新其總救災效益與總時間，如圖 4.23 所示。

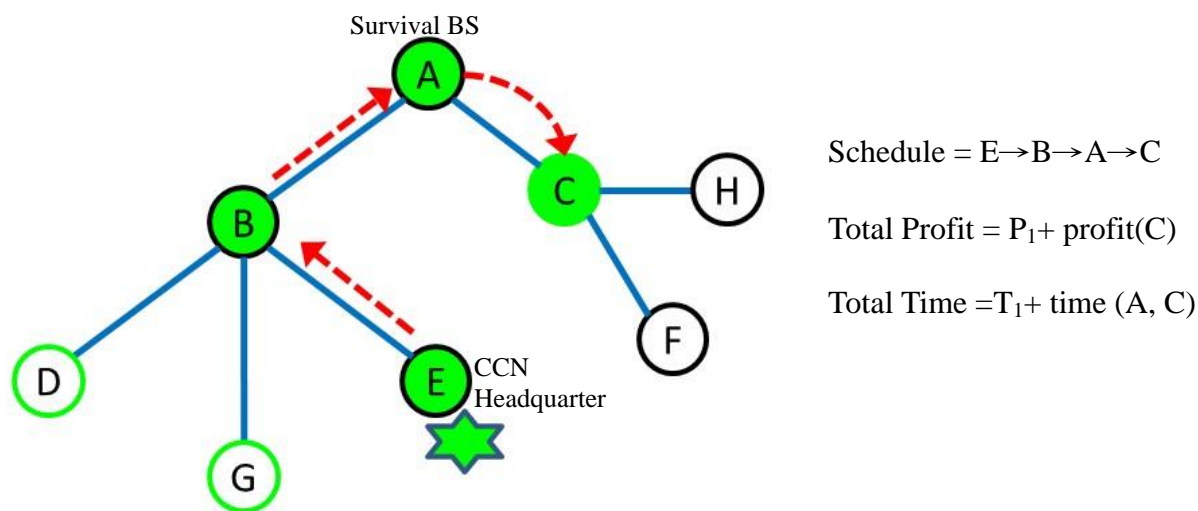


圖 4.23、DS-UCB 演算法步驟四(狀況一)

狀況二：下一節點暫時無法提供服務，以此例而言，修復 H、F 皆無法獲得 profit，如圖 4.24，此時以 H 為出發點，重複步驟二至步驟三，往父節點排程直到父節點擁有 profit 並且執行 Backtrack。重複步驟一至步驟四，直到所有節點皆被排程為止。

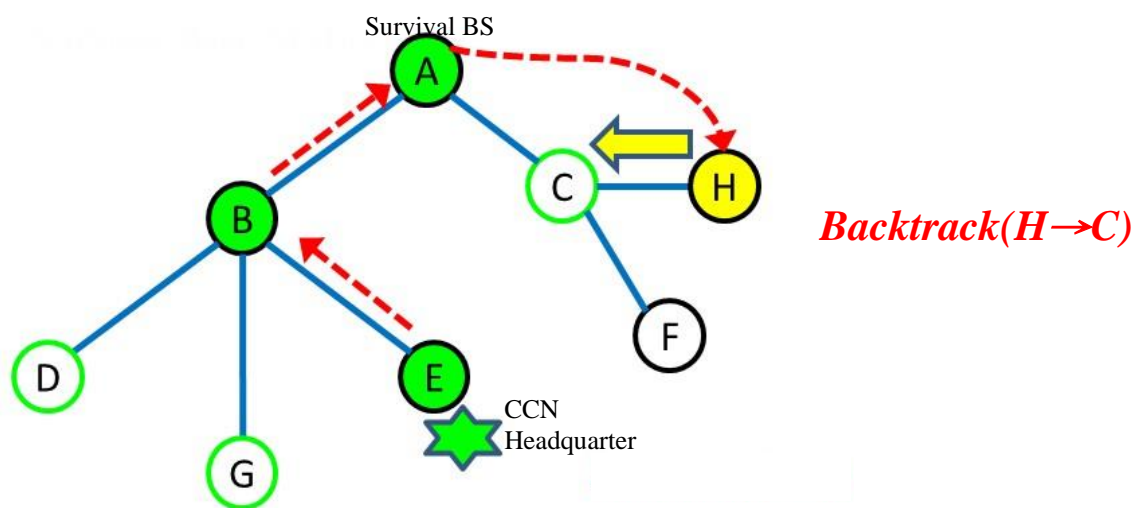


圖 4.24、DS-UCB 演算法步驟四(狀況二)

4.6.3、 Genetic Algorithm

在 3.7.3 章節中有介紹過各種後啟發式演算法(Meta-heuristics)，其中就包含了基因演算法(Genetic Algorithm)。Genetic Algorithm 理論來自達爾文進化論的觀點，物種靠不斷的演化而產生最適合生存的物種，其方法是將問題的解比擬成染色體，染色體分裂的過程，利用三個基本遺傳算子：選擇(selection)、交叉(crossover)、變異(mutation)，將優良的基因會不斷遺傳下去，把好的解不斷分裂再結合，一代一代的演化出更適應環境的基因，成為更好的答案。Genetic Algorithm 是目前在求解 Traveling Salesman Problem (TSP)中較常被使用的方法之一，基因的編碼方式，可直接使用染色體的編碼方式來代替城市的走訪順序，而本文所提出的問題模型 CCNDS-UC 欲從 CCN 拓樸中求出一組救災效益最大化的建置排程，不像 CCNDS-AC 受到 CCN 拓樸中基地台父子關係的牽制，換個角度想類似於 TSP，基地台即為所需拜訪的城市、基地台與基地台之間的路徑花費時間等同於城市與城市之間的路徑成本。因此本文試想用 Genetic Algorithm 的基因演化精神來解決 CCNDS-UC 此問題，並在實驗中與 DS-UCB 相互比較。

■ Genetic Algorithm 演算法流程圖

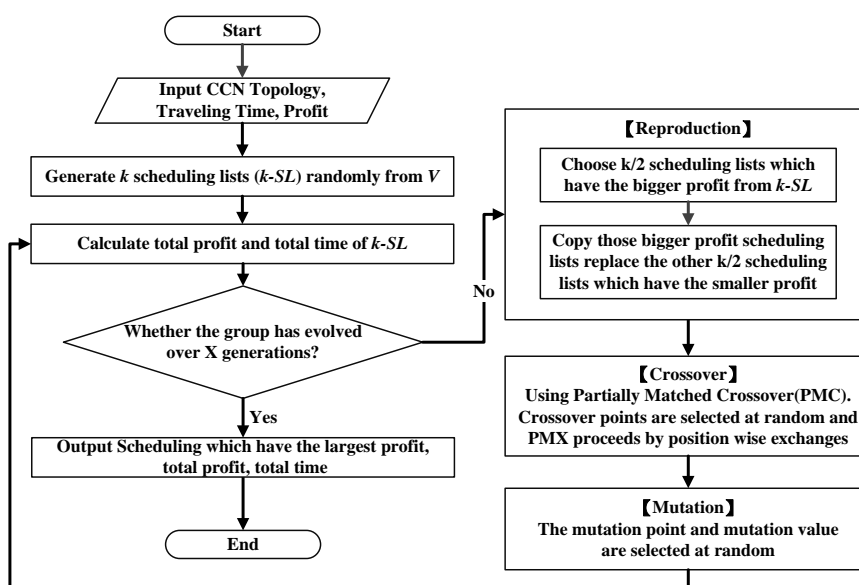


圖 4.25、 Genetic Algorithm 演算法流程圖

```

Genetic Algorithm Pseudo code:
Input CCN Topology, Traveling Time, Profit
SL=null      /*put the Scheduling Lists*/
Temp_SL=null
n=V.length()
while (SL.length<k) do
    temp_list=generate a scheduling list which is length n from V
    SL.add(temp_list)
end while
while (generation<X) do
    for each scheduling_list in SL do
        Caculate the total_profit of scheduling_list
        Caculate the total_time of scheduling_list
    end for
    Choose scheduling list which have the bigger profit from SL      /* selection */
    Delete the other scheduling list which have the smaller profit from SL
    for each scheduling_list in SL do      /*reproduction*/
        temp_list= scheduling_list
        Temp_SL.add(temp_list)
    end for
    SL.add(Temp_SL)
    for each scheduling_list in SL do      /*crossover*/
        PMC(scheduling_list_a, scheduling_list_b)
    end for
    for each scheduling_list in SL do      /*mutation*/
        rp=math.random()*n
        rv=math.random()*n
        if scheduling_list[rp]!=rv
            exchange(scheduling_list[rp],rv)
        end if
    end for
    generation=generation+1
end while

```

圖 4.26、演算法 Genetic Algorithm 之 Pseudo Code

■ Genetic Algorithm 演算法範例

我們以一個簡單的範例說明 Genetic Algorithm 演算法，由於此演算法在演算過程中並不單一考慮各基地台的救災效益、路網成本及 CCN 樹狀拓樸，而是著重在基本遺傳的選擇(selection)、交叉(crossover)、變異(mutation)，將優良的基因會不斷遺傳下去，把好的解不斷分裂再結合，每一階段基因演化結束才參照路網成本及救災效益時間總表進行救

災效益的計算。因此在此以每階段 Genetic Algorithm 中基因演化步驟如何進行講述。

如圖 4.27，第一步驟單純只看輸入資訊 CCN 樹狀拓樸中的基地台，即點的集合 A、B、C、D、E、F，由這些點隨機產生 K 組的排列組合，這些排列組合各自對應到不同的 CCN 建構排程順序，此即為初始的染色體。

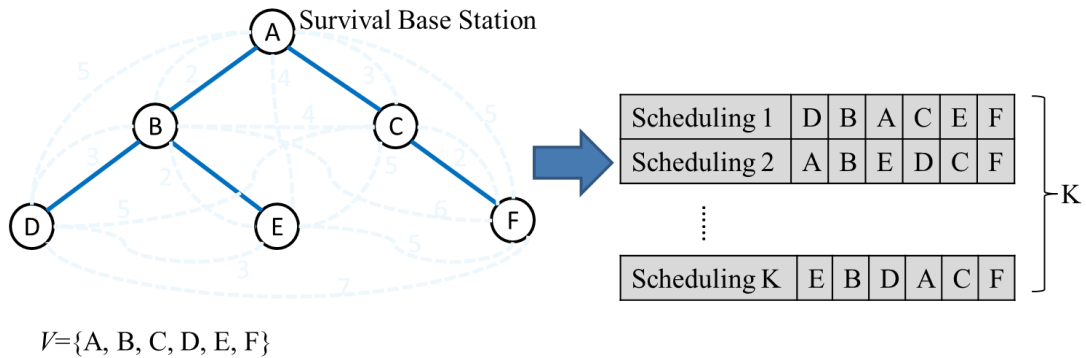


圖 4.27、Genetic Algorithm 演算法步驟一

第二步驟，由於一組排列組合即對應到一組建構排程的順序，在此步驟根據 CCN 樹狀拓樸、路徑成本以及救災效益時間總表計算 K 組建構排程各自的 total profit 及 total time。

如圖 4.28 所示，第 2 組建構排程所計算出來的 total profit 是 37。

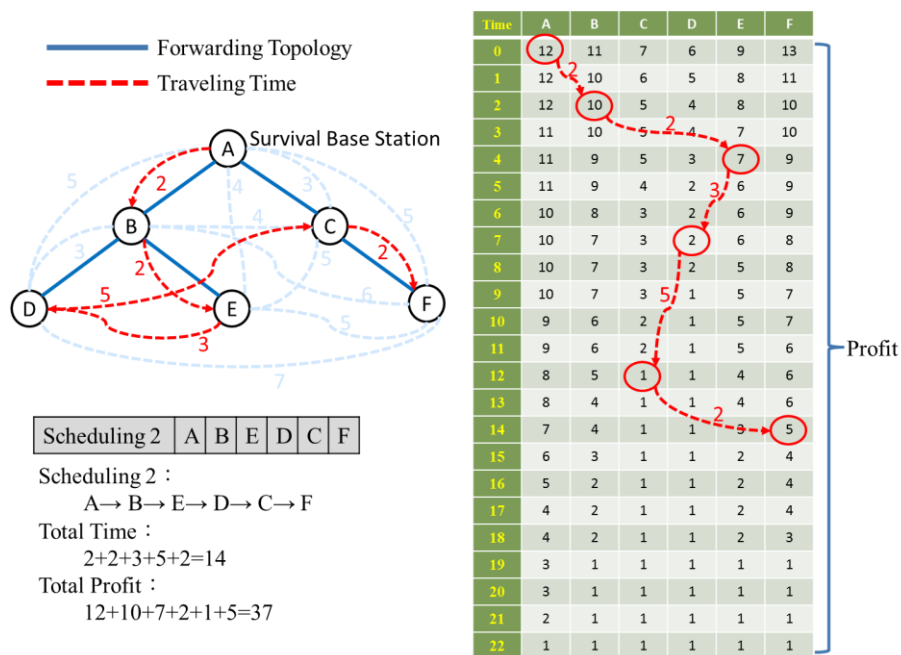


圖 4.28、Genetic Algorithm 演算法步驟二

第三步驟，由於第二步驟已將 K 組建構排程的 total profit 都計算出來，此步驟將 K 組 total profit 互相比較，選擇前二分之一較高的 total profit 保存下來，也就是將表現較好的染色體留下來，將表現差的染色體丟棄，如圖 4.29 所示，最後僅留下了 K/2 組 total profit 較高的建構排程。

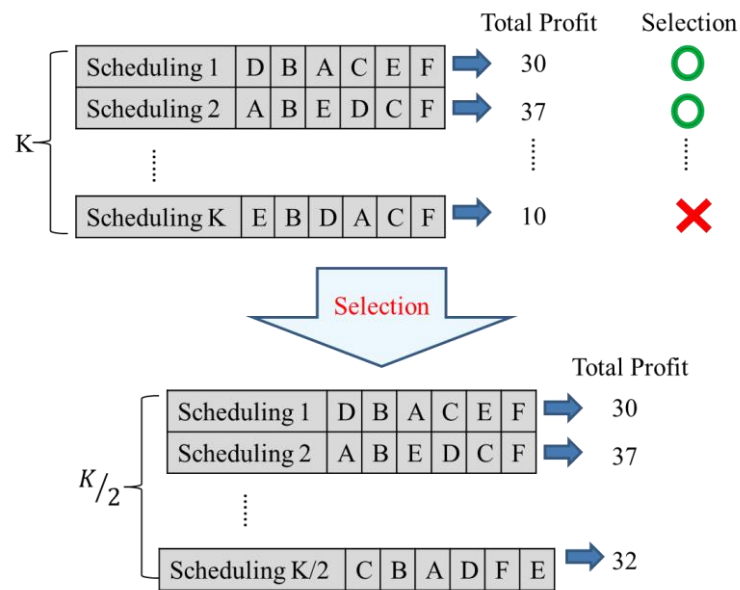


圖 4.29、Genetic Algorithm 演算法步驟三

如圖 4.30，第四步驟將第三步驟篩選過後所留下的建構排程，每一組都複製一份，建構排程的解由 K/2 組恢復到 K 組

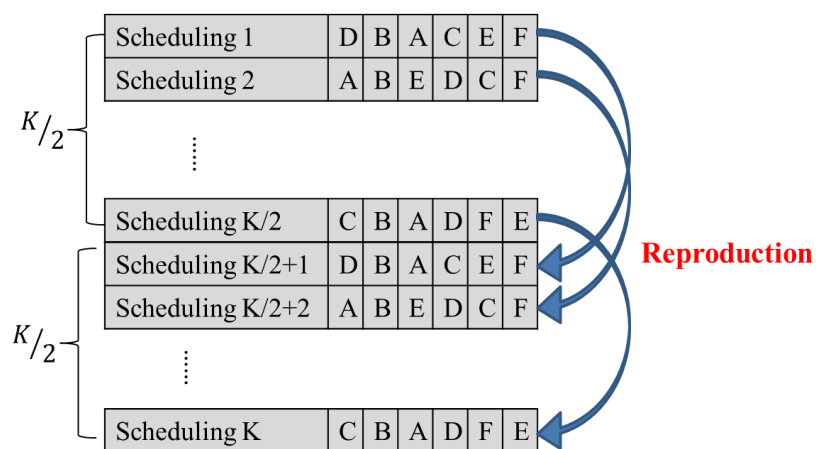


圖 4.30、Genetic Algorithm 演算法步驟四

第五步驟，為了產生出更好的解，將這些已篩選過的建構排程進行交叉(crossover)的動作，如圖 4.31 所示，當兩組建構排程進行交叉動作時，演算法會隨機產生一個值 R_i ，代表即將產生交叉的建構排程位置，圖中的例子 $R_i=5$ ，因此 Scheduling 1 位置 5 的 E 要轉換成 C，而 Scheduling 2 位置 5 的 C 要轉換成 E，但 CCN 建構排程在此具有不重複建構的特性，因此將 Scheduling 1 中原來的值為 C 的位置 4 改成 E，將 Scheduling 2 中原來的值為 E 的位置 3 改成 C，完成交叉的動作。

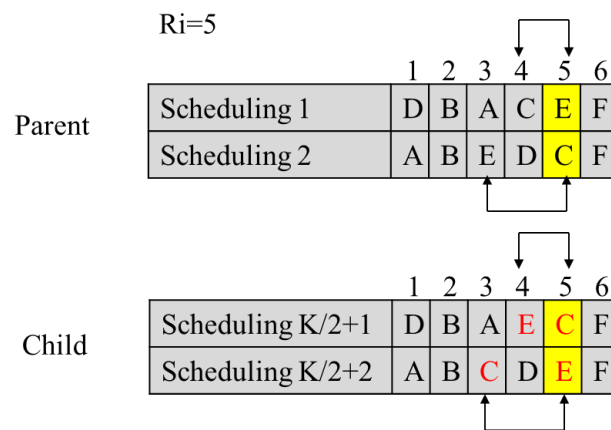


圖 4.31、Genetic Algorithm 演算法步驟五

第六步驟是突變(mutation)，隨機產生兩個值 RP 代表需變異的位置， RV 代表變異後的位置，因為不可重複的特性，在此可視為是單一建構排程內的基地台建構順序交換，如圖 4.32，位置 5 的值為 E，位置 1 的值為 D，經過突變動作後，位置 1 的值變成了 E 而位置 5 的值變成了 D。

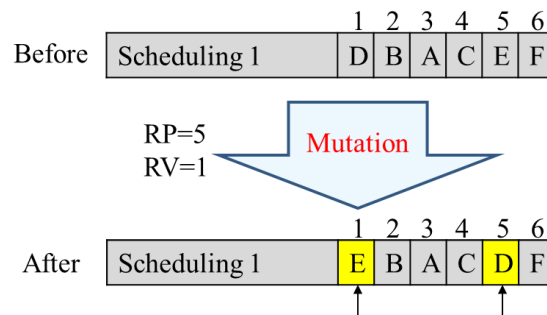


圖 4.32、Genetic Algorithm 演算法步驟六

重複第二步驟到第六步驟直到已經基因演化 X 代，演算法才算結束。以上即為 Genetic Algorithm 的演算法步驟。

4.6.4、演算法時間複雜度分析(Time Complexity)

本文進一步觀察各演算法的時間複雜度，如表 4.1，DS-G、DS-ACG 及 DS-UCB 其時間複雜度皆為多項式時間；而 Genetic Algorithm 則受到基因演化次數 X 及初始建構排程組數 K 的牽制，當基地台個數變多時，基因演化次數 X 也須跟著變多才得以發揮 Genetic Algorithm 求取較好的解；最佳解是由暴力法求出，將所有可能的解一一排列出後進行總救災效益與總救災時間之運算，因此其時間複雜度為 $O(n!)$ ，是最耗時的演算法。

表 4.1、各演算法時間複雜度

Algorithm Node Size	DS-UCB	DS-ACG	DS-G	Genetic Algorithm	Optimal Solution
Time Complexity	$O(n^3)$	$O(n^2)$	$O(n^2)$	$O(K * X * n)$	$O(n!)$

第5章、效能評估

5.1、實驗目的

我們對 CCNDS-AC 及 CCNDS-UC 此兩個問題模型所提出之演算法 DS-ACG 與 DS-UCB 進行效能評估，亦與常用於解決旅行推銷員問題、工作排程問題等問題之演算法 Genetic Algorithm 所求得之解互相比較，Genetic Algorithm 是一種優勝劣汰的進化過程的演算法，利用三個基本遺傳算子(genetic operator): 選擇(selection)、交叉(crossover)、變異(mutation) 使過程中的解，一代又一代的優化。並以暴力法求得最佳解，判斷演算法解與最佳解的差距，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 DS-ACG 及 DS-UCB 演算法的效能，分別觀察在大規模實驗與小規模實驗下與 Genetic Algorithm 解、最佳解之差距。

5.2、實驗設計

我們利用隨機的方式產生大量 CCN 拓撲，並分別針對小規模實驗及大規模實驗進行 DS-ACG、DS-UCB 及[15]所提出的演算法 DS-G、Genetic Algorithm 及暴力法運算，以求取 CCN 的建置排程。

