

國立成功大學
工業與資訊管理學系碩士在職專班
碩士論文

單根樹狀網路之最佳節線重建排程研究

Optimal Arc Restoration Scheduling for Rooted Trees

研 究 生：吳孟修
指 導 教 授：王逸琳 博士

中 華 民 國 一 百 零 六 年 六 月

國立成功大學

碩士在職專班論文

單根樹狀網路之最佳節線重建排程研究

Optimal Arc Restoration Scheduling for Rooted Trees

研究生：吳孟修

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳

李宇之
詹家成 李宇欣

指導教授：王逸琳

系(所)主管：李興敏

中華民國 106 年 6 月 14 日

摘要

強震或颱風等天然災害可能毀損路段或橋樑，導致災區居民陷入孤立無援的困境，亟需外界給予支援與救助，因此在災害發生後，「如何緊急搶修路段，以使與外界隔絕的災區能與外界儘早連通」應為搶修單位最重要的決策考量，其目標著重在儘快恢復災區與外界的連通性。過往之相關災後路網修復文獻大都直接假設欲修復之災區路段為已知，且大都以修復者的觀點尋求最早修復完成之時刻；實務上，如何自眾多毀損之路段選出應該被修復的路段至為重要，且理應自災民而非修復隊的角度來制定決策目標，亦即如何使災民等待其路網連通外界的時間越短越好，將會更貼近災民需求。而針對上述實務考量的近期文獻，雖已提出理論之整數規劃模型，但僅能處理小規模之路網，且未將各災區之災民人數列入排程考量，導致其實用性大打折扣。

本研究著重於樹狀路網的路段修復排程決策，將救災中心視為該樹狀網路的單根，此樹狀網路可被視為原路網的簡化版本；修復隊自此網路之根節點（亦即修復中心）出發，逐步依各路段與根節點的遠近來修復該樹狀路網之所有路段，以達到網路上所有節點（亦即災區）能儘早連通至根節點的目的。雖說本研究所探討的樹狀網路已為原網路之簡化版本，然而若將此樹狀網路視為由數個沒有節線交集的分枝聯集，每分枝視為一個產品，各分枝上由根節點往葉子末端方向依序之各修復路段視為該產品的加工工序，而各修復隊各為一個獨立機台的話，則本修復排程問題可被類比成欲將各產品（分枝）依其加工工序（自根或分枝節點至葉子末端由內而外）指派給多機台（修復隊）的平行機台排程問題，仍屬 NP-hard 等級。

本研究將各災區之災民數量列入排程考量，從平行機台排程角度提出整數規劃模型，設計三種（均勻選取式、比例選取式及鄰近搜尋式）單次性之貪婪演算法，與疊代性之基因演算法，並以大規模數值測試，期在短時間內能求得品質不錯之修復排程。

關鍵字：網路修復、平行機台排程、樹狀網路、整數規劃、貪婪演算法、基因演算法

Optimal Arc Restoration Scheduling for Rooted Trees

Meng-Hsiu Wu

I-Lin Wang

Department of Industrial and Information Management

SUMMARY

In the post-disaster management, how to repair the roads (arcs) to access villages (nodes) from a disaster recovery center (root) is very important. A wrong restoration scheduling leads to longer waiting time to access villages and may increase number of casualties.

This thesis focuses on restoring arcs in a rooted tree network simplified from a general network. In particular, given a rooted tree, we schedule the restoration tasks, one arc by a team at one time, such that the makespan or the weighted total waiting time for all the refugees (weights) to become accessible from the root are minimized.

We explain why this network restoration scheduling problem corresponds to an NP-hard parallel machine job shop scheduling problem, where each branch from the root or a branch node to a leaf node corresponds to a product required to go over sequential processes (restoration for arcs in the branch from the root or a branch node to a leaf node) by machines (restoration teams). We then formulate this arc restoration scheduling problem by an integer program. To further shorten the solution time, we also develop four heuristic algorithms including three greedy algorithms (GD, GDr, GR) and a genetic algorithm (GA) to calculate a schedule of good quality in shorter time. Our intensive computational experiments indicate GR has the best performance among all four algorithms.

Keywords: Network Restoration, Parallel Machine, Job Shop Scheduling Problem, Rooted Tree, Integer Program, Greedy Algorithm, Genetic Algorithm

INTRODUCTION

Natural disasters such as earthquakes, typhoons and landslides often cause considerable damages to the affected areas (a network). When a disaster damages the roads (arcs) connecting villages (nodes), those people living at an inaccessible village have to wait until at least one road (path) connecting it to the disaster recovery center (root) is restored. It is very important to decide when and which damaged road segment (arc) to be restored by which restoration team (machine). This corresponds to a network restoration scheduling problem that assigns renewable resource (number of restoration teams) to restore damaged arcs.

Without loss of generality, we can treat all the arcs are damaged. A spanning tree would be the most economic and effective way to access all nodes. If there are multiple recovery centers, we may add a dummy source node as the root and connect it to those recovery centers by dummy arcs. Thus, we only consider the arc restoration scheduling problem over a rooted tree. Although a rooted tree is a simplified version of a general road network, this scheduling problem is still NP-hard by mapping it to a parallel machine job shop scheduling problem with renewable resource constraints.

PROBLEM DESCRIPTION AND RESEARCH METHODS

Given a rooted tree, where each arc has an orientation pointing from the root to a leaf and requires to be restored by a restoration team. Each node has only 1 incoming arc from the root, and restoration always starts from an accessible node. If we want to shorten the working time for the restoration teams, we can set the objective function, denoted as MAX, to minimize the time to access the last leaf. Since the refugee side should be considered more than the restoration team side, it makes more sense to set the objective function, denoted as SUM, to minimize the total waiting time for all the refugees on all nodes to become accessible. We have tested both objective functions.

We first propose an integer program (IP), based on the resource constrained parallel machine job shop scheduling problem. We use a binary decision variable x_{ijt} to represent whether the restoration team j starts to restore the incoming arc of node i at time t ($x_{ijt} = 1$) or not ($x_{ijt} = 0$). By estimating a planning horizon T , we set constraints including: (1) all arcs have to be restored by T ; (2) each team j can at most conduct a restoration task at any time t ; (3) an arc closer to the root should be restored earlier than another arc farther to the root in the same branch; and (4) at any time t , at most R amounts of resource can be consumed in restoration. Although this IP is already compact, it is still time consuming by a state-of-the-art solver GUROBI (GRB).

To further shorten the solution time, we propose three greedy algorithms and a genetic

algorithm. The first greedy algorithm, denoted as GD, selects the most effective team with the largest ratio of weight (e.g., number of refugees at the head node of an arc) to restoration time to restore a currently accessible but damaged arc. GD is based on the greedy intuition of the shortest processing time. The second and third greedy algorithms, denoted as GDr and GR, respectively, also exploit similar intuition but adding stochastic features. In particular, for each accessible damaged arc, GDr first calculates its relative probability to be selected among all such arcs. This gives more chance but not always to select a more efficient team-arc assignment to avoid converging to a local optimum. For GR, we have implemented 3 variants, denoted as GR1, GR2, and GR3, where each corresponds respectively to a probability threshold 25%, 50%, and 75%, to indicate whether or not to select another randomly generated team-arc assignment by chance, if a generated random number is greater than a specified threshold. Therefore, GR1 (25% threshold) may change the original greedy decision more often than GR3 (75% threshold). Finally, we implement a genetic algorithm, denoted as GA, which iteratively converges to a better team-arc assignment by a 0-1 team-arc assignment encoding framework.

RESULTS AND DISCUSSION

We have conducted intensive computational experiments by measuring the quality of solutions using optimality gap and computational time over 1,620 randomly generated rooted trees. In particular, 3 settings on the number of teams and total resource upper bound, 3 settings on restoration time, 2 settings on arc resource consumption, 9 settings on the tree heights and number of arcs, and 10 random trees for each of the above $3*3*2*9=162$ combinations. For each generated rooted trees, we test GRB, GD, GDr, GR1, GR2, GR3, and GA with 2 objective functions (MAX and SUM). As expected, GRB takes much longer time (58,324ms on average), yet all the heuristics can terminate within 2ms per iteration. GDr and GA calculate optimal solutions more likely than other heuristics for the MAX cases, yet GR1, GR2, and GR3 give more optimal solutions for the SUM cases. The three GR variants give solutions of gap within 10% in all tests, whereas GD, GA, and GDr do not have such solution robustness. Therefore, we recommend GR in general.

CONCLUSION

This thesis has three contributions: First, it deals with the arc restoration scheduling problem over a rooted tree from a new viewpoint new to the literature; Second, we propose the first IP formulation for rooted trees and also suggest using SUM as objective function; Third, we develop variants of greedy algorithms and conduct intensive numerical analysis.

Among proposed heuristics, we suggest using GR, due to its robust performance in all scenarios, and is especially suitable for the SUM cases. For future research, we encourage

investigating the design of better heuristics, and extending our findings for general networks.



誌謝

回顧過去幾年求學的過程中，由最初報考的懵懵懂懂，到面對接踵而來的課業壓力，內心的惴惴不安，到如今已完成論文並撰寫誌謝，心中的滋味可謂是五味雜陳。除了白天需要面對排山倒海的業務壓力，晚上更得趕著上課、寫報告、考試及撰寫論文，這一路上除了自己要有不放棄的心態，更需要一群師長及朋友們相助才可能走到今天這一步。首先，我得感謝當初督促我要報考成功大學的禎祐學長，是他不放棄地一直催促我去考試，如果沒有他，或許我還只是上班下班，整日回到家都無所事事的人，無法去為自己的人生注入不一樣的活力；再來，我要感謝的是好朋友士敏，因為我不是寫 C 語言出身的資訊人員，在整篇的論文中，他在程式語法的經驗及建議上給我很大的幫助，讓我可以順利完成程式撰寫及最後的數據測試；再來，我要感謝的是我女朋友貞懿，從我們一開始認識她就一直督促我要加緊腳步完成論文撰寫，並取得學位，但我因為很多種種因素一直無法如期地完成，但她總是在我身邊不斷的鼓勵並鞭策我，讓我知道雖然一樣是在職專班，大家也都是白天上班，晚上上課，不過不應該拿這些理由來做為阻礙學習的藉口，要知道別人如果可以，自己也就一定可以。再來，我要感謝我的指導老師王逸琳老師，從一開始從同學口耳傳述老師非常的精實，到最後進入師門發現老師對於課業的專注程度以及付出時間都是不同於其他老師，總是會為學生想出很多解決方法及建議，並總是切中問題核心，很珍惜與老師之間的緣分，雖然你的話不多，但可以感覺到你對學生的關心，真的非常謝謝你。

再來是要感謝一路上陪伴我的同事及家人，也謝謝這一路上幫助過我的所有人，諸多恩惠永記於心，謝謝你們。

吳孟修

中華民國一百零六年六月

目錄

第一章、緒論	1
1.1 研究背景與動機	1
1.2 研究目的	4
1.3 研究範圍與限制	6
1.4 研究方法與步驟	7
1.5 論文架構	8
第二章、文獻探討	9
2.1 國內道路災害搶修作業概況與特性	9
2.1.1 道路災害搶修之定義	9
2.1.2 道路緊急搶修期程	10
2.2 災害管理相關文獻	12
2.2.1 路段修復文獻	12
2.2.2 平行機台相關文獻	14
2.3 小結	15
第三章、研究方法	16
3.1 問題描述與假設	16
3.2 問題假設	17
3.3 模型架構	18
3.3.1 模型符號定義	18
3.3.2 整數規劃模型	18
3.4 啟發式演算法	22
3.4.1 範例說明	23
3.4.2 均勻選取式貪婪演算法細部步驟	25
3.4.3 比例選取式貪婪演算法細部步驟	26
3.4.4 鄰近搜尋式貪婪演算法細部步驟	26
3.4.5 基因演算法	27
3.5 小結	34
第四章、模型建構與驗證結果分析	36
4.1 輸入資料	36
4.2 啟發式演算法比較	39
4.3 小結	45
第五章、結論與未來研究方向建議	47
5.1 結論	47
5.2 未來研究方向建議	49
參考文獻	51

圖目錄

圖 1.1 災後樹狀網路圖	3
圖 1.2 修復隊時間軸及路段修復所需時間示意圖	3
圖 1.3 網路問題最佳解示意圖	3
圖 1.4 災後道路網與修復時間對照圖	4
圖 1.5 工作隊-路段排程制定範例圖	5
圖 2.1 重大震災發生至復舊之過程	10
圖 2.2 震災各時期物流工作示意圖	11
圖 3.1 災區路段圖	16
圖 3.2 資源占用說明	19
圖 3.3 時刻檢查資源占用機制	20
圖 3.4 時刻檢查與資源占用範例圖	21
圖 3.5 災區路段與工作隊搶修排程圖	22
圖 3.6 道路網-工作隊修復時間與權重比例範例圖 1	23
圖 3.7 均勻選取式貪婪演算法作法示意圖	23
圖 3.8 比例選取式貪婪演算法作法示意圖	24
圖 3.9 鄰近搜尋式貪婪演算法作法示意圖	24
圖 3.10 道路網-工作隊修復時間與權重比例範例圖 2	25
圖 3.11 輪盤法選擇 (Roulette Wheel Selection)	29
圖 3.12 單點交配	30
圖 3.13 雙點交配	30
圖 3.14 多點交配	30
圖 3.15 均勻交配	31
圖 3.16 染色體編碼示意圖	31
圖 3.17 災後路網圖與工作隊-路段修復時間及權重比例	32
圖 3.18 初始母體示意圖	33
圖 3.19 染色體交配示意圖	33
圖 3.20 染色體突變示意圖	34
圖 4.1 災後道路網(以階層表示)	37
圖 4.2 災後道路網(以分枝表示)	37
圖 4.3 災後道路網(以數量表示)	37
圖 4.4 路網產生器	38
圖 4.5 路網產生器設定值暨圖檔產生明細示意圖	39
圖 4.6 各啟發式演算法直線圖	41
圖 4.7 各種解法之 Gap% 值折線圖	43
圖 4.8 鄰近搜尋演算法之 Gap% 折線圖分析	44

表目錄

表 1.1 工作隊-路段維修時間表.....	5
表 2.1 路段受損與搶修型式.....	9
表 2.2 路段修復相關文獻與本研究之比較.....	14
表 4.1 路網產生明細.....	38
表 4.2 不同演算法在不同網路結構下的求解表現(MAX).....	40
表 4.3 不同演算法在不同網路結構下的求解表現(SUM).....	40
表 4.4 各演算法 Gap% 值比較表(搶修路段數量 M=16).....	42
表 4.5 各演算法 Gap% 值比較表(搶修路段數量為 M=36).....	42
表 4.6 各演算法 Gap% 值比較表(搶修路段數量為 M=64).....	42
表 4.7 可行解 CPU 執行時間(單位：毫秒).....	45



第一章、緒論

本章說明本論文之研究議題：第一節為研究背景與動機；第二節為研究目的；第三節為研究範圍與限制；第四節為研究方法與步驟；第五節則為本研究之論文架構。

1.1 研究背景與動機

台灣位處環太平洋地震帶，地震發生的次數頻繁，並且常有強烈的地震發生。依據中央氣象局過去九十年的觀測資料顯示，台灣地區平均每年約發生二千二百次地震，雖然其中多數為無感地震，但有感地震每年平均約有為二百一十四次，根據以往紀錄，在一九〇〇年至一九九九年間，共有八十九次災害性地震發生，幾乎每年可能會有一件地震災害發生，其中又有二十三次為地震規模達七以上的強烈地震，由於地震災害是一種無預警性的災害，且是突如其來發生的，故其突發的程度及所造成的後果非一般災害所能比擬。

綜觀國內歷年來地震，如 1999 年 9 月 21 日集集大地震(2,415 人死亡，51,711 棟房屋全毀)、2002 年 3 月 31 日花蓮秀林外海 331 大地震(5 人死亡)、2006 年 12 月 26 日屏東恆春外海地震(2 人死亡，3 棟房屋全毀)、2009 年 12 月 19 日花蓮磯崎外海地震(17 人受傷，多棟房屋毀損)、2010 年 3 月 4 日高雄茂林地震(17 人受傷，多棟房屋毀損)、2013 年 3 月 27 日及 6 月 2 日南投仁愛地震(6 人死亡 97 傷)、2013 年 10 月 31 日花蓮瑞穗外海地震(1 人受傷)等均造成災情的發生；而對照國外地震災情，尤有甚者，1995 年日本神戶發生規模 7.2 的強震，傷亡 25,000 多人，將近 670,000 平方公尺面積被燒毀，損失金額在 9.5 兆日圓以上；2001 年 1 月底，印度更發生規模 7.8 的強震，死亡接近 3 萬人，受傷接近 6 萬人。由於地震造成極大破壞，不僅造成人員傷亡，其對運輸系統及設施之破壞，更是大幅影響居民之生活便利；尤其當運輸系統受損時，其救援效率必大打折扣；災區交通無法保持暢通，不僅災情資訊無法傳遞，搶救速度亦受牽連，因此「道路緊急搶修」實為災後救援單位首要考量的任務。Fiedrich et al.

