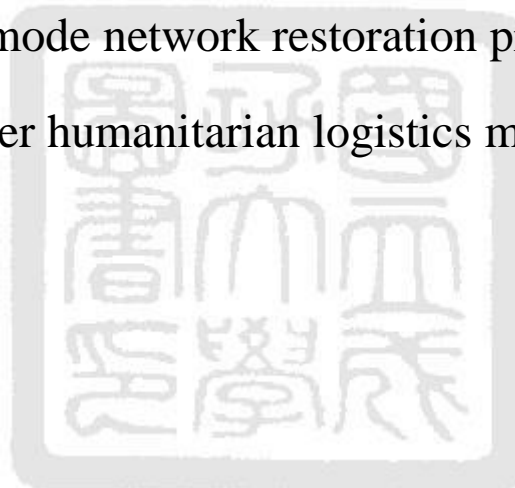


國立成功大學
工業與資訊管理學系碩士班
碩士論文

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

A multi-mode network restoration problem in
post-disaster humanitarian logistics management



研究生：黃琮閔

指導教授：王逸琳 博士

中華民國一百零四年八月

國立成功大學

碩士論文

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

A multi-mode network restoration problem in
post-disaster humanitarian logistics
management

研究生：黃琮閔

本論文業經審查及口試合格特此證明

論文考試委員：張宗勝、

王逸琳

柯東恩 李宇欣

指導教授：王逸琳

系(所)主管：

李興駿

中華民國 104 年 6 月 4 日

摘要

災後的道路網路修復問題為災後緊急救援管理的首要任務，如何指派適當的修復人力於適當時機修復適當的損壞路段，是災後能否迅速進入災區疏散、收容與撤離災民的重要關鍵。本研究將探討在掌握道路網之實際毀損情況下，如何動員現有的資源進行最有效率之緊急搶修計劃，並且決定工作隊等資源之修復路徑、各毀損道路之最佳修復方式及最佳的作業排程。有別於過去文獻，本研究首度考慮各路段有多種不同的修復方式、修復工作隊能力與可用資源亦可能不同、以及修復團隊能否合作等因素，提出具多種修復方式的緊急道路維修問題。

針對單一與無限組工作隊等兩種特例，本研究證明其可被轉換成最小展開樹問題（Minimum spanning tree, MST）及最短路徑樹問題（Shortest path tree, SPT）求解；接著，將探討介於一組至無限多組工作隊之一般情況，並以具資源限制下之專案排程問題（Resource Constrained Project Scheduling Problem, RCPSP）為基礎以提出一整數規劃模式求解此 NP-hard 問題，其主要困難點在於一般網路的路段間並不存在明顯的先後時序關係。由於該整數規劃模式在求解規模龐大的網路重建問題時經常耗時甚久，故本研究提出數個加速求解技巧與演算法來計算更佳的初始解、估算更精準的完工時間、以及能更快地收斂出品質更好的可行解。其中，本研究所設計的分段式啟發演算法先將問題依時間切成數段，再由近而遠先求解較近災點的搶通方式，以其為基礎再逐漸擴張搶通範圍直到全部災點皆搶通為止；該演算法在我們測試過的三類不同形狀的數十個隨機路網後，皆能在極短時間內得到跟最佳解品質相近的可行解，優於最佳化軟體 Gurobi 的求解速度，也遠勝另外設計的粒子群演算法之求解表現。本研究所提出之方法除了可應用於災後路網重建問題外，亦可應用於災後或一般通訊網路重建等相關的網路修復問題。

關鍵字：網路修復問題、災後緊急救援管理、整數規劃、資源限制、專案排程問題

Abstract

A multi-mode network restoration problem in post-disaster humanitarian logistics management

Tsung-Min Huang

I-Lin Wang

Department of Industrial and Information Management

SUMMARY

This thesis investigates a network restoration problem, the first step in humanitarian logistics management, where the time and resources for restoring broken roads (arcs) are given, and the objective is to access all shelters (nodes) as soon as possible. Compared with literature, we are the first to consider multiple modes of network restoration, including multiple teams and team collaboration. Our proposed integer program formulation is based on the resource constrained project scheduling problem, but is much more challenging since no clear precedence relations can be defined in a general network. Several effective heuristics are proposed to calculate good initial solutions, more accurate estimation on the bounds of planning horizon, and feasible solutions of good qualities efficiently. Computational experiments have been conducted over random network families of grid, tree, and path shapes. The results indicate our proposed sequential segment heuristics that partition the planning horizon into segments and iteratively calculate local optimal schedules for each segment can calculate a good solution of small optimality gap within much shorter time than the integer program.

Key words: Network restoration, Post-disaster, Humanitarian logistics, Integer programming, Resource constrained, Project scheduling problem

INTRODUCTION

How to evacuate people from dangerous regions affected by a disaster in minimum time is usually a top-priority task in humanitarian logistics. We consider a network restoration problem where the time and resources for restoring broken roads (arcs) have been estimated. We aim to schedule the recovery teams of different capabilities to restore roads such that all the people in shelters (nodes) can be evacuated as soon as possible. We denote the objective to minimize the completion time of accessing all nodes to be MAX, and the

objective to minimize the total waiting time for all nodes to be SUM. Note that MAX is usually easier to solve than SUM, and the optimal solution for SUM may be a good solution for MAX as well.

MATERIALS AND METHODS

We first give a network reduction procedure to shrink all non-broken roads into a single node, discuss two special cases: (1) with only one recovery team, and (2) with infinite recovery teams, and show how to solve these two cases by efficient solution methods for calculating a minimum spanning tree and a one-to-all shortest path tree. Then, we propose integer programming models called GRB1 based on theories of resource constrained project scheduling problem (RCPSP) to deal with more general cases that take team collaboration and different modes of team resources into consideration, which can be viewed as variants of multi-mode RCPSP. GRB1 is more complicated than conventional RCPSP since different sequence to restore arcs lead to different precedence relationships for the RCPSP-based formulation.

Based on GRB1, we have given formulations for two extensions: (1) a network restoration problem that aims to access part but not all nodes, and (2) to further consider the transportation time when solving for GRB1. The former problem can be done with ease, yet we give a new integer program named GRB2 for the latter problem based on Krüger and Scholl (2010). Note that GRB2 can also be used to solve our original problem by relaxing the transportation time variables, but with much worse effectiveness.

To solve GRB1 in shorter time, we have proposed “MST-*heuristic*”, a heuristics based on minimum spanning tree, to calculate a very good initial spanning tree and an estimated upper bound for the completion time T . On the other hand, the lower bound for T can be calculated by solving a one-to-all shortest path tree. We then propose a bisection method to iteratively converge the lower and upper bound for T . Unfortunately, all these heuristics still does not have sufficient effectiveness for speeding up the solution procedure for solving GRB1. To this end, we propose a sequential segment heuristic (SSH) which partitions T into several segments, treats each segment as an individual planning horizon to solve a relaxed GRB1 aiming to minimize SUM to determine a local optimal scheduling within that segment alone, then we fix the calculated schedule for that segment, shrink all the restored arcs and nodes, and continue the same scheduling procedures for the next segment until finally all nodes become accessible. We further propose a modified SSH to calculate a better local optimal schedule which solves a relaxed GRB1 on a longer segment, but only fix the scheduling decision of earlier part in that segment.

Besides SSH, we also implement two versions of particle swarm optimization algorithm (PSO) called PSO_{IP} and PSO_H , where the former solves GRB1 by Gurobi for each particle in each iteration, and the latter solves the same problem by a greedy algorithm. Both PSO algorithms start with random priorities for selecting arcs to be restored. PSO_{IP} takes longer than PSO_H , but both algorithms do not have good performance, compared with SSH.

RESULT AND DISCUSSION

Extensive computational tests have been conducted over 3 network families of different topologies: Grid, Tree, and Path, with two objectives: MAX or SUM over two scenarios: whether to collaborate or not between recovery teams of different capabilities. All the solution methods are implemented in C language, compiled by Visual C++ 2012. To validate the efficiency and solution's quality of above algorithms, we also solve the IP formulation (GRB1) by Gurobi 6.0.4. All the experiments are conducted on a personal computer with Windows 7, 16 GB RAM, and Intel(R) Core(TM) 7-4770 CPU @ 3.40GHz.

The results indicate *MST-heuristic* does effectively improve the performance of GRB1 for most cases. For the cases of single recovery team, we compare GRB1 with the FNCP by Averbakha and Pereira (2012), and conclude that FNCP has better performance for the MAX objective, but GRB1 performs better for the SUM objective. We still think GRB1 is better than FNCP in general since GRB1 can deal with the general cases of multiple recovery teams but FNCP is only limited to single recovery team. Both SSH and modified SSH can calculate good solutions in shorter time, and the modified SSH has better solution quality than SSH.

CONCLUSION

The network restoration problem is very important since it is the first step to the post-disaster humanitarian logistics management. Based on our survey, only very few literatures focus on this issue. In this thesis, we propose integer program models based on RCPSP, to simultaneously schedule which recovery team to repair which road in an optimal way that minimizes the makespan (MAX) or total waiting time (SUM). Our model is the first to address multiple modes of resource management involving possible team collaboration in such a challenging scheduling problem. We also discuss extensions such as aiming to access part but not all nodes, or taking the transportation time into consideration for scheduling the restoration. To speed up the solution procedures for our integer program GRB1, we have propose several heuristics: the *MST-heuristics* for a good initial solution and an a good upper bound estimate for the planning horizon T , a bisection method to determine a better T , two sequential segment heuristics (SSH) to calculate good

solutions in much shorter time, and two PSO algorithms PSO_{IP} and PSO_H . Results of our computational experiments indicate our proposed heuristics are effective and efficient. Although GRB1 is inferior to FNCP (Averbakha and Pereira, 2012) for the single recovery team case, GRB1 is the first IP model capable of scheduling multiple teams for restoring networks. For future research, we suggest to investigate a dynamic version of our problem, since the status of the damaged roads and recovery progress may be updated dynamically, also the quota of available recovery teams may change as well. Also, the integration of our problem to the inventory, transportation plans for storing and distributing relief supplies would be important and challenging.



誌謝

隨著論文的完成，研究所生活也跟著畫下句點，這兩年來有著許許多多的回憶，有歡笑有淚水，還有更多的痛苦，記得碩一上學期末，完成三天只睡四小時的耐成就，和剛認識的實驗室夥伴偉德及阿佰一起熬夜念書，一起吃勝利早點，再一起在早上六點回家洗澡，接著又回到學校奮戰，但那三天宛如煉獄般的生活，卻讓我們三人更加認識彼此，謝謝你們兩位陪我一起走過。碩一生活中，另外對我影響很大的是實驗室偉大的葉卿學姊，看到學姊遠赴他鄉來到臺灣念書，讓我更加督促自己的進度，謝謝葉卿學姊。碩二，實驗室來了三個新的肝，小秉、貞泰和采緹不僅為實驗室注入新的活力，也成為這一年每當我在論文進度陷入窒礙時解悶的好對象，謝謝三個一直陪我聊天桌遊射飛鏢，祝你們三個開心果在未來這一年能夠順利畢業。

這兩年來最無怨無悔幫助我的逸琳老師，謝謝你從碩零就開始壓著我們學習各種技能，其中，助教課的訓練讓我體會到要一個人學會一門科目，最快的方法就是開一個班要他去教，此外，更讓我有數次上台說話的經驗，對後續的論文發表及口試有很好的練習機會，其他要感謝老師的太多了，在這邊篇幅不夠，我就將這份感激之情放在心中，未來再報答老師了。

而這份論文能夠完成，更要感謝我的家人給我的支持，這兩年來我時常突然消失一陣子才出現，感謝你們一直默默地在背後支持我，還有在我回家的時候幫我進補，以及在臺灣另一端的小萱，謝謝這兩年來對我的體諒與包容，希望你能明白我的心意，我愛你們！最後我要謝謝我自己，辛苦了！但以後要更加努力喔！