5.2.1、實驗環境

我們利用 Java 實作 DS-ACG、DS-UCB、DS-G、Genetic Algorithm 及暴力法此幾種演算法進行比較，並隨機產生大量模擬案例，以計算各演算法在不同環境條件下之效能。使用的電腦軟硬體規格如表 5.1 所示。

表 5.1、電腦軟硬體規格

系統類型	Windows 7, 32bit
處理器	Intel(R) Core(TM)2 Duo CPU E6550 @ 2.33GHz 1.87GHz
記憶體	2.00 GB
開發環境	Eclipse JAVA

5.2.2、救災效益函數模型

基地台的救災效益是指『當基地台恢復服務時，對所在之災區所能提供的救災效益』，而救災效益函數代表基地台的救災效益隨著時間的推移而改變，將會越來越低。根據經驗，救災效益函數的下降曲線絕非固定線性函數，但在災難發生緊急時，救災單位可能無法在短時間內精確的定義救災效益函數，因此本論文參考[22]提出了三個線性模型，使用於實驗中，這些救災效益函數模型可提供救災單位作為參考。

我們設計的線性模型包含四個變數：

- P：Initial Profit，初始救災效益。
- T：Turning point，救災效益惡化率的轉換點。
- S_1 ：First deterioration rate，第一階段救災效益惡化率(斜率)。
- S_2 ：Second deterioration rate，第二階段救災效益惡化率(斜率)。

■ 救災效益函數模型一：

如圖 5.1 所示，孤立台一直維持相同的救災效益，可能此孤立台位於 CCN 拓樸重要的溝通橋樑，但當此孤立台已獲得通訊功能時(時間點 T)搶修此孤立台不會有任何效益。

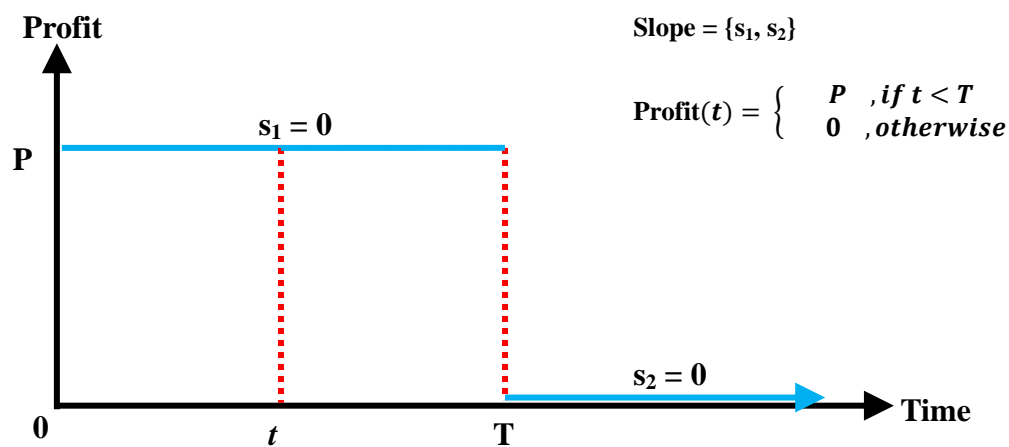


圖 5.1、救災效益函數模型一 (Profit function 1)

■ 救災效益函數模型二：

如圖 5.2 所示，救災效益從一開始就隨著時間線性下降，超過時間點 T，救災團隊在預計時間後搜救行動可完成、或超過 72 小時無生還機會、或有其他通訊方式可取代…等，此孤立台就變成沒有救災效益的情況。

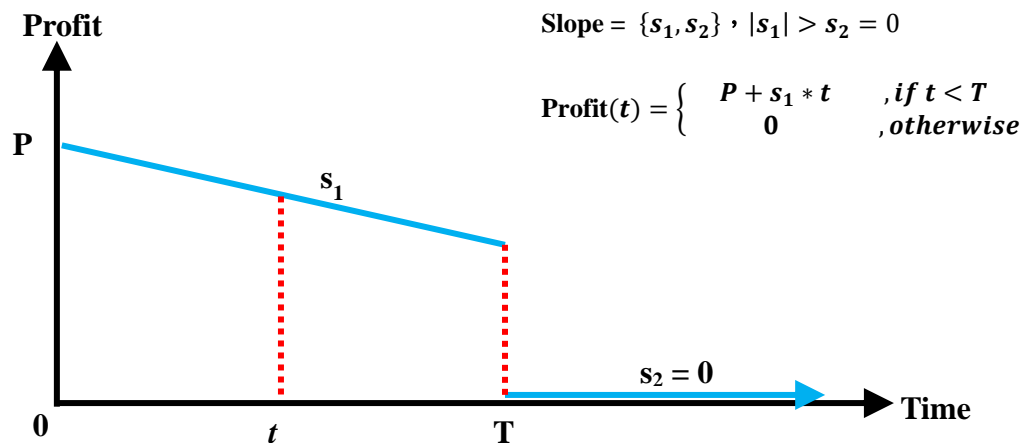


圖 5.2、救災效益函數模型二 (Profit function 2)

■ 救災效益函數模型三：

如圖 5.3，類似救災效益函數模型二，差別在於生還者在過了時間點 T 之後還有存

活機會，但隨著時間的推移下降得更快速。

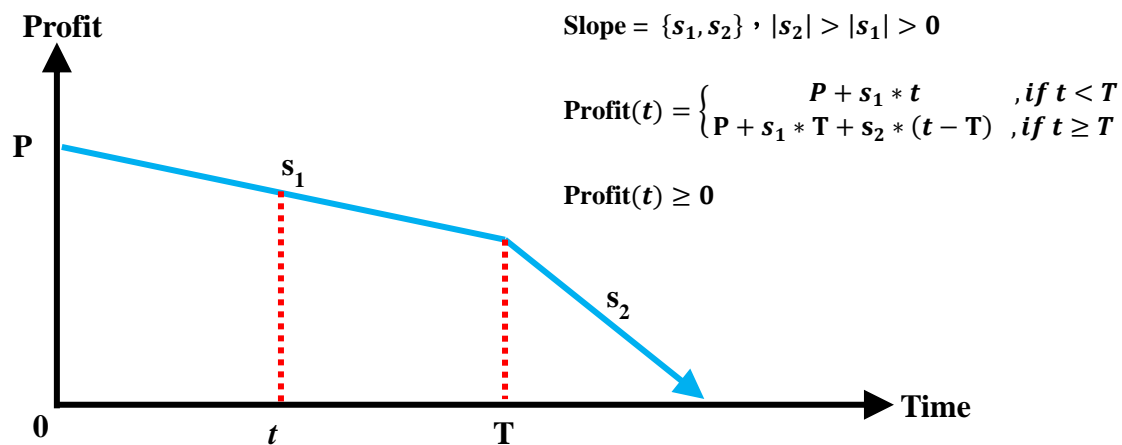


圖 5.3、救災效益函數模型三 (Profit function 3)

5.3、實驗一：小規模實驗

5.3.1、實驗參數

在小規模時間中 Genetic Algorithm 將產生 4 組的建構排程並且演化 100 次進行比較。

表 5.2、實驗一：小規模實驗參數設定

Parameters	Range of values
Initial Profit	Uniform(30, 100)
Traveling Time	Uniform (0.5, 10) hr
Forwarding Tree Size	8、9、10、11、12
Slope S_1	Uniform (-1, 0)
Slope S_2	Uniform (-1, 0)
Turning Point	Uniform (50, 80) hr
Genetic Algorithm(K, X)	(K, X)=(4, 100)

5.3.2、評估指標

實驗結果以以下之評估指標進行評估，包括：

- Total Profit：演算法修復節點之當下獲得效益之加總。
- Total Time：修復完成整個 CCN 拓樸後，所花費的總建構排程時間。
- Original Deviation：與最佳解比較總效益之誤差，如公式 2 所示。

$$\text{Original Deviation} = 1 - \frac{\text{Algorithm solution}}{\text{Optimal solution}} \dots\dots\dots(2)$$

- Normalize Deviation：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 3 所示。

$$\text{Normalize Deviation} = \frac{\text{Optimal solution} - \text{Algorithm solution}}{\text{Optimal solution} - \text{Worst solution}} \dots\dots\dots (3)$$

5.3.3、實驗結果

表 5.3 至表 5.12 及圖 5.4 至圖 5.33 為實驗一小規模實驗的實驗結果，可看出 DS-UCB 的總救災效益 Original Deviation 平均為 0.9%左右，即 DS-UCB 總救災效益相當逼近 Optimal solution，總救災時間 DS-UCB 亦較其他演算法明顯逼近 Optimal solution，大大改善了 DS-G，優於 DS-ACG。在小規模實驗中，DS-UCB 總救災效益略勝於 Genetic Algorithm 所得的總救災效益，但差距並不明顯，可見 Genetic Algorithm 確實有其優勢，但 DS-UCB 透過路程花費時間、救災效益、樹狀拓樸關係等因素考量，擺脫了 CCNDS-AC 綁手綁腳的父子關係約束，DS-UCB 由鄰近節點優先考慮的情況下，使得所求出的建構排程，不僅提高了救災效益，也降低了建構排程總花費時間亦使得建構排程更加人性化。

表 5.3、實驗一：總救災效益實驗結果與誤差比較 (8 個基地台)

Case	Total Profit						Original Deviation of Profit				Normalize Deviation of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	478.2	474.5	435.5	202.2	166.5	470.3	0.8%	8.9%	57.7%	1.7%	1.2%	13.7%	88.6%	2.6%
2	537.9	532.8	509.5	186.8	106.0	526.8	0.9%	5.3%	65.3%	2.1%	1.2%	6.6%	81.3%	2.6%
3	515.1	509.2	466.0	191.5	158.4	507.2	1.2%	9.5%	62.8%	1.5%	1.7%	13.8%	90.7%	2.2%
4	499.5	499.1	465.2	254.5	173.7	495.2	0.1%	6.9%	49.1%	0.9%	0.1%	10.5%	75.2%	1.3%
5	503.1	501.7	460.0	187.8	158.8	493.5	0.3%	8.6%	62.7%	1.9%	0.4%	12.5%	91.6%	2.8%
6	522.2	520.4	476.8	185.3	127.8	516.7	0.3%	8.7%	64.5%	1.1%	0.5%	11.5%	85.4%	1.4%
7	549.8	547.3	478.8	229.3	189.4	542.6	0.4%	12.9%	58.3%	1.3%	0.7%	19.7%	88.9%	2.0%
8	509.4	508.0	475.0	172.8	138.1	505.1	0.3%	6.8%	66.1%	0.8%	0.4%	9.3%	90.6%	1.2%
9	541.0	539.7	435.5	172.3	136.3	536.5	0.2%	19.5%	68.1%	0.8%	0.3%	26.1%	91.1%	1.1%
10	445.9	444.5	373.2	244.2	169.4	442.0	0.3%	16.3%	45.2%	0.9%	0.5%	26.3%	73.0%	1.4%
Average	510.2	507.7	457.6	202.7	152.4	503.6	0.5%	10.3%	60.0%	1.3%	0.7%	15.0%	85.6%	1.9%

Opt.=Optimal Solution. GA=Genetic Algorithm.

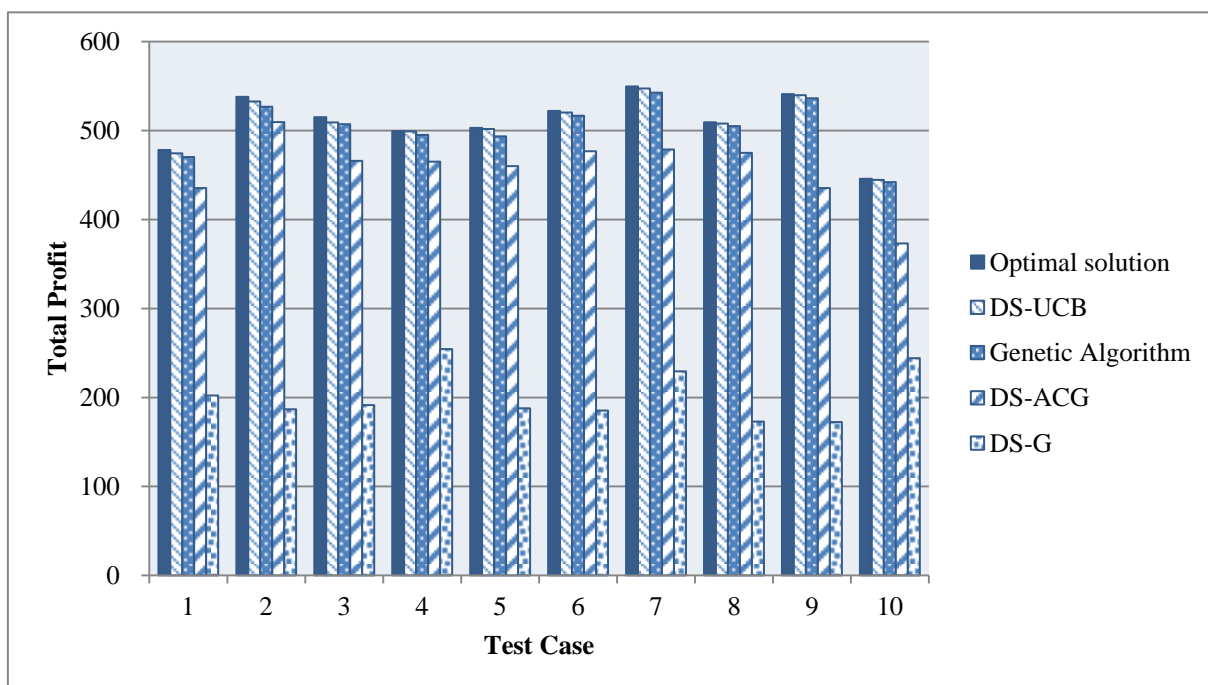


圖 5.4、實驗一：Total Profit (8 nodes)

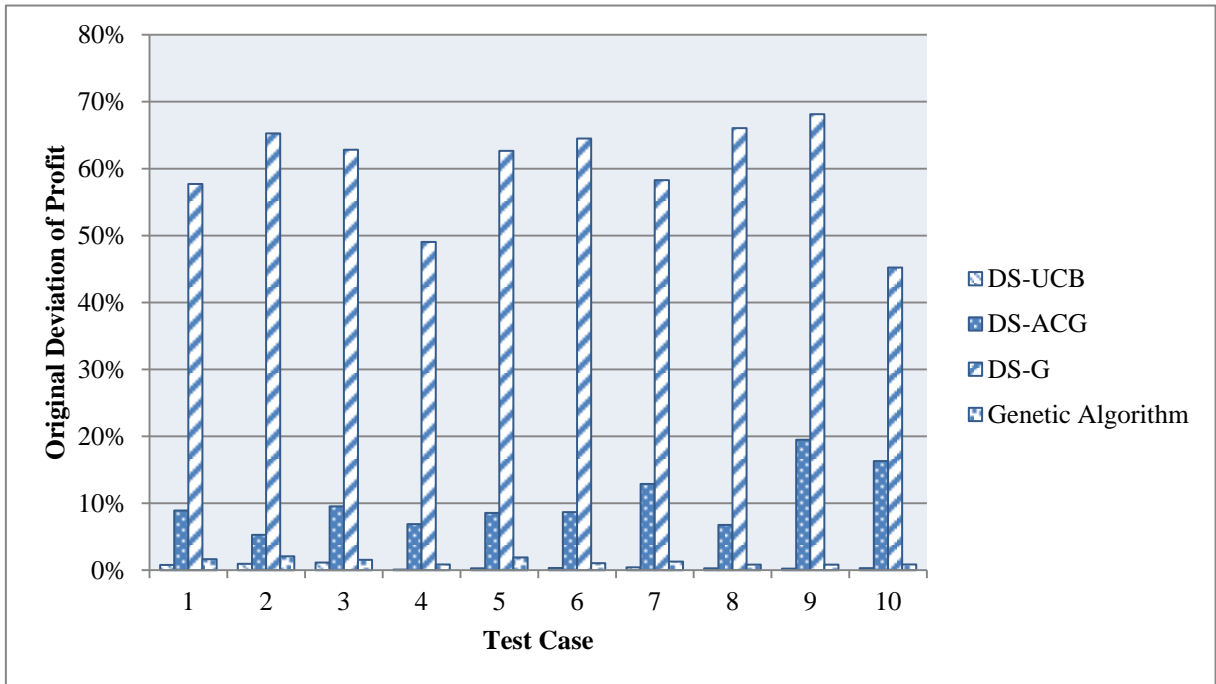


圖 5.5、實驗一：Original Deviation of Profit (8nodes)

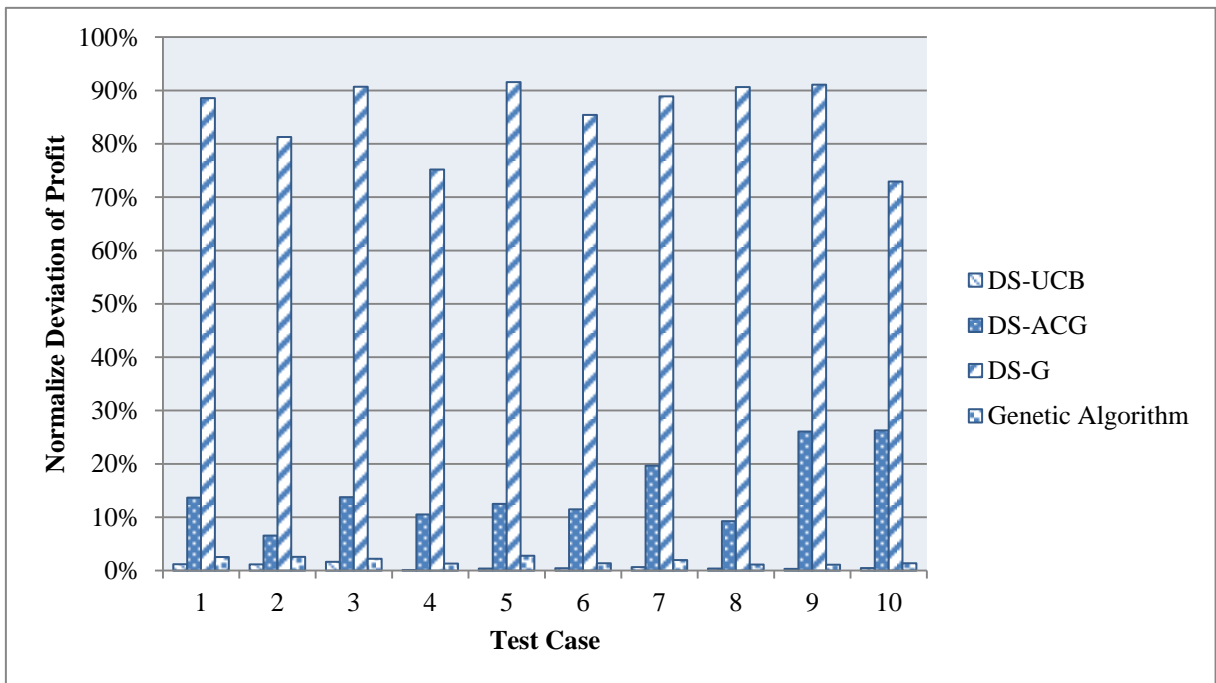


圖 5.6、實驗一：Normalize Deviation of Profit (8 nodes)

表 5.4、實驗一：總救災時間實驗結果與誤差比較 (8 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	20.5	20.8	24.8	90.9	59.1	19.7	1.6%	21.1%	343.3%	4.1%	0.9%	11.2%	182.1%	2.2%
2	19.8	20.2	23.5	97.2	54.7	21.0	1.7%	18.5%	390.0%	5.9%	1.0%	10.5%	221.7%	3.3%
3	22.0	19.7	26.7	111.6	56.1	20.3	10.6%	21.2%	407.4%	7.6%	6.8%	13.7%	262.7%	4.9%
4	21.3	20.0	25.8	87.7	57.8	20.3	6.3%	21.1%	311.2%	4.7%	3.7%	12.3%	182.0%	2.7%
5	20.2	19.2	25.7	109.5	54.6	21.7	5.0%	27.3%	442.8%	7.4%	2.9%	16.0%	259.5%	4.4%
6	20.3	18.7	25.7	108.0	58.4	20.7	8.2%	26.2%	431.4%	1.6%	4.4%	14.0%	230.6%	0.9%
7	20.5	17.2	24.3	96.0	58.5	20.7	16.3%	18.7%	368.2%	0.8%	8.8%	10.1%	198.6%	0.4%
8	21.5	20.8	26.8	120.6	55.5	19.7	3.1%	24.8%	460.9%	8.5%	2.0%	15.7%	291.3%	5.4%
9	20.0	20.7	27.0	112.9	56.7	21.3	3.3%	35.0%	464.6%	6.7%	1.8%	19.1%	253.3%	3.6%
10	19.0	17.3	27.2	90.2	57.8	21.3	8.8%	43.0%	375.0%	12.3%	4.3%	21.0%	183.6%	6.0%
Average	20.5	19.5	25.8	102.5	56.9	20.7	6.5%	25.7%	399.5%	0.7%	3.6%	14.4%	226.5%	0.4%

Opt.=Optimal Solution. GA=Genetic Algorithm.