(2000)等人亦將緊急搶修視為救災工作可以減少傷亡的主要工作之一，並視搶修為修復運輸系統生命線，提升災區內外聯繫之可及性。藍武王(2000)認為大地震發生後，災區範圍廣泛，且多樣性的災害會伴隨發生，由於路網同時有多處損害致使交通中斷，將嚴重影響災民搶救、緊急物資的運送，以及復建效率。

然參考國內針對地震等天然災害的應變措施，主要依據民國八十九年七月頒行的「災害防救法」運作，條文中雖明定各類災變下相關單位之組織及職掌，以為各類災害之緊急應變指導原則，惟針對路段搶救的應變機制，僅律定高速公路權責單位由高速公路局維護，快速路段、省道由公路總局維護，縣道、鄉道由各縣市政府維護，其餘一般市區路段、村里路段、產業路段由各鄉鎮市公所維護。雖明訂有各相關權責單位負責搶修的路段，卻無法提供指派與決策的應變方法，面對諸多毀損路段的搶修順序與資源的指派更是鮮少著墨。另以過去實際搶修經驗中，工程單位搶修與復建係先勘災、掌握災情，調查損害情形後，實施緊急搶通與修復路段工作，往往由決策者依經驗進行，普遍缺乏系統化的數理分析結果當後盾；故決策雖可行，但並非最佳及最有效率之決策。因此，如何在最短時間內制定出有效率的緊急路段搶通計劃，訂定所欲搶修之路段、搶修順序及機具、人力等資源的搶修排程，即為本研究探討的重點。

假設某天災造成某一地區對外通聯之路段及橋樑等均受損，可知搶修計劃在現實情境下，應為搶救居民生命財產為主要目標，故標的物皆以大量群聚之鄉鎮村里進行規劃。過去諸多文獻均將受災區域設定為一連通網路 $G=(N, A, C)$ ； N 為網路內節點(nodes) 集合，即為災點，可分為三類：起點、可對外通連節點及不可對外通連節點，也因此可將其歸類為災點搶通計劃。而本研究係以路段搶修為主要標的，同樣地，在本研究中也即可將兩個相鄰災點間的路段搶通完畢視為災點間的連通。

除搶通災點外，時效性亦為搶修計劃中關注的焦點之一，以決策者之立場，是希望整體完工時刻越早越好，也就是最小化最晚搶通災點的時刻；然而以災民的立場來看，則希望自己所在之災點能迅速獲得支援，因此可能是最小化各個災點搶通時刻之總和較為合理。

搶修計劃制定同樣也取決於災後受損道路網的型態，無論受損之路網如何連結，可預期的是緊急搶通路段應該會是「樹狀」網路，此乃因樹狀網路是能保證節點連通的最基礎連結形態。因此本研究直接針對樹狀路網加以探討，亦即搶修計劃均由單一起點出發，將與起始點連接之任一分枝來進行搶修，如圖 1.1 所示。

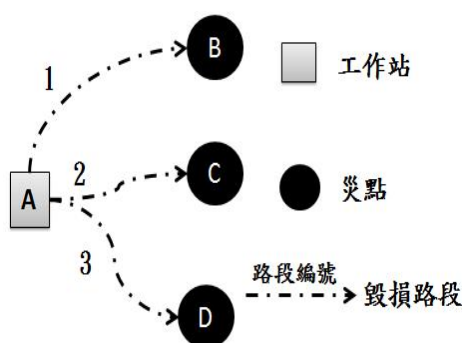


圖 1.1 災後樹狀網路圖

假設任一救災隊能力相同，且同時刻只能修復一路段(節線)，結束後方可修復其它路段，當路段數小於救災隊數量時，可知最佳之搶修計劃是在時刻等於 0 時為每一條需要修復的路段分配一個救災修復隊，此時所有路段中最晚修復完成的時刻，就是整個問題的最早完成時刻。當路段數大於救災隊數量時，以圖 1.2 做說明，假設現有兩個修復隊以及三條需要修復的路段，灰色方格的長度代表所需的修復時間，可知最快完成此問題的方法是讓修復隊 1 修復路段 1 及路段 2、讓修復隊 2 修復路段 3，如圖 1.3 所示。

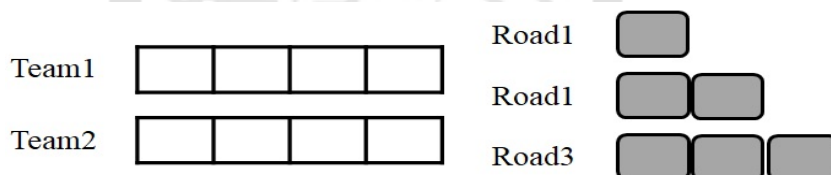


圖 1.2 修復隊時間軸及路段修復所需時間示意圖

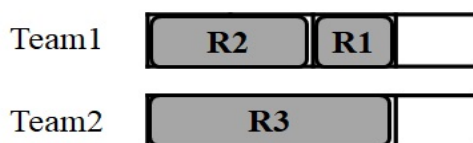


圖 1.3 網路問題最佳解示意圖

經由此分析，本研究係針對災後道路網屬單根樹狀網路內全數路段進行搶修排程的制定，並以平行機台排程方式求解有限工作隊時之最佳路段修復排程。本研究可進

一步使用具資源限制的平行機台排程手法，來解決網路中具其它有限資源(譬如電力、水、載具等)限制時之最佳排程。

1.2 研究目的

災後救援之最終目標是於黃金 72 小時內至災區內進行救援，故如何在極少時間內運用有限資源並決定最佳效益之道路緊急搶修排程甚為重要。若將個別欲修復之路段視為工件，而救災之人、物力資源視為處理工件的機台，則如何在最短時間內搶通所有的災點即為一複雜的「具資源限制之平行機台排程問題」。

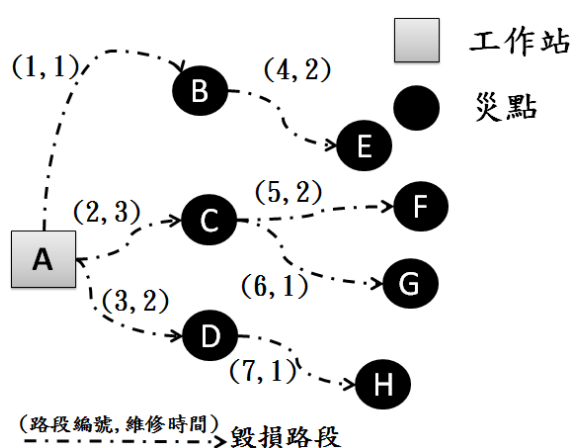


圖 1.4 災後道路網與修復時間對照圖

本研究僅探討單根樹狀之毀損網路，假設有一災後道路網如圖 1.4 所示，節點 A 為緊急搶修工作隊出發起始點，而節點 B、C、D、E、F、G、H 各為災點，虛線段則為損壞之路段，虛線上括號左邊為路段之編號，右邊則為工作隊維修時間，由此可知如從工作站 A 出發，要搶通災點 B、C 及 D，需先搶通編號 1、2 及 3 之路段，也因此代表以本研究著眼之道路網，要制定搶修排程之順序，必先以與起點有相連通路段為優先考量。

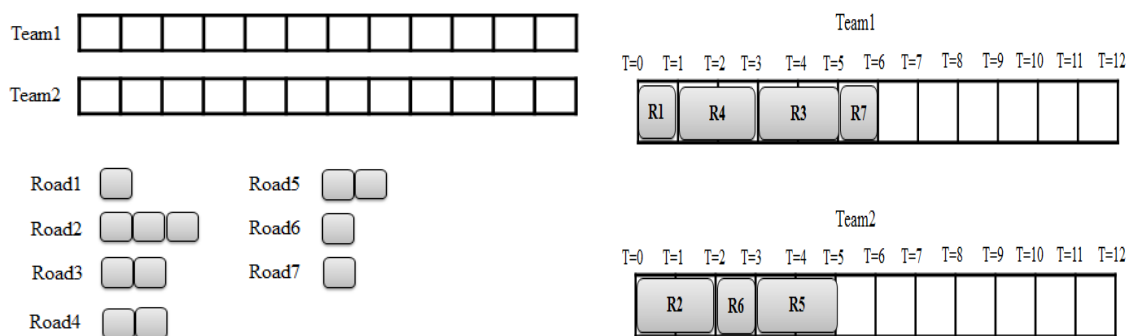


圖 1.5 工作隊-路段排程制定範例圖

在現實情境中，路段修復亦須考慮時序性。以上述圖 1.4 之災後道路網為例，如要搶通災點 E 則需修復路段編號 4，須先確認工作隊是否已完成災點 B 之搶通，亦即編號 1 之路段必須修復完畢。假設已知工作隊數量為兩組（工作隊編號 1 及 2），會先選取與起點相連通之路段編號 1、2 及 3 進行搶修，相關搶修排程制定如圖 1.5 所示。為儘早完成修復任務，直覺上將儘量避免工作隊閒置，因此在時刻 1 時，工作隊編號 1 已閒置，而當下僅有路段編號 4 可供修復，便立即將路段 4 安排給工作隊 1。此外，考量到路段的連結順序，在同一分枝內，唯有在較近根節點之前一路段搶修完畢後，才可繼續修復離根節點較遠的路段。

表 1.1 工作隊-路段維修時間表

路段 編號	工作隊維修時間		
	工作隊 A	工作隊 B	工作隊 C
1	2	3	1
2	3	4	6
3	2	5	4

然而實際搶修作業使用之資源種類繁多，其中人力為主要之資源類別，並常搭配其他輕重型機具組成一組搶修工作隊，決策單位再依次指揮各組工作隊遂行各災點的搶修作業。實務上，不同之工作隊可能對不同路段有不同之維修時間。假設現有三組工作隊 A、B 及 C，其能力均不同，則路段維修模式可如表 1.1 所示，各路段皆可由各工作隊搶修。在本研究中不限制某些路段僅限定給某工作隊修復，而將此特性應用於平行機台的理論上，即可將其視為非等效平行機台多模式排程問題進行求解。另外，

在考慮現實情境下，於天災發生後，所能動用之資源不多，因此應盡可能發揮救災資源的最大效益。而效益應該同時考量修復時刻與各災點的重要性（或可用各災區的災民數量代表），故在本研究於各路段加入權重概念，以修復時間乘以權重，當權重值越高時，代表路段的搶修比重越高，假設有兩個路段 A 及 B 需要搶修，路段 A 修復時間為 5，可救援的災民為 10 人；而路段 B 修復時間為 10，可救援的災民為 1000 人，則 A 路段搶修權重為 50，B 路段為 10000，兩相比較下，應以 B 路段為優先搶修對象。

因此本研究不僅是探討搶修排程問題，亦決定各欲修復路段的最佳緊急修復方式，並可用多重作業方式及資源限制下之平行機台問題建構數學模型。然在災後路段搶修網規模龐大時，以數學模型建構需耗時甚久，無法有效解決問題之急迫性，故除了建構模型外，本研究亦發展四個啟發式演算法，期望於較短時間內得到一近似解，以供決策單位於災後緊急救援搶修階段，獲得更高效率之重建方案。

1.3 研究範圍與限制

有關本研究之研究範圍與限制如下：

1. 本研究之災後道路網屬單根樹狀網路圖，係基於搶修哪條路段已知的情境下，對災區內路段制定搶修排程，以提供決策單位一套搶修路段之決策方法。
2. 就長遠之角度來看，於本研究中工作隊之移動時間相較於修復時間來得短，也就是說當工作隊閒置後，對可分配之路段進行指派時，對於從現有路段移動至搶修路段中間的移動時間可忽略不計。
3. 於本研究中設定每個路段相對於工作隊來說均有其各自之修復時間與使用資源多寡，因此不考慮工作隊可合作修復同一路段，亦即一個路段僅可分配給一個工作隊進行搶修。且於研究中設定每一期時刻之使用資源上限，去計算當期時刻下各工作隊搶修路段使用資源之總和，並據以限制之。

4. 在問題中所描述之有限資源，係指可更新資源(renewable resource)，即當工作隊使用該資源搶修路段時，該資源於開始時刻便被該作業占用，直到作業結束時刻後方可繼續使用，譬如：搶修人力資源或機具設備等等。

1.4 研究方法與步驟

本研究透過災後道路網已知給定後進行模型的建置，係以平行機台方式探討最小化總完工時刻與總等待時間之災後單根樹狀路網搶修排程問題，並發展四個啟發式演算法，據以優化本問題之處理時效，最後依據實驗數據進行驗證分析與討論，因此本研究可主要分成四部分：

1. 單根樹狀網路路段搶修排程模型建構：

給定工作隊及所需要修復的路段，將樹狀網路上每一分枝視為一個產品，分枝上由根節點(起始點)往葉子末端方向依序之各特修復路段視為該產品的加工工序，而各修復隊均視為一個獨立機台，並依據路段-工作隊最佳修復模式、當期時刻使用資源上限及路段搶修時序性，據以建構最小化總完工時刻及總等待時間之模型。

2. 三種貪婪演算法模型建構：

依同樣路網自起始點出發，從當下可修復路段及空間的工作隊中，選擇以「路段搶修時間/權重」最小者為優先搶修路段，每搶修一條路段完畢便搜尋可繼續執行之為完成路段，並將閒置之工作隊立即指派，並同樣計算最小化總完工時刻及總等待時間，直至路網搶修完畢。

3. 基因演算法模型建構：

依據貪婪演算法所計算之最佳解作為基因演算法之初始母體，染色體編碼之原則係將路段依修復完工時刻順序進行排列，以總完工時刻做為適應性函數，透過複製、交配及突變尋求最佳解，藉以優化問題處理之時效性。

4. 數值測試實驗驗證

藉由上述之模型建構，求解實驗數據結果，並進行驗證與分析。

1.5 論文架構

本篇論文結構如下：第二章回顧國內外防救災體系及路段修復等相關文獻；第三章敘述研究方法，模擬以單根樹狀之災區道路網為基礎下，對全數路段進行最佳修復順序排定；第四章為實驗結果與分析，最後第五章總結本論文與建議未來研究方向。



第二章、文獻探討

台灣近年來天災頻傳，不僅造成人員及財產的損害，對於道路網之破壞更造成災區救援及災民疏散之困難，而災區路段無法暢通，則災情資訊無法有效傳遞，搶救時效性亦受牽連，最終可能造成人命及財產難以彌補的損失。故災後道路網路搶修決策研究十分重要。而本研究係針對災害發生後，道路網遭受破壞時如何以最短的時間成本緊急修復路段以連通各災點，使物資能在最快的時間內送達。本章安排為 2.1 節說明道路災害搶修之定義，並介紹國內災害搶修工程單位之概況與搶修特性；2.2 節彙整國內、外學者所提出之災後路段搶修作業相關文獻。

2.1 國內道路災害搶修作業概況與特性

2.1.1 道路災害搶修之定義

依據「災害防救標準作業手冊」，災害緊急搶修係為養護單位於路段遭受破壞後，為維持行車安全或恢復路段之原有功能，及提供良好服務水準所採取之工程措施。

李志華(2003)基於公路總局轄管包含一般路段、橋樑與隧道等三類型公路設施，彙整不同類型公路設施之受損型式及緊急搶修方法，如表 2-1 所示。

表 2.1 路段受損與搶修型式

道路系統組成	災損型式	緊急搶修方法
一般公路	1.路面龜裂。 2.路基隆起或下陷、缺口及流失。 3.落石坍方。 4.其他。	1.路面整平。 2.路基固定填平。 3.落石坍方、土方障礙移除。
橋樑	1.落橋。 2.橋墩傾斜、裂縫。 3.橋面位移。 4.其他。	搶搭便橋或搶築便道。
隧道	1.出現裂縫。 2.內壁掉落。 3.落實掩埋封閉。 4.倒塌(全毀、半毀)	1.落石坍方移除。 2.倒塌移除。

資料來源：李志華(2003)

王在莒(2004)則表示緊急搶修工作為臨時性作業、階段性任務，其要求須至少完成一車道之搶通工作，可供工程及救援車輛進行。

吳心琪(1997)說明路段搶修工程包含障礙清除、工作便橋之搭建(搶修工作)至土建工程完成，並至少有一車道可容許人客車或卡車通過為止，但不討論鋪面、植被等完全復舊工程，而楊宗岳(2000)將「搶修」解釋為對已受損之路段、橋樑進行緊急搶修，以達到災害聚落或生產地之聯外運輸至少可供小型車通行之目的。

2.1.2 道路緊急搶修期程

吳心琪(1997)和施佑林(2005)將震災發生至復舊之過程分為以下幾個時期：發震期、混亂期、避難期、大火期、搶修期以及復舊期，如圖 2-1 所示。大型震災發生後，必先進行事故區滅火而後方可進行倖存傷患搶救、事故現場清除，之後才進行災後搶修及重建工作，其工作範圍涵蓋醫療、消防、警力及維修工程等方面，相對於消防、救護等救災工作，工程搶修與復舊工作其緊急性較低。其中公路系統在搶修復舊時期擔負物資人力運送流通之主要責任，依日本神戶地震經驗與國內省公路局於風災後之實務經驗可知，公路網若同時多處受損，應先緊急搶修，待全面搶通後方可進行復舊工程至恢復原貌。由圖 2-1 可知，道路緊急搶修作業在震後第三天開始進行搶修，全面搶通後方可進行復舊工程至原貌，大約為震後三個月左右開始從事復舊工作。



圖 2.1 重大震災發生至復舊之過程

資料來源：吳心琪(1997)、施佑林(2005)

黃亦琇(2000)和陳信宇(2000)則將防災計畫分為災前、搶救、維生及復舊四個階段，如圖 2-2 所示，在地震發生前即做好事前規劃，將地震可能造成的傷害降至最低，因此災前即有防災工作；而地震發生後，欲降低大型震災可能導致的損害，在規劃上可分為搶救、維生及復舊三階段，也就是以時刻區隔工作流程，而交通設施的搶通視為在搶救期和維生期所該從事的作業，如圖 2-2 所示，各階段分述如下：