目錄

摘要	I
Abstract	II
誌謝	VI
目錄	VII
圖目錄	X
表目錄	XII
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究目的	5
1.3 研究範圍	8
1.4 論文架構	8
第二章 文獻回顧	9
2.1 災害管理相關文獻	9
2.2 資源限制下之專案排程問題 (RCPSP) 相關文獻	14
2.2.1 RCPSP 數學模式	17
2.2.2 RCPSP 相關演算法	20
2.3 小結	21
第三章 模式建構	22
3.1 問題描述	22
3.2 問題假設	23
3.3 網路縮減	24
3.4 一組及無限多組工作隊之特殊情況	25

3.5	數學模式	29
3.5.1	符號定義	29
3.5.2	整數規劃模式	30
3.6	模式延伸	34
3.6.1	延伸一：修復特定災點	34
3.6.2	延伸二：考量工作隊移動時間	34
3.7	小結	39
第四章	求解方法	40
4.1	模式初始解與上下界	41
4.1.1	啟發式初始解及上界設定	41
4.1.2	下界設定	42
4.2	完工時間 T 之預估	42
4.3	分段式啟發演算法 (Sequential segment heuristic, SSH)	43
4.3.1	步驟流程	44
4.3.2	範例說明	47
4.3.3	演算法修正	50
4.4	粒子群最佳化演算法 (PSO)	51
4.4.1	PSO_{IP} 步驟流程	52
4.4.2	PSO_H 步驟流程	54
4.5	小結	55
第五章	數值分析	56
5.1	輸入資料	56
5.1.1	測試網路	56
5.2	GRB1 模式測試	57
5.2.1	實驗情境一：不考慮工作隊合作	57

5.2.2	實驗情境二：考慮工作隊合作.....	60
5.2.3	模式比較.....	62
5.3	GRB2 模式測試.....	63
5.4	分段式啟發演算法測試.....	65
5.3.1	實驗情境一：不考慮工作隊合作.....	65
5.3.2	實驗情境二：考慮工作隊合作.....	69
5.5	PSO 演算法測試.....	70
5.6	小結.....	71
第六章	結論與未來研究.....	73
6.1	結論.....	73
6.2	未來研究.....	76
參考文獻	78



圖目錄

圖 1.1 災害管理流程圖	1
圖 1.2 災後網路示意圖	3
圖 1.3 以路段所需修復時間為依據之搶修計劃	4
圖 1.4 最佳搶修路徑與其作業排程	4
圖 1.5 決策三：降低工作隊之閒置情況	5
圖 1.6 災後毀損網路示意圖及對應之 PERT 網路圖	6
圖 1.7 無修復方向限制之災後毀損網路示意圖	6
圖 2.1 作業時序限制示意圖	11
圖 2.2 PERT 網路圖	15
圖 2.3 MPCPSP 對應之 PERT 網路圖及最佳作業排程	16
圖 2.4 PERT 網路圖	19
圖 2.5 MRCPSP 修正後之 RCPSP 示意圖	21
圖 3.1 九二一大地震公路受損示意圖	23
圖 3.2 具無毀損子網路之網路圖	24
圖 3.3 修正後之網路圖	25
圖 3.4 災後損壞網路示意圖	26
圖 3.5 最佳搶修路徑為 MST	27
圖 3.6 最佳搶修路徑為 SPT	28
圖 3.7 錯誤的修復路徑範例	31
圖 3.8 未做網路縮減之網路及迴圈路徑	32
圖 3.9 資源限制說明	32
圖 3.10 虛擬節點與節線示意圖	35
圖 3.11 資源運輸示意圖	36

圖 3.12 有迴圈之修復路徑	37
圖 4.1 完工時間 T 之預估流程	43
圖 4.2 分段式啟發演算法求解流程	46
圖 4.3 範例網路圖	47
圖 4.4 各節點之最快修復路徑（網路之 SPT）及縮減後之網路	48
圖 4.5 時間區段 I_1 內之搶修路徑及作業排程	48
圖 4.6 投入時間區段 I_2 之網路	49
圖 4.7 時間區段 I_2 內之作業排程	49
圖 4.8 時間區段 I_3 內及最後整體作業排程	50
圖 4.9 各工作隊之搶修路徑	50
圖 4.10 最佳搶修作業排程	51
圖 4.11 各工作隊之最佳搶修路徑	51
圖 4.12 粒子示意圖	52
圖 4.13 PSO _{IP} 演算法流程圖	52
圖 4.14 粒子示意圖	54
圖 5.1 網路結構示意圖	56
圖 5.2 網狀網路圖下 MST-HEURISTIC 的改善比較	59
圖 5.3 樹狀網路圖下之 MST-HEURISTIC 的改善比較	60
圖 5.4 路徑網路圖下之 MST-HEURISTIC 的改善比較	60
圖 5.5 GRB1 與 FNCP 之求得最佳解次數比較	63
圖 5.6 GRB1 與 FNCP 之 $B-GAP$ 比較	63
圖 5.7 修正前後的演算法與 GRB1 模式求解時間比較	67
圖 5.8 修正前後的演算法求解誤差比較（格狀網路）	68
圖 5.9 修正前後的演算法求解誤差比較（樹狀網路）	68
圖 5.10 修正前後的演算法求解誤差比較（路徑網路）	68

表目錄

表 1.1 路段修復模式	7
表 2.1 災後緊急搶修作業相關文獻與本研究之比較	13
表 3.1 多種修復方式說明	33
表 4.1 GRB1 之模式規模	40
表 4.2 GRB1 之模式大小估算	40
表 4.3 各路段之在不同工作隊下的修復時間	47
表 5.1 測試網路之規模資訊	57
表 5.2 GRB1 模式在實驗情境一下的求解表現 (MAX)	58
表 5.3 GRB1 模式在實驗情境一下的求解表現 (SUM)	58
表 5.4 GRB1 模式在實驗情境二下的求解表現 (MAX)	61
表 5.5 GRB1 模式在實驗情境二下的求解表現 (SUM)	61
表 5.6 GRB1 與 FNCP 求解表現比較 (MAX)	62
表 5.7 GRB1 與 FNCP 求解表現比較 (SUM)	62
表 5.8 GRB2 模式在實驗情境一下的求解表現	64
表 5.9 分段式啟發演算法參數設定	65
表 5.10 分段式啟發演算法求解結果 (實驗情境一)	65
表 5.11 修正後分段式啟發演算法求解結果 (實驗情境一)	66
表 5.12 測試之網路對象	68
表 5.13 在不同路段修復時間下的求解結果	69
表 5.14 分段式啟發演算法求解結果 (實驗情境二)	70
表 5.15 修正後分段式啟發演算法求解結果 (實驗情境二)	70
表 5.16 PSO_H 之測驗結果	71

第一章

緒論

1.1 研究背景與動機

全球環境變遷導致全球災害的頻率增加與規模增強，而臺灣地區被列為是全世界最容易發生災害的地區之一，主要因為臺灣的地理環境，位處於歐亞大陸板塊與菲律賓海板塊的交界處，經常引發臺灣地區頻繁的地震活動，且氣候屬於副熱帶季風區，天氣型態複雜而多變，又位於颱風路徑要衝，因此每年常遭受颱風、地震侵襲而形成洪水及土石流等天然災害，伴隨著全球環境變遷，災害發生的強度增強，使得緊急災害管理日顯重要。



圖 1.1 災害管理流程圖

緊急災害管理工作可分為災前整備與減災、災後應變及災後重建三階段：

1. 災前整備與減災 (Preparedness & Mitigation)

在災害發生前的準備工作，就災害可能發生的影響先行準備，透過各種因應措施，防止災害之發生或減輕災害的影響，以降低災難所帶來的衝擊。

2. 災後應變 (Response)

在災難發生後，於第一時間搶救人民生命及財產，並能迅速進入災區疏散、收容與撤離災民，因此，如何短時間的動員現有的資源進行緊急救援工作，是此一階段的最大挑戰。

3. 災後重建 (Recovery)

屬於災後的重建工作，並讓社會與經濟恢復正常運作之長期政策，配合減災行動使災害後恢復至災害前或比災害前更佳狀態的對策。

臺灣最常經歷地震與颱風等大規模的災害，而這些災害大多帶來「面」的影響，對災害地區道路網造成多處破壞，造成災民無法疏散、災區無法救援，亦使得救援單位無法確切掌握災區內實際災害情形，使得救援單位無法即時擬定接下來的救援計劃，進而影響災後救援的效率，影響所及，人命及財產因黃金救援時間的虛耗造成難以彌補的損失。

一般而言，緊急搶修策略主要是確保救援運輸道路暢通，對於受阻路段除應緊急搶通外，若損壞路段數量龐大，導致人力、機具等可能不敷使用時，決策單位將評估各路段是否該優先修復。因此，如何在短時間內擬定出最有效率的搶通計劃，涵蓋欲搶修的路段、搶修順序及人力、機具等資源的搶修排程，將是值得探討的重點。

搶修計劃的目標，以決策單位的角度來看，會希望整體的完工時間越短越好，即最小化最晚打通災點之打通時間。以災民的角度來看，則會希望支援能夠盡早進入各災區，亦即最小化各災點的打通時間總和。雖然對災民來說，皆希望自己的災區能夠優先被打通，對此本研究假設決策單位看待各災區的優先權皆平等無差異。然而在有限的資源下，亦有可能發生無法在設定之時限內打通所有的災點，此時決策單位將依各災區的災情程度做優先程度的比較，並且優先打通災情狀況較嚴重的災區，這種情況將在本論文之第三章加以討論。

在現實的搶修計劃中，因資源運輸時間較各路段的維修時間相對短許多，本研究

將假設在尚未毀損的路段上所耗費的移動時間為 0，故可將原網路中已經相通的子網路縮減成一個虛擬節點，借以縮減網路規模；經此網路縮減程序後，僅剩有毀損的節線（路段）才會保留下來。但當網路規模非常龐大時，資源運輸所耗費的時間則應納入排程考量，本研究將在第三章中的延伸問題部分討論考量資源運輸時間的網路重建問題。

搶修計劃主要可分兩個階段，第一階段為「搶修路徑決策」，第二階段為「作業排程與資源指派」，且此二個階段會相互影響，但在過去研究中，多在已知修復路段為哪些的情況下，去進行作業排程與資源指派，以下將說明此二階段如何互相影響：

救援單位在做路徑決策時，常以路段所需修復時間為決策依據，例如：

決策一：只選擇所需修復時間最短的路段 或

決策二：針對每一災點選擇能最快連通的搶修路徑；

儘管從修復時間來看，在這兩種決策依據下，可排除修復時間長的路段，但卻未考慮後續可能因資源的稀少性，使得整體作業無法在預估的完工時間內完工。

以圖 1.2 的網路圖來說明，假設現有兩組工作隊，欲修復一毀損道路網如圖 1.2，圖中節線上的數字代表修復該條節線所需的修復時間。依照上述兩種決策方法，可得其對應之搶修路徑及作業排程如圖 1.3(a)及(b)所示：右側各方格代表各搶修路段作業，其中文字代表其對應的節線，長度表示修復所花費的時間，寬度表示所占用工作隊數，故由圖可知完工時間皆為 10，然而實際最佳搶修計劃應如圖 1.4，完工時間為 9；由此例發現若不考量資源稀少性而直接決定搶修路徑下的搶修計劃可能不為最佳。