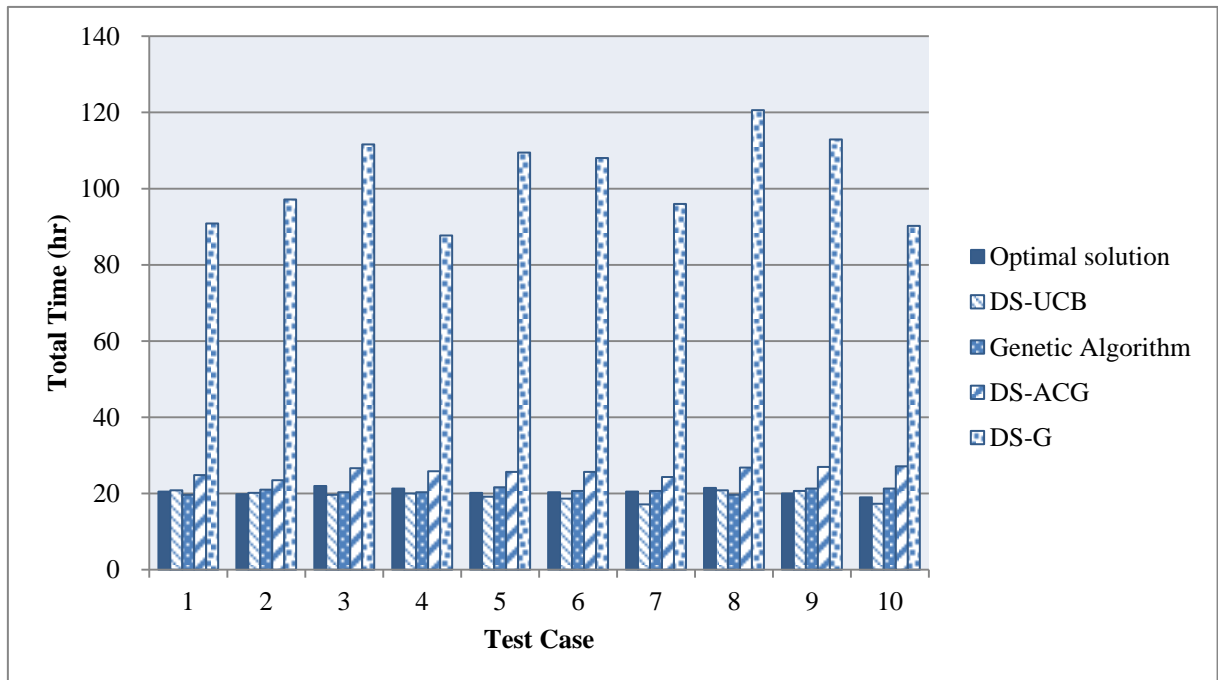


圖 5.7、實驗一：Total Time (8 nodes)

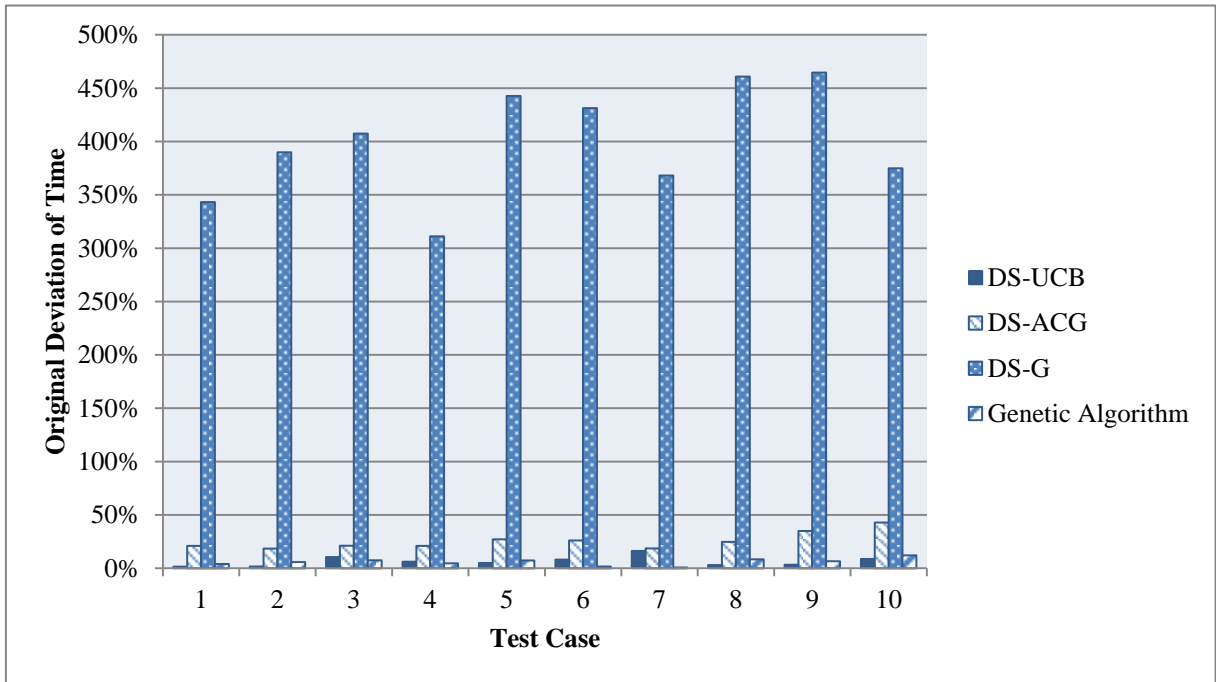


圖 5.8、實驗一：Original Deviation of Time (8 nodes)

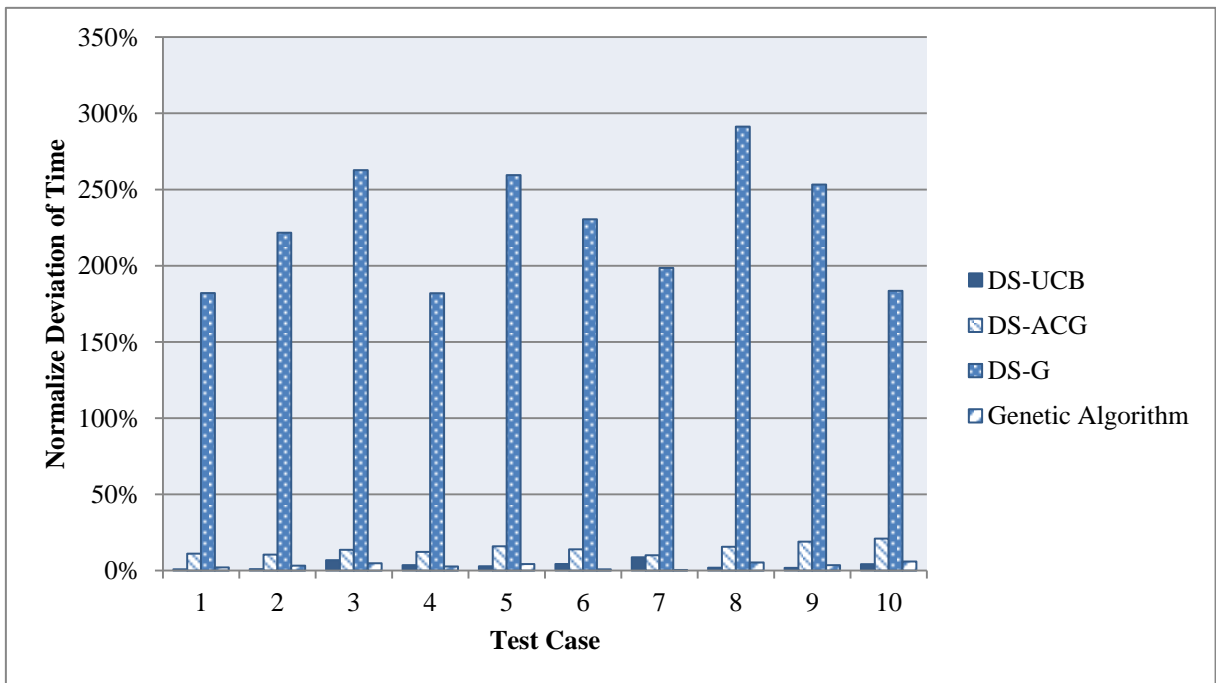


圖 5.9、實驗一：Normalize Deviation of Time (8 nodes)

表 5.5、實驗一：總救災效益實驗結果與誤差比較 (9 個基地台)

Case	Total Profit						Original Deviation of Profit				Normalize Deviation of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	598.6	593.4	547.3	468.0	263.4	582.3	0.9%	8.6%	21.8%	2.7%	1.6%	15.3%	39.0%	4.9%
2	567.0	559.9	531.7	430.3	257.7	547.2	1.3%	6.2%	24.1%	3.5%	2.3%	11.4%	44.2%	6.4%
3	573.0	569.4	542.5	432.3	269.8	554.3	0.6%	5.3%	24.5%	3.3%	1.2%	10.1%	46.4%	6.2%
4	572.5	559.5	495.3	460.3	253.9	514.6	2.3%	13.5%	19.6%	10.1%	4.1%	24.2%	35.2%	18.2%
5	563.0	556.7	504.5	426.8	244.1	545.5	1.1%	10.4%	24.2%	3.1%	2.0%	18.3%	42.7%	5.5%
6	584.7	580.6	541.0	393.8	277.0	572.0	0.7%	7.5%	32.6%	2.2%	1.3%	14.2%	62.0%	4.1%
7	618.3	613.5	579.7	469.2	231.1	608.3	0.8%	6.2%	24.1%	1.6%	1.2%	10.0%	38.5%	2.6%
8	583.7	579.7	538.3	454.3	250.0	567.2	0.7%	7.8%	22.2%	2.8%	1.2%	13.6%	38.8%	5.0%
9	627.3	622.8	574.5	440.3	302.9	612.1	0.7%	8.4%	29.8%	2.4%	1.4%	16.3%	57.6%	4.7%
10	553.2	547.0	524.0	419.8	202.7	541.9	1.1%	5.3%	24.1%	2.0%	1.8%	8.3%	38.1%	3.2%
Average	584.1	578.2	537.9	439.5	255.3	564.5	1.0%	7.9%	24.7%	3.4%	1.8%	14.2%	44.2%	6.1%

Opt.=Optimal Solution. GA=Genetic Algorithm.

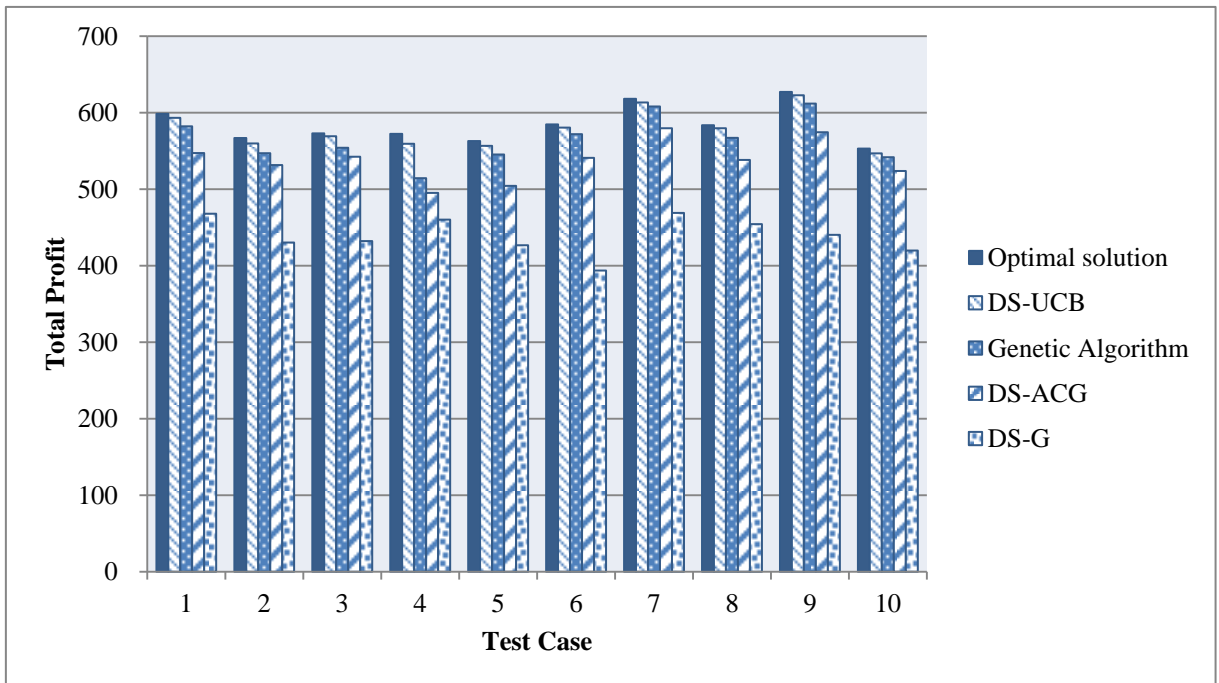


圖 5.10、實驗一：Total Profit (9 nodes)

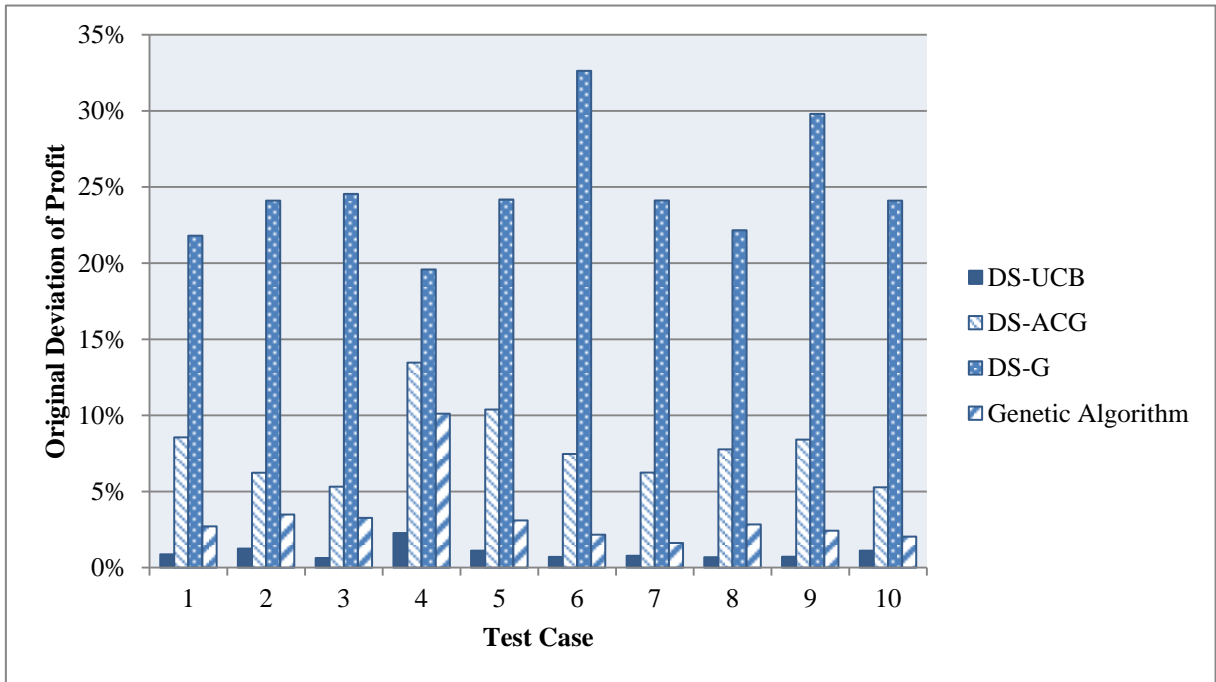


圖 5.11、實驗一：Original Deviation of Profit (9 nodes)

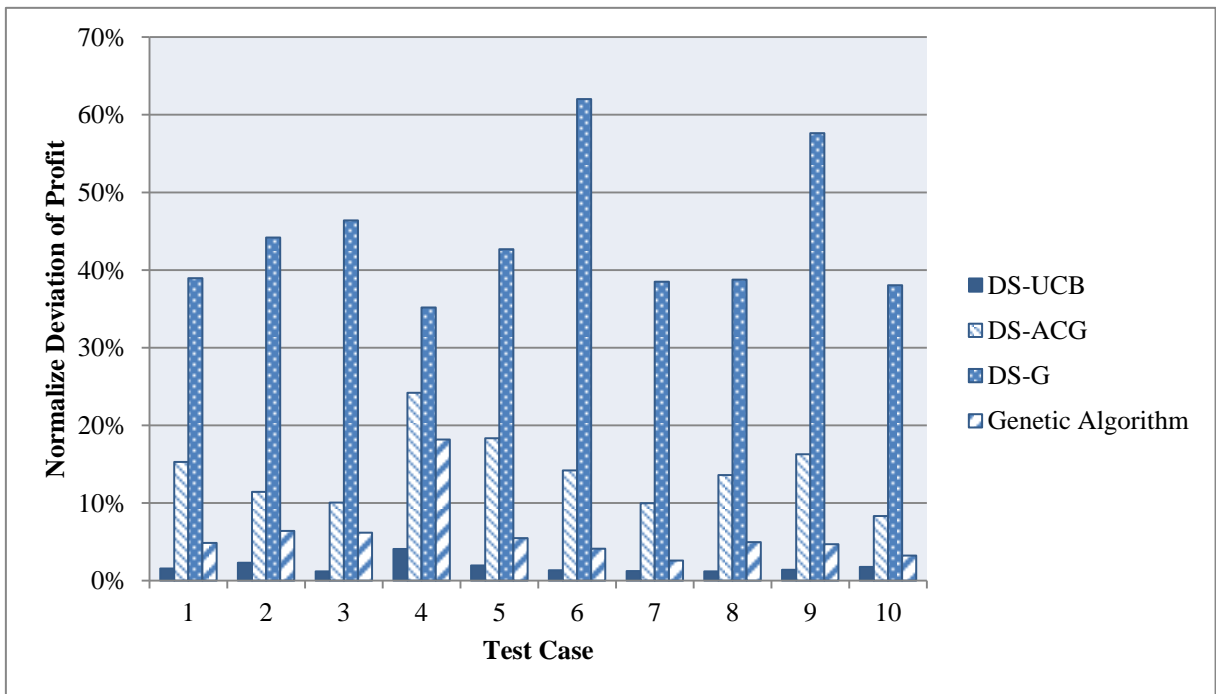


圖 5.12、實驗一：Normalize Deviation of Profit (9 nodes)

表 5.6、實驗一：總救災時間實驗結果與誤差比較 (9 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	21.2	21.8	26.7	53.7	39.3	20.7	3.1%	26.0%	153.5%	2.4%	3.7%	30.3%	178.9%	2.8%
2	19.3	20.3	25.7	50.5	39.7	22.5	5.2%	32.8%	161.2%	16.4%	4.9%	31.1%	153.3%	15.6%
3	17.2	18.2	24.8	58.0	40.0	22.2	5.8%	44.7%	237.9%	29.1%	4.4%	33.6%	178.8%	21.9%
4	20.0	24.2	32.2	55.2	39.3	21.3	20.8%	60.8%	175.8%	6.7%	21.6%	62.9%	181.9%	6.9%
5	18.3	22.5	28.3	57.8	40.3	23.3	22.7%	54.5%	215.5%	27.3%	18.9%	45.5%	179.5%	22.7%
6	20.2	19.8	25.5	71.2	41.0	22.7	1.7%	26.4%	252.9%	12.4%	1.6%	25.6%	244.8%	12.0%
7	19.7	20.5	24.5	53.8	39.0	23.5	4.2%	24.6%	173.7%	19.5%	4.3%	25.0%	176.7%	19.8%
8	21.2	20.2	28.2	55.7	38.7	22.7	4.7%	33.1%	163.0%	7.1%	5.7%	40.0%	197.1%	8.6%
9	18.3	21.7	26.8	62.2	40.0	20.8	18.2%	46.4%	239.1%	13.6%	15.4%	39.2%	202.3%	11.5%
10	20.2	23.5	25.5	56.8	39.3	22.7	16.5%	26.4%	181.8%	12.4%	17.4%	27.8%	191.3%	13.0%
Average	19.6	21.3	26.8	57.5	39.7	22.2	10.3%	37.6%	195.4%	13.7%	9.8%	36.1%	188.5%	13.3%

Opt.=Optimal Solution. GA=Genetic Algorithm.