1. 災前：在地震發生之前，主要是針對防災與減災等兩項工作，目的在於降低災害的潛在損失，防止災害發生。
2. 搶救：地震發生後的一個星期內，第一時間儘速完成救援行動，以降低損失，保障人民生命財產並防止第二次災害的發生。
3. 維生：地震發生後一星期後至三個月內為維生階段，主要工作為維持災區民眾的基本民生需求。
4. 復舊：地震發生後的三個月之後為復舊階段，主要工作為儘速恢復災區民眾的生活、重建設施、社會恢復正常及各種復原工作。

重建設施、社會恢復正常及各種復原行動。

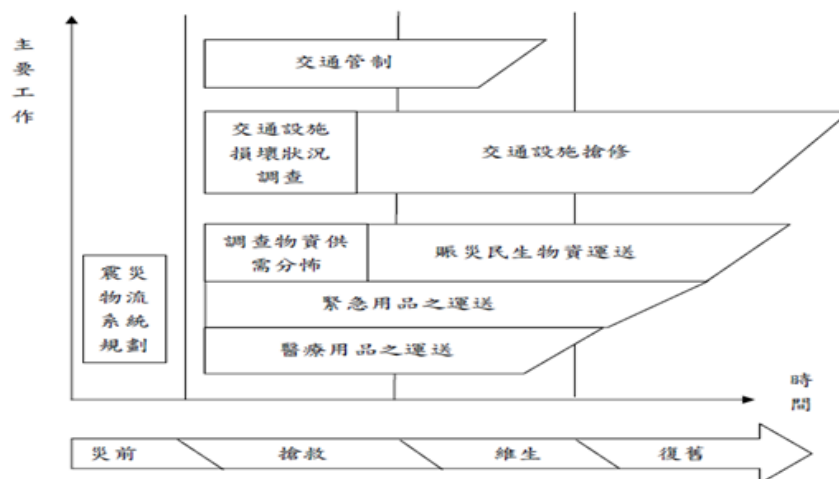


圖 2.2 震災各時期物流工作示意圖

資料來源：陳信宇(民 89)

綜上所述，本研究探討之問題係為災後道路網緊急搶通問題，係於災後三天至一星期內開始作業之行動，與災後重建復原的道路網問題具體差異為前者屬期程短暫的

搶修計劃，講求時效性及急迫性，而後者屬長期規劃的修復計劃，而如表 2-1 所示，路段搶修主要任務包括障礙清除、搭建便橋與搶築通道，至少可容一通道之車輛通行為止，使支援得以運送，而復原重建則包括路段鋪面與植被等復舊工程。

災後緊急搶修問題在過去幾年亦有廣泛的討論，以下將依本研究特性蒐整相關文獻進行分析探討：

2.2 災害管理相關文獻

2.2.1 路段修復文獻

以路徑為基礎(path-based)建構數學模型之文獻，如 Wilson et al. (2014)認為於災害發生後，當務之急便是物資供應網(Supply Chain Network, SCN)之暢通與否，而物資輸送之時效又取決於路段毀損狀況，因此提出兼具最小化物資運輸時效與整體路段復原時間為目標之災後供應鏈網路重建排程，研究定義路網中僅有供應點(Supply Point)、需求點(Demand Point)及維修站(Repair Crews Locations)三個節點，維修隊均自 Repair Crews Locations 各自出發搶修路段，以迅速搶通 Supply Point 及 Demand Point，將維修隊(Repair Crew)及物資(Load)之起訖點及方向性均以 Arc 表示，而 Arc 間的交集(Intersection)則為節點(Node)，重新訂定以路徑為基礎之網路圖，並以具連續性時間下之線性整數規劃模型求解，研究亦指出因為模型處理效率受限於節點及路段數量，且未有啟發式演算法處理，故容許之最大數量為 7 個節點或 19 個路段。

Aksu & Ozdamar (2014) 認為災後路段重建問題之目標在於最大化路網各節點之連通性，以使災區內人員及路段上之碎石可迅速撤離與清除，其方法係於每一災點至人員撤離點及碎石臨時集中地點之最短路徑，以這些最短路徑為基礎建構整數規劃模型，在每一期提供之維修資源有限，且機具移動時間可忽略之情況下，盡可能越早搶通這些路徑越好。運用動態路徑與 Divide and Conquer 的觀念建構一演算法，將原路網分割成數個區域，並用不同資源量加以限制，針對每一災損區域進行求解(divided)，得到各災損區域於不同資源量下之具體維修計劃，最後，再以總體資源上限，尋找不

違反每一期資源上限之各災損區域的維修計劃組合(conquer)。

Matisziw et al. (2010) 則認為緊急修復作業中最主要之目標係維持特定的某些節點間路徑連通性及流量之上限，譬如救災指揮中心及災損點，若欲維持路徑連通性，則須修復路徑上之全部節線，並依此觀點建構一整數規劃模型，將整體修復計劃分為數期，在依循著每期搶修成本預算控制下，據以決定每期要搶修之節線，目標在最大化每一個起訖點(Origin-destination, O-D)間之流量與最小化總搶修成本。

黃琮閔(2015) 則以多重作業方式及資源限制下之專案排程問題 (Multi-mode Resource Constrained Project Scheduling Problem, MRCPSP) 為基礎下，探討在有限資源下，兼顧「路徑決策」、「作業排程與資源指派」與「多重修復方式」之路段搶修方法，並據以建構整數規劃模型，而當此類最佳化問題在問題規模龐大時，求解效率亦顯不彰，故該研究亦透過貪婪式演算法及粒子群最佳化演算法藉以路網問題龐大所造成之求解效率不佳之問題。

林秉錚(2015)以物資補給需求滿足的總等待時間極小化對地下管線網路修復問題進行探討，將整體網路修復視為專案，節線修復為專案中之子任務，而工作隊即為有限之救災資源，並將問題定義為具網路架構之特殊「具資源限制下之專案排程問題」，研究中指出子任務無先後關係規範，但不同的修復順序卻對目標有不同的影響。於研究中利用 Branch-and-Cut (B&C)架構與多商品網路流量(Multi-Commodity Network Flows, MCF)建立整數規劃模式，並對單一工作隊及多工作隊之特例分別設計貪婪樹生成法與基因演算法以解決當網路規模變大時導致求解緩慢之問題。

從過往路段修復文獻中可發現，以災後道路網而言，未有如本研究所探討之單根樹狀網路圖，而就問題而言，也少有將「路段搶修排程」、「資源使用限制」及「多重修復方式」等做一次性考量，如表 2.2。其中，林秉錚(2016)所探討的問題假設與本研究雷同，且道路網中亦有測試樹狀網路的數據，然因該研究方向係針對地下管線，可隨時選擇任一毀損節線來修復；Aksu & Ozdamar (2014)是以修復隊觀點最小化總完工時刻，且通往各災點之搶修路徑做為決策依據，再進行搶修排程訂定，但其研究中

所探討之災後道路網並非樹狀網路圖，且未如本研究中提供多重修復方式，以更符合實際災後救援情境；而黃琮閔(2014)所探討的問題假設均與本研究相同，均兼顧災後路徑修復、搶修排程的制定、使用資源限制及工作隊多重修復等觀念，該研究可處理一般路網，但其求解效能不彰，因此本研究係基於該架構下，單獨探討其簡化後之單根樹狀網路圖型之搶修排程制定，並期望當遇一般路網規模較大時，可將道路網拆解為多個樹狀網路圖，以彌補求解效能不彰之問題。

表 2.2 路段修復相關文獻與本研究之比較

文獻	路徑修復	搶修排程	資源限制	多重修復
Wilson et al.,(2014)	✓	✓		
Aksu & Ozdamar (2014)	✓	✓		
Matisziw et al.,(2010)	✓	✓		
黃琮閔(2015)	✓	✓	✓	✓
林秉錚(2016)	✓	✓	✓	
本研究	✓	✓	✓	✓

2.2.2 平行機台相關文獻

針對災害管理文獻，亦有些文獻是以平行機台之概念進行探討，如 Edrisi & Nadi (2014)以災後能迅速完成各災點之救援需求為主要目標進行探討，研究提出將整個災區切割為數個區域(region)，且將數個區域分配給工作隊，其主要功能不在搶救與輸送物資，是在對該區域內之災點實施僅一次之繞徑，將區域設定為工件(works)，而工作隊設定為機台(machine)，則此問題是結合車輛繞徑問題(Vehicle Routing Problem, VRP)與平行機台排程(Parallel Machine Scheduling, PMS)特點據以產生之數學式以求解，藉由多個工作隊同時於各個區域進行作業，以最小化最大之工作隊最後完成時刻為其目標。

Felix et al. (2014) 探討災後如何整合救援單位並分配指派搶救每一個災點，並將

其定義為救援單位指派與排程問題(Rescue Unit Assignment and Scheduling Problem, RUASP)，作者運用平行機台方式將每一個救援單位視為機台，而各個災點代表一個事件(incident)，可視為個別獨立處理之工件，而機台處理的工件中則包括救援單位移動至災點的時間、災點受損嚴重程度權重及搶救處理時間，同時考慮是否可合作處理工件，定義出 RUASP 及 RUASP/C 兩個數學模式，以最小化處理時間為目標進行求解，該研究亦發展出貪婪演算法以強化運算之速度。

Nurre et al. (2012) 開發能夠供決策者參考的災後緊急修復交通網的系統，首先，他們發展出的啟發式配置的規則 (Heuristic Dispatching Rule)，參考平行機台排程的配置規則，並將其轉換到適合網路修復的模式，藉由此配置規則，得到下一個可處理任務之集合再用整數規劃模式求解，實際測試系統後，證實此系統對於災後緊急重建或是長遠的重建計畫都有不錯的結果。

2.3 小結

災後緊急搶修作業，在過去文獻中，未有以單根樹狀網路圖做為災後道路網基礎，且同時將「路段搶修排程」、「資源使用限制」及「多重修復模式」等限制因素做通盤考量。而參考過往文獻及綜合以上條件，可知當欲求解之路網越大、救災隊伍的數量越多、修復方式及時間越多時，也因數學模型規模太大造成求解時間過長，對此本研究除使用平行機台概念建構數學模型外，亦將設計有效率之求解演算法，以利決策單位得到更有效率之搶修排程計劃。後面章節將介紹本研究之數學模型及設計之啟發式演算法步驟流程。

第三章、研究方法

本章將透過單根樹狀路網的建置，以有限的工作隊及使用資源來制定路段搶修的排程，並以整數規劃模型將排程目標區分為「最小化作業總完工時刻」及「最小化各災點總等待完成搶通時間」等兩類。當路網規模越複雜，該模型求解時間往往耗時甚久，故又發展三種貪婪演算法及遺傳演算法，以在更短時間內獲得不錯的可行解。

3.1 問題描述與假設

災害發生後對社會所造成的影響，往往是全面性的。而災區範圍廣泛，路網可能同時有多處損害發生，為使路網早日恢復暢通，全面搶通作業完成時刻越早越好。因此路段搶修排程決策至為重要。無論受損之路網如何連結，可預期的是緊急搶通路段應該會是「樹狀」網路，此乃因樹狀網路是能保證節點連通的最基礎連結形態，故本研究探討災後單根樹狀網路之路段搶修作業排程問題。

災後救援之最主要目的為搶通災點，盡可能使救援隊進入災點以搶救災點上之人員性命。本研究是以路段搶修為基礎之排程制定，係以路段搶修完畢視為其末端災點之搶通。針對每一個災點而言，自外界（救災中心）欲聯通該災點僅需一條聯通路徑即已足夠，但是自該災點再出發去搶通之路段則可有多個，此連結架構便形成本研究所探討之單根樹狀路網，自根節點（救災中心）出發，聯通至各災點，以路段的搶通方向而言，剛好一律皆自根節點往外分枝擴散開來，如圖 3.1 所示。

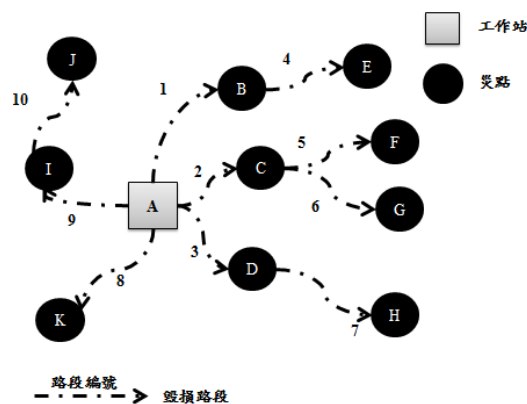


圖 3.1 災區路段圖

在問題求解限制中亦可加入有限資源之概念，如前述，本研究主要係針對可更新資源（譬如人力或機具設備）進行討論。廣義而言，各工作隊也有可能因配備的機具不同而有不同的搶修能力或效率，而各路段針對不同的工作隊可能也有一種以上(含)以上之被搶修方式。

此外，我們考慮兩種不同之排程目標式：(1)最小化作業總完工時刻，以 MAX 表示；(2)最小化各災點等待搶通之時間總和，以 SUM 表示。

3.2 問題假設

本研究之前提假設如下：

1. 災害發生後災區通訊極可能因此損壞，外界難以完全得知災區整體情形，故災害發生後，災害防救單位應先了解災區整體情況，建立災情資訊系統。因此假設災區資料已蒐集完全，以利救災單位整體考量並做出最有利之災害搶救決策。
2. 災害發生後，路段搶修工具及人力多寡於現實生活中會影響搶修時間長短與否，假設每條路段僅允許一個工作隊進行修復，且針對各個工作隊在各路段之搶修時間皆為已知（但可不同）。各工作隊一旦開始某路段之修復工程就不能半途暫停，直至該路段完工後才能再執行下一段工程。
3. 本研究假設工作隊進駐之救災中心為已知，以該中心為起點排定路段修復排程搶通所有災點；如果有多個救災中心可同時進駐，則可建構一虛擬之總中心聯通各救災中心，因此僅以單一根節點（救災中心）來簡化問題並不失其一般性。
4. 為模擬實際災害情況之發生，本研究於各個災點增加權重（可代表為對該災點搶通之等待人數）之概念，當權重乘以修復時間之值越高代表該災點緊急程度越高，須優先處理；由於每個災點僅有一條連入路段，故亦可將該災點的權重轉嫁至該連入路段，如此較方便計算該災點之加權總等待時間（即該權重乘以該路段被完成修復的時刻）。
5. 因本研究為搶修路段之排程計畫，且路段修復所花的時間（可能以天、小時計）

大多比其打通後之通行時間（可能以分鐘計）大許多，因此我們判斷修復排程之決策不應受到路段通行時間太多影響。為降低模式複雜度，在此我們假設工作隊的移動僅依賴已打通的路段，且不考慮工作隊於災點間已打通路段之通行時間。

6. 本研究僅考慮待修之路段，且假設這些路段構成一個單根節點的樹狀路網。已知之情況下建構模型，故須針對可分配之工作隊對災區內全數路段完成搶通。

3.3 模型架構

此節將說明本研究之數學規劃模式，首先介紹模型參數定義，再說明目標式與限制式。

3.3.1 模型符號定義

集合

N ：路段之集合 $i \in N = \{1, 2, \dots, n\}$

M ：工作隊之集合 $j \in M = \{1, 2, \dots, m\}$

參數

d_i ：執行路段 i 修復所需時間

T ：結束時刻上限

r_i ：執行路段 i 修復所需資源量

R ：每時刻資源使用上限

w_i ：路段 i 之權重（在本研究中等同於該路段箭頭端節點之災民人數）

決策變數

$X_{ijt} = \begin{cases} 1, & \text{路段 } i \text{ 是否在時刻 } t \text{ 由工作隊 } j \text{ 開始執行，是則其值為 } 1，\text{反之為 } 0 \end{cases}$

3.3.2 整數規劃模型

當要求解之問題為 MAX（極小化最晚被打通路段的完工時刻）時，如式(3.1)：

$$\text{Minimize } Y = \max_{i \in N} \left\{ \sum_{j=1}^m \sum_{t=0}^{T-1} (t + d_i) \cdot x_{ijt} \cdot w_i \right\} \quad (3.1)$$

當要求解之問題為 SUM（極小化各路段搶通前之等待時間總和）時，如式(3.2)：

$$\text{Minimize } Y = \sum_{i=1}^n \sum_{j=1}^m \sum_{t=0}^{T-1} (t + d_i) \cdot x_{ijt} \cdot w_i \quad (3.2)$$

限制式

式(3.3)係定義個別路段開始執行時刻之限制式，使每路段 i 在所有工作隊、所有時刻中只有一個開始執行的時刻。

$$\sum_{j=1}^m \sum_{t=0}^{T-1} x_{ijt} = 1, \forall i \in N \quad (3.3)$$

式(3.4)為個別工作隊之限制式，使每個工作隊 j 在每個時刻中只能執行一項路段修復工作，可將每個時刻 t 的執行機會視為一種資源，其上限為 1；另外也確保當一條路段由該工作隊開始執行後，該資源由開始執行時刻至完成時刻將無法被共用。

$$\sum_{i=1}^n \sum_{s=\max\{0, t-d_i\}}^{t-1} x_{ijs} \leq 1, \forall j \in M, t = 1, \dots, T \quad (3.4)$$