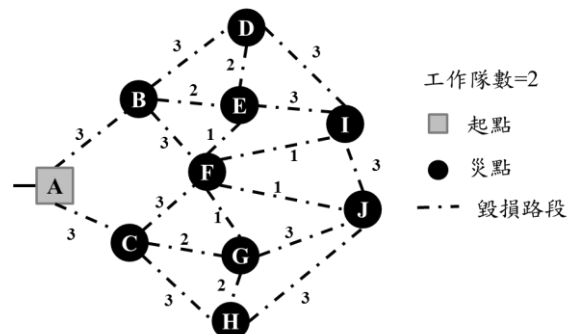
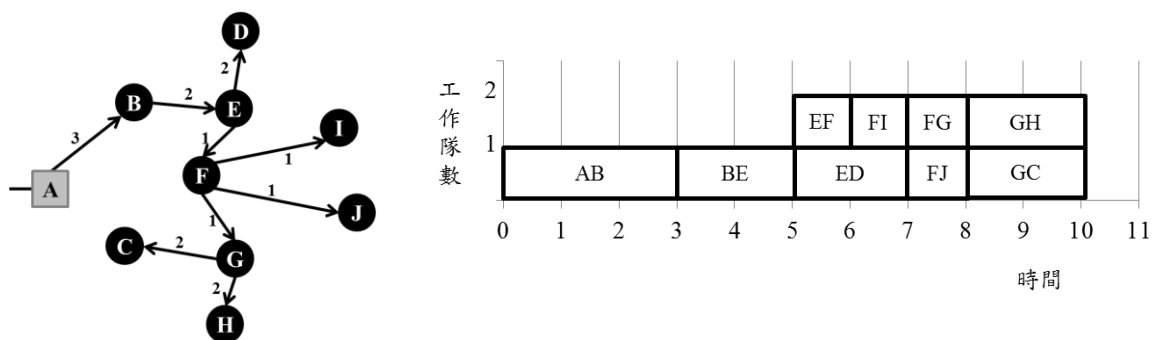
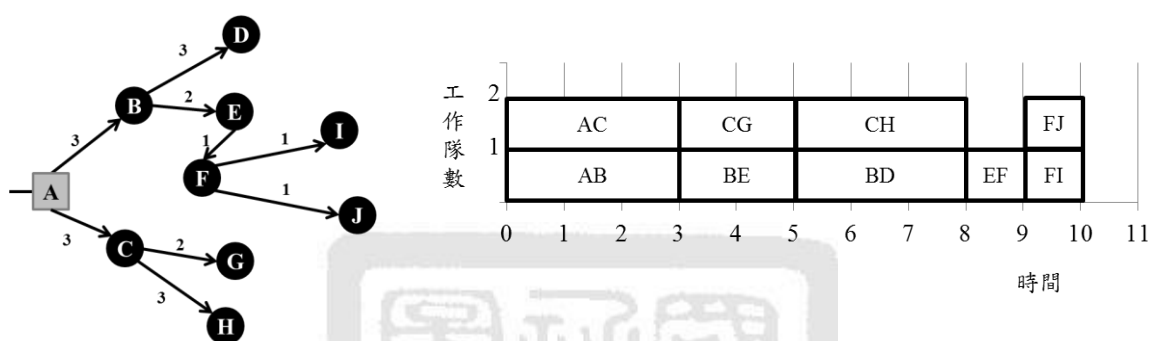


圖 1.2 災後網路示意圖



(a) 決策一：只選擇所需修復時間最短的路段



(b) 決策二：針對每一災點選擇能最快通往的搶修路徑

圖 1.3 以路段所需修復時間為依據之搶修計劃

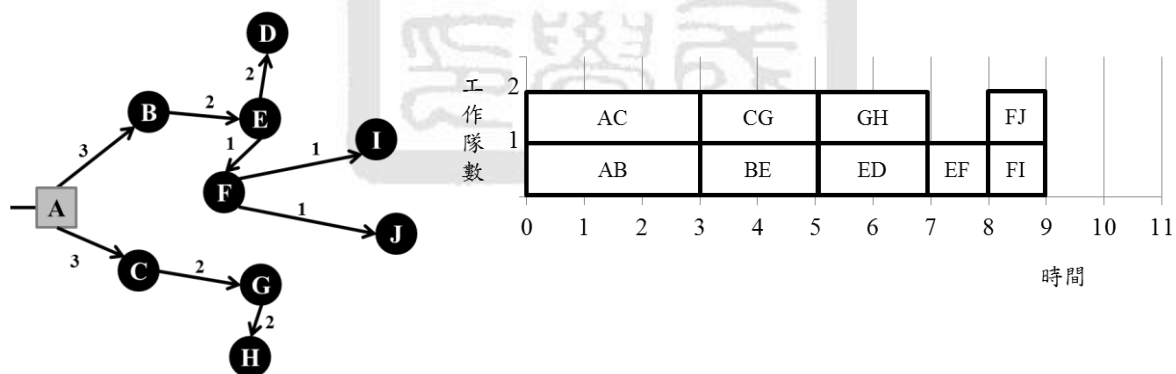


圖 1.4 最佳搶修路徑與其作業排程

除了上述兩種決策依據外，亦有救援單位是以極大化工作隊之使用率為決策依據，例如：**決策三**：降低工作隊之閒置情況；即針對每一組工作隊，每當其完成一路段之搶修作業後，隨即再指派另一作業。但此種貪婪式決策通常忽視了各路段被修復後可能得到更多修復時間較短的路段可能性。以圖 1.2 的網路圖為例，決策三下所得的搶修路徑及作業排程如圖 1.5，其排程結果工作隊 2 在修復完路段(G, H)後，因不允許閒

置，便立即修復路段(G, J)；但若允許工作隊 2 閒置並等待工作隊 1 修復完路段(E, F)，則工作隊 2 可藉由修復時間為 1 之路段(F, J)來取代修復時間為 3 之路段(G, J)以降低完工時間。

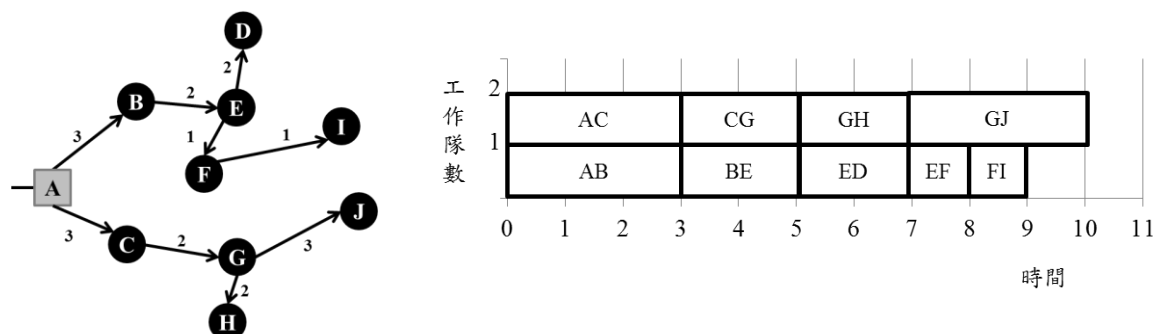


圖 1.5 決策三：降低工作隊之閒置情況

在過去工務單位實際搶修經驗中，搶修指派一般由決策者根據其經驗來主觀地訂定決策，缺乏全面系統性的客觀學理基礎。其決策就算可行，經常未必最佳；倘若決策失當，恐造成人命損失等難以彌補的後果。故本研究將建構數學規劃模式，其目標為最小化「所有災點搶通前的等待時間總和」或「整體搶修計劃的完工時間」，並可同時考量「路徑決策」及後續「作業排程與資源指派」模式，提供工務單位決策者一個可行的災後救援緊急應變指派方法，希望藉此得到比以往研究更佳之搶修計劃。

1.2 研究目的

災後救援的最大目標是在黃金 72 小時內搶進災區內進行救援，因此如何在短時間內善用有限的資源並決定出最有效益的道路網搶修排程甚為重要。若將整個道路搶修視為一個專案，則如何利用現有的救災人、物力資源以在最短的時間內搶通所有的災難點即為一個複雜的「具資源限制之專案排程問題」(Resource Constrained Project Scheduling Problem, RCPSP)，旨在探討如何於各項資源有限的情況下，建立最佳排程，以達到專案資源使用最佳化。在救災部分的道路網搶修排程專案，應該以能於最短時間內搶通所有災難點為首要之務。



圖 1.6 災後毀損網路示意圖及對應之 PERT 網路圖

假設有一災區道路網如圖 1.6(a)，節點 A 為搶修工作隊出發點，節點 B、C、D、E、F 為災點，虛線段為損壞之路段，虛線上之數字為修復該路段所需的時間，且修復方向須依照箭頭方向，找一可使網路圖內所有節點互相連通之搶修計劃，可藉由下述方式將圖 1.6(a)之網路圖轉成圖 1.6(b)之 PERT 網路圖：將修復所有毀損節線視為一修復作業，由圖 1.6(a)可明顯看出各作業之時序限制（亦即路段被修復的先後順序限制），依照時序限制便可畫出其對應之 PERT 網路圖，而此例題即可被視為 RCPSP。值得注意的是，圖 1.6 之路網為 Tree 型態，因此同一各分枝上的路段剛好必須滿足時序限制；然而，針對非 Tree 型態的路網（如圖 1.7 所示）而言，其搶修計劃中之工作則不見得有時序限制，如此一來傳統具時序限制的 RCPSP 數學模式即必須加以修改，不能直接使用。以圖 1.7 為例，其道路無修復方向之限制，故其時序限制將受每條損壞道路的修復方向而有所不同，舉例來說，修復節點 B、E 及 C 的順序可為 A-B-E-C 亦可為 A-C-E-B，注意到路段 C-E 及 B-E 的修復方向在兩個路徑中互為相反。最後，本研究所討論之搶修計劃，目的並不見得一定要修復所有的損壞道路（此為災後復原的範疇），而是盡快搶修通往各災點的路徑，故只需修復部分損壞道路，直到可打通所有的災點為止。

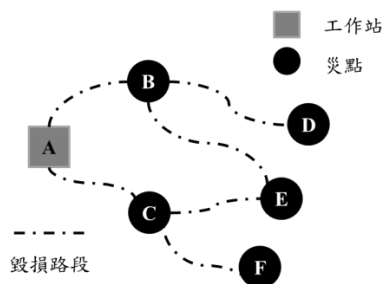


圖 1.7 無修復方向限制之災後毀損網路示意圖

表 1.1 路段修復模式

模式	使用的工作隊		所需維修時間
	工作隊 A	工作隊 B	
1	✓		5
2		✓	3
3	✓	✓	1

實際搶修作業中，所使用的資源種類非常多，其中以人力為主要的資源種類，並且配備其他輕重型器具組成一組工作隊，決策單位再指揮各組工作隊進行各地的搶修作業。而工作隊之間的修復能力，可能因配備不同而有所不同，因此各路段由不同工作隊進行搶修，便有不同的修復時間。舉例來說，假設有兩組工作隊 A、B，其修復能力不同，則路段的維修模式可如表 1.1 所示：模式 1 表示指派工作隊 A 進行該路段的搶修作業需要 5 單位時間；模式 2 表示指派工作隊 B 需要 3 單位時間；而模式 3 則表示兩隊合作需要 1 單位時間。另外，本研究亦考慮部分工作隊可能因配備不足，而無法處理部分路段（例如沒有配備大型挖具設備的工作隊無法處理受巨大落石阻礙的路段）。故本研究不僅探討搶修排程之問題，亦將決定各欲修復路段之最佳修復方式，並以 RCPSP 之延伸問題－「多種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）的觀點建構數學模式。然而，在問題規模龐大時，直接對數學模式虛耗時數日，無法滿足問題的急迫性，因此，除了建構數學模式外，本研究亦從問題本身發展一個分段式啟發演算法來進行求解，期望在短時間內求得一近似最佳解。

整體而言，本研究將以 MRCPSP 為基礎，建構在有限資源下，兼顧「路徑決策」、「作業排程與資源指派」與「多種修復方式」之整數規劃模式，並且發展一分段式啟發演算法求解，以提供工務單位在災後救援緊急搶修階段，可獲得更高效率的重建方案。

1.3 研究範圍

本研究的研究範圍如下：

1. 建立緊急災後救援中，指揮單位一套搶修道路網的決策方法，目標打通災區內所有災點或需支援的地點，讓救援單位得以進入並進行疏散等救援行動。
2. 在時間緊迫及資源有限的條件下，決定搶修作業的排程、工作隊的指派、以及每一修復路段之最佳修復模式。