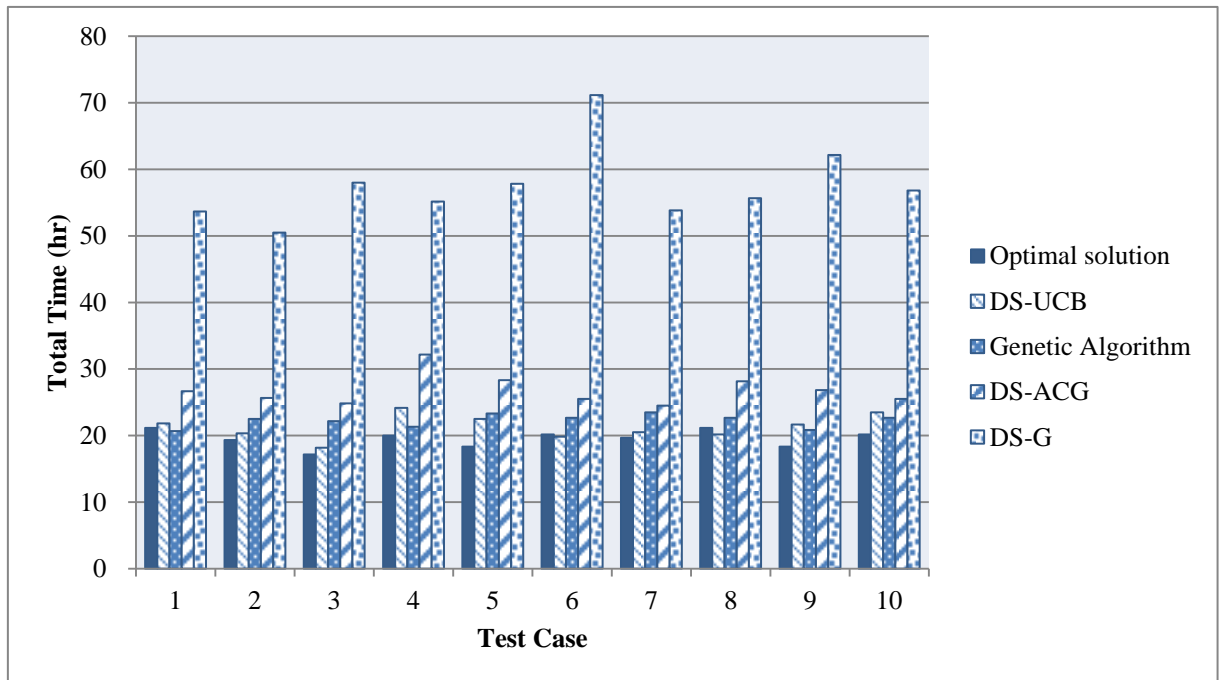


圖 5.13、實驗一：Total Time (9 nodes)

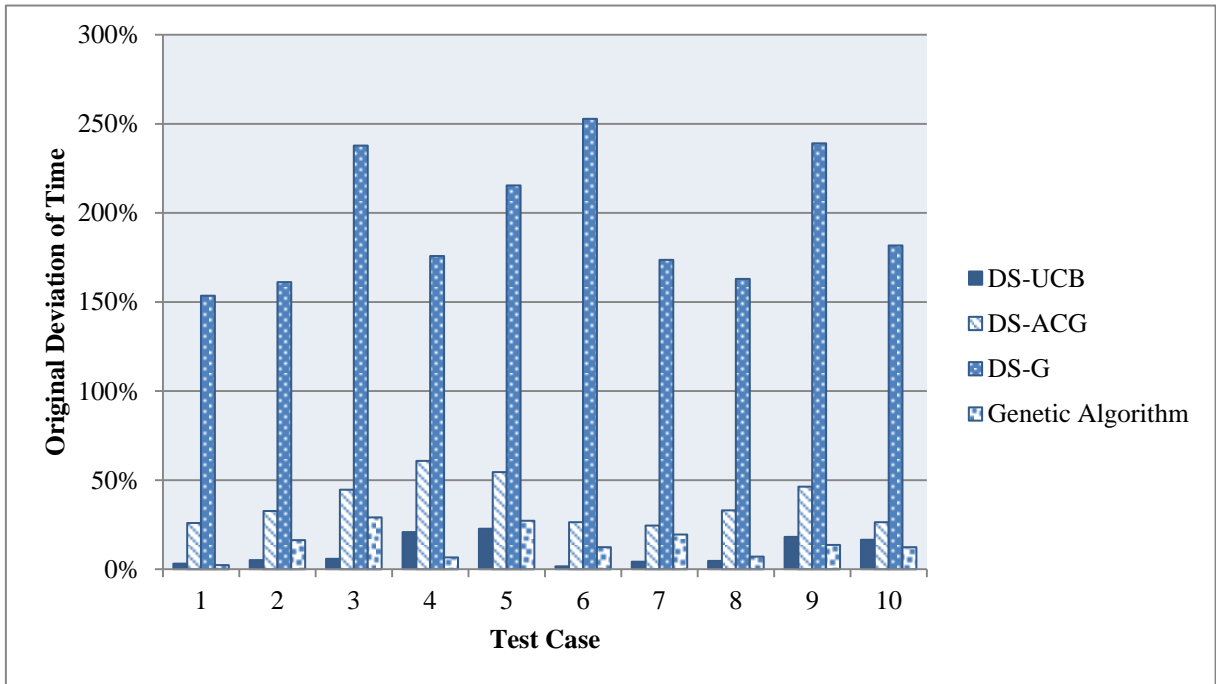


圖 5.14、實驗一：Original Deviation of Time (9 nodes)

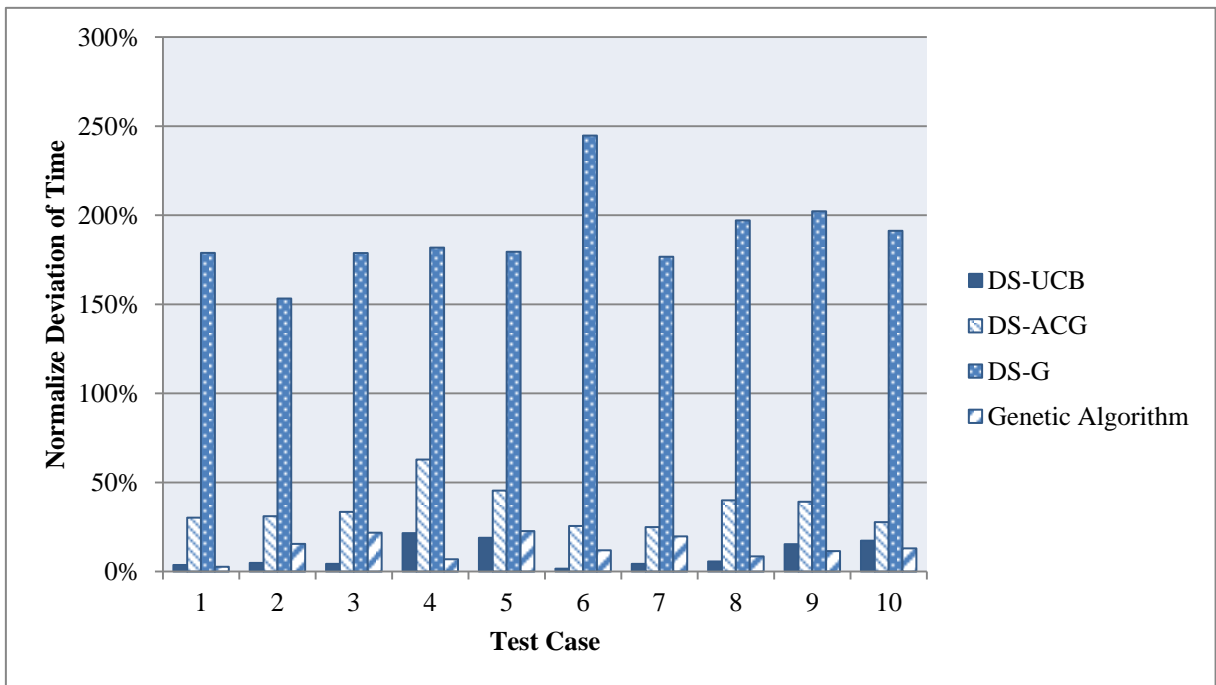


圖 5.15、實驗一：Normalize Deviation of Time (9 nodes)

表 5.7、實驗一：總救災效益實驗結果與誤差比較 (10 個基地台)

Case	Total Profit						Original Deviation of Profit				Normalize Deviation of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	724.1	711.7	617.2	415.8	217.8	680.4	1.7%	14.8%	42.6%	6.0%	2.5%	21.1%	60.9%	8.6%
2	533.5	526.7	520.7	319.7	186.7	523.5	1.3%	2.4%	40.1%	1.9%	2.0%	3.7%	61.7%	2.9%
3	565.0	559.6	538.3	361.0	252.1	555.0	0.9%	4.7%	36.1%	1.8%	1.7%	8.5%	65.2%	3.2%
4	589.1	584.4	546.5	348.8	258.6	577.5	0.8%	7.2%	40.8%	2.0%	1.4%	12.9%	72.7%	3.5%
5	615.0	607.7	526.3	338.3	210.4	596.1	1.2%	14.4%	45.0%	3.1%	1.8%	21.9%	68.4%	4.7%
6	662.4	659.2	588.7	341.8	199.0	643.2	0.5%	11.1%	48.4%	2.9%	0.7%	15.9%	69.2%	4.1%
7	666.4	659.6	611.2	371.0	225.1	658.0	1.0%	8.3%	44.3%	1.3%	1.5%	12.5%	66.9%	1.9%
8	641.0	636.0	568.2	393.2	285.2	611.2	0.8%	11.4%	38.7%	4.7%	1.4%	20.5%	69.7%	8.4%
9	681.9	676.0	617.5	398.3	314.5	675.3	0.9%	9.4%	41.6%	1.0%	1.6%	17.5%	77.2%	1.8%
10	617.7	610.8	548.7	351.7	243.0	608.8	1.1%	11.2%	43.1%	1.4%	1.8%	18.4%	71.0%	2.4%
Average	629.6	623.2	568.3	364.0	239.2	612.9	1.0%	9.5%	42.1%	2.6%	1.6%	15.3%	68.3%	4.1%

Opt.=Optimal Solution. GA=Genetic Algorithm.

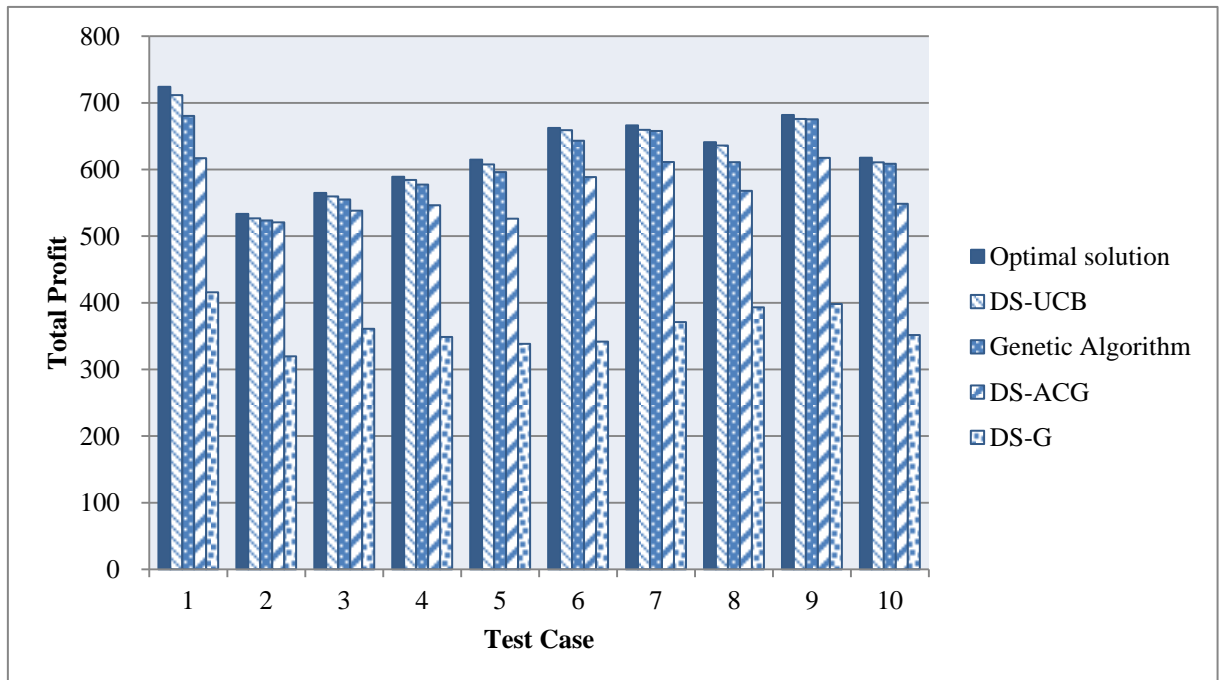


圖 5.16、實驗一：Total Profit (10 nodes)

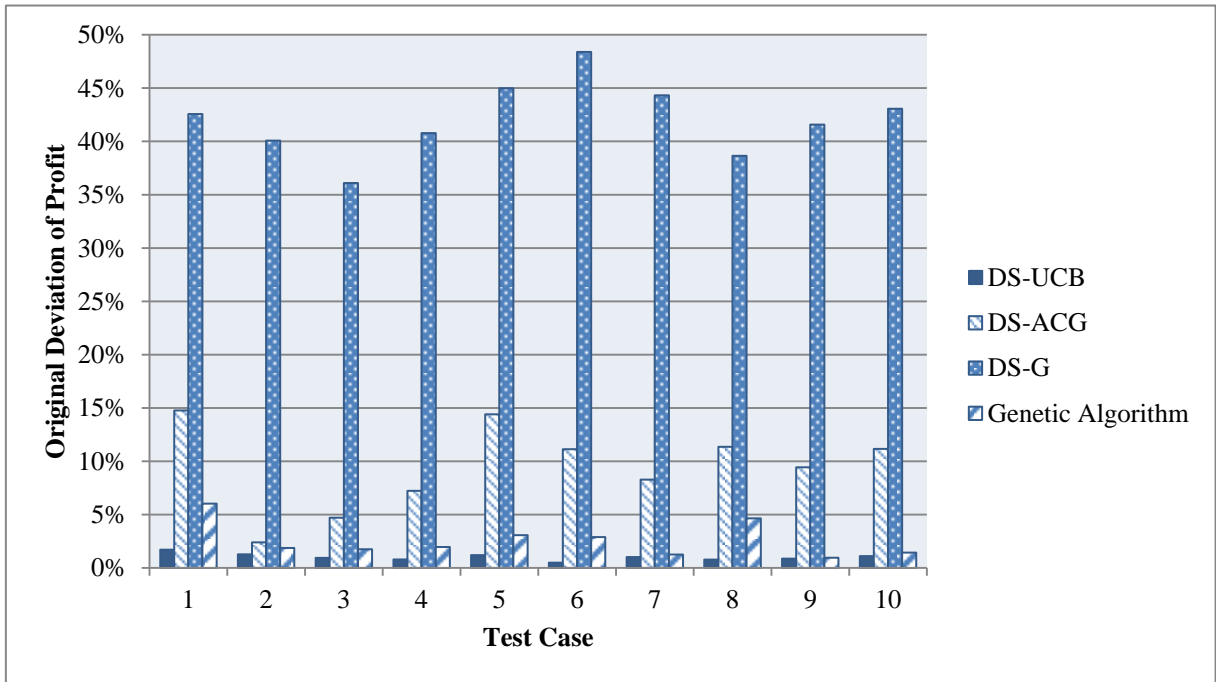


圖 5.17、實驗一：Original Deviation of Profit (10 nodes)

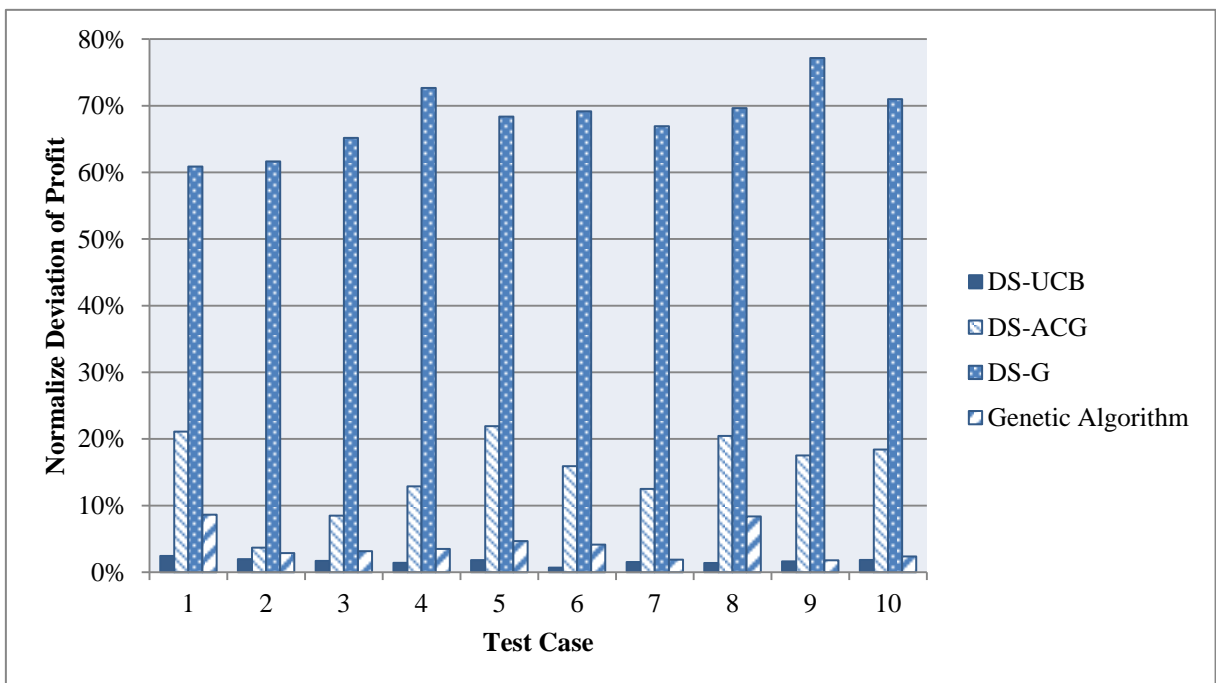


圖 5.18、實驗一：Normalize Deviation of Profit (10 nodes)

表 5.8、實驗一：總救災時間實驗結果與誤差比較 (10 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	17.8	22.5	32.3	87.2	46.0	24.3	26.2%	81.3%	388.8%	36.4%	16.6%	51.5%	246.2%	23.1%
2	18.7	22.2	24.5	87.8	47.0	23.7	18.8%	31.3%	370.5%	26.8%	12.4%	20.6%	244.1%	17.6%
3	18.0	21.0	25.5	81.7	47.3	25.0	16.7%	41.7%	353.7%	38.9%	10.2%	25.6%	217.0%	23.9%
4	17.7	20.2	24.8	84.0	46.3	24.2	14.2%	40.6%	375.5%	36.8%	8.7%	25.0%	231.4%	22.7%
5	18.5	20.3	30.7	85.2	46.3	23.3	9.9%	65.8%	360.4%	26.1%	6.6%	43.7%	239.5%	17.4%
6	16.3	19.7	28.5	93.8	45.3	25.3	20.4%	74.5%	474.5%	55.1%	11.5%	42.0%	267.2%	31.0%
7	18.5	21.3	25.2	80.8	47.0	23.2	15.3%	36.0%	336.9%	25.2%	9.9%	23.4%	218.7%	16.4%
8	18.3	19.8	24.0	72.3	46.7	24.0	8.2%	30.9%	294.5%	30.9%	5.3%	20.0%	190.6%	20.0%
9	16.5	20.2	26.7	74.7	47.7	23.3	22.2%	61.6%	352.5%	41.4%	11.8%	32.6%	186.6%	21.9%
10	18.7	18.0	28.0	82.7	47.0	24.2	3.6%	50.0%	342.9%	29.5%	2.4%	32.9%	225.9%	19.4%
Average	17.9	20.5	27.0	83.0	46.7	24.1	15.5%	51.4%	365.0%	34.4%	9.5%	31.7%	226.7%	21.4%

Opt.=Optimal Solution. GA=Genetic Algorithm.