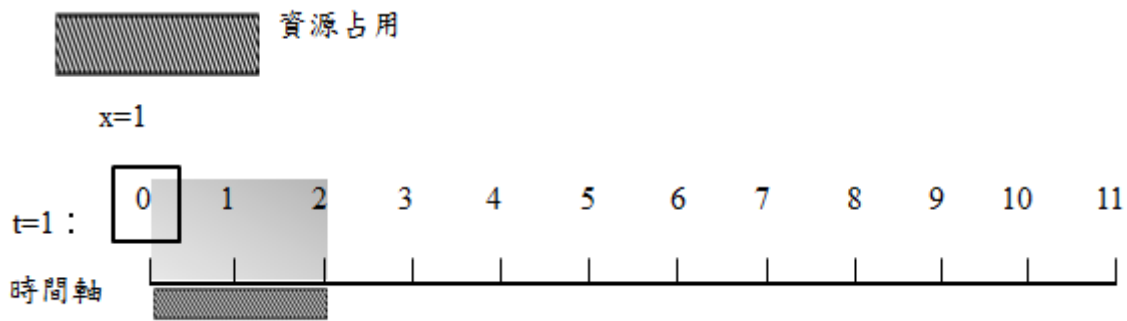


圖 3.2 資源占用說明

在本研究中，假設 $x_{110} = 1$ ，代表路段 1 由工作隊 1 在 $t=0$ 的時刻開始搶修，如果修復時間為 2，也就是 $d_i = 2$ ，則工作隊占用情形如圖 3.2 所示，代表從時刻 $t = 0$ 開始占用，至時刻 $t = 2$ 釋放該工作隊的使用。

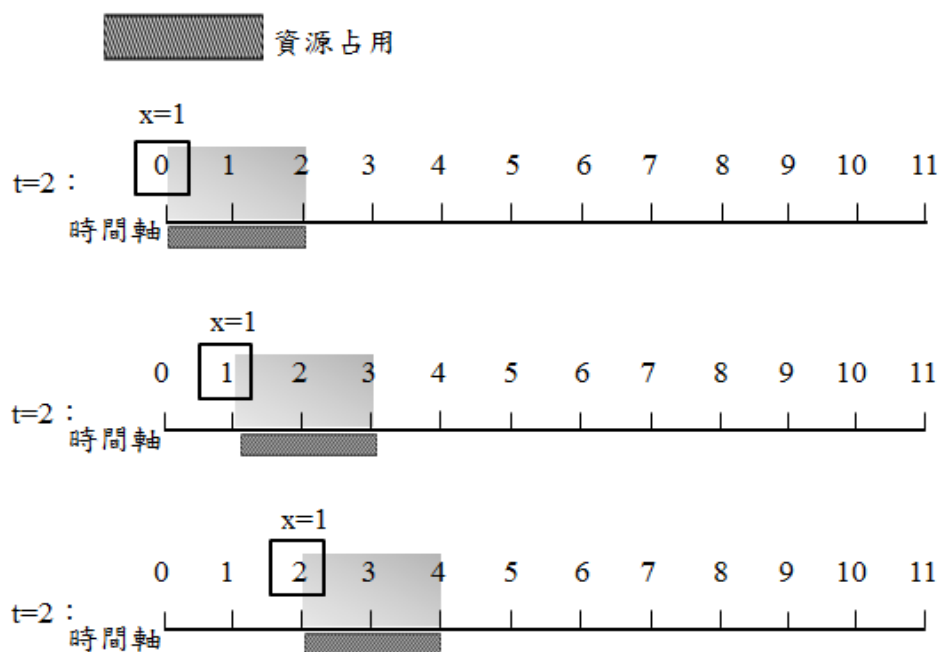


圖 3.3 時刻檢查資源占用機制

同上，因此式(3.4)之時刻 t 係用於該時刻下是否有工作隊占用之情形，同樣以路段 1 由工作隊 1 實施搶修，如果要檢查時刻 $t=2$ 時是否有占用情形，則以圖 3.3 所示，不應該考慮 $x_{110} = 1$ 的變數，因為從時刻 $t=0$ 開始搶修，到時刻 $t=2$ 時已釋放資源，故只須考慮 x_{111} 及 x_{112} 這兩個變數，並限制其 $x_{111} + x_{112} \leq 1$ 。

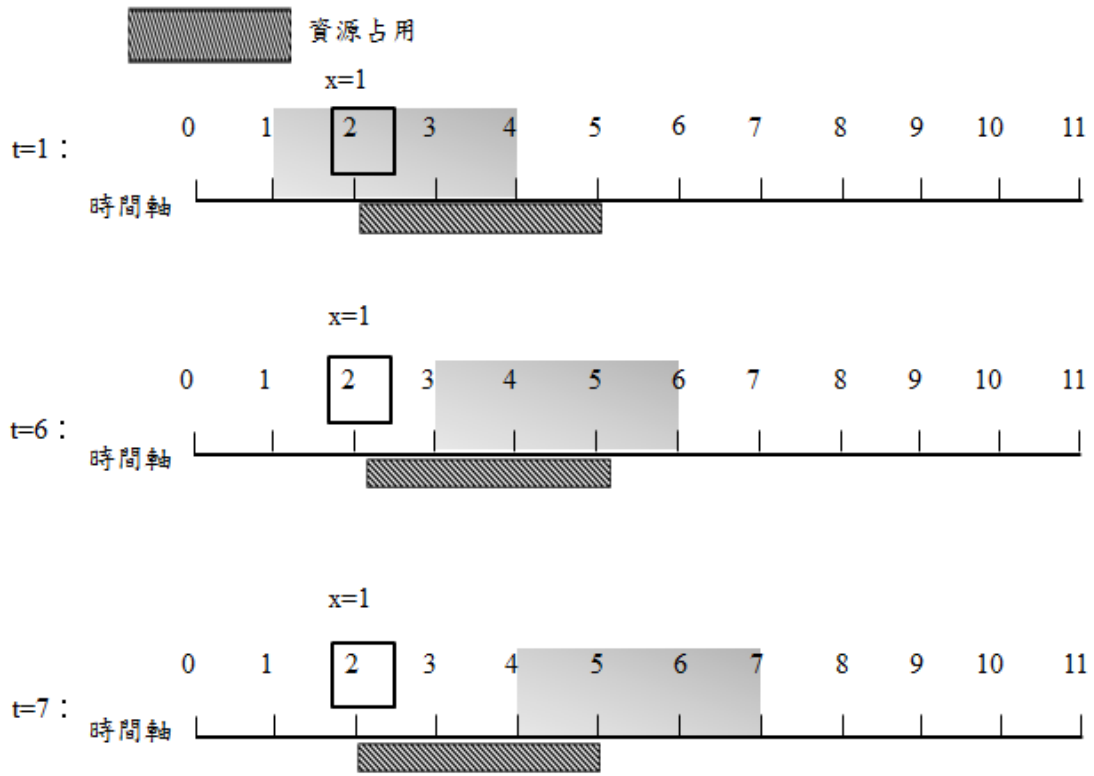


圖 3.4 時刻檢查與資源占用範例圖

再次舉例說明，假設有一搶修作業，所需耗費修復時間為 3，於時刻 2 開始搶修，因此代表此作業會占用一組工作隊，而被占用情況可如圖黑色斜紋長條圖所示。若欲檢查時刻 t 下工作隊被作業占用之情形，則須檢查該作業之開始時間是否介於 $Range_t = [\max\{0, t - d_i\}, t - 1]$ 。圖 3.4 分別以三種情況作為範例討論，當時刻 $t=4$ 時，需檢查灰色長條圖，也就是 $Range_4$ 之範圍，可發現該作業開始時刻係位於 $Range_4$ 內，表示時刻 $t=4$ 時，該作業占用到工作隊；當時刻分別為 $t=6$ 及 $t=7$ 時，其作業開始時刻並未位於 $Range_6$ 及 $Range_7$ 內，代表該作業未占用到工作隊，且因時刻 $t=6$ 為搶修作業結束時刻，其占用之工作隊於該時刻下，可立即提供給其他作業使用。

式(3.5)係定義路段修復前後關係，假設在某條自根節點發散出去之分枝上先經過災點 i 再經過災點 i' ，則較尾端的路段（進入災點 i' ）之開始修復時刻均會大於或等於較前端路段（進入災點 i ）之完成修復時刻（其開始修復時刻加上維修時間）。

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{t=0}^{T-1} (t + d_i) x_{ijt} \leq \sum_{i'=1}^n \sum_{j=1}^m \sum_{t=0}^{T-1} t x_{i'jt}, \forall i \text{ before } i' \quad (3.5)$$

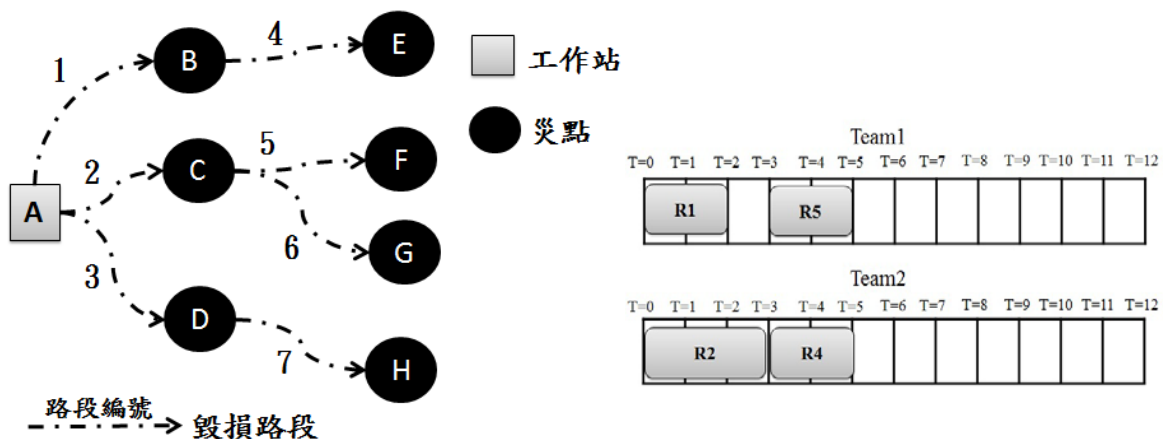


圖 3.5 災區路段與工作隊搶修排程圖

因本研究係將路段視為工件，而工作隊視為機台，但考慮到實際上路網之搶修時序性，須設定後面修復的路段開始時刻均會大於或等於前一條路之修復完工時刻之限制，如圖 3.5 所示，假設 Team1 在時刻 $t=2$ 時閒置，照前述應立即將路段 5 排入到 Team1 搶修，但如同路網所示，路段 5 須等待路段 2 搶修完畢後方可開始，故應等待到時刻 $t=3$ 時才可啟動。

式(3.6)係定義於該時刻，任一路段由任一工作隊執行時，從開始到結束所占用之可用資源不可大於該資源在任一時刻之上限。

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{s=\max\{0, t-d_i\}}^{t-1} r_i \cdot x_{ijt} \leq R, t = 1, \dots, T \quad (3.6)$$

3.4 啟發式演算法

有鑑於路網規模越複雜，以整數規劃模型求解最佳解經常耗時甚久，故本節闡述本研究發展之三類貪婪式演算法及遺傳式演算法。首先於 3.4.1 節將以實際範例說明三類貪婪式演算法之差異；而於 3.4.2 節至 3.4.5 節則分別敘述此四類啟發式演算法細部流程步驟。

3.4.1 範例說明

本研究所發展之貪婪式演算法計有「均勻選取式」、「比例選取式」及「鄰近搜尋式」等三類，以下將以圖 3.6 說明各演算法具體作法差異：

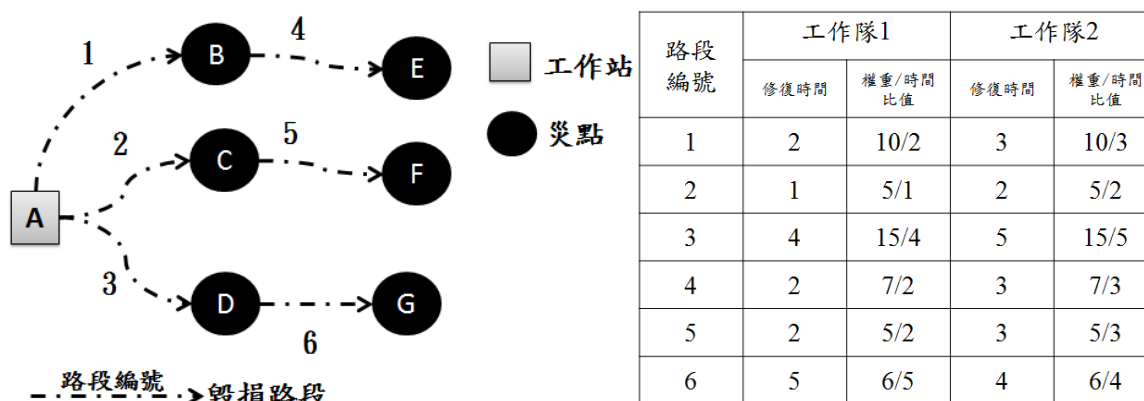


圖 3.6 道路網-工作隊修復時間與權重比例範例圖 1

【均勻選取式貪婪演算法】以「路段權重/修復時間」之比值最高者優先選取，如遇比值相同者，則以均勻機率隨機選取之。其搶修排程之順序如圖 3.7 所示，而路段 1 及路段 2 因其「權重/修復時間」比值相同，隨機挑選路段 2 修復。

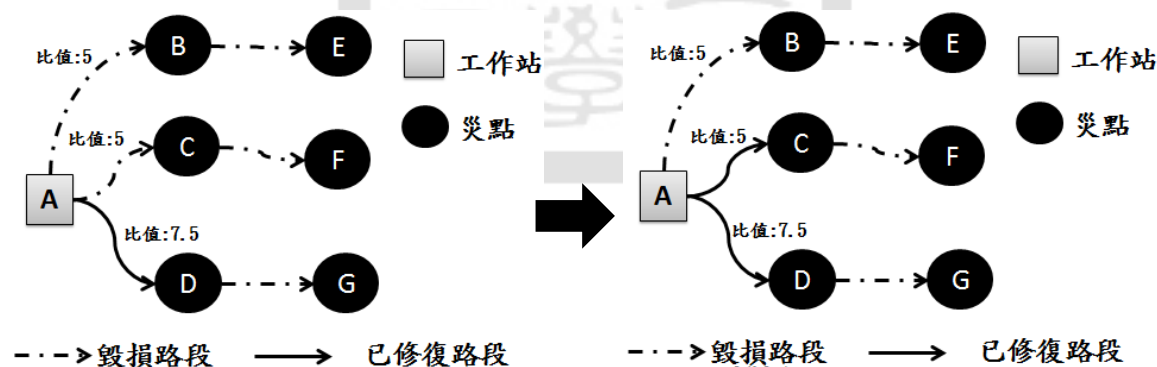


圖 3.7 均勻選取式貪婪演算法作法示意圖

【比例選取式貪婪演算法】將所有可修復路段個別之「路段權重/修復時間」比值除以「全部路段之該比值加總」，依此「正規化」後之比例當成其被選取之機率，再隨機挑選路段進行修復，此演算法旨在仍然保留「路段權重/修復時間」比值較高者優先被選取的貪婪原則，但較先前的均勻式選取法更有彈性，因為均勻式選取法永遠會挑到比值較高之路段進行修復，但比例式選取法仍有機會透過機率選到比值較低的路

段來修復，避免造成局部最佳解的問題產生，其搶修排程之順序如圖 3.8 所示。

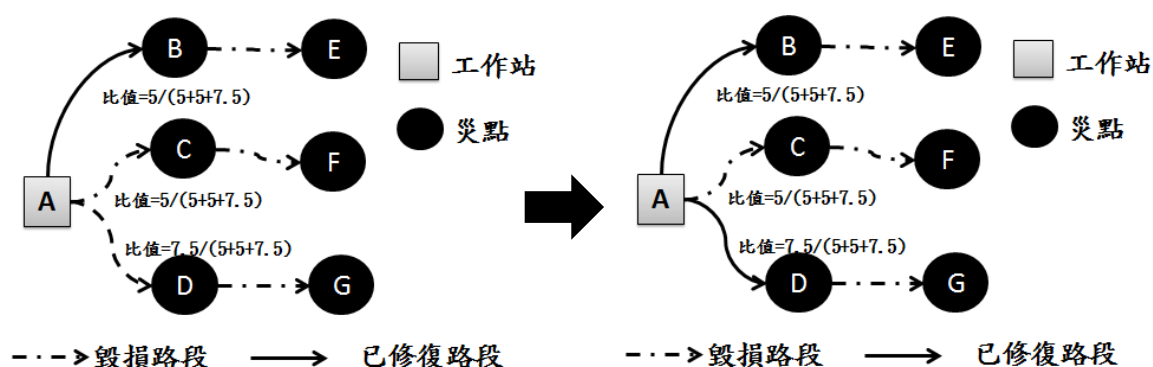


圖 3.8 比例選取式貪婪演算法作法示意圖

【鄰近搜尋式貪婪演算法】以「路段權重/修復時間」之比值最高者優先選取，會設定一個介於[0,1]間的「固定更換機率門檻值」，當每條路段正要被指派工作隊來修復時，將產生一個介於[0,1]間的隨機亂數，當該亂數值大於設定之門檻值時，則更換由其他工作隊修復，因此若該門檻值越小將越可能更換原先由貪婪法選取之排程決策，如圖 3.9 所示。