1.4 論文架構

本論文之架構如下：第二章為文獻回顧，針對災害管理相關文獻做整理分類，並回顧 RCPSP 的相關文獻。第三章將討論問題之特例與延伸，並提出本研究之數學模式。第四章進行模式的求解，針對問題提出求解演算法。第五章為數值分析，測試本研究提出之數學模式及演算法，在不同的網路型態（如格狀網路）下的求解表現，並與過去文獻的方法進行比較。第六章總結本研究成果及提供未來可研究方向和建議。

第二章

文獻回顧

在災後緊急道路網搶修作業中，該以何種修復路徑以搶通各個災點，為影響整體成效好壞的最大因素；但在資源限制下，修復作業的排程及修復方式，亦為影響完工時間的不確定因素。若不考慮後續的作業及資源排程，而以修復時間之長短來決定修復路徑，此路徑可能會因資源的限制，而不為最佳（時間最短）之修復路徑。近幾年來災後工程搶修相關文獻頗多，在過去文獻中，「修復路徑決策」及「作業排程與資源指派」常被分開來討論，甚少有將此二問題合併探討之文獻，且較少考慮路段具「多種修復方式」的可能性，本節將針對災後道路網搶修作業相關文獻進行回顧。另外，本研究運用 RCPSP 概念來建構一整數規劃模式，故亦針對 RCPSP 相關文獻作回顧，做為研究之參考。

2.1 災害管理相關文獻

災害管理目前已廣為討論，根據 Altay and Green (2006)的分類，目前的文獻大致可分為四個方向：

1. **減災**：目的在預防可能轉變成災害的危險或降低災害可能帶來的影響。過去的文獻約有 44%在探討此一方向的問題，是四個方向最廣為討論的。
2. **整備**：過去文獻在災前整備主要探討的問題著重於如何使災前的準備行動能更有效率的支援災後反應，例如：緊急避難處的選址問題及支援物資的配置問題等等。
3. **反應**：災後反應的目的在於管理目前的支援物資，使其能夠有效率的投注至災區救援，減小災區人命及財產的損失，過去文獻討論的主題有物資配送問題、災區緊急疏散問題、醫院指派問題及道路網緊急搶修問題等等。

4. **重建復原：**重建復原是將災區的各项系統，恢復至原狀態或更佳的状态及穩定災區居民的生活，過去文獻探討的問題有災後廢棄物處理問題、災後道路網重建問題等等。

本研究探討的問題係屬「災後反應」的道路網緊急搶修問題，和重建復原的災後道路網重建問題的差異在於，前者屬短期的搶修計劃，具有急迫性，主要包括包括障礙清除、工作便橋、工作通道之搭建，復原程度只需達可容大客車或卡車通過為止，使支援得以運送；而後者屬長期的修復計劃，包括鋪面、植被等完全復舊工程。

災後反應在過去幾年亦有廣泛的討論，其主題多集中於災後物資配送問題，災後道路網緊急搶修問題則相對較少，且多將修復路徑決策、作業排程及資源指派分開討論，並常以最佳化方法來進行求解，以下將繼續針對災後緊急搶修作業做文獻探討。

路徑決策問題，即決定整體修復路徑所包含的路段有哪些，探討此議題的文獻如 Meshkovskiy and Rokotyan (1992) 研究災後通訊網路重建問題，其研究容許在資源不足的情況下，可允許網路中一些節點不被連通，亦即只需連通特定的節點，並在已知各節線修復成本的情況下，找尋連通網路中特定節點的修復方式。該研究亦提到當資源充足的情況下，允許連通所有節點時，此問題可視為找尋一最小展開樹(Minimum spanning tree, MST)；而在資源不充足的情況下，只連通目前最迫切的節點，此問題便可視為在此網路圖中找尋一史坦納樹(Steiner Tree, ST)，又找尋 ST 為一 NP-hard 問題，對此 Meshkovskiy and Rokotyan (1992) 提出可在多項式時間下求得近似解的演算法。Liberatore et al. (2014) 則認為路徑決策應以後續賑災物資運送為最大考量，故以網路流量提出一數學模式，其模式有數個考量的目標，涵蓋各災區救援物資需求滿足程度、後續賑災物資物流路線之可靠度、搶修工時、搶修成本及搶修風險，且主要限制整體的修復成本不得超過預設之預算上限。

若只單以修復時間為依據來決定修復路徑，於 1.1 節所提之決策一與決策二兩種範例可知，並不一定是最佳搶修路徑，除了資源稀少以外，搶修作業間也會有時序上的限制。如圖 2.1 所示，需先修復完路段(A, B)，才可修復路段(B, D)及(B, C)，故搶

修作業的排程亦為研究的重點之一，對此我們已在第一章說明如何將此問題轉化為 RCPSP 來求解。

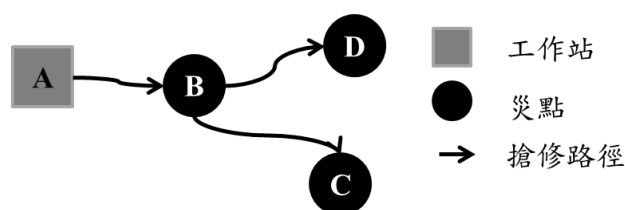


圖 2.1 作業時序限制示意圖

通常搶修時程下的每一時間點，所能使用的搶修資源是有限的，以人力資源為例，每一個時間點所能使用的人力是有限的。故除了原本作業之間的時序性，資源的稀少性亦為作業排程的一大限制，因此需將資源的指派納入考量。

Averbakha (2012) 討論網路結構為路徑 (path) 的道路網搶修問題，此類結構為路徑的網路常出現於隧道或礦坑，由於單一路徑的網路沒有分枝及迴圈，因此通往每一個災點的方法皆只有一種，故要打通全部災點勢必需修復所有的毀損路段。Averbakha (2012) 允許各個工作隊的能力不同，並假設工作隊合作的能力是具相加性的。亦即若工作隊 1 可每小時修復 0.5 公里，工作隊 2 可每小時修復 1.5 公里，則工作隊 1 及 2 合作可每小時修復 2 公里。除此之外，工作隊的起始位址亦可不同，並以動態規劃來求解此問題。

Averbakha and Pereira (2012) 提出網路重建作業排程問題 (Flowtime network construction problem, FNCP)，開始探討無特定網路結構的道路網，但其工作隊數上限只有一組，並且證明在此情形下的 FNCP 為一 NP-hard 問題。但若限制網路圖結構為樹 (tree) 時，可將問題轉化為一般的排程問題，並且可在 $O(n \log n)$ 下求得最佳解 (Brucker, 2007)。而對於無特定網路結構的網路圖，則利用單一商品流問題 (single-commodity flow) 及最小展開樹的概念來建構整數規劃模式，並以分支界限法模式求解。從其測試結果顯示，在求解節點個數大於 30 個以上的網路圖，FNCP 模式可能無法在 30 分鐘內求得最佳解。

Yan and Shih (2009) 同樣也是討論無特定網路結構的道路網，但其目標在於修復所有的損壞點，且認為修復道路網時，需將後續賑災物資物流配送之物資運送會耗費的運輸時間納入搶修計劃擬定考量因素之一，故透過時空網路圖 (time space network) 的設計，建構一可同時考量災後工程搶修作業及賑災物流排程之數學模式，目標最小化整體搶修工時及後續的物資運送時間，但此數學模式在網路圖較龐大時，依舊需花費大量的時間才可找到最佳解。

Nurre et al. (2012) 提出完整網路設計及排程問題 (Integrated Network Design and Scheduling, INDS)，主要探討的是災後系統修復之作業排程，此問題將災後損壞之系統結構模式化為一網路圖，網路中之節點視為系統中的設施，節線表示設施間的服務關係且具方向性。例如：設施 A 及 B 間有一節線(A, B)表示設施 A 提供服務給設施 B，節線上的流量表示設施間的服務量，如電力等。系統的損壞可視為移除網路圖中之節線，故 INDS 便可視為一個探討在有限修復資源下，如何重新安置節線的問題。該研究以單一商品流問題為基礎提出一整數規劃模式，來決定欲修復哪條節線，以最小化各節點未滿足的需求量。

以上臚列之文獻大多以網路中各兩相鄰節點的鏈接(link-based)為基礎來建構數學模式，此外亦有研究是以路徑為基礎(path-based)的觀點來建構數學模式。

Matisziw et al. (2010) 認為緊急修復作業中最重要的目標是確保特定節點間的路徑之連通性及流量上限。例如賑災中心及災點，若欲維持路徑的連通性，則必須修復路徑上的所有節線。該研究依此提出一整數規劃模式，將整體計劃分為數期，並在不違反每期修復成本預算下，決定每一期要修復的節線，以最大化每組起訖點 (Origin-destination, O-D)間的流量並最小化總修復成本。Aksu et al. (2014) 則先針對每一災點找一可通往撤離點或臨時集中處之最短路徑，並修復這些候選路徑。該研究以這些最短路徑建構一整數規劃模式，在每期修復資源有限的情況下，盡可能地越早打通這些路徑越好。運用 Divide and Conquer 的觀念設計一演算法，將原網路切割成數個區域，並以不同資源量限制，對每一區域進行求解(divide)，得到各區域在不同資

源量下之修復計劃，接著再以整體資源上限，來尋找不違反每期資源上限之各區域的修復計劃組合(conquer)。

從過去災後緊急搶修作業相關文獻可發現，較少有研究是一次性的考量「路徑決策」、「作業排程」及「資源指派」等三大議題。如表 2.1 所示，其中，Averbakha and Pereira (2012) 所探討的問題及其問題假設雖與本研究所探討的問題環境相同，但其研究只探討一組工作隊的情形，較不符合現實的道路搶修計劃；而 Averbakha (2012)、Nurre et al. (2012)及 Aksu et al. (2014) 在研究中雖同時考慮路徑決策、作業排程及資源指派，但 Averbakha (2012) 只能解決網路結構為路徑的網路圖；Nurre et al. (2012) 所探討的網路圖性質則不符合道路網路的性質（道路網中的節線不具有方向性，屬無向節線）；而 Aksu et al. (2014)則是以選擇能最快通往每一災點的搶修路徑為決策依據，再去進行作業排程與資源指派，但此種方式所得之搶修路徑並不一定是最佳路徑（本研究已於 1.1 節說明之）。

表 2.1 災後緊急搶修作業相關文獻與本研究之比較

文獻	路徑決策	作業排程	資源指派	多種修復
Meshkovskiy and Rokotyan, (1992)	✓			
Liberatore et al., (2014)	✓			
Averbakha (2012)	✓	✓	✓	✓
Averbakha and Pereira, (2012)	✓	✓		
Yan and Shih, (2009)		✓	✓	
Nurre et al., (2012)	✓	✓	✓	
Matisziw et al., (2010)	✓	✓		
Aksu et al., (2014)	✓	✓	✓	
本研究	✓	✓	✓	✓

若不同時考量「路徑決策」及「作業排程」，儘管可得到最佳作業排程，但因路徑不為最佳搶修路徑，所得之整體搶修計劃亦非最佳。而「作業排程」及「資源指派」兩問題為 RCPSP 所討論的範疇，故本研究的問題可看成是由「路徑決策」與 RCPSP

兩問題的組和問題。除此之外，本研究亦加入各路段具多種修復模式的可能性，並探討其對搶救時程之影響。下一小節將繼續針對 RCPSP 作文獻回顧。

2.2 資源限制下之專案排程問題 (RCPSP) 相關文獻

近幾年來，專案排程問題(Project Scheduling Problem, PSP)受到學術上及實務上很大的重視，且有相當豐富的相關文獻。1950 年代末期，便已發展出針對 PSP 之解法，如計劃評核術(Program Evaluation and Review Technique, PERT)及要徑法(Critical Path Method, CPM)；但 PERT 及 CPM 僅考量時間因素，並未將資源限制納入考量，造成其實用性受到限制。