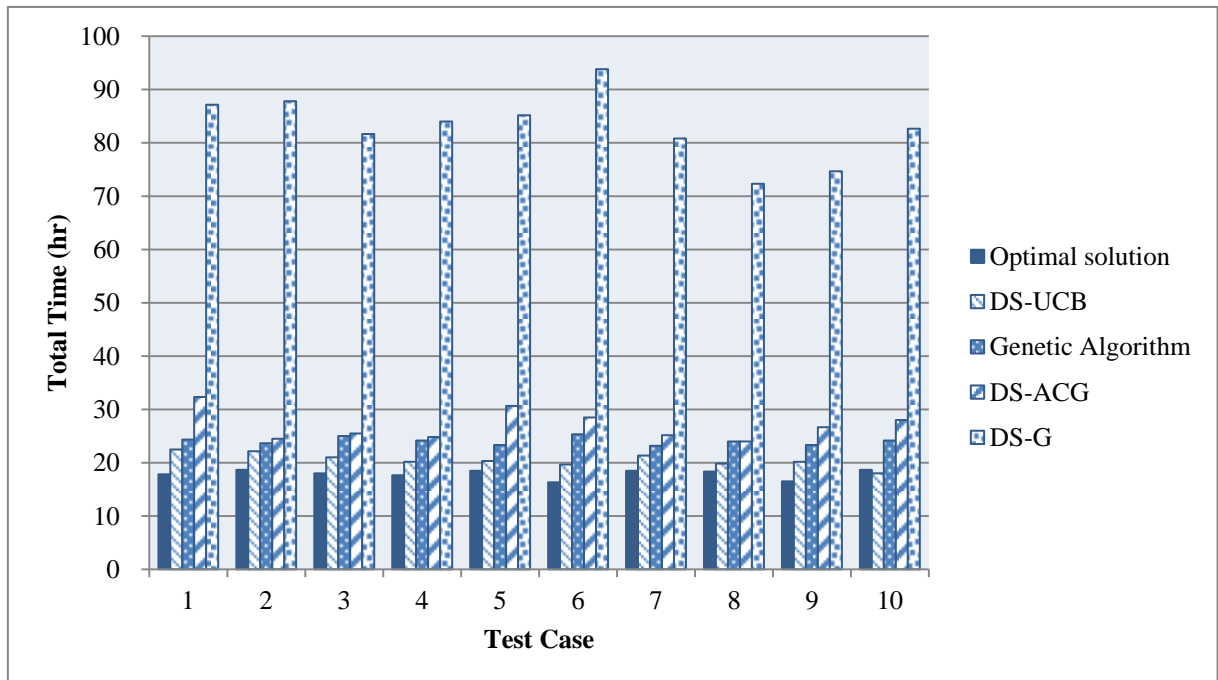


圖 5.19、實驗一：Total Time (10 nodes)

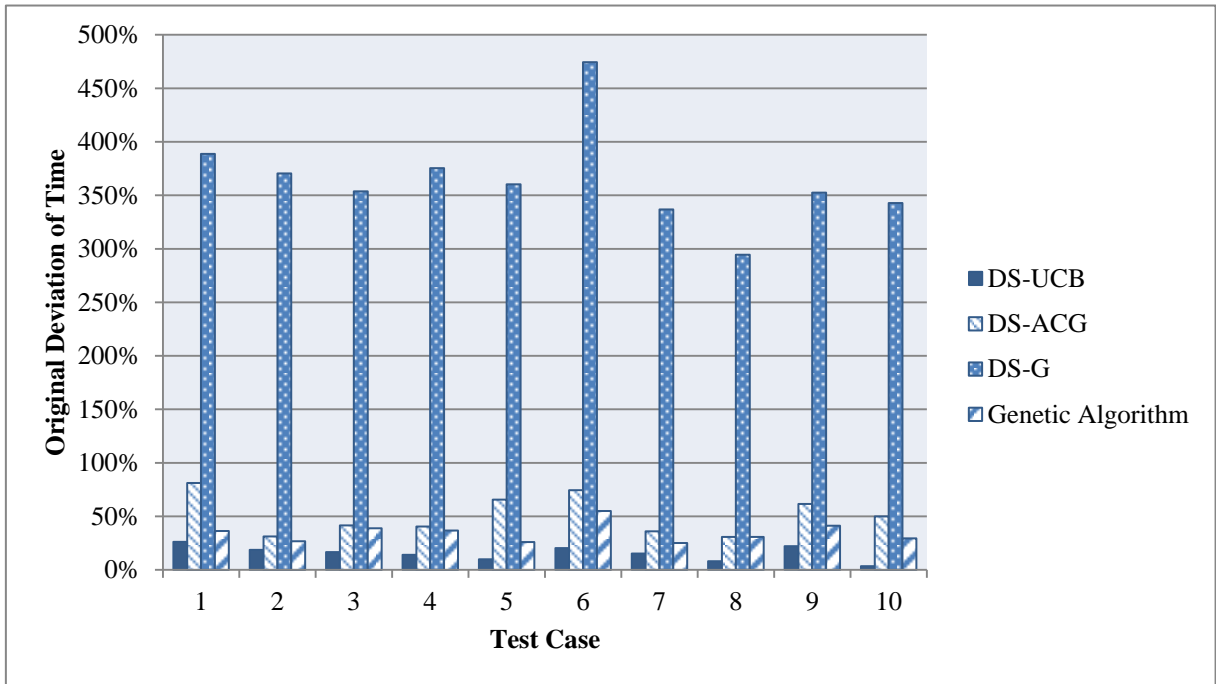


圖 5.20、實驗一：Original Deviation of Time (10 nodes)

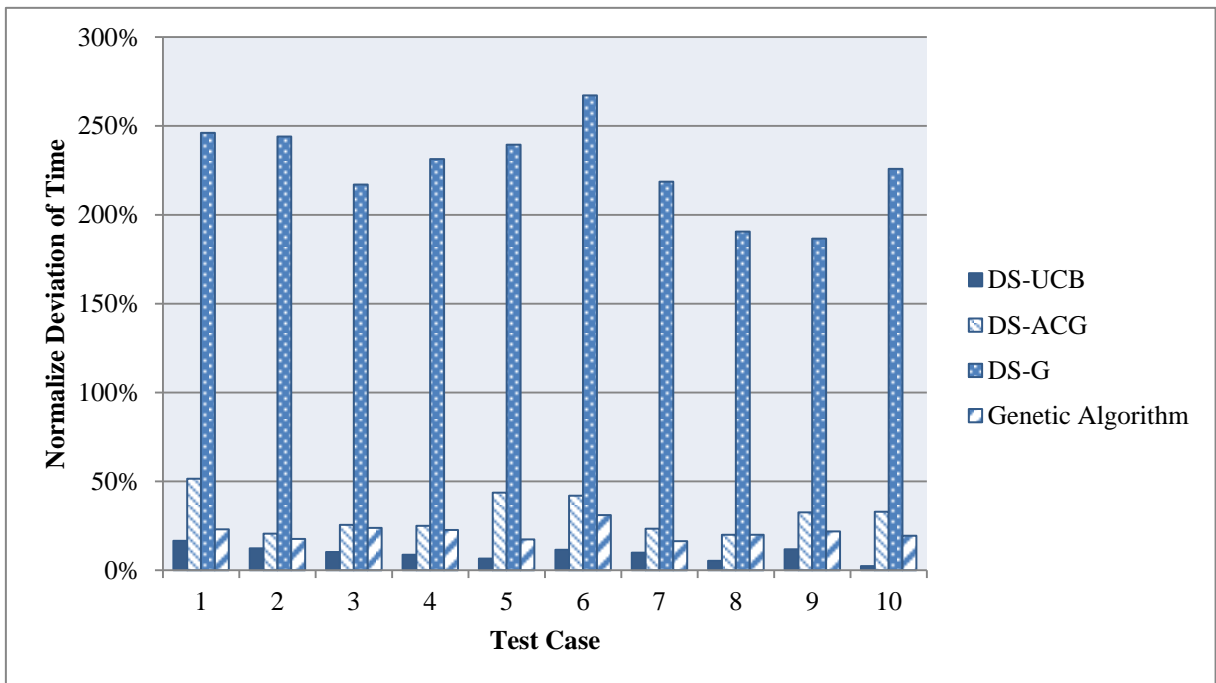


圖 5.21、實驗一：Normalize Deviation of Time (10 nodes)

表 5.9、實驗一：總救災效益實驗結果與誤差比較 (11 個基地台)

Case	Total Profit						Original Deviation of Profit				Normalize Deviation of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	616.8	614.1	545.3	321.3	161.5	589.2	0.4%	11.6%	47.9%	4.5%	0.6%	15.7%	64.9%	6.1%
2	669.0	666.2	626.2	326.2	239.6	645.8	0.4%	6.4%	51.2%	3.5%	0.7%	10.0%	79.8%	5.4%
3	699.4	688.3	649.5	318.0	322.6	669.5	1.6%	7.1%	54.5%	4.3%	2.9%	13.2%	101.2%	7.9%
4	649.2	637.9	570.3	308.5	137.0	620.9	1.7%	12.1%	52.5%	4.4%	2.2%	15.4%	66.5%	5.5%
5	705.2	696.7	650.5	367.8	210.5	696.2	1.2%	7.8%	47.8%	1.3%	1.7%	11.1%	68.2%	1.8%
6	615.6	610.1	565.8	300.5	223.5	591.3	0.9%	8.1%	51.2%	3.9%	1.4%	12.7%	80.4%	6.2%
7	689.0	682.8	650.8	308.0	247.1	665.8	0.9%	5.5%	55.3%	3.4%	1.4%	8.6%	86.2%	5.3%
8	751.3	743.7	652.5	383.2	224.1	733.1	1.0%	13.2%	49.0%	2.4%	1.4%	18.7%	69.8%	3.5%
9	729.0	723.4	681.3	333.5	206.4	710.9	0.8%	6.5%	54.2%	2.5%	1.1%	9.1%	75.7%	3.5%
10	766.4	758.5	730.8	319.7	212.6	745.2	1.0%	4.6%	58.3%	2.8%	1.4%	6.4%	80.7%	3.8%
Average	689.1	682.2	632.3	328.7	218.5	666.8	1.0%	8.3%	52.2%	3.3%	1.5%	12.1%	77.3%	4.9%

Opt.=Optimal Solution. GA=Genetic Algorithm.

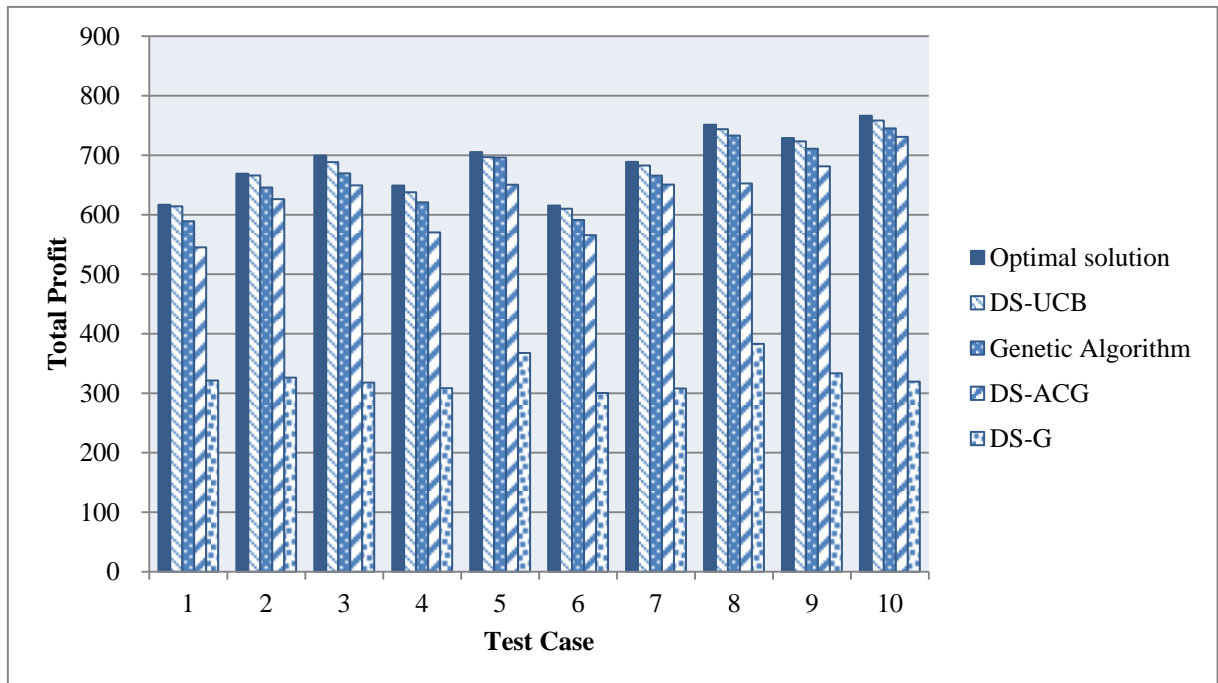


圖 5.22、實驗一：Total Profit (11 nodes)

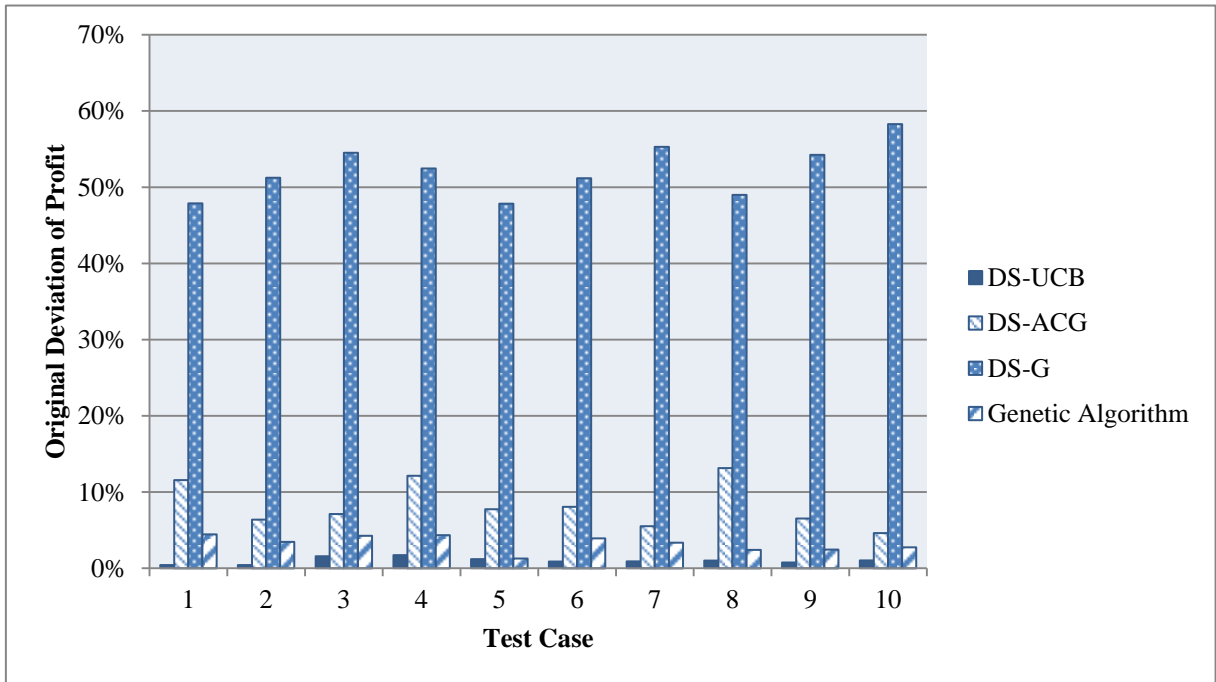


圖 5.23、實驗一：Original Deviation of Profit (11 nodes)

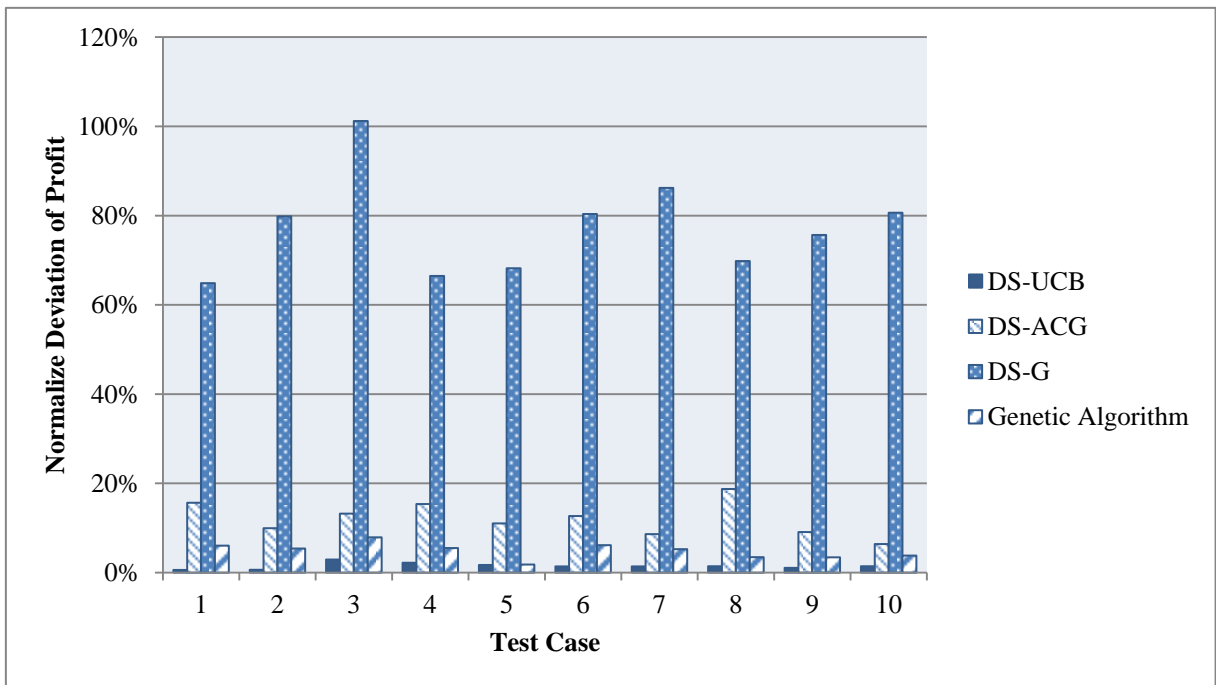


圖 5.24、實驗一：Normalize Deviation of Profit (11 nodes)

表 5.10、實驗一：總救災時間實驗結果與誤差比較 (11 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	23.0	23.7	28.0	94.3	54.7	24.2	2.9%	21.7%	310.1%	5.1%	2.1%	15.8%	225.3%	3.7%
2	18.5	19.5	25.0	107.7	54.3	27.2	5.4%	35.1%	482.0%	46.8%	2.8%	18.1%	248.8%	24.2%
3	21.2	26.2	24.2	119.2	54.3	25.2	23.6%	14.2%	463.0%	18.9%	15.1%	9.0%	295.5%	12.1%
4	17.5	23.3	26.5	104.2	54.3	24.2	33.3%	51.4%	495.2%	38.1%	15.8%	24.4%	235.3%	18.1%
5	21.2	23.3	24.3	100.5	54.0	24.5	10.2%	15.0%	374.8%	15.7%	6.6%	9.6%	241.6%	10.2%
6	20.0	23.7	27.0	121.3	54.0	25.7	18.3%	35.0%	506.7%	28.3%	10.8%	20.6%	298.0%	16.7%
7	19.5	21.8	22.8	116.3	54.0	27.0	12.0%	17.1%	496.6%	38.5%	6.8%	9.7%	280.7%	21.7%
8	18.0	22.3	27.7	101.2	54.7	24.7	24.1%	53.7%	462.0%	37.0%	11.8%	26.4%	226.8%	18.2%
9	20.8	20.3	25.8	115.5	54.3	24.8	2.4%	24.0%	454.4%	19.2%	1.5%	14.9%	282.6%	11.9%
10	19.0	22.0	23.0	110.8	54.3	25.0	15.8%	21.1%	483.3%	31.6%	8.5%	11.3%	259.9%	17.0%
Average	19.9	22.6	25.4	109.1	54.3	25.2	14.8%	28.8%	452.8%	27.0%	8.2%	16.0%	259.5%	15.6%

Opt.=Optimal Solution. GA=Genetic Algorithm.