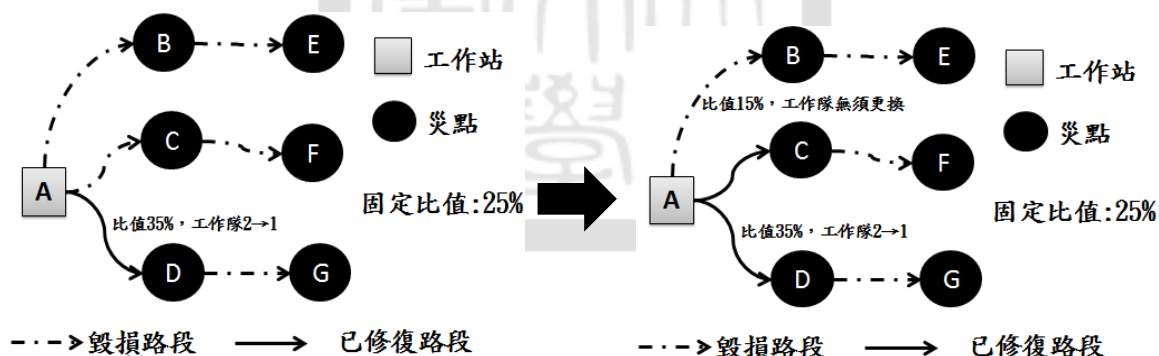


圖 3.9 鄰近搜尋式貪婪演算法作法示意圖

3.4.2 均勻選取式貪婪演算法細部步驟

該演算法之每一個步驟只求局部最佳解，僅計算當前可修復路段之權重比值最高者，如遇比值相同者，則再以平均機率選取。假設有一路網與修復時間如圖 3.10 所示，本研究使用之主要步驟如下：

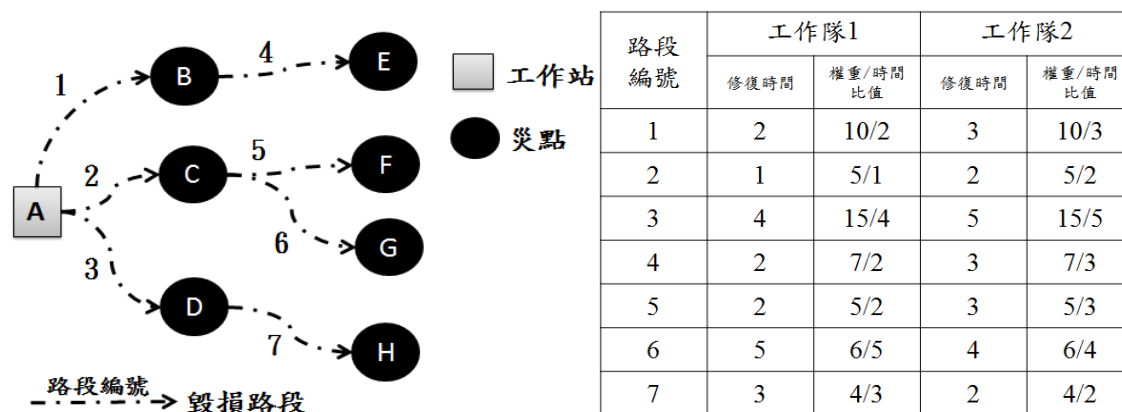


圖 3.10 道路網-工作隊修復時間與權重比例範例圖 2

【步驟一】因本研究之道路網為單根樹狀網路圖，如圖 3.10 所示，此為災後道路網分佈，以起點開始搜尋可修復的路段，並以陣列 $A=\{\text{arc1}, \text{arc2}, \text{arc3}\}$ 存放，陣列 $B=\{\text{arc4}, \text{arc5}, \text{arc6}, \text{arc7}\}$ 則存放尚不可修復之路段，陣列 $C=\{\}$ 則存放已完成修復之路段。

【步驟二】計算陣列 A 中可修復路段之權重/修復時間比值，如圖 3.6 所示，以三條路段兩個工作隊共六個比值組合進行比較，取比值最大者優先進行指派，如遇比例相同，則以平均機率隨機指派，因此可知工作隊 A 對 arc1 及 arc2 比值最大，假設將工作隊 A 指派給 arc1，則接著比較 arc2 及 arc3 對於工作隊 B 的時間權重比值，則可得須將 arc3 指派給工作隊 B 搶修。

【步驟三】假設 arc1 及 arc3 已搶修完畢，則同步更新陣列 A、B 及 C 內資料，此時陣列 C 為 $\{\text{arc1}, \text{arc3}\}$ ，而陣列 A 中因 arc1 及 arc3 已搶修完畢，則代表 arc4 及 arc7 可加入考量，則陣列 A 為 $\{\text{arc2}, \text{arc4}, \text{arc7}\}$ ，陣列 B 為 $\{\text{arc5}, \text{arc6}\}$ ，如此重複步驟二及三之動作，直至陣列 A 及 B 均已無路段可搶修，程式方執行完畢。

3.4.3 比例選取式貪婪演算法細部步驟

該演算之發展目的係為打破均勻選取式貪婪演算法過度注重當下權重比值最高者必被選取的貪婪法則可能造成局部最佳解的問題，故當有工作隊閒置時，便將當前可修復之個別「路段權重/修復時間」比值再除以全數路段之該比值加總，以佔有比例隨機挑選，使得權重比例較低者亦有一定機率可能被選取進行修復。

【步驟一】如圖 3.10 所示，此為災後道路網分佈，與各路段權重及對各工作隊之修復時間，以起點開始搜尋可修復的路段，並以陣列 $A=\{\text{arc1}, \text{arc2}, \text{arc3}\}$ 存放，陣列 $B=\{\text{arc4}, \text{arc5}, \text{arc6}, \text{arc7}\}$ 則存放尚不可修復之路段，陣列 $C=\{\}$ 則存放已完成修復之路段。

【步驟二】計算陣列 A 中可修復路段之「權重/修復時間」比值，如圖 3.6 所示，以三條路段兩個工作隊共六個比值組合進行比較，在此處以 P_{1A} 代表工作隊 A 修復路段編號 1 之權重/修復時間比值，因此可知有 P_{1A} 、 P_{1B} 、 P_{2A} 、 P_{2B} 、 P_{3A} 及 P_{3B} 共六種組合，計算每一情況之權重時間比值，再除以全部路段之該比值加總，以 $W_{1A}=P_{1A}/(P_{1A}+P_{1B}+P_{2A}+P_{2B}+P_{3A}+P_{3B})$ ，分別計算 W_{1B} 、 W_{2A} 、 W_{2B} 、 W_{3A} 及 W_{3B} 之比例，再按照比值大小隨機選取工作隊及其修復路段，以此方法不由比例大者優先修復，藉由設定執行次數以求解數據是否較接近最佳解。

【步驟三】假設經過上述方法，arc1 及 arc3 已搶修完畢，則同步更新陣列 A、B 及 C 內資料，此時陣列 C 為 $\{\text{arc1}, \text{arc3}\}$ ，而陣列 A 中因 arc1 及 arc3 已搶修完畢，則代表 arc4 及 arc7 可加入考量，則陣列 A 為 $\{\text{arc2}, \text{arc4}, \text{arc7}\}$ ，陣列 B 為 $\{\text{arc5}, \text{arc6}\}$ ，如此重複步驟二及三之動作，直至陣列 A 及 B 均已無路段可搶修，程式方執行完畢。

3.4.4 鄰近搜尋式貪婪演算法細部步驟

該演算法選取路段修復之方式與均勻選取式相同，以「權重/修復時間」比值最高者優先選取，如遇比值相同，則平均機率隨機選取。惟此處加入「交換機制」藉以調整工作隊修復路段之順序，以解決局部最佳解問題，於該路段可修復時，隨機產生

一介於[0,1]間之隨機亂數，當該亂數大於當前程式所設定之介於[0,1]間的「固定更換機率門檻值」時，即改由其他工作隊進行修復。

【步驟一】如圖 3.10 所示，此為災後道路網分佈，與各路段權重及對各工作隊之修復時間，以起點開始搜尋可修復的路段，並以陣列 $A=\{\text{arc1}, \text{arc2}, \text{arc3}\}$ 存放，陣列 $B=\{\text{arc4}, \text{arc5}, \text{arc6}, \text{arc7}\}$ 則存放尚不可修復之路段，陣列 $C=\{\}$ 則存放已完成修復之路段。

【步驟二】計算陣列 A 中可修復路段之「權重/修復時間」之比值，如圖 3.6 所示，以三條路段兩個工作隊共六個比值組合進行比較，取比值最大者優先進行指派，如遇比值相同，則以同樣機率隨機指派，因此可知工作隊 A 對 arc1 及 arc2 比值最大，假設將工作隊 A 指派給 arc1，惟此時加入可能不依貪婪法則選取之機制，於每條路段指派給工作隊時隨機產生一介於[0,1]之隨機亂數，以決定是否需要指派給其餘工作隊，假設演算法初始設定「固定更換機率門檻值」為 25%，若 arc1 隨機產生之亂數為 35% (>25%)，則將其隨機調換指派給非原工作隊，藉由設定執行次數、更換比例及更換次數以求解數據是否較接近最佳解。

【步驟三】假設 arc1 及 arc3 已搶修完畢，則同步更新陣列 A、B 及 C 內資料，此時陣列 C 為{arc1, arc3}，而陣列 A 中因 arc1 及 arc3 已搶修完畢，則代表 arc4 及 arc7 可加入考量，則陣列 A 為{arc2, arc4, arc7}，陣列 B 為{arc5, arc6}，如此重複步驟二及三之動作，直至陣列 A 及 B 均已無路段可搶修並達到程式所設定之條件，程式方執行完畢。

3.4.5 基因演算法

因貪婪演算法僅為單次性且易落入局部最佳解，與本研究所提出之平台機台搶修模式所求解之最佳解差異甚大，故以具疊代性之基因演算法持續收斂至更佳之解。

基因演算法有三組基本程序需遵循，第一個程序為保留機制，係基於達爾文之「適者生存」的理論發展而成，保留較佳群體藉以提升族群基因優劣；其次為選擇機制，

根據達爾文之「物競天擇」理論，以適應環境能力較佳之染色體，有更高機率被選擇去繁衍下一代，也就是基因染色體被選擇之機率；最後為基因多樣性，係源於交配及突變機制，避免產生局部最佳解之情況。Michalewicz (2013)整理之基本步驟及功能簡介，分述如下：

1. 編碼：每一染色體以一組或一個基因表示，基因數目代表求解問題之參數數目。

編碼架構設計之優劣，長決定性影響求解效率。而編碼主要區分三種形式：

- (1) 二元編碼 (Binary coded genetic algorithm): 以二進位字串 0、1 表示決策變數，屬於傳統之編碼方式。

- (2) 實數編碼 (Real coded genetic algorithm): 與上述編碼原則差異在於，二元編碼需透過編碼與解碼之過程，而此形式以實數表示決策變數，或可優化二元編碼之效率且較適用於連續型最佳化問題。

- (3) 符號編碼：以符號表示決策變數。

2. 初始母體產出 (Initial population): 產出任意數量之初始解，後再進行演算搜尋。產生方式可分為特定或隨機的方式。

3. 適應性函數 (Fitness function): 用於評估染色體對於環境之適應值。假如個體適應力越強，代表在環境中存活的機率越高，則越有可能被選擇繁衍下一代。求解問題前，先將目標函數轉換為適應函數，以目標函數判斷解答接近目標程度，譬如目標函數值若越高則代表解答越接近最後目標，也更有機會進一步再次搜尋出更佳之解答。

4. 藉由基因運算子 (Genetic operator)產出子代：

常見基因運算子為以下三種：

- (1) 複製 (reproduction): 主要為保留策略及選擇機制。保留策略係於任意數目之母體中搜尋較佳適應值之染色體，並保留至下一代，此功能主要根據達爾文「適者生存」原則建立，適應函數值較高的個體有較高之機率保存至下一世代族群，常見為菁英策略 (Elitist Strategy)；第二為選擇機制，主要來自於達

爾文「物競天擇」之概念，主要之選擇機制有以下三個：

- i. 輪盤法選擇 (Roulette Wheel Selection)：如圖 3.11 所示，於演化過程中將每一個染色體適應值加總，總適應值則以整個輪盤表示，再依照各染色體適應值分配至輪盤上所佔區域，區域越大將擁有較高機率放入交配池，其主要目的可使擁有較佳適應值染色體有較高之機率繁衍下一代；而缺點則為當某一個染色體之適應值遠遠大於其他染色體時，其重複被選擇之機率增加，將導致收斂成局部最佳解。

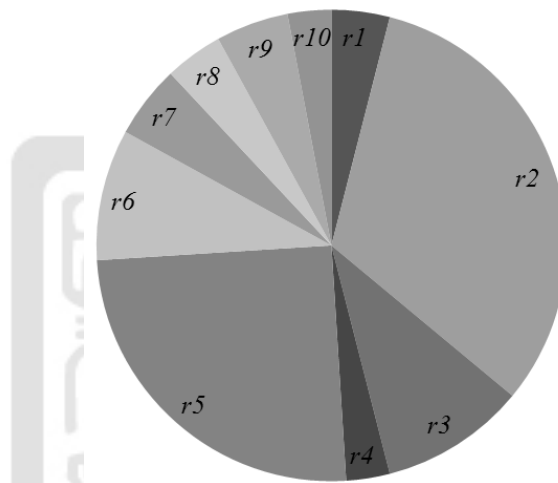


圖 3.11 輪盤法選擇 (Roulette Wheel Selection)

- ii. 排列機制 (Rank-Based Mechanism)：將全部染色體之適應值列表並且排序，適應值越高則排序越前面；染色體被選擇之機率是根據族群中適應值排序決定，可平衡每一個染色體被選擇之機率且改善輪盤法缺點。
 - iii. 競爭式選擇 (Tournament Selection)：每次實施競爭之數量是決定最終結果的最重要因素。最簡單的方法為隨機選擇兩條染色體並比較適應值，較高者可放入交配池中，而競爭數量越大，則越容易尋找到較佳解，如此反覆進行此動作直至達到交配量為止。
- (2) 交配(crossover)：從交配池內隨機挑選兩組染色體配對，將部份資訊交換並交配出子代，其目的在於產生新解，透過這種方式重新編成個體基因組合，並

擴展搜尋空間，而子代之染色體同樣將會一部份的遺傳父代染色體優點。交配方式主要區分為四種：

- i. 單點交配(One-Point Crossover)：於染色體內隨機選取一點，保留前段基因，再將後段基因與另一染色體之後段基因結合。如圖 3.12 所示，假設隨機將染色體 A 及 B 隨機以基因 5 為基準，對後段基因進行交配。

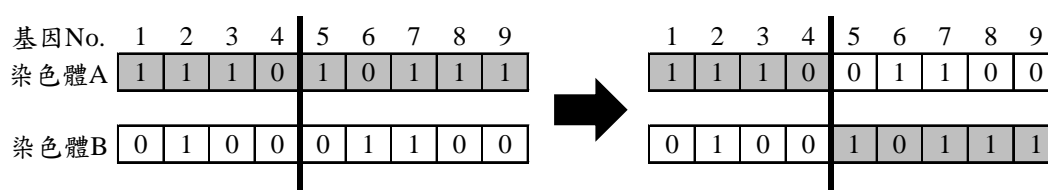


圖 3.12 單點交配

- ii. 雙點交配(Two-point Crossover)：於染色體內隨機選定兩點，保存前段與後段之基因，再與另一染色體之中段基因實施調換成為新染色體。如圖 3.13 所示，假設染色體 A 及 B 隨機以基因 3 與基因 5 為基準，將介於基因 3 與 5 之中段基因進行交換。

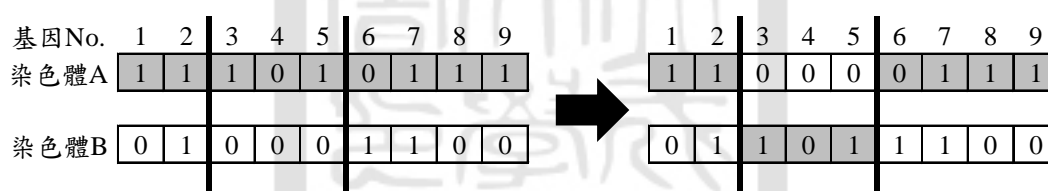


圖 3.13 雙點交配

- iii. 多點交配(Multi-Point Crossover)：類似雙點交配之方式，於染色體上隨機選取兩個以上之交配點，再與另一染色體基因實施交換變成新的染色體。如圖 3.14 所示，在染色體 A 及 B 隨機選擇四個點，各為基因 2、3、6 及 8。將介於基因 2 至 3 及介於基因 6 至 8 的基因段互相交換。

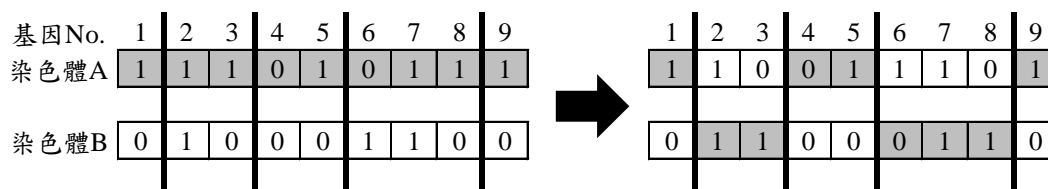


圖 3.14 多點交配

- iv. 均勻交配(Uniform Crossover)：隨機產出 0, 1 字串，假設染色體之基因為 1，則代表將基因都遺傳至第一個子代；為 0 時，則遺傳至第二個子代。如圖 3.15 所示，隨機產生(0, 1, 1, 1, 0, 1, 0, 0, 1)後進行基因交換。