而資源限制下專案排程問題 (Resource Constrained Project Scheduling Problem, RCPSP) 便可改善 PERT 及 CPM 未將資源限制納入考量的缺點，且因其較能符合業界之實際情形，在過去數十年廣為國內外研究與討論，其中一項延伸為「多種作業方式及資源限制下專案排程問題」(Multi-mode Resource Constrained Project Scheduling Problem, MRCPSPP)，即利用不同資源需求量導致不同作業之執行時間，有不同作業方式可作為選擇，使研究更加貼近實務。以下為 RCPSP 問題描述：

RCPSP 可看作一有 J 個作業 (activity) 之專案，編號 j 從 1 至 J ，每個作業 j 有其對應之執行時間 p_j (processing time)，每個作業都有其先行作業(predecessor activities) 集合 P_j ，即作業 j 需於作業 $i \in P_j$ 皆完成後才可開始進行；在專案中有 K 種資源，每個資源在每一期有一資源上限 R_k ，而每一作業 j 對每一資源 k 有需求 r_{jk} 。依上述，可將專案轉化為作業之 PERT 網路圖：考慮一網路圖 $G=(N, A)$ ，其中 G 為有向無循環網路圖(acyclic directed network)， N 為作業節點之集合，每一個節點代表一個作業， A 代表有向節線之集合，一個有向節線代表作業間之優先順序，例如：節線 (i, j) 表示作業 j 須待作業 i 完成後，才可開始進行，故依據專案中所有作業之前置作業集合，便可繪出一 PERT 網路圖，最後在網路圖前後各加一虛擬節點，分別為虛擬開始節點及虛擬完工節點，如圖 2.2，節點 1 至 8 為專案之作業，節點 0 及 9，分別表示虛擬開

始節點及虛擬完工節點，節線表示作業間之優先順序。

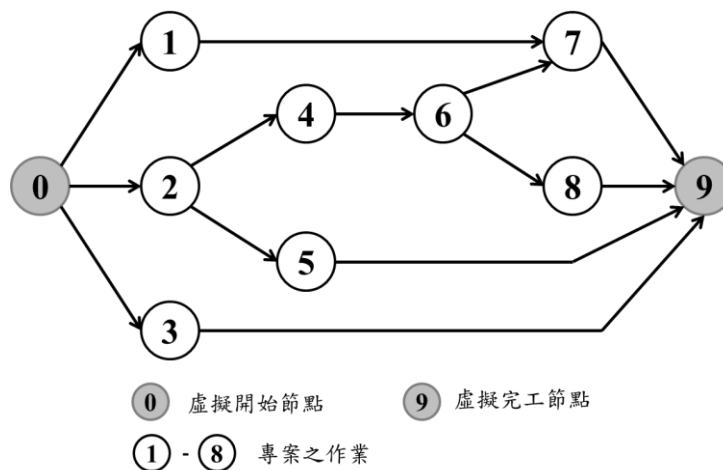


圖 2.2 PERT 網路圖

說明：節線表作業間之優先順序

此外，各作業之開始執行時間亦受到其前置作業之所需執行時間及資源需求影響，亦即資源使用量受到限制，故可定義 RCPSP 為「設法將一種以上的有限資源，隨著時間的進行分配至專案中之作業，並且達到專案之目標」；而通常專案之目標，可根據決策者所需分為：專案完成時間最小化、專案總成本最小化、最大化資源使用率等。

在此類問題中之有限資源，可概分為兩類，一為可更新資源(renewable resource)，即原先被先前作業占用，該作業完成後可被重複使用之資源，例如：人力資源、機械設備等等；另一則為不可更新資源(nonrenewable resource)，即以整個專案作為考量，不可重複使用之資源，例如：資金預算、原物料等。

以一例子說明，考慮一包含 5 個作業、1 種可更新資源且每期上限為 4 單位、1 種不可更新資源且資源上限為 8 單位之專案，各作業間的順序限制如圖 2.3(a)之 PERT 網路圖所示，各作業節點下為該節點之可能作業模式，以作業 1 為例，作業 1 有兩種作業模式，其中一種需使用 2 單位的可更新資源和 3 單位的不可更新資源，且需花費 4 單位時間；另一則需使用 4 單位的可更新資源和 1 單位的不可更新資源，且需花費 6 單位時間，而其最佳之作業排程下，可更新資源在各個時間點下的佔用情況，如圖 2.3(b)所示，其中方塊之長度表示該作業花費的時間長度，寬度表示該作業使用

可更新資源的使用量，由此圖可得知最小專案完工時間為 7，而所使用之不可更新資源量為 8。

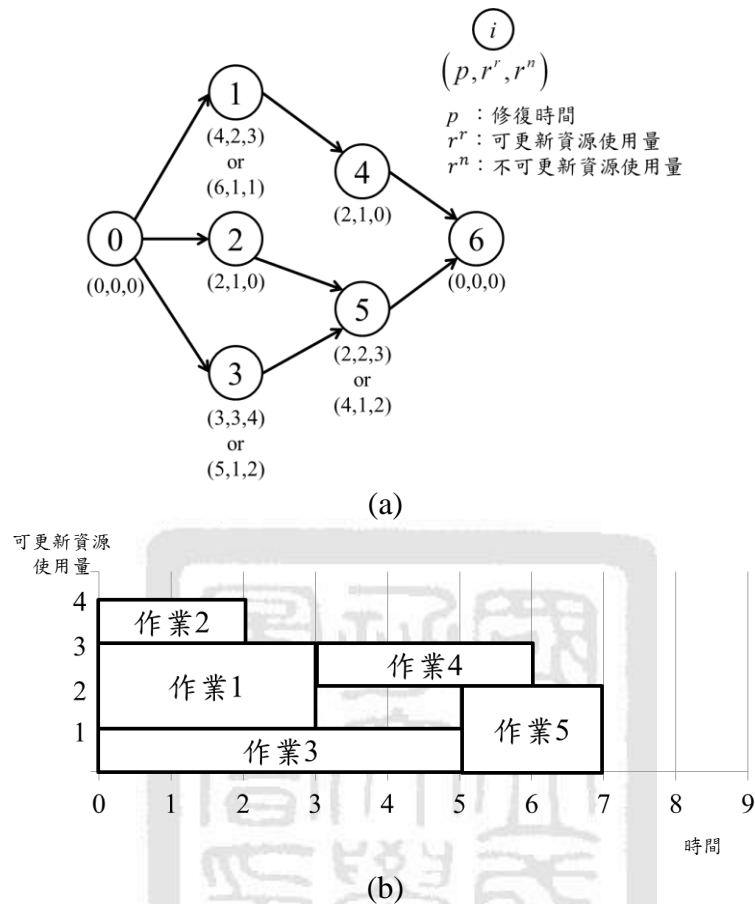


圖 2.3 MPCPSP 對應之 PERT 網路圖及最佳作業排程
【資料來源：Coelho and Vanhoucke (2011)】

根據過去文獻，本研究茲整理 RCPSP 及 MRCPSP 特性與假設如下：

1. 作業具有不可中斷性(non-preemptive)，即作業一旦開始執行後，不可中斷。
2. 在 RCPSP 中，各作業在任何時間點之執行時間及資源需求量固定且已知。
3. 在 MRCPSP 中，各作業在任何時間點之執行時間及資源需求量是由其選擇之作業方式決定，且各作業方式之執行時間及資源需求量固定且已知。
4. 資源之可用上限、作業之執行時間及資源需求量为整數。

1983 年，Blazexicz et al. (1983) 證明 RCPSP 為一 NP-hard 問題，其延伸問題 MRCPSP 亦因其作業可由不同作業方式加以執行，其複雜的程度更勝於 RCPSP。對

此，過去之研究發展了許多數學模式及求解演算法。本研究將繼續針對 RCPSP 之數學模式與求解演算法做文獻探討。

2.2.1 RCPSP 數學模式

Talbot (1982)，提出一 MRCPSPP 之數學模式，該數學模式中探討之專案可描述如下：專案中有 n 個作業，編號 1 至 n ，每一作業皆有多種作業方式，且具不可中斷性；專案中之有限資源有可更新資源及不可更新資源兩種；此數學模式將決定該專案之作業排程、作業方式及每期之資源分配，並將專案轉成 PERT 圖，在給定專案完工時間上限後，計算每個作業之最早及最晚執行時間。

以下將針對該模式進行說明：

模式之各符號定義

- N 節點之集合，包含一虛擬起始節點 (dummy start node) 及虛擬完成節點 (dummy finish node)，各別編號 0 及 $n+1$
- A 節線之集合，每一節線表兩作業間之優先順序關係
- R^r 可更新資源之集合， $r \in R^r$
- a_k^r 每一期之可更新資源 k 之資源上限
- R^n 不可更新資源之集合， $l \in R^n$
- a_l^n 整個專案下之不可更新資源 l 之資源上限
- M_i 作業 i 之作業方式集合， $m=1, 2, \dots, M_i$
- $d_{i,m}$ 作業 i 在其作業方式 m 下之所需作業時間
- $r_{i,m,k}^r$ 作業 i 在其作業方式 m 下之對可更新資源 k 之需求量
- $r_{i,m,l}^n$ 作業 i 在其作業方式 m 下之對不可更新資源 l 之需求量

T 專案完工時間之上限， $t=0,1,...,T$

es_i 作業 i 之最早可執行時間

ls_i 作業 i 之最晚可執行時間

$x_{i,m,t}$ 若等於 1 代表作業 i 在時間點 t 以作業方式 m 開始執行；反之等於 0。

數學模式

Talbot (1982) 提出之數學模式專案求取最小之整體完工時間，其目標式為：

$$\text{Minimize } \sum_{t=es_{n+1}}^{ls_{n+1}} tx_{n+1,m,t} \quad (2.1)$$

限制式

專案之優先順序限制：PERT 網路圖中之每一節線 (i, j) ，代表作業 j 須待作業 i 完成後，方可開始執行，故作業 j 之開始執行時間，必大於或等於作業 i 之完工時間，如式(2.2)。

$$\sum_{m=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + d_{i,m}) x_{i,m,t} \leq \sum_{m=1}^{M_j} \sum_{t=es_j}^{ls_j} tx_{i,m,t} \quad \forall (i, j) \in A \quad (2.2)$$

式(2.3)限制專案中之每一作業 i ，在整體專案下，只會有一作業方式及一開始執行時間。

$$\sum_{m=1}^{M_i} \sum_{t=es_i}^{ls_i} x_{i,m,t} = 1 \quad \forall i \in N \quad (2.3)$$

式(2.4)-(2.5)為專案下之資源限制。式(2.4)限制在專案下之每一時間點 t ，每一種可更新資源 k 所占用之資源量，不可超過其資源上限 a_k^r 。

$$\sum_{i=1}^n \sum_{m=1}^{M_i} r_{i,m,k}^r \sum_{s=\max\{t-d_i, es_i\}}^{\min\{t-1, ls_i\}} x_{i,m,s} \leq a_k^r \quad \forall k \in R^r, t=1,...,T \quad (2.4)$$

式(2.5)則限制每一種不可更新資源 l ，在整體專案下之使用量，不可超過其資源上限 a_l^n 。

$$\sum_{i=1}^n \sum_{m=1}^{M_i} r_{i,m,l}^n \sum_{t=es_i}^{ls_i} x_{i,m,t} \leq a_l^n \quad \forall l \in R^n \quad (2.5)$$

最後，定義變數類型及其值域範圍。

$$x_{i,m,t} \in \{0,1\} \quad \forall i \in N, m=1,\dots,M_i, t=1,\dots,T \quad (2.6)$$

此模式雖為 MRCPSP 之數學模式，然而藉由設定 MRCPSP 中每一作業之作業方式只有一種，亦可針對 RCPSP 進行求解。另外，Talbot (1982)亦提出一可窮舉所有可行解之技巧，試圖找尋一最小總完工時間之作業排程。