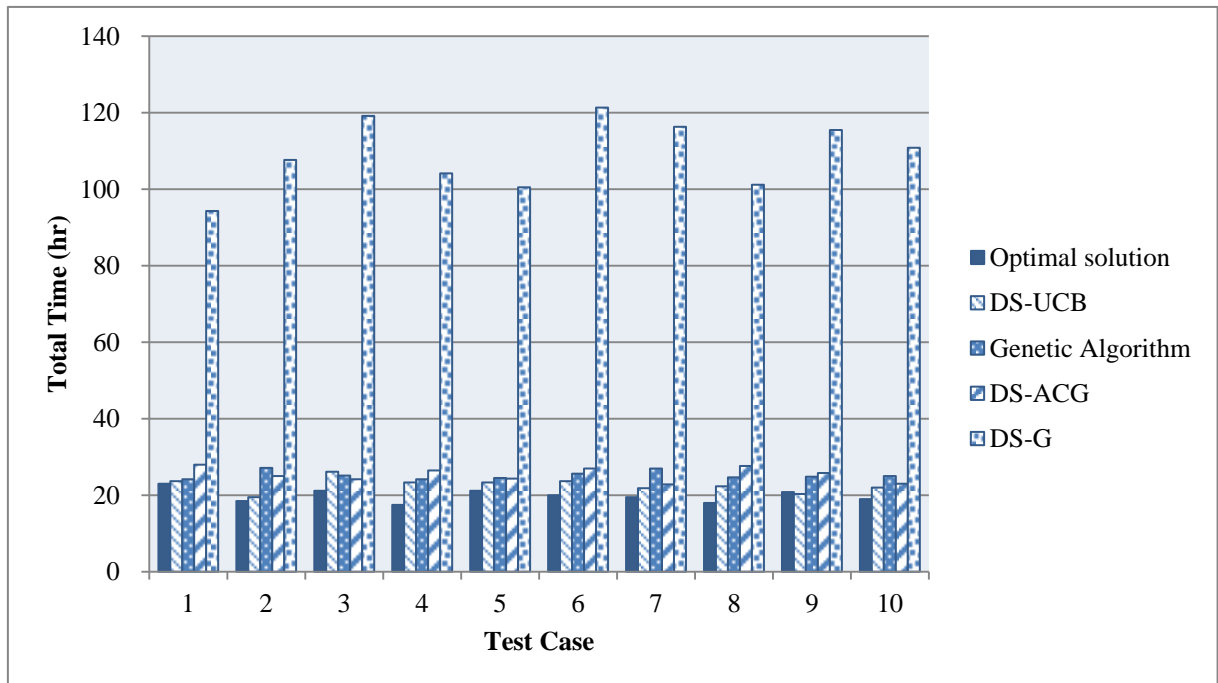


圖 5.25、實驗一：Total Time (11 nodes)

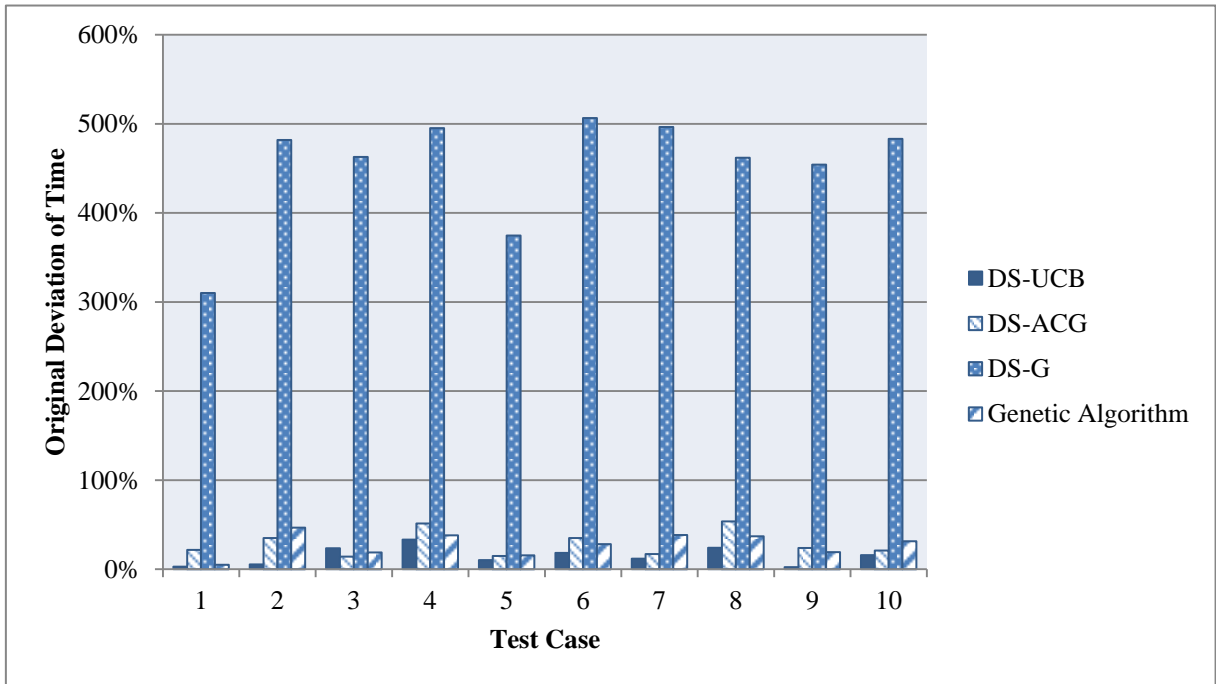


圖 5.26、實驗一：Original Deviation of Time (11 nodes)

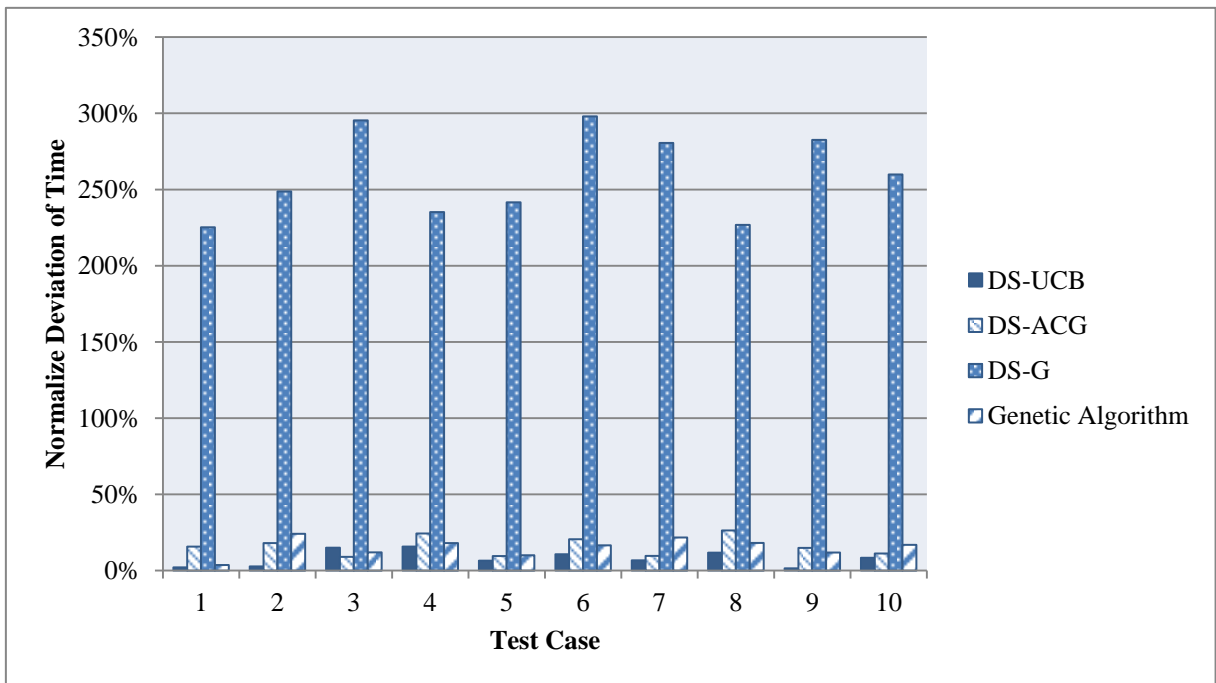


圖 5.27、實驗一：Normalize Deviation of Time (11 nodes)

表 5.11、實驗一：總救災效益實驗結果與誤差比較 (12 個基地台)

Case	Total Profit						Original Deviation of Profit				Normalize Deviation of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	696.7	695.0	642.2	304.2	218.3	675.5	0.2%	7.8%	56.3%	3.0%	0.3%	11.4%	82.1%	4.4%
2	719.1	716.1	687.5	375.0	240.3	706.6	0.4%	4.4%	47.8%	1.7%	0.6%	6.6%	71.9%	2.6%
3	791.3	783.8	681.3	547.8	276.7	773.5	1.0%	13.9%	30.8%	2.3%	1.5%	21.4%	47.3%	3.5%
4	713.6	703.9	637.0	389.7	222.9	687.3	1.4%	10.7%	45.4%	3.7%	2.0%	15.6%	66.0%	5.4%
5	756.3	749.9	635.0	465.8	194.2	735.5	0.8%	16.0%	38.4%	2.8%	1.1%	21.6%	51.7%	3.7%
6	794.6	779.7	711.3	393.3	285.0	766.9	1.9%	10.5%	50.5%	3.5%	2.9%	16.3%	78.7%	5.4%
7	702.9	697.4	627.5	381.7	229.6	667.6	0.8%	10.7%	45.7%	5.0%	1.2%	15.9%	67.9%	7.5%
8	758.1	752.7	668.3	405.5	239.7	733.2	0.7%	11.8%	46.5%	3.3%	1.0%	17.3%	68.0%	4.8%
9	751.1	720.8	714.8	508.8	208.9	713.0	4.0%	4.8%	32.3%	5.1%	5.6%	6.7%	44.7%	7.0%
10	692.4	687.2	617.3	376.5	145.0	671.7	0.8%	10.8%	45.6%	3.0%	1.0%	13.7%	57.7%	3.8%
Average	737.6	728.6	662.2	414.8	226.1	713.1	1.2%	10.2%	43.9%	3.3%	1.7%	14.7%	63.6%	4.8%

Opt.=Optimal Solution. GA=Genetic Algorithm.

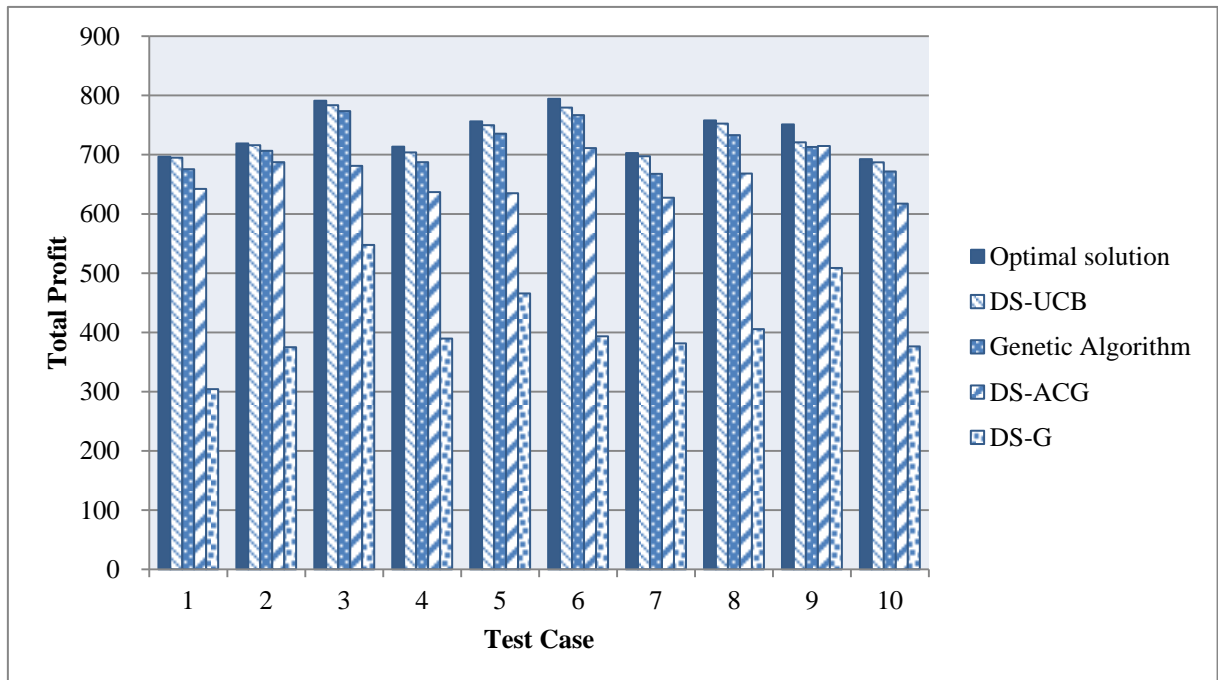


圖 5.28、實驗一：Total Profit (12 nodes)

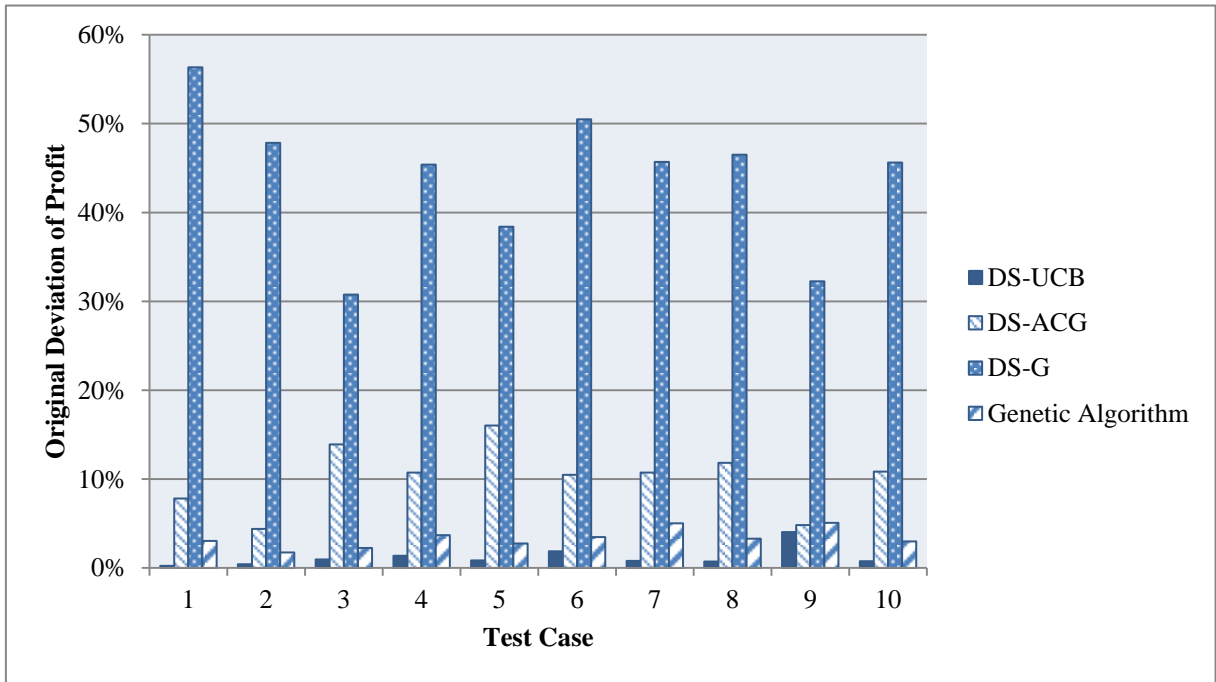


圖 5.29、實驗一：Original Deviation of Profit (12 nodes)

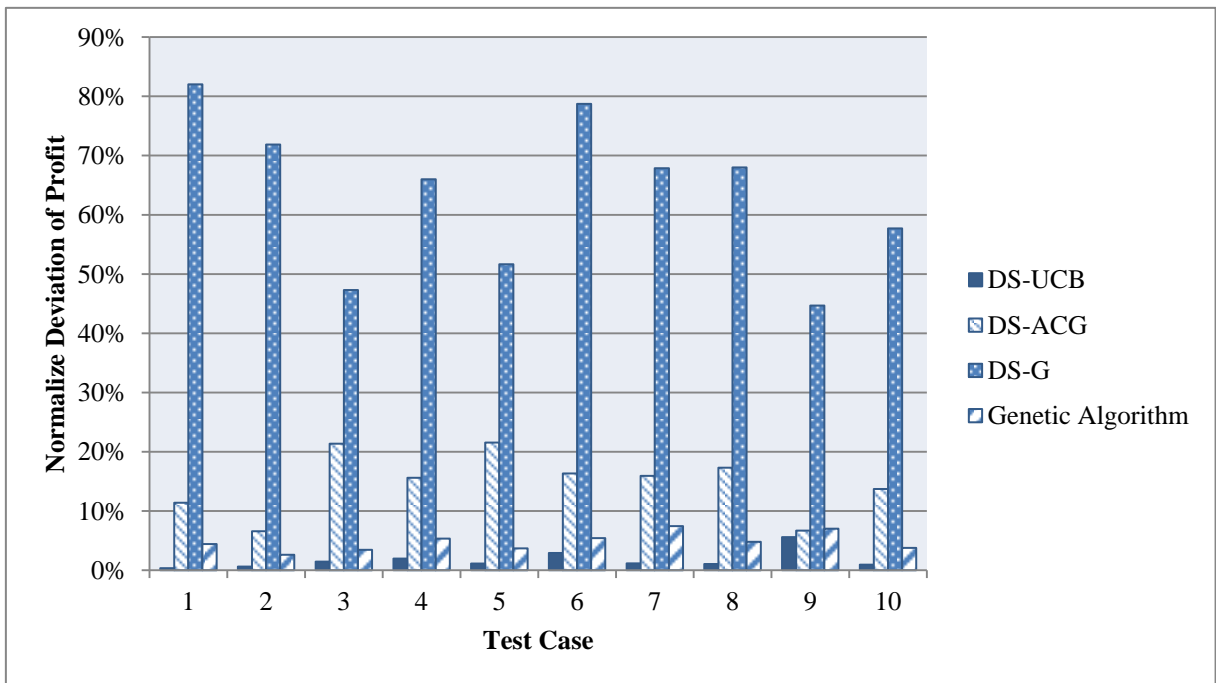


圖 5.30、實驗一：Normalize Deviation of Profit (12 nodes)

表 5.12、實驗一：總救災時間實驗結果與誤差比較 (12 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	18.8	20.0	25.8	113.2	60.3	24.3	6.2%	37.2%	500.9%	29.2%	2.8%	16.9%	227.3%	13.3%
2	19.3	22.0	23.5	107.0	60.0	23.8	13.8%	21.6%	453.4%	23.3%	6.6%	10.2%	215.6%	11.1%
3	22.3	23.8	29.7	67.2	60.7	23.5	6.7%	32.8%	200.7%	5.2%	3.9%	19.1%	117.0%	3.0%
4	22.5	24.8	31.8	97.8	61.0	24.7	10.4%	41.5%	334.8%	9.6%	6.1%	24.2%	195.7%	5.6%
5	26.3	26.7	33.8	78.7	60.3	24.8	1.3%	28.5%	198.7%	5.7%	1.0%	22.1%	153.9%	4.4%
6	19.7	22.3	29.7	106.7	60.3	22.7	13.6%	50.8%	442.4%	15.3%	6.6%	24.6%	213.9%	7.4%
7	19.3	19.3	32.0	96.8	59.7	25.0	0.0%	65.5%	400.9%	29.3%	0.0%	31.4%	192.1%	14.0%
8	19.3	20.2	30.7	87.7	60.0	24.2	4.3%	58.6%	353.4%	25.0%	2.0%	27.9%	168.0%	11.9%
9	19.8	20.8	26.3	79.5	60.7	21.8	5.0%	32.8%	300.8%	10.1%	2.4%	15.9%	146.1%	4.9%
10	19.8	22.2	28.7	101.3	59.0	22.8	11.8%	44.5%	410.9%	15.1%	6.0%	22.6%	208.1%	7.7%
Average	20.7	22.2	29.2	93.6	60.2	23.8	7.3%	41.4%	359.7%	14.6%	3.8%	21.5%	183.8%	7.7%

Opt.=Optimal Solution. GA=Genetic Algorithm.