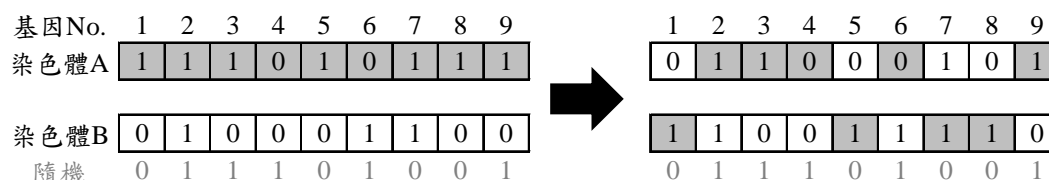


圖 3.15 均勻交配

- (3) 突變(mutation)：為隨機發生之動作，功能在於藉由隨機改變在染色體內之基因，導入未曾使用過之基因型態，以增加改變母體變異程度，可使在求解之過程中能尋找到新的解，避免在複製及交配過程中，因遺漏重要之基因訊息，而陷入局部最佳解。
- (4) 停止條件：因演化過程皆為迴圈，故須先設定停止條件。通常以時間成本與解答之收斂程度為主，而時間成本通常會設定固定值；而收斂程度則判斷當多少世代後染色體適應值函數均趨近一致後，便可停止演化。

以下將以 Wang et al. (2015)運用之基因演算法作為基礎，進而發展本研究所使用之步驟：

指派順序	1	2	3	4	5	6	7	8	9	10
路段編號	2	3	1	5	4	7	8	9	6	10
工作隊	2	1	3	1	3	3	2	1	1	2

圖 3.16 染色體編碼示意圖

【步驟一】進行染色體編碼，如圖 3.16 所示，染色體基因包括路段編號(如路段編號 2、3……)與各工作隊(Machine1、2 及 3)之修復關係，以工作隊 1 之排程計畫為範例，即自起點出發後從路段 3 搶修，依序為路段編號 5、9 及 6，餘工作隊同義類推，本研究是以路段修復結束時刻作為染色體編碼之排序基準。因此對於有 n 個路段與 m 個工作隊搶修之排程而言，染色體長度均為 n 個路段基因，亦即為 n 個位元。

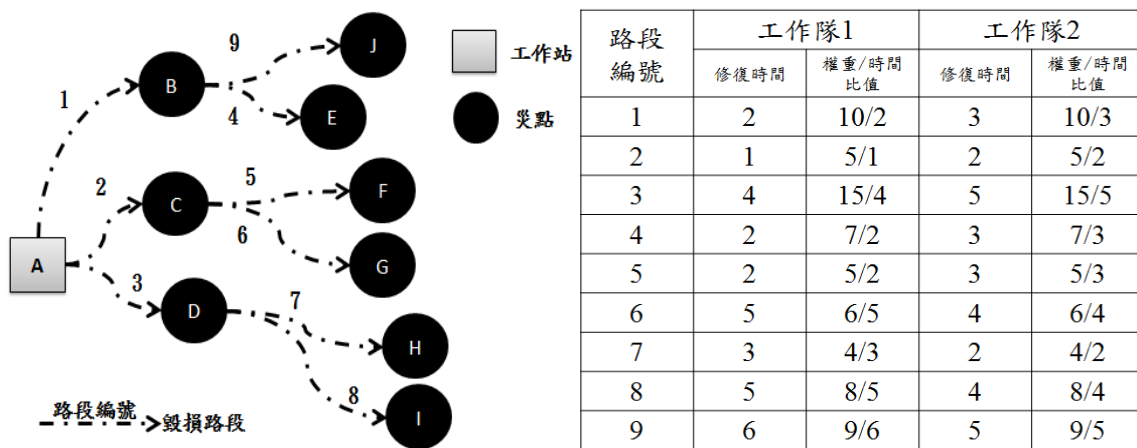


圖 3.17 災後路網圖與工作隊-路段修復時間及權重比例

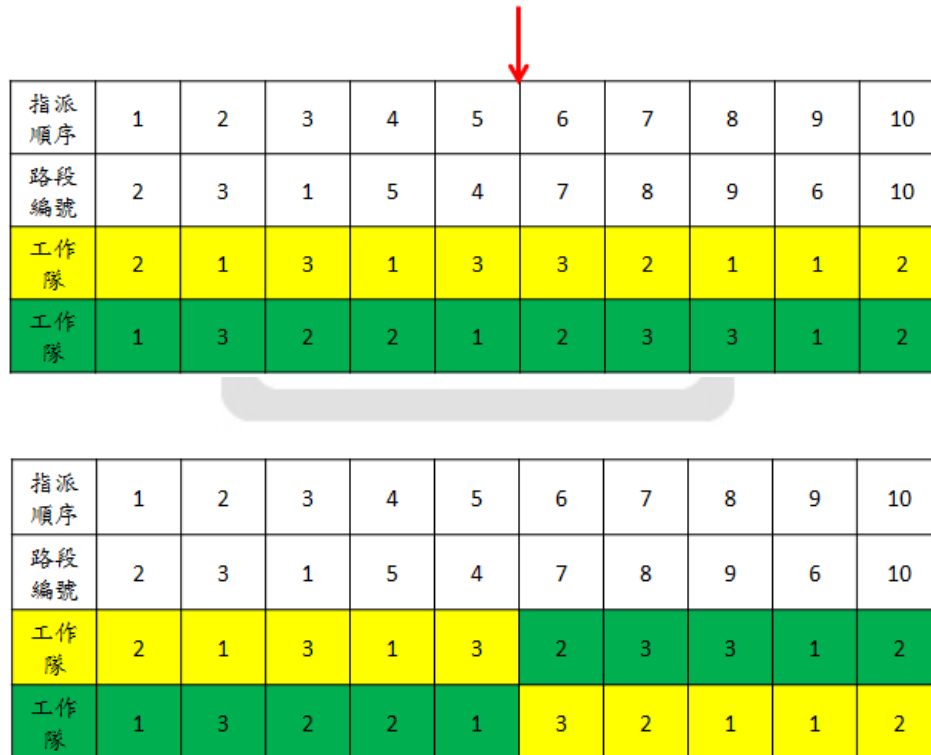
【步驟二】本研究起初是以貪婪演算法之最佳解作為基因演算法之染色體編碼原則，但依據編碼原則進行複製、交配及突變後，因為貪婪演算法屬於局部最佳解，並未對路網作通盤考量，且編碼後路段順序固定，因此基因演算法求解之值並非最佳，故在此以 3.4.2 節之均勻選取式貪婪演算法進行求解，期以產出較佳之路段順序。假設有一災後道路網與工作隊-路段修復時間及權重比例如圖 3.17 所示，從起點出發，優先選擇路段編號 1、2 及 3 進行修復，而工作隊有 2 組，在此處以 P_{1A} 代表工作隊 A 修復路段編號 1 之權重/時間比例，因此可知有 P_{1A} 、 P_{1B} 、 P_{2A} 、 P_{2B} 、 P_{3A} 及 P_{3B} 共六種組合，而後使用輪盤法計算比例，以 $W_{1A} = P_{1A} / (P_{1A} + P_{1B} + P_{2A} + P_{2B} + P_{3A} + P_{3B})$ ，分別計算 W_{1B} 、 W_{2A} 、 W_{2B} 、 W_{3A} 及 W_{3B} 之比例，並根據比例隨機決定要修復哪條路段，直至完成全數路段修復並計算適應值，由於是隨機方式，故在程式方面亦可輸入「執行次數」，假設輸入 10000 次，便是執行上述隨機方式 10000 次，每尋找到比之前更佳之適應值便更新，直至次數執行完畢。

指派 順序	1	2	3	4	5	6	7	8	9	10
路段 編號	2	3	1	5	4	7	8	9	6	10
工作 隊	2	1	3	1	3	3	2	1	1	2
工作 隊	1	3	2	2	1	2	3	3	1	2
工作 隊	2	1	1	2	3	1	3	2	3	2
：										

圖 3.18 初始母體示意圖

【步驟三】以上述步驟二產生之最佳解作為染色體之編碼原則，路段修復順序不變動，僅隨機異動工作隊，產生一定數量之初始母體，並同時計算各適應值，如圖 3.18 所示。

【步驟四】設定競爭比例，將初始母體數量*競爭比例之數目，隨機挑選該數量之染色體進行適應值比較，最優者放入交配池中，直至染色體數量達標為止。



指派 順序	1	2	3	4	5	6	7	8	9	10
路段 編號	2	3	1	5	4	7	8	9	6	10
工作 隊	2	1	3	1	3	3	2	1	1	2
工作 隊	1	3	2	2	1	2	3	3	1	2

指派 順序	1	2	3	4	5	6	7	8	9	10
路段 編號	2	3	1	5	4	7	8	9	6	10
工作 隊	2	1	3	1	3	2	3	3	1	2
工作 隊	1	3	2	2	1	3	2	1	1	2

圖 3.19 染色體交配示意圖

【步驟五】隨機自交配池中選取兩組染色體並實施單點交配，如圖 3.19 所示。

指派 順序	1	2	3	4	5	6	7	8	9	10
路段 編號	2	3	1	5	4	7	8	9	6	10
工作 隊	2	1	3	1	3	3	2	1	1	2

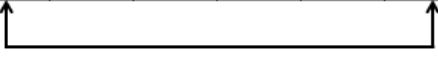


圖 3.20 染色體突變示意圖

【步驟六】隨機挑選某一染色體內之兩個基因，設定當比例高於突變率時便進行突變，直至產生規定數量之染色體，如圖 3.20 所示。

【步驟七】將父代染色體依適應值大小做升冪排列，並依據菁英比例保留父代染色體，將該數量之染色體與子代染色體進行適應值比較，挑選適應值較優之染色體數量進行下一世代之演化。

【步驟八】於本程式設定終止條件，當演化 15 次世代或者當代全數染色體適應值皆為相同時，演算法便終止。

3.5 小結

本章在 3.1 節描述問題與假設，以單根樹狀網路為基礎下制定路段搶修之排程，將路段搶修完畢視為災點之搶通。每個災點之進入路段來源僅有一個，但對外出去之分枝則可有多個，而出發點（根節點）只有往外連通各災點之路段；將問題目標式分為最小化作業總完成時刻(MAX)及最小化各災點等待搶通之時間總和(SUM)兩種

3.2 節條列七項假設前提，藉以限縮本研究可能於實際情況中所遇到之情況，以確保模式考量之完整。3.3 節說明本研究之目標式及限制式，於 3.3.1 節定義模型符號並設定整數規劃模型之限制式與目標式。3.4 節說明本研究所設計之啟發式演算法：均勻選取式、比例選取式及鄰近搜尋式貪婪演算法，再以範例說明三項演算法之差異；之後再解釋本研究使用之基因演算法。

因本研究所使用之貪婪演算法均透過比值(權重/修復時間重)進行比較以決定該選擇哪條路段優先修復，也因此針對同樣比值之例外狀況設計「均勻選取式貪婪演算

法」，其主要作法係以可修復路段之比值最高者先選取，而當遇同比例之路段時，則等機率隨機選取，藉以判斷路段修復的順序。上述作法，有可能造成局部最佳解之問題，故設計出「比例選取式貪婪演算法」，其作法為將單一路段比值除以全數可修復路段之比值加總，以所佔比例方式隨機選取。而為了增加路段修復順序的擾動效果，則設計出「鄰近搜尋式貪婪演算法」，其作法除遵循均勻選取式外，更於程式中加入介於[0,1]間的「固定更換機率門檻值」，當每一條路段選定後則隨機產生一隨機亂數與更換門檻值相比較，當大於門檻值時，便將此路段由原工作隊改由其餘任一工作隊修復。最後則是設計出「基因演算法」，透過染色體編碼、初始母體建置、適應函數計算、複製及交配，藉以產生更佳之可行解。下章將測試本研究之數學模型及啟發式演算法在不同網路結構下的求解表現與比較。



第四章、模型建構與驗證結果分析

為驗證本研究建構之數學模式與求解方法在不同的網路圖下皆能有效操作，本章將以七種不同啟發式演算法測試樹狀網路的表現，並依據路段階層、數量及工作隊區分 9 組測試例。本研究以 Microsoft Visual Studio 2013 為程式語言介面，以 C++ 程式語言撰寫程式，並配合 Gurobi 6.5.2 版求解數學模式；同時在個人電腦執行，測試平台為 Microsoft Windows 10 Professional 作業系統，搭配之處理器及記憶體為 Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 與 16GB RAM。最後，本研究將依據各項啟發式演算法與整數規劃模型進行數據分析，藉以判斷何種方案為最佳。

4.1 輸入資料

本研究測試之網路圖以樹狀為主，以 Microsoft Visual Studio 2013 撰寫路網產生器，並以「路段階層」、「分枝」及「路段數量」來決定路網規模大小以及形狀，以圖 4.1(a)二階災後道路網及(b)三階災後道路網進行說明，只與工作站(起點 A)相連結之災點屬第一階層，而與第一階層災點所相連結之災點則屬第二階層。依此類推，在圖 4.1(a)，路段 1、2 及 3 屬第一階層，而路段 4、5、6 及 7 屬第二階層；在圖 4.1(b)，路段 1、2 及 3 屬第一階層，路段 4、5、6 及 7 屬第二階層，而路段 8 及 9 屬第三階層。在圖 4.2(a)，路段 1 及 4 屬分枝 A，路段 2 及 5 屬分枝 B，路段 2 及 6 屬分枝 C，路段 3 及 7 屬分枝 D；在圖 4.2(b)，路段 1、4 及 8 屬分枝 A，路段 2 及 5 屬分枝 B，路段 2 及 6 屬分枝 C，路段 3、7 及 9 屬分枝 D。在圖 4.3(a)有路段 1 至 7 共 7 條；在圖 4.3(b)有路段 1 至 9 共 9 條，則依上述三項設定因素可分別形成圖 4.1 二階層、分枝 4 條與路段數量 7 條，以及圖 4.3 三階層、分枝 4 條與路段數量 9 條之災後道路網。

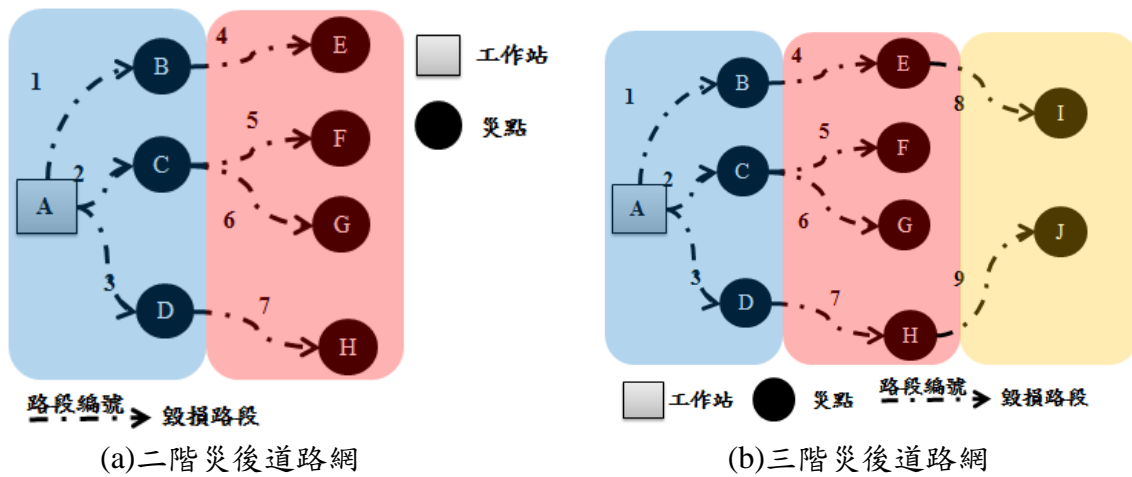


圖 4.1 災後道路網(以階層表示)

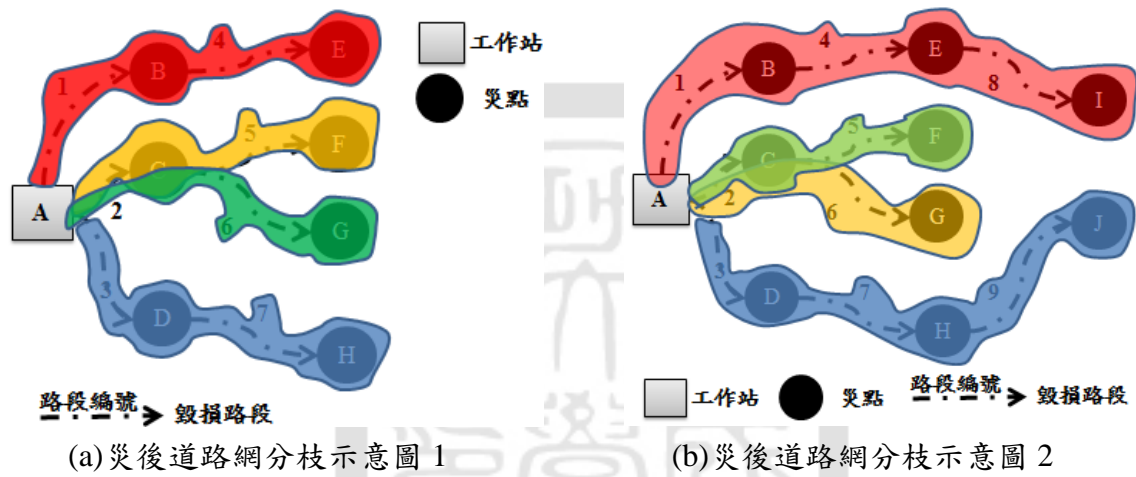


圖 4.2 災後道路網(以分枝表示)