Mingozi et al. (1998) 則對 RCPSP 提出另一種整數規劃模式，首先在所有作業中，找尋可平行作業的作業集合，即該集合須滿足下列條件：

- (1) 集合內之作業不違反每期之可更新資源限制。
- (2) 集合中之作業互相不具有優先順序關係。

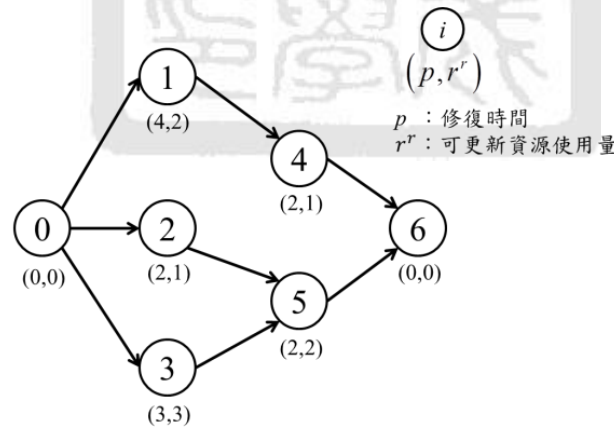


圖 2.4 PERT 網路圖

以圖 2.4 之 PERT 網路圖為例，假設可更新資源上限為 4，其中，作業 1 及作業 2 可平行執行，因兩者無優先順序限制，且同時執行只需花費 3 單位的可更新資源，而作業 1 及作業 3 則不可同時執行，因需花費 5 單位的可更新資源，超過資源上限，

違反第一個條件。另外，作業 1 及作業 4，因有順序限制的關係存在，亦不可同時執行。其數學模式便是決定每一個時間點下要執行的作業集合為哪些，且須滿足各作業之所需的執行時間、作業間優先順序及資源限制，並以分枝界限法(Branch and Bound, B&B)進行求解。

Krüger and Scholl (2010) 則將資源之運輸時間納入問題考量，即每當完成作業時，將該作業所占用之可更新資源運輸至下一作業所在位置時，需耗費一運輸時間，使 RCPSP 更具一般性，並且提出一混整數數學模式予以求解。

2.2.2 RCPSP 相關演算法

Li and Willis (1992) 探討在 RCPSP 之資源使用效率問題，認為若增大資源之使用效率便可有效降低專案之完工時間，其交互應用前推式作業表單串聯法(Forward scheduling procedures)及後推式作業表單串聯法(Backward scheduling procedures)，試圖找尋一在有限資源下，最快完工時間之作業排程。

Debels and Vanhoucke (2005) 則以 Li and Willis (1992) 之前推式/後推式作業表單串聯法為解碼方式，設計一雙族群之基因演算法(Bi-population Based Genetic Algorithm, BPGA)求解 RCPSP。Van Peteghem and Vanhoucke (2010) 再依此 BPGA 演算法為基礎，設計一演算法，並可求解 MRCPSPP 及可允許作業中斷之 P-MRCPSPP。

Coelho and Vanhoucke (2011) 以 Talbot (1982) 所提出之數學模式為基礎，先將 MRCPSPP 轉變成為 RCPSP。以圖 2.2(a)為例，將具多作業方式之作業節點，如圖 2.2(a)中之節點 1、3、5，各依其作業方式拆成獨立之作業節點並重新編號，如原節點 1 拆成新節點 1、2，且新作業節點 1、2 只需擇一執行，結果如下頁圖 2.5 所示；接著將此新 RCPSP 視為一可滿足性問題 (Satisfiability Problem, SAT)，以 SAT solver 演算法進行求解，並與過去文獻中，求解表現最佳之演算法 (Van Peteghem and Vanhoucke, 2010) 進行比較，雖然 SAT solver 演算法平均所花費的時間較長，但卻可找到更佳的可行解。

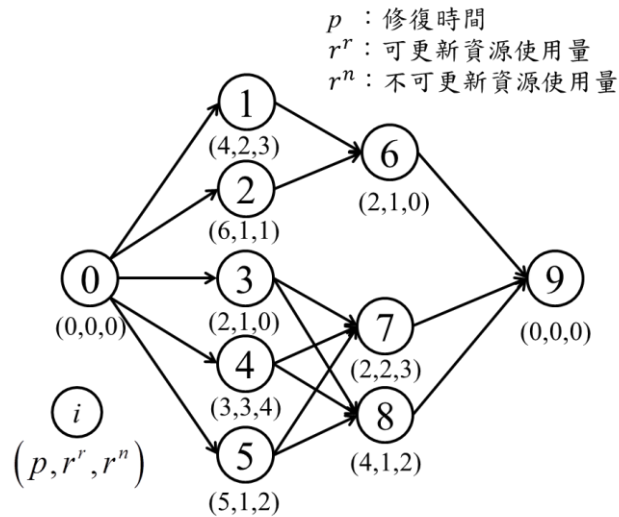


圖 2.5 MRCPS 修正後之 RCPSP 示意圖
 【資料來源：Coelho and Vanhoucke (2011)】

2.3 小結

災後緊急搶修作業，在過去文獻中，大多將搶修時程最小化做為主要考量目標，決定欲將修復之路段，未考慮後續可能因工作隊之指派，而無法達到預設之目標。近年來雖已將搶修工作隊之指派納入考量，但大多在已決定修復那些路段情況下，所做之作業排程規劃，並運用最佳化方法建構數學模式來求解，其數學模式常以旅行推銷員問題、時空網路圖或單一商品流量問題等來建構，但常因問題過於龐大，造成求解時間過長，雖有研究對此設計求解演算法，但皆只能找到可行解。對此本研究在考量「欲修復路段決策」及「搶修作業排程與資源指派」二者連續關係下，以 RCPSP 之概念來建構數學模式，希望在 RCPSP 的基礎上，能夠找到更有效率之解法。

而過去文獻較少考量修復路段有多種修復方式之可能性，即所需修復時間可能因投入到該修復路段之資源組合不同而有所不同，例如：工作隊能力不同的影響或允許工作隊合作，雖然加入多種修復方式會造成問題更加複雜，但此一考量能使問題更加貼近實務，以利決策單位能夠得到夠有效率之搶修計劃。後面章節將開始介紹本研究之數學模式。

第三章

模式建構

本章節將進行本研究之問題描述，並討論該問題之兩個特例：(1) 搶修工作隊只有一組、(2) 搶修工作隊有無限組，說明為何可以最小展開樹問題及最短路徑樹問題來有效的求解。介於一組及無限多組之間之一般性問題，則以本研究提供之數學模式求解；最後，將討論延伸問題，並提出延伸問題之數學模式。

3.1 問題描述

如何於災後在最短時間內進入災區進行救援或疏散等救援工作，一直是災後救援工作之重要議題，因此，道路網搶修計劃之好壞是影響災後救援效率的關鍵之一。本研究探討的是在一災後網路下之搶修作業與資源的排程問題，可定義如下：

假設有一連通網路 (connected network) $G=(N, A, C)$ ；其中 N 為網路內之節點 (nodes) 集合，且節點分為三種，起點、可對外連通節點及不可對外連通節點。起點，即工作站位置，為工作隊出發之節點；可對外連通節點，為在災難發生後，可經由無損毀道路通往之節點；不可對外連通節點，即搶修計劃下目標打通之災點，如圖 3.1 中黑色節點，因周遭路段皆毀損，而無法與外界連通。 A 為網路中所有無向節線 (undirected arcs) 之集合，節線分為兩種，一為可通行，即完好無損壞可供車輛通行之節線，另一為不可通行，即損壞或受阻之不可通行節線，如圖 3.1 中虛線所示之交通受阻路段，又因屬無向節線，故各路段之修復方向或通行方向，皆屬雙向； C 為節線修復所需時間成本，其中可通行之節線之時間成本為 0；本研究之網路修復問題描述如下：在不違反資源限制下，找到一連通子網路 $G_I=(N, A_I, C)$ ，其中 $A_I \in A$ ，且其整體修復時間成本為最小，並決定何時要指派哪組工作隊去修復哪條不可通行節線。

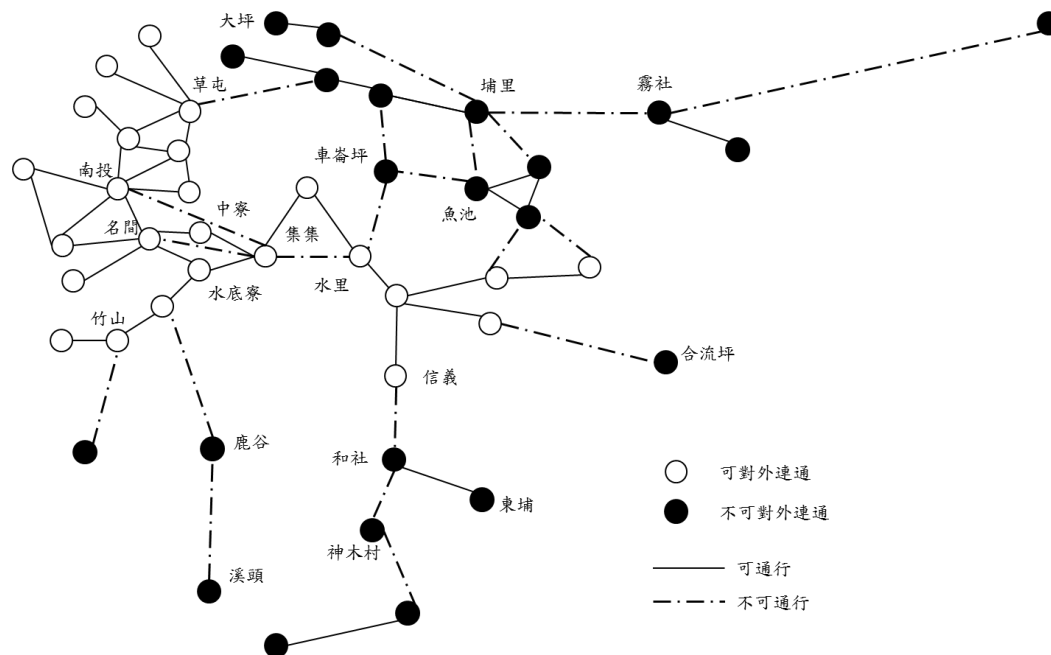


圖 3.1 九二一大地震公路受損示意圖
【資料來源：交通部公路局】

相同於 RCPSP，此問題中的有限資源亦可概分為可更新資源及不可更新資源，但本研究主要討論可更新資源，即人力與機具資源，且人力與機具可組合為一組工作隊 (recovery team)，故不同工作隊可能因配備不同而修復能力有所不同，而各路段的修復方式也因此可能為一種以上。

問題目標式則可分為兩種：

1. 目標為最小化最後完工時間，以 **MAX** 代表。
2. 目標為最小化各災區等待打通的時間總和，以 **SUM** 代表。

3.2 問題假設

為適當簡化模式的發展與界定模式的使用限制，本研究有下列基本假設：

1. 外界可能因災區內之運輸網路及通訊網路之損壞，而難以完全掌握災區整體情形，故災害發生初期，須先蒐集整體災害資訊。因此，本研究假設災區損害資訊已完整蒐集，且各工作隊之搶修能力皆可正確掌握，使決策單位可在整體考量下作出決策。

2. 當災害發生時，原位在災區內之部分工作隊便可立刻進行搶修作業，對此本研究不加考量，意即本研究皆假設當搶修計劃擬定完成才開始指派工作隊進行搶修作業，而搶修時程亦是由此時開始計算。另外，實務上，決策者可將已指派之工作隊，視為已知資料，進行求解。
3. 假設資源運輸時間遠小於各道路修復時間，因此可將資源運輸時間視為零。
4. 搶通各災區的優先權皆相同，即對決策單位而言，各災區皆一視同仁。
5. 所有損壞路段之各修復方式，所需搶修工時、工作隊數皆為已知，且不隨搶修之時程而有所變動。
6. 各工作隊之修復能力不隨時間而有所減弱，即在整體搶修時程下，各工作隊之能力固定，故不考量工作隊於搶修完一個災點後，需要回工作站補充修復能量之情況，且一旦工作隊開始進行某路段之搶修工作，便不允許工作之中斷。