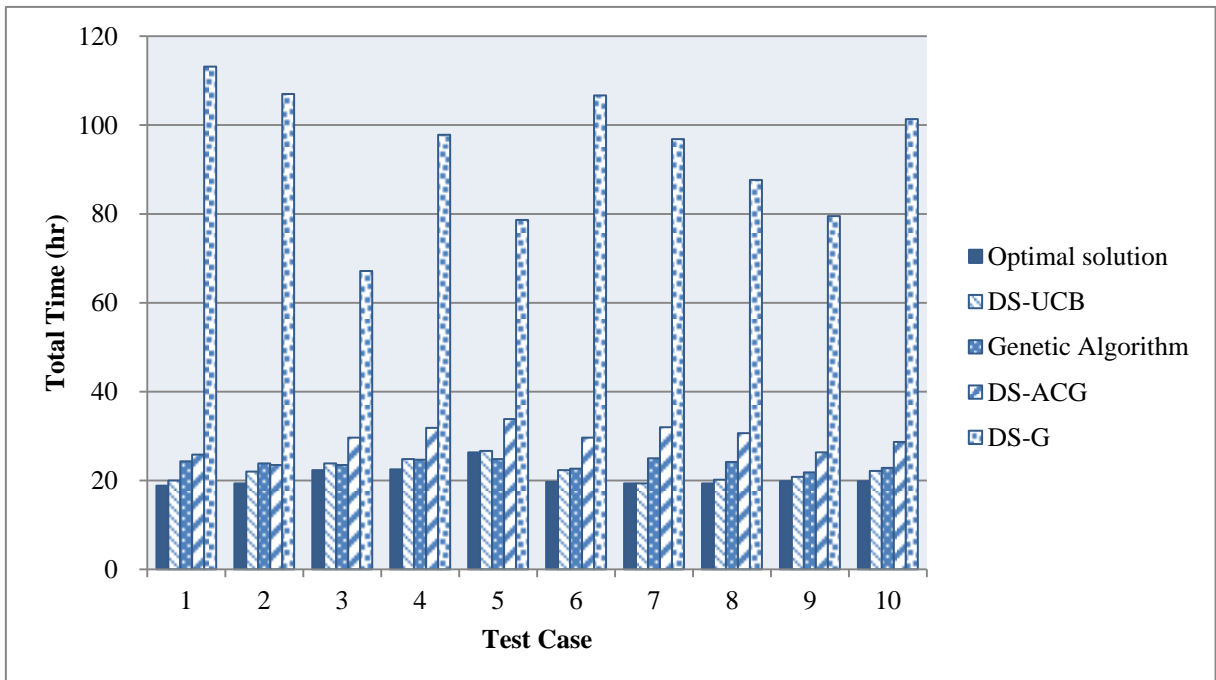


圖 5.31、實驗一：Total Time (12 nodes)

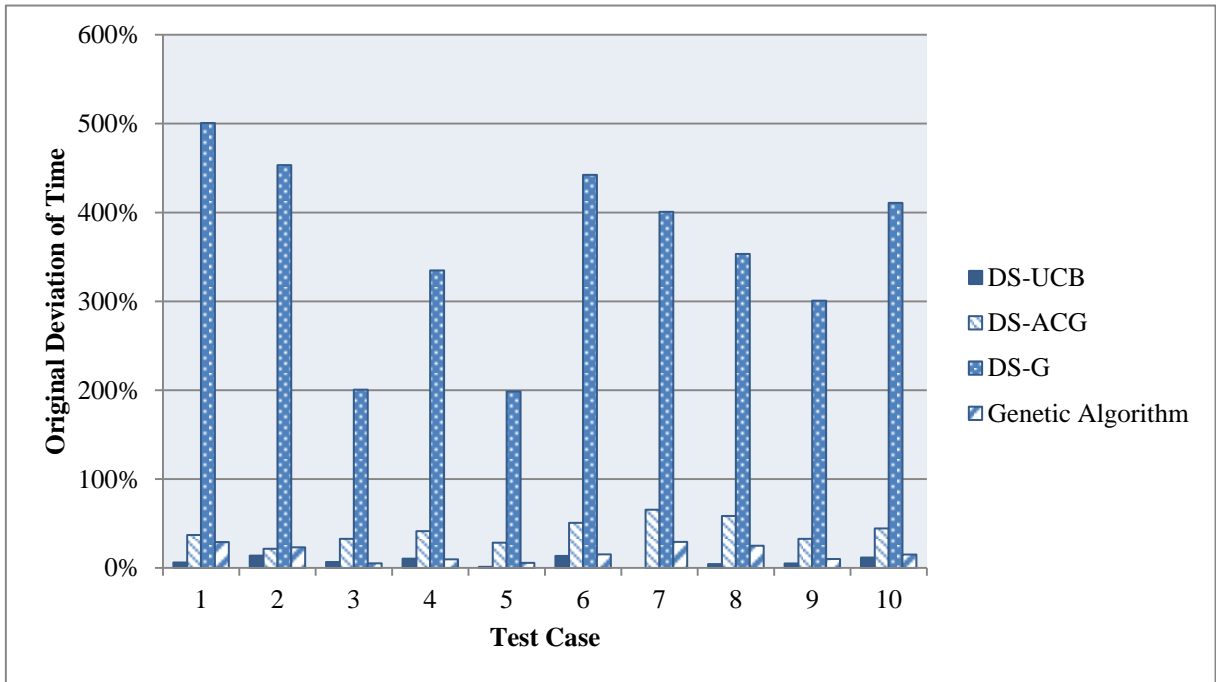


圖 5.32、實驗一：Original Deviation of Time (12 nodes)

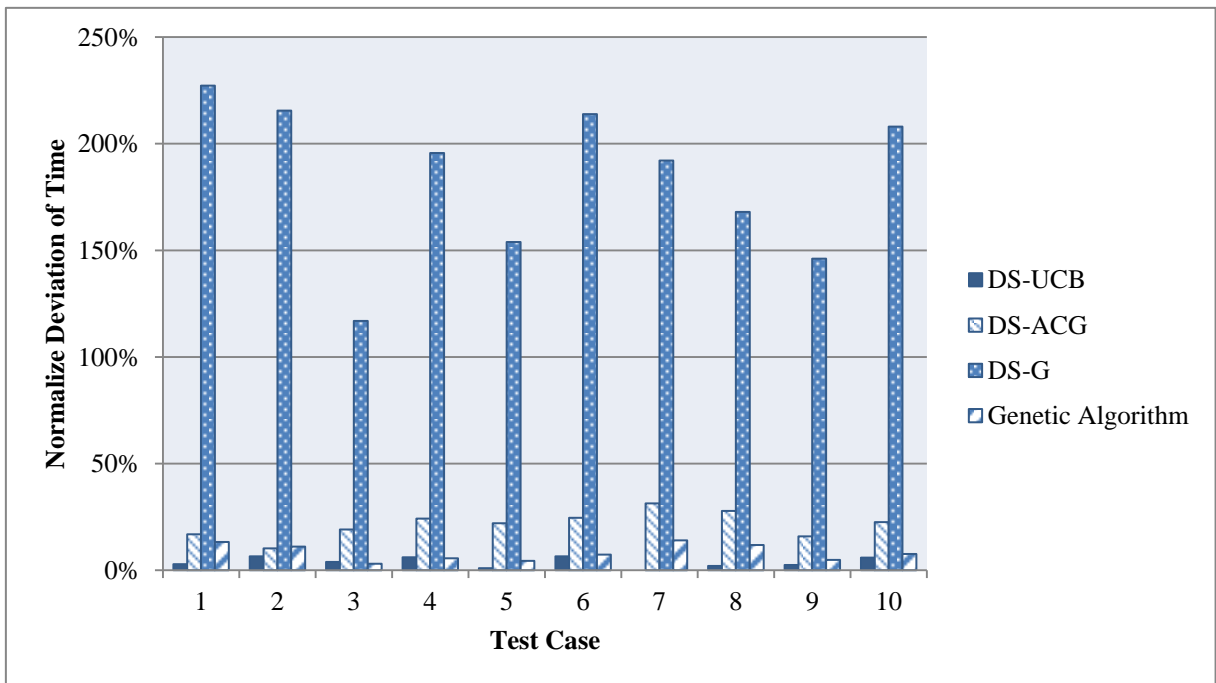


圖 5.33、實驗一：Normalize Deviation of Time (12 nodes)

5.4、實驗二：大規模實驗

5.4.1、實驗參數

表 5.13、實驗二：大規模實驗參數設定

Parameters	Range of values
Initial Profit	Uniform(30, 120)
Traveling Time	Uniform (0.5, 10) hr
Forwarding Tree Size	50
Slope S_1	Uniform (-1, 0)
Slope S_2	Uniform (-1, 0)
Turning Point	Uniform (50, 100) hr
Genetic Algorithm(K, X)	(K, X)=(8, 100000)

5.4.2、評估指標

實驗二中我們將進行大規模案例之實驗，但由於基地台個數過多，最佳解無法在短時間內計算出結果，因次改而採用 1000 萬個解中的最佳解與最差解，作為準最佳解(pseudo optimal solution)與準最差解(pseudo worst solution)，來評估 DS-UCB、DS-ACG、DS-G 及 Genetic Algorithm 在大規模實驗下之總救災效益與總救災時間。Genetic Algorithm 將產生 8 組的建構排程並且演化 10 萬次進行比較。實驗結果以以下評估指標進行評估：

- Total Profit：演算法修復節點之當下獲得效益之加總。
- Total Time：修復完成整個 CCN 拓樸後，所花費的總建構排程時間。
 - Original Deviation (Error)：與最佳解比較總效益之誤差，如公式 4 所示。

$$\text{Original Deviation (Error)} = 1 - \frac{\text{Algorithm solution}}{\text{Pseudo optimal solution}} \dots\dots\dots(4)$$

- Normalize Deviation (Error)：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 5 所示。

$$\text{Normalize Deviation (Error)} = \frac{\text{Pseudo optimal solution} - \text{Algorithm solution}}{\text{Pseudo optimal solution} - \text{Pseudo worst solution}} \dots\dots(5)$$

5.4.3、實驗結果

大規模實驗中，由表 5.14 及圖 5.34 可看出 DS-UCB 的總救災效益明顯優於其他演算法，DS-ACG 與 DS-G 因為需建構的基地台變多而又受到本身的 model 的限制，使得兩者無法發揮其功用，總救災效益表現不佳，而 Genetic Algorithm 的總救災效益雖在小規模實驗中緊次於 DS-UCB，但大規模實驗中，Genetic Algorithm 可能因規模過大表現不佳，使得基因演算可能需演化更多世代才能有機會突破；DS-UCB 不因為規模變大，反發揮其演算法由鄰近節點建構排程，並且由 Backtracking 加強其演算法建構排程之功能，計算出樹狀拓樸的建置排程，獲得更好的救災效益。將由圖 5.35 與圖 5.36 可看出 DS-UCB 的 Original Deviation (Error) of Profit 與 Normalize Deviation (Error) of Profit 皆優於 DS-ACG、DS-G 與 Genetic Algorithm，顯現出 DS-UCB 演算法的強大優勢。根據表 5.15、圖 5.37 和圖 5.38，總救災時間的表現，DS-UCB 能更節省更多的時間。

表 5.14、實驗二：總救災效益實驗結果與誤差比較 (50 個基地台)

Case	Total Profit						Original Deviation (Error) of Profit				Normalize Deviation (Error) of Profit			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	1935.6	2599.4	1322.8	303.0	43.4	795.6	-34.3%	31.7%	84.3%	58.9%	-35.1%	32.4%	86.3%	60.2%
2	2030.6	2493.6	1576.7	400.8	42.8	764.1	-22.8%	22.4%	80.3%	62.4%	-23.3%	22.8%	82.0%	63.7%
3	2021.1	2250.4	1571.8	293.3	7.1	916.0	-11.3%	22.2%	85.5%	54.7%	-11.4%	22.3%	85.8%	54.9%
4	2179.0	2290.3	1747.7	286.7	17.1	895.5	-5.1%	19.8%	86.8%	58.9%	-5.1%	20.0%	87.5%	59.4%
5	2071.8	2386.1	1623.2	354.3	38.6	704.2	-15.2%	21.7%	82.9%	66.0%	-15.5%	22.1%	84.5%	67.3%
6	1924.3	2108.1	1586.2	314.5	26.8	1145.8	-9.6%	17.6%	83.7%	40.5%	-9.7%	17.8%	84.8%	41.0%
7	2057.7	2203.0	1451.5	308.3	37.7	1096.4	-7.1%	29.5%	85.0%	46.7%	-7.2%	30.0%	86.6%	47.6%
8	1719.9	2018.4	1296.5	220.8	8.7	1216.6	-17.4%	24.6%	87.2%	29.3%	-17.4%	24.7%	87.6%	29.4%
9	2092.6	2734.7	1837.2	516.2	18.9	762.0	-30.7%	12.2%	75.3%	63.6%	-31.0%	12.3%	76.0%	64.2%
10	2099.9	2378.3	1836.5	389.0	34.2	1089.7	-13.3%	12.5%	81.5%	48.1%	-13.5%	12.8%	82.8%	48.9%
Average	2013.2	2346.2	1585.0	338.7	27.5	938.6	-16.7%	21.4%	83.2%	52.9%	-16.9%	21.7%	84.4%	53.7%

Opt.=Optimal Solution. GA=Genetic Algorithm.

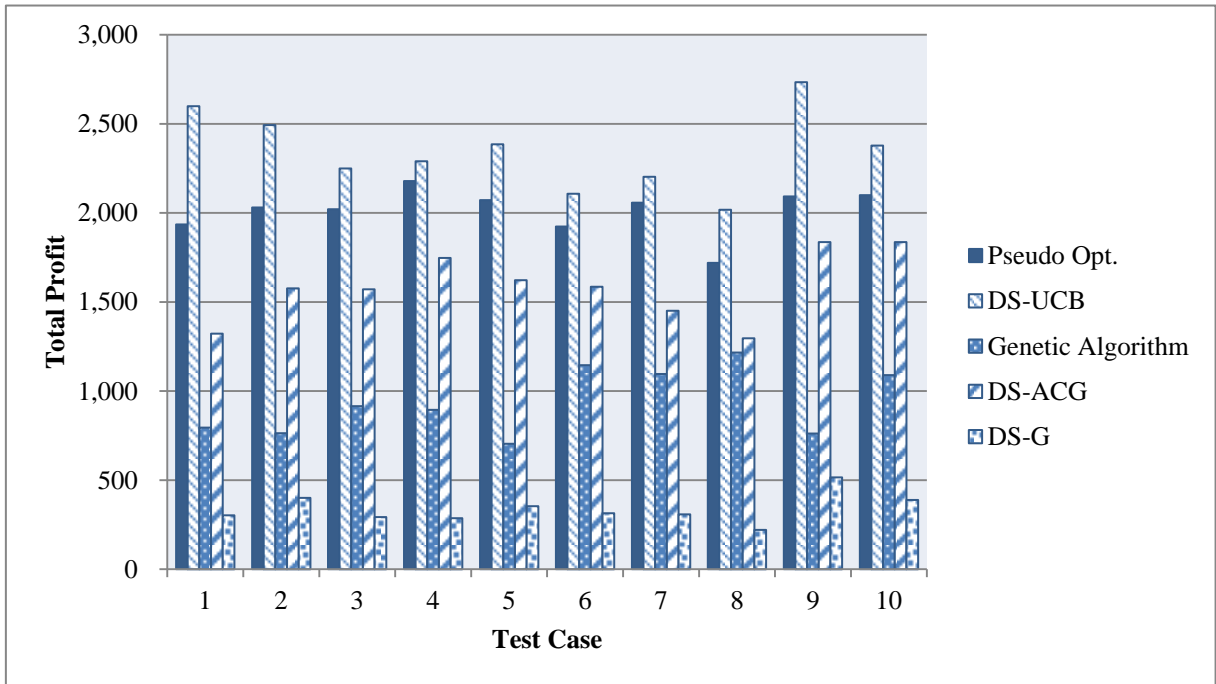


圖 5.34、實驗二：Total Profit (50 nodes)

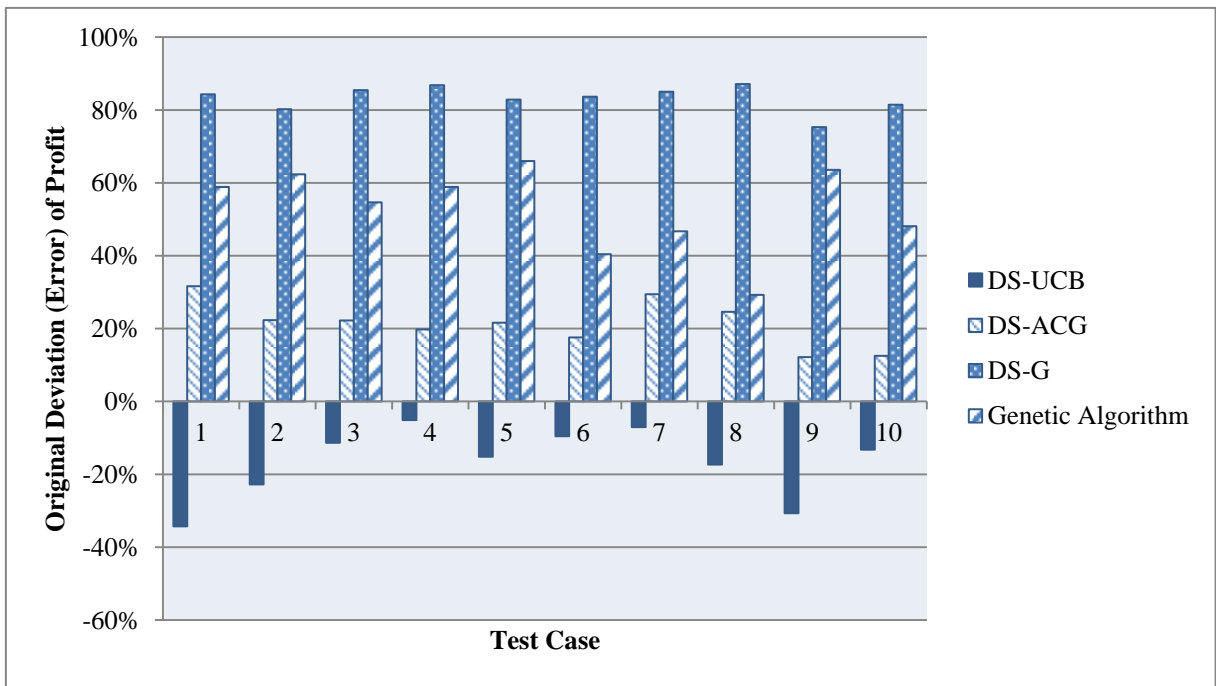


圖 5.35、實驗二：Original Deviation (Error) of Profit (50 nodes)

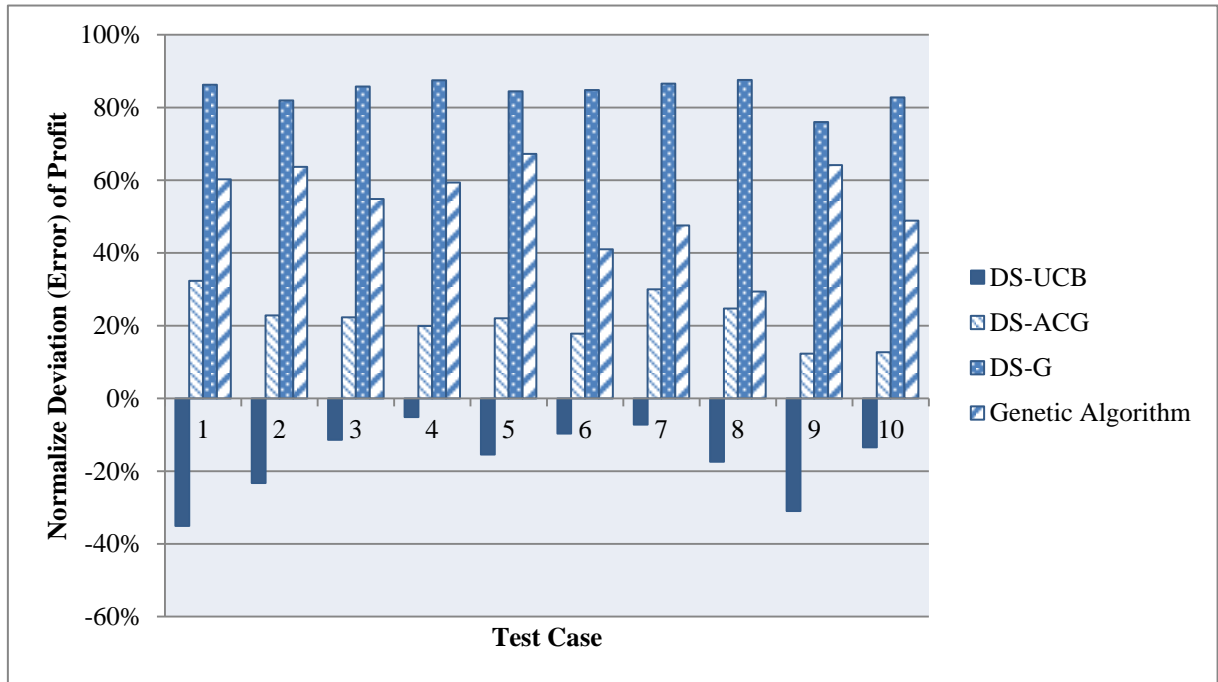


圖 5.36、實驗二：Normalize Deviation (Error) of Profit (50 nodes)

表 5.15、實驗二：總救災時間實驗結果與誤差比較 (50 個基地台)

Case	Total Time (hr)						Original Deviation of Time				Normalize Deviation of Time			
	Opt.	DS-UCB	DS-ACG	DS-G	Worst	GA	DS-UCB	DS-ACG	DS-G	GA	DS-UCB	DS-ACG	DS-G	GA
1	155.3	108.3	167.2	651.3	202.5	285.0	30.3%	-7.6%	-319.3%	-83.5%	-99.6%	25.1%	1051.6%	274.9%
2	158.3	128.0	168.3	612.5	203.7	264.3	19.2%	-6.3%	-286.8%	-66.9%	-66.9%	22.1%	1001.8%	233.8%
3	156.8	130.8	163.0	676.5	214.3	268.3	16.6%	-3.9%	-331.3%	-71.1%	-45.2%	10.7%	903.8%	193.9%
4	154.7	128.2	140.7	628.7	195.8	268.7	17.1%	9.1%	-306.5%	-73.7%	-64.4%	-34.0%	1151.4%	276.9%
5	159.0	120.5	145.7	627.2	207.5	264.3	24.2%	8.4%	-294.4%	-66.2%	-79.4%	-27.5%	965.3%	217.2%
6	160.2	132.7	155.7	672.3	213.7	262.3	17.2%	2.8%	-319.8%	-63.8%	-51.4%	-8.4%	957.3%	191.0%
7	156.7	142.2	177.3	622.5	208.8	270.8	9.3%	-13.2%	-297.3%	-72.9%	-27.8%	39.6%	893.0%	218.8%
8	159.2	131.5	178.3	736.7	207.3	261.5	17.4%	-12.0%	-362.8%	-64.3%	-57.4%	39.8%	1199.0%	212.5%
9	155.2	115.8	136.0	399.3	207.5	273.7	25.3%	12.4%	-157.4%	-76.4%	-75.2%	-36.6%	466.6%	226.4%
10	157.7	129.8	152.2	532.8	206.2	255.7	17.7%	3.5%	-237.9%	-62.2%	-57.4%	-11.3%	773.5%	202.1%
Average	157.3	126.8	158.4	616.0	206.7	267.5	19.4%	-0.7%	-291.4%	-70.1%	-62.5%	1.9%	936.3%	224.8%

Opt.=Optimal Solution. GA=Genetic Algorithm.