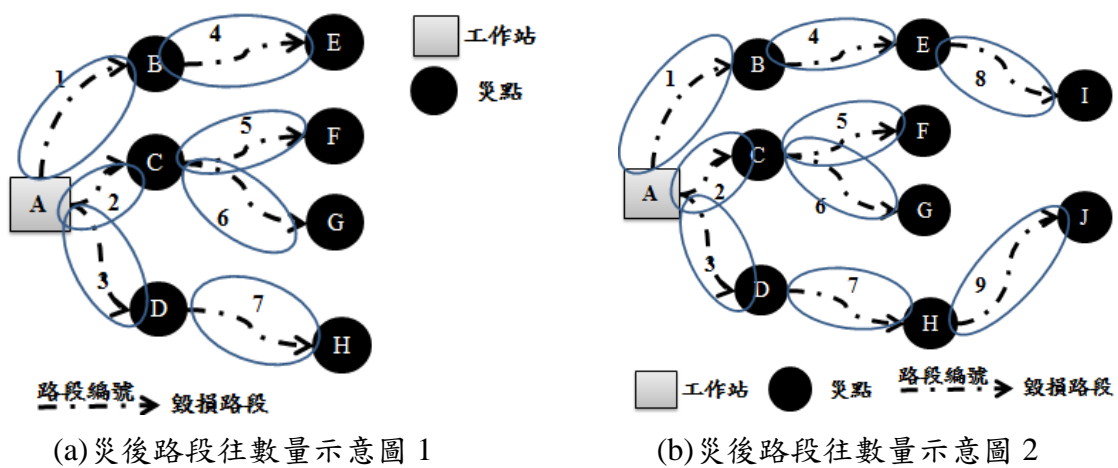


圖 4.3 災後道路網(以數量表示)

所使用之路網產生器如圖 4.4 所示，並透過「L(路段階層)」等 8 個參數設定，依圖 4.2 路網產生明細將網路圖 1 至 9，分別以工作隊及資源上限組合 3 組*路段修復時間組合 3 組*路段資源組合 2 組*網路圖 9 組*各網路圖各 10 組圖檔，據以產生 $3*3*2*9*10=1,620$ 個圖檔進而實施數據測試與分析，相關設定明細詳如表 4.1。

圖 4.4 路網產生器

表 4.1 路網產生明細

	L(階層)	M(道路數量)	R(工作隊數量), E(總資源上限)	T(路段修復時間)	E(各路段資源 需求量)	W(路段權重)
網路圖 1	2	16	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 2	4	16	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 3	8	16	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 4	3	36	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 5	6	36	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 6	12	36	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 7	4	64	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 8	8	64	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	
網路圖 9	16	64	2, 30	1, 1~3, 1~6	5~10, 12~15	5~10
			4, 40	1, 1~3, 1~6	5~10, 12~15	
			8, 120	1, 1~3, 1~6	5~10, 12~15	

本研究測試例之路網產生器設定值如圖 4.5(a)所示，將產生之圖檔明細為圖 4.5(b)，第一列由左至右分別為路段分枝數量、路段數量、工作隊數量及當前時刻之資源上限，第二列至第四列代表分枝數量：第二列由左至右分別代表路段數量及路段編號；第三列第二欄為路段編號 2，第三欄為路段編號 4，而此分枝之順序則由路段編號 2 至 4，也因此工作隊修復順序需依此規則實施；第 5 列至第 8 列代表路段編號 1 至 4 之修復時間及使用資源，由左至右依序為該路段由工作隊 A、工作隊 B 及工作隊 C 負責之修復時間，其次為由工作隊 A、工作隊 B 及工作隊 C 負責所使用之資源數量。

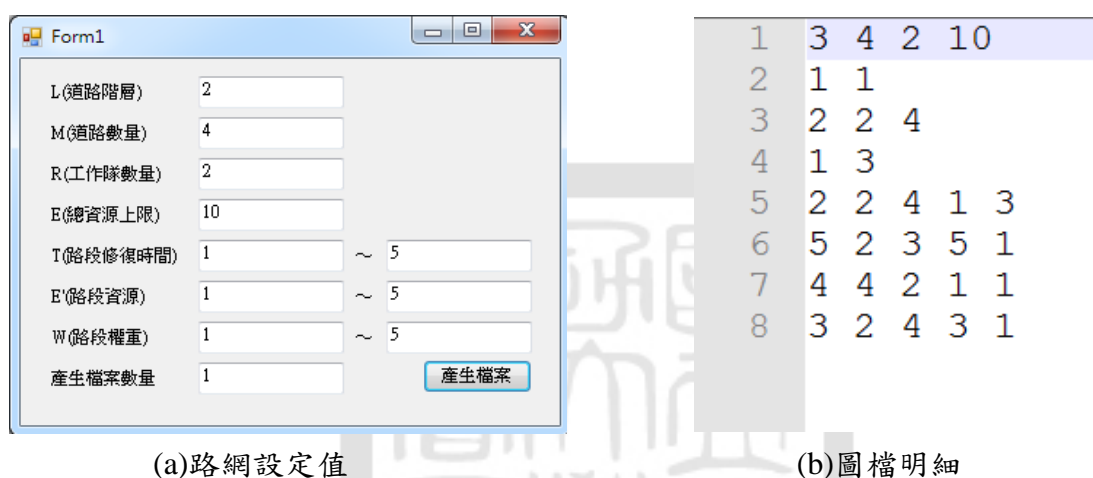


圖 4.5 路網產生器設定值暨圖檔產生明細示意圖

4.2 啟發式演算法比較

此小節將測試不同路網結構在不同演算法下的求解表現。以 1,620 個圖檔為實驗標的，依據 3.4.2 節之均勻選取式貪婪演算法(以下簡稱為 GD)執行 1 次、3.4.3 節之比例選取式貪婪演算法(以下簡稱為 GD_r)執行 50,000 次、3.4.4 節之鄰近搜尋貪婪演算法分別以更換比例為 25%(以下簡稱為 GR1)、50%(以下簡稱為 GR2)、75%(以下簡稱為 GR3)，未設定執行更換上限並執行 50,000 次，以及 3.4.5 節之基因演算法(以下簡稱為 GA)設定競爭比例 20%、突變率 10%及菁英比例 20%共執行 50,000 次世代，依所述之步驟求解各項數據結果。表 4.2 及 4.3 分別為問題目標式 MAX 及 SUM 的測試結果，每個演算法均以 1,620 個圖檔進行測試，並以# *opt* 代表在該例中求得最佳解的次數。由測試結果來看，當路網規模增大，以路段數量 16(A16)依序至路段數量

36(A36)及路段數量 64(A64)之# *opt* 最佳解之數量會隨著遞減。因此可判斷出真正影響最佳解數量的因素為路段數量，而非路段階層。就 6 種啟發式演算法之求解結果表現，在 MAX 問題中以 GDr 及 GA 最好，在 SUM 問題中則以 GR1 及 GR2 最好，故無法藉此種方法判定何者較優。

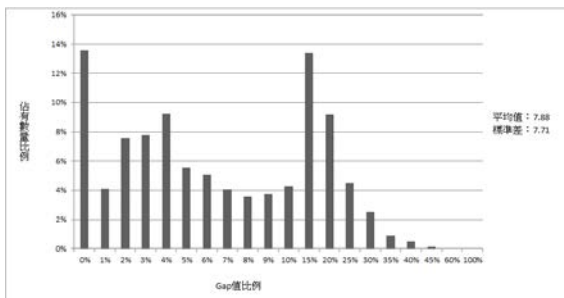
表 4.2 不同演算法在不同網路結構下的求解表現(MAX)

路網結構	測試個數	GD	GDr	GR1	GR2	GR3	GA
		# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>
L2A16	180	69	109	61	69	69	85
L4A16	180	74	124	66	67	74	121
L8A16	180	96	140	96	95	85	139
A16	540	239	373	223	231	228	345
L3A36	180	47	60	54	56	52	50
L6A36	180	44	61	43	41	40	53
L12A36	180	54	71	51	51	51	69
A36	540	145	192	148	148	143	172
L4A64	180	29	44	34	32	40	31
L8A64	180	18	51	20	22	23	43
L16A64	180	35	52	38	34	37	48
A64	540	82	147	92	88	100	122
各演算法加總		466	712	463	467	471	639
備註：L 代表路段階層；A 代表路段數量							

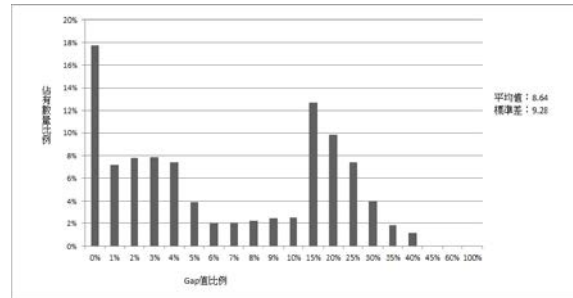
表 4.3 不同演算法在不同網路結構下的求解表現(SUM)

路網結構	測試個數	GD	GDr	GR1	GR2	GR3	GA
		# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>	# <i>opt</i>
L2A16	180	45	29	97	97	97	16
L4A16	180	39	65	65	65	65	69
L8A16	180	79	113	100	100	84	112
A16	540	163	207	262	262	246	197
L3A36	180	0	0	3	2	3	0
L6A36	180	10	20	17	17	18	20
L12A36	180	30	40	55	55	55	40
A36	540	40	60	75	74	76	60
L4A64	180	0	0	0	0	0	0
L8A64	180	0	0	0	0	0	0
L16A64	180	17	20	34	34	34	20
A64	540	17	20	34	34	34	20
各演算法加總		220	287	371	370	356	277
備註：L 代表路段階層；A 代表路段數量							

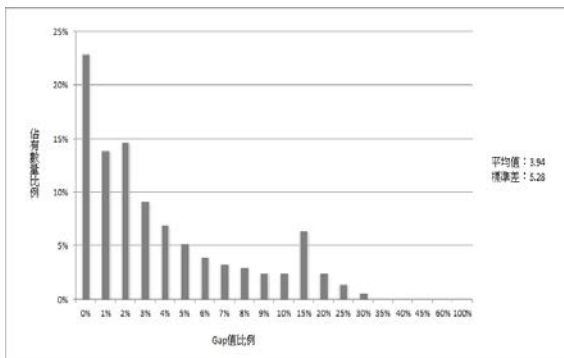
而圖 4.6(a)至(f)分別為問題目標式 SUM 分別由 6 種演算法執行 1,620 個路網圖檔後所計算之 Gap 值分佈圖，可依此計算其 Gap 平均值與標準差。



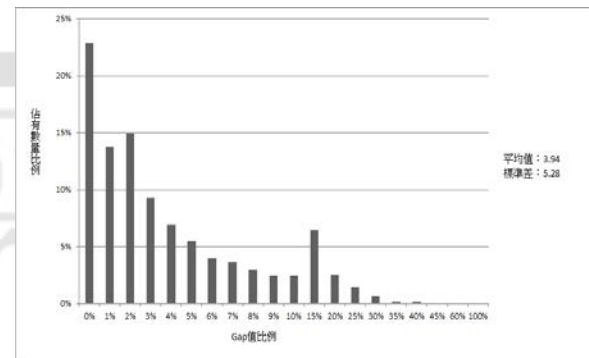
(a) GD 直線圖



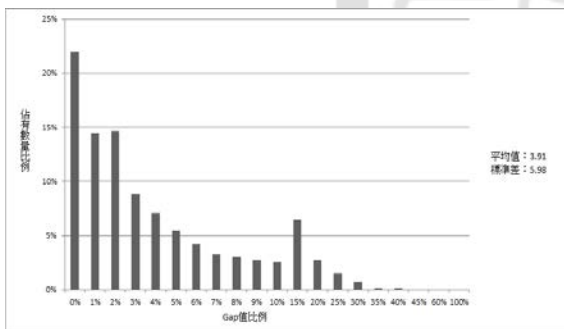
(b) GD_r 直線圖



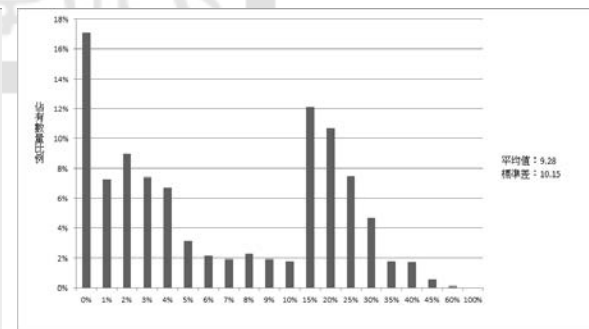
(c)GR1 直線圖



(d)GR2 直線圖



(e)GR3 直線圖



(f)GA 直線圖

圖 4.6 各啟發式演算法直線圖

由測試結果來看，在問題目標式 SUM 的求解結果中，以 GR1、GR2 及 GR3 之標準差 5.28~5.98 為最低，而計算 Gap 值小於 5%之測資比例，由大至小分別為 GR1=73%、GR2=73%、GR3=72%、GD_r=52%、GA=51%及 GD=48%，故以趨近於最

佳解的比例來說則以 GR1、GR2 及 GR3 最佳。

以搶修路段數量對 1,620 個路網圖進行三項大分類，並以路段階層(L)、路段數量(M)及工作隊數(R)進行細部分類，並依據 GD、GDr、GR1、GR2、GR3 及 GA 分析比較 Gap 值，如表 4.4 至表 4.6 所示：

表 4.4 各演算法 Gap% 值比較表(搶修路段數量 M=16)

T(L,M,R)	GRB	GD	GDr	GR1	GR2	GR3	GA
T(2,16,2)	0	1.22	2.21	0.50	0.50	0.50	1.97
T(2,16,4)	0	2.78	3.59	0.93	0.93	0.93	3.71
T(2,16,8)	0	6.62	5.13	1.96	1.96	1.96	22.81
T(4,16,2)	0	4.81	10.76	3.26	3.26	3.26	10.78
T(4,16,4)	0	11.45	1.93	6.38	6.38	6.38	1.85
T(4,16,8)	0	6.61	2.71	1.71	1.71	1.71	3.44
T(8,16,2)	0	8.61	1.38	6.77	6.77	6.77	1.28
T(8,16,4)	0	6.39	1.12	3.88	3.88	3.88	1.16
T(8,16,8)	0	0.78	1.12	0.13	0.13	0.13	1.08

表 4.5 各演算法 Gap% 值比較表(搶修路段數量為 M=36)

T(L,M,R)	GRB	GD	GDr	GR1	GR2	GR3	GA
T(3,36,2)	0	3.33	7.59	2.02	2.02	2.01	7.70
T(3,36,4)	0	5.02	12.47	2.13	2.13	2.13	12.29
T(3,36,8)	0	6.82	15.51	1.55	1.56	1.56	15.53
T(6,36,2)	0	7.81	6.72	5.61	5.61	5.61	6.45
T(6,36,4)	0	12.33	10.49	6.08	6.10	6.09	10.53
T(6,36,8)	0	11.10	11.44	3.23	3.25	3.23	11.32
T(12,36,2)	0	10.56	5.53	7.86	7.86	7.86	5.44
T(12,36,4)	0	12.58	5.64	8.11	8.11	8.11	5.73
T(12,36,8)	0	4.56	6.71	1.34	1.34	1.34	6.53

表 4.6 各演算法 Gap% 值比較表(搶修路段數量為 M=64)

T(L,M,R)	GRB	GD	GDr	GR1	GR2	GR3	GA
T(4,64,2)	0	5.07	11.48	3.37	3.37	3.36	11.40
T(4,64,4)	0	5.62	17.76	2.68	2.70	2.69	17.65
T(4,64,8)	0	6.86	21.06	2.26	2.26	2.25	21.06
T(8,64,2)	0	8.41	10.89	5.70	5.72	5.72	10.90
T(8,64,4)	0	10.42	16.51	4.90	4.91	4.90	16.39
T(8,64,8)	0	14.34	19.47	5.12	5.15	5.21	19.57
T(16,64,2)	0	11.09	10.56	7.59	7.59	7.59	10.52
T(16,64,4)	0	15.76	13.13	9.37	9.37	9.36	13.39
T(16,64,8)	0	8.11	11.45	1.52	1.52	1.51	11.05

而圖 4.7(a)至(c)則是根據表 4.4 至表 4.6 之 Gap 值繪製之折線圖，藉此觀察其分佈情形，從中可發現只有 GR1、GR2 及 GR3 三種演算法其 Gap 值在大分類上均未超過 10%，表現為最佳。另就大分類上，Gap 最大值與最小值之差異若越小（亦即折線圖之高低起伏越小）則代表該演算法之求解表現越穩定一致。在搶修路段為 16 時，以 GR1、GR2 及 GR3 約 6 為最佳；在搶修路段為 36 時，以 GR1、GR2 及 GR3 約 6.1 為最佳；在搶修路段為 64 時，仍以 GR1、GR2 及 GR3 約 7 為最佳。可判斷就同一個圖型而言，GR1、GR2 及 GR3 此三種演算法之求解表現最穩定。