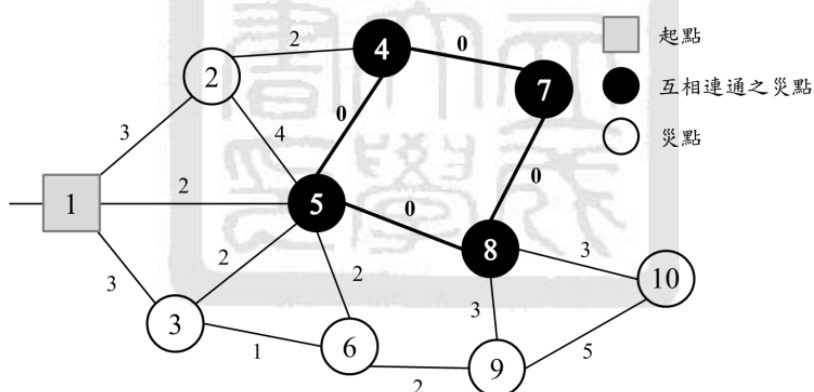


圖 3.2 具無毀損子網路之網路圖

3.3 網路縮減

在現實之災後道路網，可能會在網路中找到一子網路 $G' = (N', A', C)$ ，其中 $N' \in N/S$ ， $A' \in A$ 且 $C=0$ ，意即在 G' 內之節點，雖不與起點連通卻可互相連通，如圖 3.2 中，其中節點 4、5、7、8 及節線所構成之子網路（節線上為所需修復時間）。

故當打通子網路 G' 中任意一點，便可視同打通在 G' 其它節點，又本研究不考慮

資源運輸的時間，故若將 G' 以一虛擬節點代替之，並不會影響最後的求解結果，除此之外，此簡化可將節點及節線數量縮小，縮減問題大小，使模式求解效率提升，因此，在進行求解前先對網路作以下縮減步驟：

【步驟一】 對網路圖中所有節點，依照編號順序，輪流自每一個節點進行廣度搜尋法 (Breadth-first search, BFS)，找尋 $c=0$ 的節線。

【步驟二】 步驟一結束後，可得到許多棵樹 (BFS tree)，計算各棵樹所含節點數。

【步驟三】 節點數大於一之 BFS tree，表示該 BFS tree 中的節點可互相連通，打通任一節點，其它節點亦可視同打通，故可將該 BFS tree 合併為一節點 d ，並依據原圖建立新節線 (d, j) ，其中，若節點 j 存在一條以上之節線 (d, j) ，則可將之合併為一，其修復方式亦可合併，成為多種修復方式。接著，重複進行步驟一至步驟三，直到無法找到節點數大於一之 BFS tree 為止。

圖 3.2 修正後之網路圖，如圖 3.3 所示。

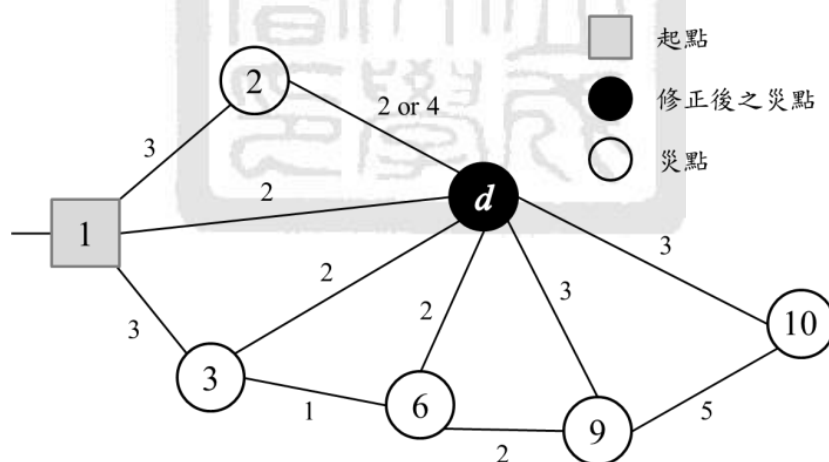


圖 3.3 修正後之網路圖

3.4 一組及無限多組工作隊之特殊情況

因本研究只探討最快將網路修復至各節點皆可互相通往的情況，並不需要修復所有不可通行節線，因此有以下最佳解性質：

【最佳解性質】最佳搶修計畫下的搶修路徑，其結構必為一展開樹 (Spanning Tree)。

說明：假設有一 n 個節點的網路， a_i 表第 i 個節點。由於修復路徑必須自起點連通所有節點，故修復路徑必為一展開連結圖。接著，假設最佳解性質不成立，則最佳搶修路徑必含迴圈 $a_1 - a_2 - a_3 - \dots - a_\ell - a_1$ ，共 ℓ 條節線。若從該迴圈移除一條節線，即不修復該條路段，則此路徑也可連通所有災點，且有較小的完工時間。故由以上可知，最佳搶修路徑為一不含迴圈之展開連結圖，即最佳路徑之結構為展開樹。

【觀察 1】若網路圖結構為樹 (Tree-network)，則道路搶修問題等同 RCPSP 問題。

說明：因網路圖本身結構為樹，若要打通所有的節點，便須修復所有的節線，故網路本身便為最佳的搶修路徑，且可容易從樹的結構看出各路段的修復方向，作業之間的順序限制便可得知，在第一章中已說明如何在已知各作業間的順序限制下，繪出對應之 PERT 網路圖，故藉由 PERT 網路圖，便可將問題轉化為 RCPSP 進行求解。

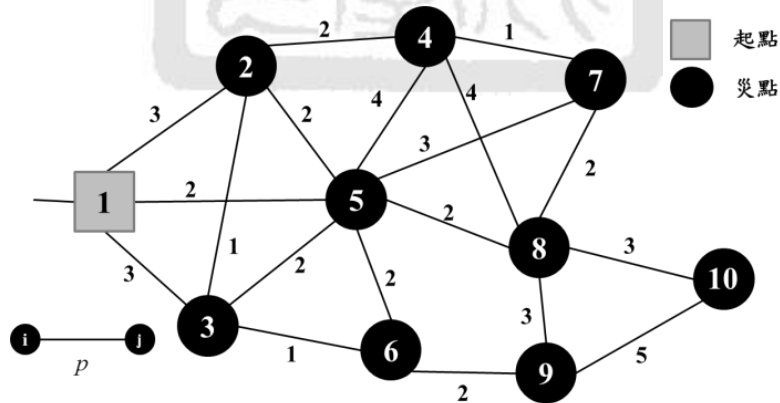


圖 3.4 災後損壞網路示意圖

若假設所有工作隊之能力皆相同，且各不可通行路段只能由一組工作隊修復，從最佳解性質可以有以下定理：

【定理 1】 當問題目標為 MAX 且工作隊只有一組時，最佳修復路徑為該網路之最小展開樹 (Minimum Spanning Tree, MST)。

說明：在只有一組工作隊可進行作業的情況下，每次作業都需等該工作隊結束上一作業才可開始進行，故整體完工時間即等於所有作業的作業時間總和，且由最佳性質已知最佳路徑為一展開樹，若將作業時間視為網路中各節線之權重，且因目標為最小化最後完工時間，故等同在網路中找最小權重總和之展開樹，此為常見之 MST 問題。以圖 3.4 為例，圖 3.4 為一損壞道路網，網路中所有路段皆毀損，而路段上數字表其修復所需時間，則在只有一組工作隊的情況下，最佳修復路徑如圖 3.5，完工時間為 16 單位。MST 問題目前已有許多演算法可在多項式時間下求解，如：Prim's 演算法及 Kruskal's 演算法。

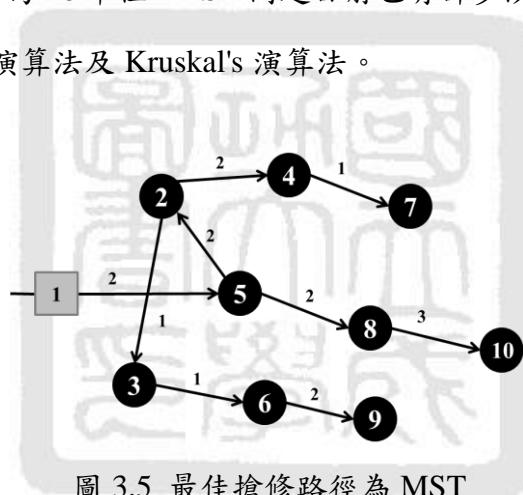


圖 3.5 最佳搶修路徑為 MST

【定理 2】 當問題目標為 SUM 且工作隊只有一組時，問題為 NP-hard。

說明：見 Averbakha and Pereira (2012)。

【觀察 2】 當問題目標為 SUM 且工作隊只有一組時，若網路圖結構為樹 (tree-network)，則問題可在 $O(n \log n)$ 下求得最佳解。

說明：觀察 1 說明若網路圖結構為樹，可以由網路圖直接定義各作業間的順序限制關係，問題便可轉化為單一資源下，目標最小化 n 個作業完工時間加總的作業排程問題，且有一演算法可在 $O(n \log n)$ 下求得最佳解 (Brucker, 2007)。

【定理 3】 當問題目標為 **MAX** 或 **SUM** 且工作隊組數無限時，最佳修復路徑為該網路之最短路徑樹 (Shortest path tree, SPT)。

說明：假設有夠充足的工作隊，並且針對每一個節點指派一組工作隊負責打通，由最佳解性質已知最佳路徑為展開樹，可知最佳搶修路徑會有一條以上的分支，在每條分支末端節點的打通時間等於該分支上所有路段修復時間的加總，因目標為最小化，所以每一條分支必為該分支末端節點的最短路徑，又最短路徑上的子路徑亦為最短路徑，故各節點的打通時間等同於由起點到該點之最短路徑上所有路段修復時間加總。若將作業時間視為網路中各節線之權重，則問題可轉化為找尋起點對其他節點的一對多最短路徑問題；其中，當問題目標為 **MAX** 時，最後完工時間為最長之最短路徑末端節點的打通時間，如圖 3.6，最長之最短路徑為 1-5-8-10，故當問題目標為 **MAX** 時，最後完工時間為 7。若今網路中有一個以上之可通行節點，則只需加入一虛擬節點，連往所有初始起點，再以虛擬節點為起點，找尋 SPT。SPT 問題目前已有許多演算法可在多項式時間下求解，如：Dijkstra 演算法等。

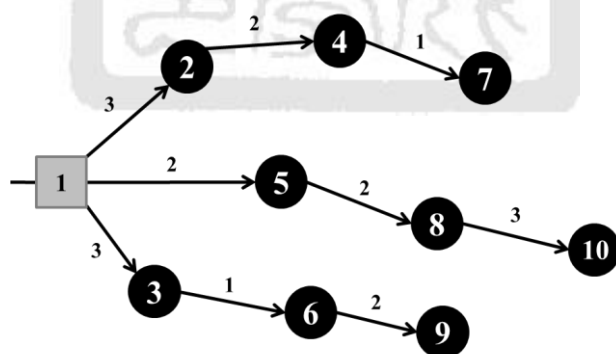


圖 3.6 最佳搶修路徑為 SPT

然而現實中工作隊組數是介於一組及無限多組之間，而後續「搶修作業排程與資源指派」對於最後完工時間會有很大的影響，如第一章所提到，搶修路徑若以路段所需修復時間為決策依據：「只選擇所需修復時間最短的路段」，其路徑為 MST，雖在一組工作隊下為最佳搶修路徑，但在兩組工作隊下，卻非最佳搶修路徑；或「針對每

一災點選擇能最快通往的搶修路徑」，其路徑為 SPT，同樣在兩組工作隊下非最佳搶修路徑。然而，雖已知搶修路徑為一展開樹，但要窮舉出一網路圖中所有的展開樹所耗費的成本會非常龐大，並不具時間效益，故針對工作隊組數介於一組及無限多組之一般情況，本研究將建構一數學模式進行求解。