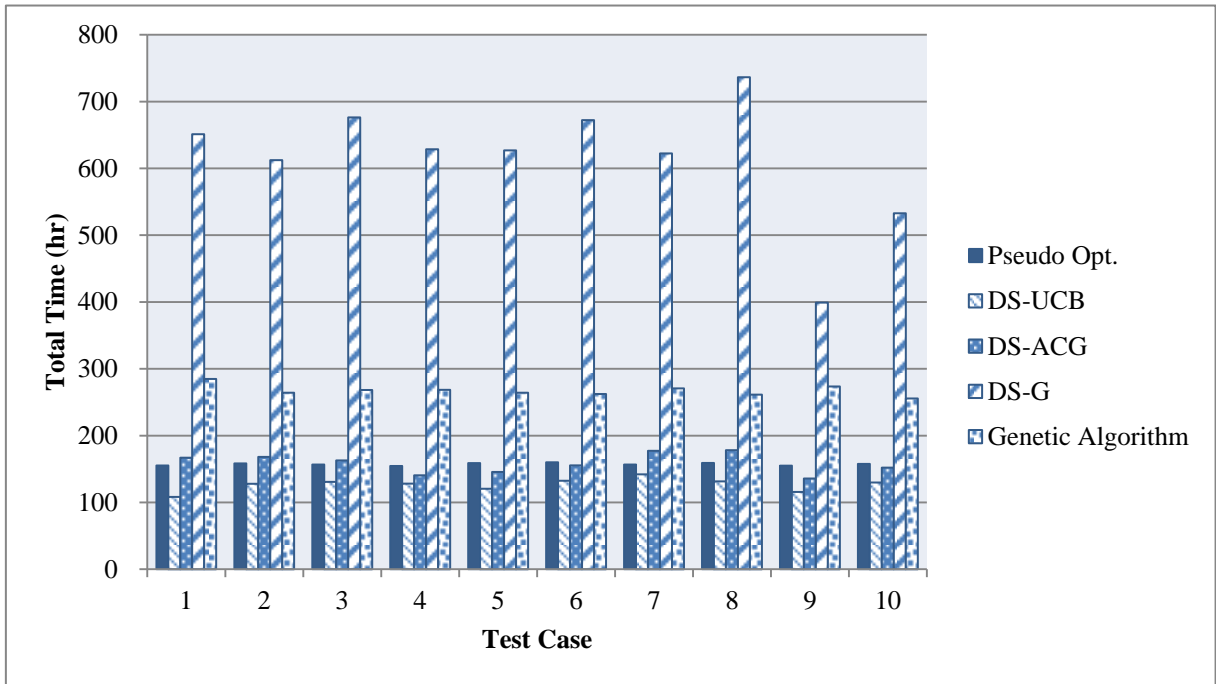


圖 5.37、實驗二：Total Time (50 nodes)

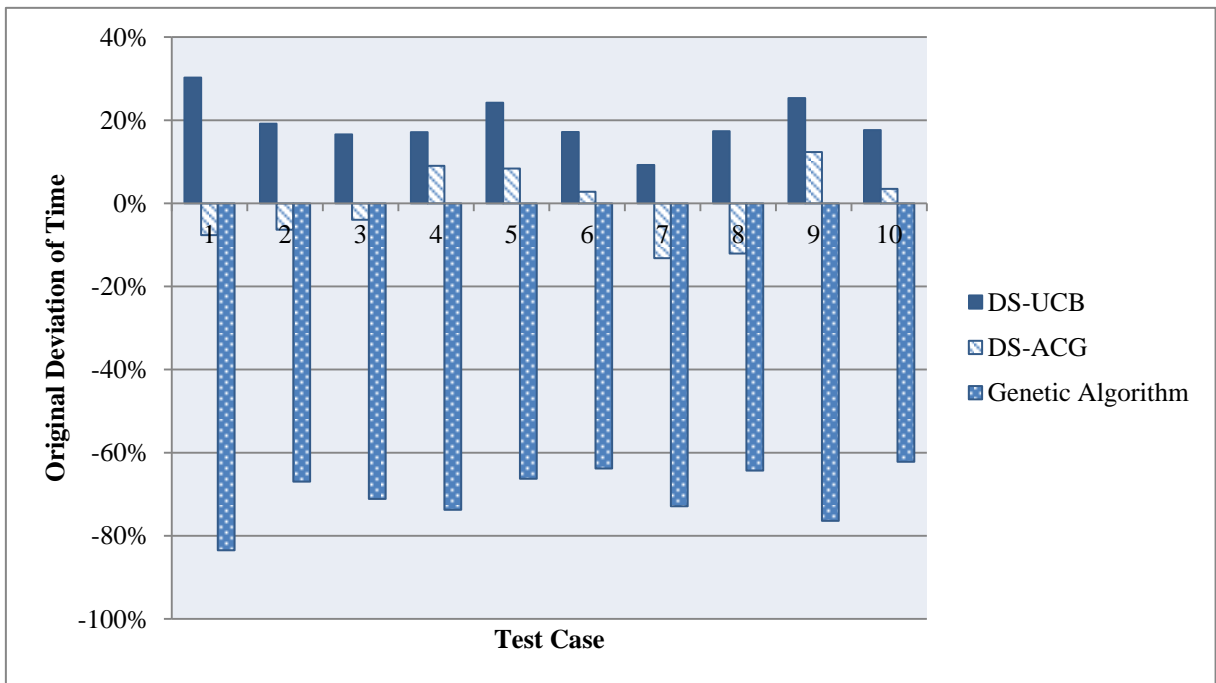


圖 5.38、實驗二：Original Deviation of Time (50 nodes)



圖 5.39、實驗二：Normalize Deviation of Time (50 nodes)

5.5、實驗總結

本論文根據[15]所提出之救災排程並參考較貼近實際狀況之因素提出演算法 DS-ACG 與 DS-UCB。由實驗結果可知，[15]所提出的演算法 DS-G 沒有考慮基地台間的相對距離，使得總救災時間皆高於本論文所提出的 DS-ACG 與 DS-UCB，也因此 DS-G 的總救災效益亦低於 DS-ACG 與 DS-UCB 相當多。

在實驗一的小規模實驗中，DS-UCB 的總救災效益平均 original deviation、平均 normalized deviation 表現皆優於第一個問題模型 CCNDS-AC 所提出的演算法 DS-ACG，也略勝於 Genetic Algorithm，但 DS-UCB 與 Genetic Algorithm 所得的總救災效益差距並不明顯，可見 Genetic Algorithm 運用基因演算的方式解決 CCNDS-UC 問題確實有其優勢，在某些 Test Case 可以發現 DS-UCB 與 DS-ACG 的總救災效益相去不遠，這可能是因為救災總部的的位置較靠近於連網台周圍，抑或是連網台周圍之孤立台災情嚴重，相對的救災效益較高，使得 DS-UCB 的特點無法顯現出來，但若進一步比較總救災時間可以

發現，DS-UCB 較 DS-ACG 節省更多的時間，DS-UCB 的 normalized deviation of profit 平均皆在 2% 左右，這意味著 DS-UCB 秉持著將鄰近的節點一併建構的特性，使得運載交通工具及時間可以更加的節省，且能達到逼近於最佳解的總救災時間與總救災效益。

在實驗二大規模實驗中，Genetic Algorithm 由於需建構排程的基地台數量變多，需基因運算演化的世代也隨之增加，但進行 10 萬次基因演算就需花費將近 1 小時的運算成本，且其總救災效益表現不彰，Genetic Algorithm 的 original deviation (error) of profit 低於 pseudo optimal solution 約 52.9%，因此在大規模實驗中 Genetic Algorithm 無法好好發揮其效能。演算法 DS-UCB 發揮其優勢，鄰近且救災效益高的節點一併建構，利用 Backtracking 進一步檢查周遭節點，使得總救災時間表現不錯，相較於 pseudo optimal solution，DS-UCB 的 original deviation of time 少約 19.4%，總救災效益有很好的表現，DS-UCB 的 original deviation (error) of profit 相較於 pseudo optimal solution 高出約 16.7%，因此 DS-UCB 是目前解決 CCNDS-UC 最好的啟發式演算法。DS-ACG 因受限於父節點優先建構，使其所得之總救災效益較差，DS-ACG 的 original deviation (error) of profit 低於 pseudo optimal solution 約 21.4%，但卻能比 DS-G 更逼近於 pseudo optimal solution。實驗結果可證實本論文所提出的兩大模型 CCNDS-AC 與 CCNDS-UC 其演算法 DS-ACG 與 DS-UCB 對於 CCN 建構排程有相當的改善，而 CCNDS-UC 雖因捨棄了限制條件使得可能的狀況複雜許多，卻使 CCNDS-UC 較 CCNDS-AC 考慮更多可能的排程讓整個建構排程得到最大的救災效益。

由圖 5.40 與表 5.16 可看出，求得最佳解所需花費之時間呈現指數上升的趨勢，甚至到了大型實驗竟然需要將近 3 個小時才得以求得最佳解，Genetic Algorithm 也因為演化次數的增加導致運算時間大為增加，將近 1 小時才得以求初期基因演算解，而其他的演算法均可在短時間內執行完畢。但救災行動受時間所逼迫，建構 CCN 之工作亦相當緊急，在 CCN 規劃階段所花費的時間可以越短越好，才可盡快佈建好 CCN，提供救災通訊，協助救災之進行。

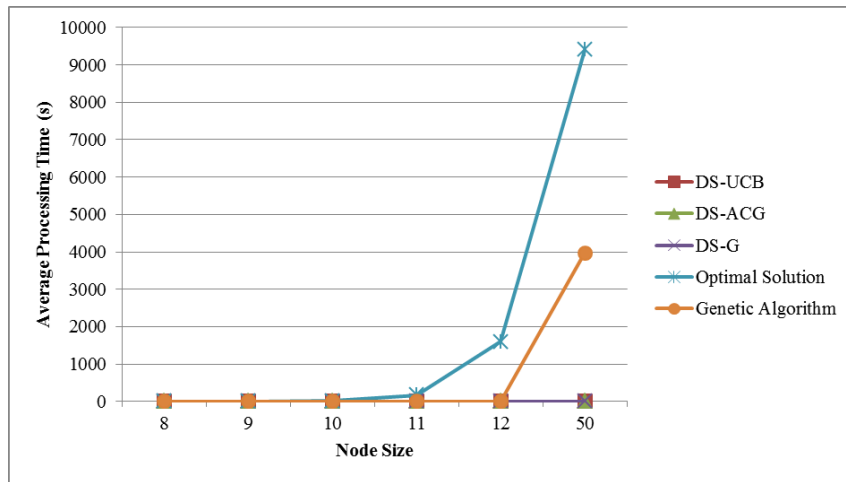


圖 5.40、Average Processing Time

表 5.16、各演算法平均執行時間(單位:秒)

Algorithm Node Size	DS-UCB	DS-ACG	DS-G	Genetic Algorithm	Optimal Solution
8	0.15	0.13	0.14	0.16	0.45
9	0.17	0.15	0.14	0.15	1.96
10	0.14	0.13	0.12	0.14	15.01
11	0.13	0.12	0.12	0.14	167.19
12	0.12	0.10	0.10	0.15	1599.29
50	0.22	0.21	0.36	3972	9402.25

第6章、結論與未來展望方向

當大型自然災害發生時，造成傷亡慘重，通訊系統常會癱瘓無法使用，使得救災的進行受到阻礙。然而有效運作的通訊系統卻是災情傳遞、資源調度及救災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)，由於各地災情狀況不完全相同，CCN 的建構順序必須考慮災區的輕重緩急、時間的急迫等因素依序建構。因此當 CCN 拓樸規劃完成後，根據 CCN 拓樸、各地災情嚴重程度以及拓樸中基地台間的相對距離進行基地台建構排程規劃，以達到最大的總救災效益。

本論文根據[15]進行改良，考慮基地台間的相對距離、時間與基地台救災效益，提出兩個適合 CCN 拓樸樹狀結構的考慮資源運輸路徑之最佳化排程模型，進而求取總救災效益的最大化。這些模型可供救災指揮單位系統化的解決 CCN 建構排程問題。各基地台建構完成可獲的之救災效益由救災單位根據各災區之災情自行定義，本論文所提出之考慮資源運輸路徑之建構排程模型與建構排程演算法可決定 CCN 拓樸中基地台的建構優先順序。

考慮資源運輸路徑之 CCN 建置排程問題是更為複雜的 Single Machine Scheduling Problem，本論文證明該問題屬於 NP-Hard 問題。但當發生突發性大型天然災害時，時間相當急迫，可容許的計算時間相當短暫，因此我們提出了兩個快速的啟發式演算法 DS-ACG 與 DS-UCB，在短時間內求出一組相當逼近於最佳解的建構排程順序。DS-ACG 演算法根據 CCN 拓樸、基地台建置完成所獲得之救災效益以及基地台間相對距離時間，以父節點優先於子節點建構的考量下，找出最大化總救災效益的建構排程順序；為了改善父節點優先於子節點建構所衍生之運載資源浪費、時間的延宕導致救災效益下降等問題，提出了 DS-UCB 演算法，撤除了父節點優先建構的限制，將鄰近且救災效益高的節

點一併加入建置排程，以節省更多運載資源，降低總救災時間，並且提升總救災效益。結果顯示我們所提出之演算法 DS-ACG 與 DS-UCB 皆優於 DS-G，在小規模實驗中，DS-UCB 可求得與最佳解的總救災效益 Original Deviation 平均約 0.9% 的近似最佳解之建構排程；大規模實驗中 DS-UCB 發揮其優勢，鄰近且救災效益高的節點一併建構，使得 DS-UCB 的 original deviation of time 少於 pseudo optimal solution 約 19.4%，也因此總救災效益有很好的表現，DS-UCB 的 original deviation (error) of profit 高於 pseudo optimal solution 約 16.7%。

本論文所提出之模型尚有改善的空間，例如：多組建構團隊所運行之建構排程順序、運載工具負載 CRP 數量有限之分群問題、考慮各種不同的演算法(基因演算法、旅行家問題相關演算法)進行改良。此外，當 CCN 拓樸有多個連網台時，意即不在只是單純的樹狀結構時，多組建構團隊運行的建構排程之順序抑是一項重要之研究課題。在有限的時間內，盡快地規劃且完成 CCN 拓樸建置，發揮 CCN 對救災之通訊協助，期以搶救更多生命。

參考文獻

- [1] Association of Public-Safety Communications Officials International, Project 25, <http://www.apcointl.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] Yong Bai, Wencai Du, Zhengxin Ma, Chong Shen, Youling Zhou and Baodan Chen, "Emergency communication system by heterogeneous wireless networking," *2011 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 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] Jean-Charles Bazin, Hongdong Li, In So Kweon, Cédric Demonceaux, Pascal Vasseur and Katsushi Ikeuchi, "A Branch and Bound Approach to Correspondence and Grouping Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Tokyo, Japan, Dec. 2012.
- [6] 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.
- [7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms," Third Edition, Cambridge, Mass.: The MIT Press, 2009.
- [8] 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.
- [9] Weimin Dong, et al., Chi-Chi, "Taiwan Earthquake Event Report, Risk Management Solutions," Inc., https://www.rms.com/Publications/Taiwan_Event.pdf, retrieved Mar. 2010.

- [10] 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.
- [11] 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.
- [12] 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.
- [13] M.R. Garey, and M.D. Johnson. "Computers and Intractability: A Guide to the Theory of NP-Completeness," 1979. ISBN 0-7167-1045-5.
- [14] Jyh-Shyan Huang, Yao-Nan Lien and Yu-Chieh Huang, " Network Topology Planning for Contingency Cellular Network," *Proceedings of 17th Mobile Computing Workshop*, Aug. 2012, Taoyuan, Taiwan, NSC 100-2221-E-008.
- [15] Jyh-Shyan Huang, Yan-Song Wang and Yao-Nan Lien, "Deployment Scheduling for Contingency Cellular Network for Disaster Operations," *Proceedings of the 15th Asia-Pacific Network Operations and Management Symposium*, Sep. 2013, Hiroshima, Japan, NSC 101-2420-H-004-006-MY3.
- [16] ITR-RESCUE, Robust Networking and Information Collection Project, <http://www.itr-rescue.org/research/networking.php>, retrieved Feb. 2010.
- [17] 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.
- [18] Hung-Chin Jang, Yao-Nan Lien and Tzu-Chieh Tsai, "Rescue Information System for Earthquake Disasters Based on MANET Emergency Communication Platform", *Proc. of the ACM International Workshop on Advanced Topics in Mobile Computing for Emergency Management: Communication and Computing Platforms (MCEM 2009)*, June, 2009, Leipzig, Germany, pp. 623-627.
- [19] 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.

- [20] J. Kennedy, and R. C. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Network*, vol. 4, Perth, Australia, Nov. 1995, pp. 1942-1948.
- [21] Tae-Ho Lee and Taesang Choi, "Self powered wireless communication platform for disaster relief," *2011 Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Sep. 2011.
- [22] Yao-Nan Lien and Yung-Chuan Wun, "QoS-Aware Packet Scheduling by Looking Ahead Approach", *Proc. of The 19th Workshop on Object-Oriented Technology and Applications*, HuWei, Taiwan, Sep. 2008.
- [23] Yao-Nan Lien, Li-Cheng Chi and Yuh-Sheng Shaw, "A Walkie-Talkie-Like Emergency Communication System for Catastrophic Natural Disasters," *Proc. of ISPAN09*, Kaohsiung, Taiwan, Dec. 2009.
- [24] 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.
- [25] Yao-Nan Lien, Li-Cheng Chi and Chih-Chieh Huang, "A Multi-hop Walkie-Talkie-Like Emergency Communication System for Catastrophic Natural Disasters", *Proceedings of International Conference on Parallel Processing Workshop (on Applications of Wireless Ad Hoc and Sensor Networks)*, San Diego, CA.
- [26] Yao-Nan Lien, Hung-Chin Jang and Tzu-Chieh Tsai, "Design of P2Pnet: An Autonomous P2P Ad-Hoc Group Communication System", *Proceedings of The First International Workshop on Mobile Peer-to-Peer Information Services (MP2PIS)*, May 18-21, 2009, Taipei, Taiwan.
- [27] 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.
- [28] Kelly T. Morrison, AT&T, "Rapidly Recovering from the Catastrophic Loss of a Major Telecommunications Office," *IEEE Communications Magazine*, vol.19, no.1, Jan. 2011. pp. 28-35.
- [29] E. Natalizio, "The practical experience of implementing a GSM BTS through open

software hardware," *2010 International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, Nov. 2010.

- [30] 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. 36-45.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] Zhenhong Shao, Yongxiang Liu, Yi Wu and Lianfeng Shen, "A Rapid and Reliable Disaster Emergency Mobile Communication System via Aerial Ad Hoc BS networks," *2011 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Sep. 2011.
- [36] 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.
- [37] The Weather Channel, "Earthquake Deaths to Reach 3.5 Million by 2100," <http://www.weather.com/news/science/earthquake-deaths-dramatic-rise-20130221>, retrieved Feb. 2013.
- [38] 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.

- [39] Search Underwood, "Improving Disaster Management," *Comm. of ACM*, vol. 53, no. 2, Feb. 2010, pp. 18-20.
- [40] Misako Urakami, Yuya Okada, Yasuyuki Niwa, Hisaya Motogi and Hiroshi Matsuno, "Construction of Wireless Network for Information Communication for a Disaster-affected Island," *IEEE International Conference on Advanced Information Networking and Applications Workshops*, Perth, Australia, Apr. 2010.
- [41] 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.
- [42] 3GPP, Specifications, <http://www.3gpp.org/Specifications>, retrieved Nov. 2011.
- [43] 3GPP, TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access" version 11.0.0.
- [44] 林進豐, "行動衛星通訊," 五南出版社, ISBN : 9789571150062, Dec. 2007.
- [45] 姚國章, "應急管理信息化建設," 北京大學出版社, ISBN : 9787301155806, Sep. 2009.
- [46] 孫玉, "應急通信技術總體框架討論," 人民郵電出版社, ISBN : 7115208328, 2009.
- [47] 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*, Tao-Yuan, Taiwan, Dec.2010.
- [48] 張雪麗等, "應急通信新技術與系統應用," 機械工業出版社, ISBN : 9787111292982, Jan. 2010.
- [49] 楊永年, "八八水災救災體系之研究," *公共行政學報*, vol. 32, pp.143-169, Sep. 2009.
- [50] 交通部電信總局, "九二一震災災後重建電信問答手冊," <http://kbteq.ascc.net/archive/dgt/dgt01.html>, retrieved Dec. 2011.
- [51] 高抗災通信平臺, http://88flood.www.gov.tw/committee_news_detail.php?cn_id=506, retrieved Dec. 2011.
- [52] 國家通訊傳播委員會, "高雄縣莫拉克颱風災後通訊傳播設施改善實施計畫," Jun. 2010.