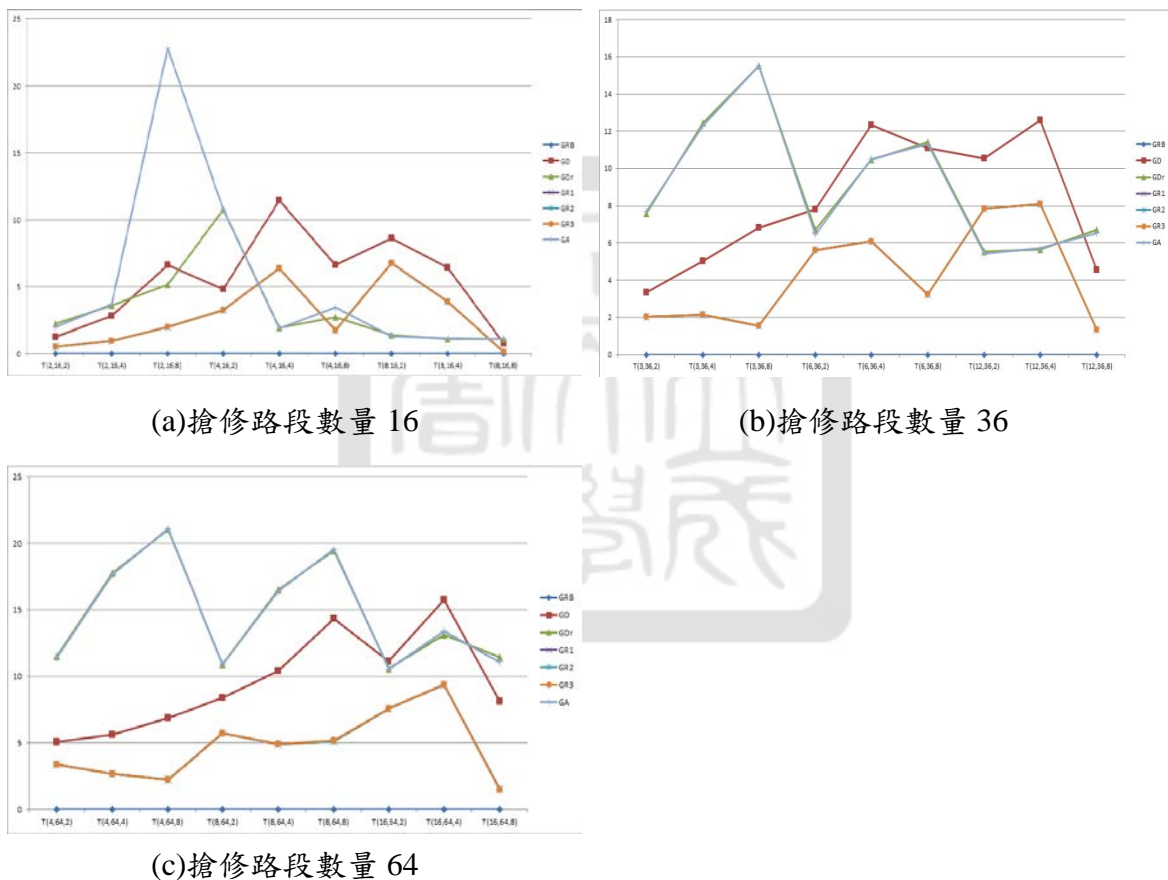


圖 4.7 各種解法之 Gap% 值折線圖

若將 GR1、GR2 及 GR3 再單獨分析，如圖 4.8(a)至(c)所示，可發現若以 Gap 值為 2% 為基準，不管在路段數量為 16、36 或者 64 的情況下，階層分別為 2、3 及 4 時，其 Gap 值均有達到最初設計啟發式演算法所希望達到的目標 ($\text{Gap} \leq 2\%$)。但隨著同樣路段及階層數量，但工作隊數量增加的情況下，如以 T(8,64,2)、T(8,64,4)及

T(8,64,8)與 T(16,64,2)、T(16,64,4)及 T(16,64,8)分類比較，則可發現在工作隊 2 及 4 其 Gap 值均未如工作隊數量為 8 時達到設定之期望值 ($\text{Gap} \leq 2\%$)；類似情況也發生在路段數量為 16 及 36 上，故可判斷 GR1、GR2 及 GR3 演算法在同階層及路段數量的路網下，工作隊數量增加將可縮小 Gap 值的差距。



圖 4.8 鄰近搜尋演算法之 Gap%折線圖分析

而表 4.7 則為各演算法單次計算可行解之 CPU 執行時間，由下表可發現單次執行之啟發式演算法均優於 Gurobi，且各演算法之間其 CPU 執行時間差距均不大，有達到設計啟發式演算法之快速求解目標。此外，除 GD 外，其它五種演算法(GDr、GR1、GR2、GR3、GA)均含隨機機制，可依需求調整求解回合次數（譬如 10,000 次等），隨著求解回合次數增加，可收斂得品質更佳之解。

表 4.7 可行解 CPU 執行時間(單位：毫秒)

Map_L_M	GRB	GD	GDr	GR1	GR2	GR3	GA
Map_2_16	57.48	0.77	0.66	0.3	0.19	0.3	0.45
Map_3_36	435.21	0.57	0.5	0.27	0.17	0.27	0.31
Map_4_64	35267.8	0.48	0.44	0.35	0.21	0.33	0.38
Map_4_16	79.48	1.96	1.78	1.12	0.72	1.12	1.26
Map_6_36	1549.13	1.23	1.3	0.98	0.63	0.98	1.31
Map_8_64	199734.3	1.39	1.2	1.1	0.68	1.08	1.89
Map_8_16	92.80	3.89	3.6	3.17	2.06	3.22	3.56
Map_12_36	1134.24	2.88	2.84	2.7	1.75	2.7	2.98
Map_16_64	58324.14	2.68	2.63	2.76	1.77	2.7	2.69
平均值	32963.84	1.76	1.66	1.42	0.91	1.42	1.65

4.3 小結

在此章節，首先於 4.1 節對本研究影響災後道路網大小及規模之因素：「路段階層」、「分枝」及「路段數量」三類以實例圖形進行說明，其次對研究使用之路網產生器說明如何以「L(路段階層)」等 8 個參數產製研究所需使用之 1,620 個路網圖，最後則說明每一路網圖檔細部各列及各欄分別代表何種參數，以供後續 Gurobi 及 6 類啟發式演算法使用。

4.2 節則分別以 1,620 個路網圖檔對 GD 執行 1 次，及 GDr、GR1、GR2、GR3 與 GA 執行 50,000 次、共六種啟發式解法測試。首先針對 MAX 問題計算六種啟發式解法之最佳值次數，以 GDr 及 GA 為最佳；而 SUM 問題則以 GR1、GR2 及 GR3 表現最佳。若將六種啟發式解法之可行解與 Gurobi 最佳解比對 Gap 值，按其數量比例繪製直線圖，就其比例數量、標準差及平均值而言，則以 GR1、GR2 及 GR3 為最佳。最後以路段階層、路段數量及工作隊進行細部分類並繪製折線圖，則可判斷出以 GR1、GR2 及 GR3 演算法於 Gap 值上均未超過 10%，表現最佳；另以折線圖繪出 Gap 最大值與最小值之差異，在搶修路段為 16、36、64 時，皆以 GR1、GR2 及 GR3 的差異（分別約為 6、6.1、7）為最小，代表其求解表現最穩定。最後則以路網圖分類比較

Gurobi 及六種啟發式解法之單次執行 CPU 時間，可得知啟發式演算法皆可在平均 3 秒內求得可行解，其 CPU 執行時間皆明顯遠小於 Gurobi 整數規劃模型。



第五章、結論與未來研究方向建議

天然災害的發生常造成生命與財產的重大損失，因此如何制定一套有效率之搶修排程，以迅速完成災區路段修復至為重要。本研究針對一般網路的簡化版本—單根樹狀網路加以探討，並說明本問題等價於一個特殊之平行機台排程問題，希望本問題之求解結果可被用於求解一般網路的修復問題，以能儘量縮短災民之總等待時間為目的。本研究歸納出結論與建議如下：

5.1 結論

本研究之目的在於求解有限資源下之單根樹狀網路修復問題，而為更貼近實務，亦於研究中加入多重修復方式，使每條路段對於各工作隊均有不同之修復時間；針對於災難發生後，周邊可調配使用之工具資源亦有限之情況下，亦加入每期時刻各工作隊可使用資源之總上限；研究中亦導入實際路段連結之先後順序，以確保不會有後面路段先行修復之不合理情況發生，而透過多方面之限制與規劃，理應比過去文獻得到更全面之決策方法。

於第三章中，本研究首先說明問題路網的型態為單根樹狀路網，係對每一個災點而言進入之路段來源僅有一個，但是出去之分枝則可有多個，起點是屬於無須修復之工作隊出發點。數學規劃模型可依目標式分成兩類：最小化作業總完工時刻(MAX)、最小化各災點等待搶通之時間總和(SUM)。將各工作隊視為各機台、待修路段視為加工程序、樹狀路網之各分枝（自根節點或分枝節點連至端末葉節點）視為循序必須被加工之過程或產品，則本問題可被視為一個特殊的具資源限制與加工順序規範之平行機台排程問題，因過往文獻已驗證此問題為 NP-hard，故我們亦發展啟發式演算法，以能更有效率地求解大型問題。而演算法主要有四種：第一為「均勻選取式貪婪演算法」(GD)，其方法為於每次有可用之工作隊時，便計算當前所有可修復路段之權重/修復時間比例，以最大者優先修復，如於相同比例即以同等機率選取修復；第二種為「比例選取式貪婪演算法」(GDr)，亦遵循均勻選取式貪婪演算法之貪婪原則，讓更

符合貪婪目的的配對方式有更高機率被選取，以防止演算法陷入局部最佳解；第三種為「鄰近搜尋式貪婪演算法」(GR1、GR2、GR3)，亦遵循均勻選取式貪婪演算法之貪婪原則，但與比例選取式貪婪演算法之隨機選取方式不同，此法先產生一個介於[0,1]間的「固定更換機率門檻值」(25%、50%、75%)，主要步驟依均勻選取式貪婪演算法，但當路段選定後便隨機產生一介於[0,1]間之隨機亂數，若該亂數大於「固定更換機率門檻值」，即改由其他工作隊進行修復，否則仍依均勻選取式貪婪演算法的配對法則來選工作隊與維修路段。第四種為「基因演算法」，使用比例選取式貪婪演算法計算出最佳可行解作為染色體基礎，並進行複製、交配及突變以求得最佳解。

第四章針對本研究中使用之路網圖形說明「階層數」、「路段數」及「分枝數」等參數如何影響路網規模大小，解釋 1,620 個圖檔之產生方式。而針對啟發式演算法亦對均勻選取式貪婪演算法(GD)執行 1 次(因其選取方式固定不變，不似之後的演算法具有隨機特性)、比例選取式貪婪演算法(GDr)執行 50,000 次、鄰近搜尋式貪婪演算法分別以更換比例 25%(GR1)、50%(GR2)及 75%(GR3)執行 50,000 次之測試，基因演算法(GA)設定競爭比例 20%、突變率 10%及菁英比例 20%共執行 50,000 次世代。透過直線圖、平均值與標準差、折線圖及 CPU 求解時間等綜合分析，歸納出以鄰近搜尋式貪婪演算法(GR1, GR2, GR3)執行效果為最佳。

整體而言，對於緊急網路重建問題，本研究導入平行機台排程、路徑修復、資源限制及多重修復模式等多方面考量，首度針對一般路網之樹狀網路簡化版本設計更簡化之數學規劃模型與數種啟發式演算法。實際測試結果顯示最佳之啟發式演算法(鄰近搜尋式貪婪演算法 GR1、GR2、GR3)表現最佳，但其誤差值與 Gurobi 軟體(GRB)之最佳解誤差仍達 8%，故後續仍有一些持續改善的空間。

綜合以上結論，本論文的具體貢獻歸納如下：

- 1. 改善過往平行機台路段搶修文獻未考慮路段順序之問題：**過往平行機台路網搶修文獻均將路段視為獨立之工作進行排程制定，但未對路網圖形、起始點及路段順序加以說明，而本研究發現如為單根樹狀網路圖型，起點均為單一起點，而連通

各災點之路段只有一條，如此則可明確定義該災點之搶通路徑。

2. **提出兩種目標式之排程問題求解：**過往平行機台文獻大多探討如何縮短供給端之完工時間，以最小化最後完工時刻(MAX)為排程目標，並未全面考量需求端災民等待聯通外界之時間。本研究認為滿足需求端災民應更為重要，因此對各路段設定權重（可使用其一端點之災民個數），藉由權重乘以修復時間來表示各路段之加權等待修復時間，從而規劃第二種模型建構目標式-最小化總等待時間(SUM)。
3. **發展啟發式演算法：**本研究之排程最佳化問題會隨著問題規模變大，求解時間亦大幅提升，故須仰賴演算法來執行更有效率的求解。對此，本研究設計三種貪婪式演算法及基因演算法，經由測試結果顯示以「鄰近搜尋式貪婪演算法」(GR1、GR2、GR3)的表現最佳。

5.2 未來研究方向建議

至目前為止，本研究雖對災後緊急路網重建問題做整體探討，但仍有未臻完善之處，以下列舉幾個未來可延伸之議題：

1. **啟發式演算法求解效果不佳：**本研究之基因演算法，係使用比例選取式貪婪演算法透過「執行次數」，以找出次數內最佳可行解作為染色體基礎，並進行複製、交配及突變以求得最佳解，而本研究測試後發現因為染色體編碼方式如 3.4.3 節所述，使用路段搶修順序固定，工作隊變動之方式實施，研究中均設定演化世代為 10,000 次，然皆未找出比初始母體（由貪婪演算法所求得）更佳之解。研判為染色體編碼方式之問題，故後續須針對此項問題進行改善。而針對貪婪演算法，因整體 Gap 值未收斂至 2%，最大者尚有 40%，故亦須將執行步驟加以修正，方可改善。
2. **基礎路網設定值過小：**本次基礎路網圖最大路段數量僅設定為 64 條，而 Gurobi 軟體求解整數規劃模型平均時間均不足 50 秒，故後續亦須加大路網比例，以測試模型建構與啟發式演算法分別求解之時間效率為多少。
3. **初始解導入模型以提升求解效率：**承上，基礎路網若持續擴大，而求解效率不佳

時，亦須設想導入一初始解，以提升 Gurobi 軟體求解效率，並比較與啟發式演算法何者為優。

4. **路網適用範圍太狹隘：**本研究路網為單根樹狀網路圖，就實際災難發生後，路網有可能是千變萬化的，故後續須拓展適用範圍，可使不同路網均可使用此一模型建構，以更貼近實務現況。



參考文獻

中文文獻

吳心琪，(1997)，震災後公路網搶修工程排程之研究，國立交通大學交通運輸研究所，碩士論文。

陳信宇，(2000)，震災物流系統之決策模式，交通大學交通運輸研究所，碩士論文。

黃亦琇，(2000)，震災路段系統評估指標之建立，交通大學交通運輸研究所，碩士論文。

楊宗岳，(2000)，參與行政院 921 災後重建推動委員會交通組之後感，臺灣公路工程，第二十六卷，第十二期，頁 26-40。

藍武王，(2000)，大地震的防災應變：路段管理相關課程，臺灣公路工程，第二十六卷，第九期，頁 2-8。

李志華，(2003)，基因演算法於震災路網搶修排程問題之研究，國立成功大學交通管理學系碩，碩士論文。

王在莒，(2004)，非都會區公路之震災緊急搶修排程，交通大學交通運輸研究所，博士論文。

施佑林，(2005)，災後工程搶修作業暨賑災物流排程之研究，國立中央大學土木工程研究所，碩士論文。

黃琮閔，(2014)，考慮多種修復方式之網路重建問題-以災後緊急救援物流管理之路網重建為例，國立成功大學工業與資訊管理學系，碩士論文。

林秉錚，(2015)，災後管理之管線網路最佳節線修復排程研究，國立成功大學資訊管理研究所，碩士論文。

交通部公路總局，(2007)，災害防救標準作業手冊。

英文文獻

Aksu, D.T. & Ozdamar, L., (2014), "A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation," *Transportation Research Part E: Logistics and Transportation Review*, 61, 56-67.

- Edrisi, A. & Nadi, A., (2014), “An appropriate approach to routing disaster relief assessment teams,” *2nd International Congress on Structure , Architecture and Urban Development*,16-18.
- Felix, W., Guido, S., Stefan, F. & Dirk, N., (2014), ” Emergency response in natural disaster management: Allocation and scheduling of rescue units” *European Journal of Operational Research*,235, 697-708.
- Fiedrich, F., Gehbauer , F., & Rickers, U. , (2000),” Optimized resource allocation for emergency response after earthquake disasters,” *Safety Science*,**35(1-3)**, 1-14.
- Matisziw, T.C., Murray, A.T. & Grubescic, T.H., (2010), “Strategic network restoration,” *Networks and Spatial Economics*, 10(3), 345-361.
- Michalewicz, Z (2013) Genetic algorithms + data structures = evolution programs Springer Science & Business Media.
- Nurre, S.G., Cavdaroglu, B., Mitchell, J.E., Sharkey, T.C. & Wallace, W.A., (2012), “Restoring infrastructure systems: An integrated network design and scheduling (INDS) problem,” *European Journal of Operational Research*, 223(3), 794-806.
- Wang, Y.C., Wang, M.J., Lin, S.C., (2015), “Selection of cutting conditions for power constrained parallel machine scheduling,” *Robotics and Computer-Integrated Manufacturing*,43(2017),105-110.
- Wilson, F.A., Alejandro, V., Héctor, J.C., Suzanna, L., Tom, S., & Steve, C. ,(2014), “A mathematical model for supply chain network infrastructure restoration,” *Industrial and Systems Engineering Research Conference*,78-85.

網站資料

全國法規資料庫，(2010)，災害防救法，
<http://law.moj.gov.tw/LawClass/LawAll.aspx?PCode=D0120014>