3.5 數學模式

以下將針對本研究之問題建構一整數規劃模式，並稱作 GRB1，探討在路段具有多種修復方式情況下的道路搶修問題，模式如下：

3.5.1 符號定義

集合

- N 表所有災點（節點）構成之集合， $i, j \in N$
- A 表所有路段（節線）構成之集合， $(i, j) \in A$
- S 表起點（節點）構成之集合， $S \in N$
- R 表工作隊的種類， $k \in R$
- $M_{(i,j)}$ 表修復路段 (i,j) 之路段修復模式， $m=1, 2, \dots, M_{(i,j)}$

參數

- $p_{(i,j),m}$ 表在修復模式 m 下，修復路段 (i,j) 所需修復時間
- $r_{(i,j),m,k}$ 表在修復模式 m 下，指派修復路段 (i,j) 的工作隊 k 數量
- \bar{r}_k 表每期工作隊 k 的使用數量上限， $k \in R$
- T 表搶修完工時間上限， $t=0,1,\dots,T$
- B 表一極大常數值

決策變數

$$x_{(i,j),m,t} = \begin{cases} 1, & \text{於時間 } t \text{ 開始以修復模式 } m \text{ 修復路段 } (i,j) \\ 0, & \text{O.W.} \end{cases}$$

F_i 災點 i 的打通時間

3.5.2 整數規劃模式

當問題為 **MAX** 時，即目標最小化最後完工時間，目標式如式(3.1)：

$$\text{Minimize } \max_{i \in N} F_i \quad (3.1)$$

當問題為 **SUM** 時，即目標最小化各災點的打通等待時間總和，目標式如式(3.2)：

$$\text{Minimize } \sum_{i \in N} F_i \quad (3.2)$$

限制式

式(3.3)-(3.4) 定義每一個節點的打通時間。

$$F_j = \sum_{\substack{(i,j) \in A \\ i \in N, i \neq j}} \sum_{m=1}^{M_{(i,j)}} \sum_{t=0}^T (t + p_{(i,j),m}) x_{(i,j),m,t} \quad \forall j \in N/S \quad (3.3)$$

$$F_j = 0 \quad \forall j \in S \quad (3.4)$$

式(3.5) 是針對起點的限制，其限制在整體修復時程下，不會有修復方向通往起點的修復作業。

$$\sum_{\substack{(i,j) \in A \\ i \in N/S, j \in S}} \sum_{m=1}^{M_{(i,j)}} \sum_{t=0}^T x_{(i,j),m,t} = 0 \quad (3.5)$$

式(3.6) 為各修復作業之時序限制。針對所有節線 $(i,j) \in A$ ，若該節線從時間 t 開始修復，則到該節線之修復作業完成前，不會有任何修復工作是從節點 j 出發。

$$\sum_{(j,i') \in A} \sum_{m'=1}^{M_{(j,i')}} \sum_{s=0}^{\min\{t+p_{(i,j),m}-1, T\}} x_{(j,i'),m',s} \leq (1 - x_{(i,j),m,t}) B \quad \forall (i,j) \in A, m=1, \dots, M_{(i,j)}, t=0, 1, \dots, T \quad (3.6)$$

式(3.7) 限制所有非起點之節點接需被打通，且只會有一個修復作業去打通該節點，以此避免不必要之搶修作業。

$$\sum_{\substack{(i,j) \in A \\ i \in N}} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} = 1 \quad \forall j \in N/S \quad (3.7)$$

式(3.6)-(3.7)亦可限制修復路徑之結構需為展開樹（因從最佳解性質已知最佳解必為網路中一展開樹），避免修復路徑出現迴圈或多餘的節線；若有搶修路徑有迴圈發生，如圖 3.7(a)之路徑，可發現節點 A 應於 $t=10$ 時，藉由搶修節線(C, A)來打通，但於 $t=2$ 時，卻有一搶修作業從 A 開始，明顯不合理，故藉由式(3.6)之時序限制，便可排除修復路徑有迴圈發生；而圖 3.7(b)，違反式(3.7)，因有兩個修復作業方向通往節點 B，造成不必要之搶修作業。

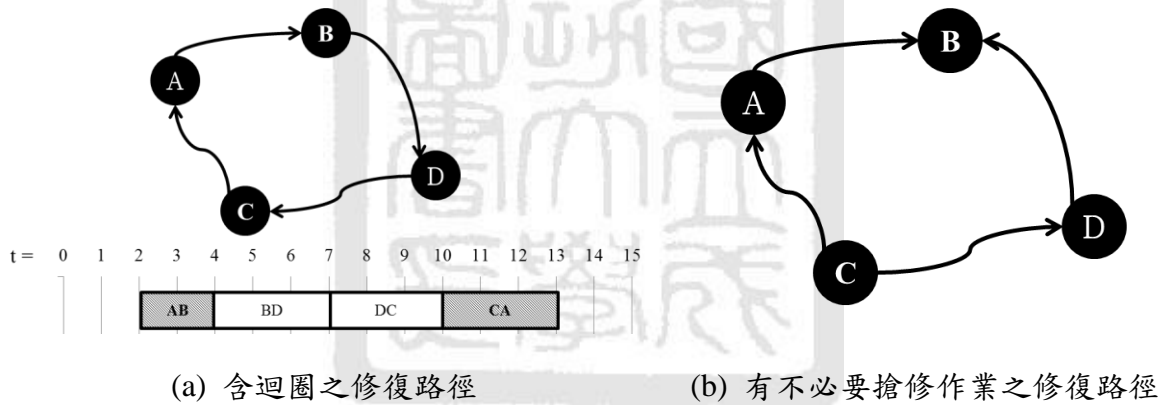


圖 3.7 錯誤的修復路徑範例

而在節 3.3 的縮減網路步驟，除了可縮減問題大小外，若網路中存在該節中描述的子網路 G' （即網路中有節點可互相連通），會使式(3.6)之功能無效，造成搶修路徑出現迴圈。以圖 3.8(a)為例，節線旁為該節線之修復時間，其中網路中之 4-5-7-8 形成的子網路，彼此之間的節線所需修復時間為 0（無毀損），若有一如圖 3.8(b)的搶修路徑出現，此種搶修路徑雖然明顯不為可行解（因修復路徑有迴圈產生，且該迴圈並不外界連通），但式(3.6)並無法將此種路徑排除，使得模式所求的路徑出現迴圈，故在以模式求解前，必須先進行節 3.3 之網路縮減步驟。

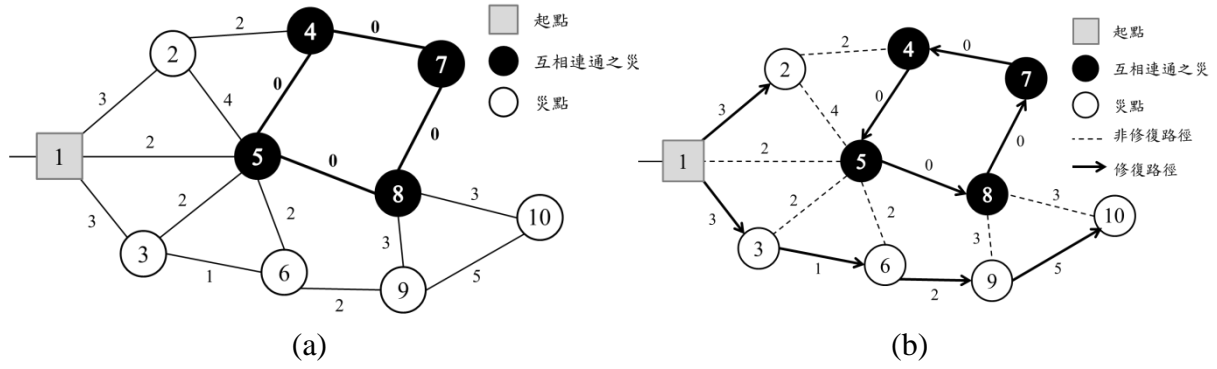


圖 3.8 未做網路縮減之網路及迴圈路徑

式(3.8)，為資源限制。此條限制式是從 Talbot (1982) 提出 MRCPSPP 之數學模式中，針對可更新資源限制延伸而來。其限制在整個搶修時程下之每一時間點下進行的搶修作業所占用之工作隊 k 數量，總占用量不可超過該工作隊數上限。

$$\sum_{\substack{(i,j) \in A \\ i \in N, j \in N/S}} \sum_{m=1}^{M(i,j)} r_{(i,j),m,k} \sum_{s=\max\{t-p_{(i,j),m}+1, 0\}}^t x_{(i,j),m,s} \leq \bar{r}_k \quad \forall k \in R, t=1, 2, \dots, T \quad (3.8)$$

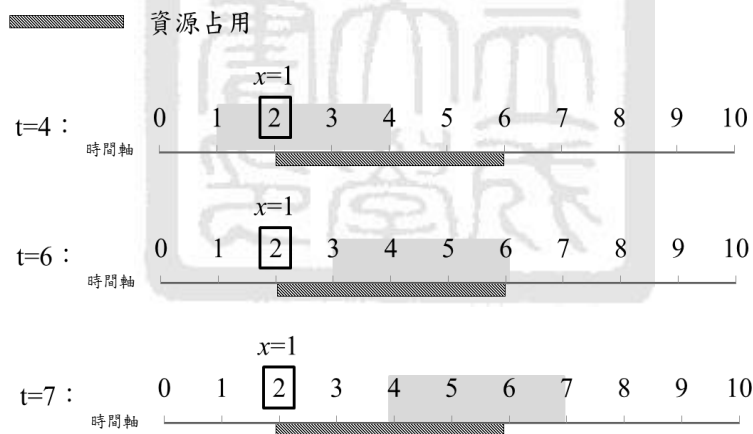


圖 3.9 資源限制說明

假設有一作業，其所需修復時間為 4，開始修復時間為 2，並且會占用一組工作隊，則該工作隊在時間軸下的被占用情形可如圖 3.9 黑色斜線長條圖表示。若欲檢查時間 t 下作業占用工作隊的情形，則須查看該作業開始時間是否介於 $Range_t = [\max\{t - p_{(i,j),m} + 1\}, t]$ 範圍內。圖 3.9 分別討論三種情況，當檢查時間點 $t=4$ 時，需檢查圖中灰色長條圖，即 $Range_4$ 之範圍，可發現作業開始時間位於 $Range_4$ 內，表

示 $t=4$ 時，該作業有占用到工作隊；當檢查時間點 $t=6$ 及 $t=7$ 時，作業開始時間並未在其對應之 $Range_6$ 及 $Range_7$ 內，表示該作業並未占用到工作隊，其中因 $t=6$ 為作業完成時間點，其占用的工作隊在該時間點下，便可立即供其他作業使用。

式(3.9)-(3.10)，則定義變數類型及其值域範圍。

$$x_{(i,j),m,t} = \{0,1\} \quad \forall (i,j) \in A, m=1,\dots,M_{(i,j)}, t=0,1,\dots,T \quad (3.9)$$

$$F_j \geq 0 \quad \forall j \in N \quad (3.10)$$

若所有節線之修復方式皆只有一種，即對所有節線 (i,j) 而言， $M_{ij}=1$ ，則此模式亦可求解單一種修復方式且無資源運輸時間之情況。實際上，依照路段之損壞狀況，某些路段只能由限定之工作隊來進行修復作業，則只需將之視為不同種之修復方式即可。若要考慮工作隊之合作情形，亦只需將之當作各損壞路段之修復方式之一，依此便可處理多種修復方式相關延伸問題，舉例來說，假設有一路段的修復模式如表 3.1，代表該路段能由工作隊 A 和 B 獨自進行搶修，但不能由工作隊 C 獨自搶修，而合作搶修則是能由工作隊 A、B 合作或工作隊 A、C 合作。

表 3.1 多種修復方式說明

編號	使用的工作隊			所需維修時間
	工作隊 A	工作隊 B	工作隊 C	
1	✓			5
2		✓		3
3	✓	✓		1
4	✓		✓	2

另外，若欲求得一整體完工時間與無限多組工作隊情況相同之工作組數下限，可先由此模式針對無限多組工作之情況進行求解，得到最小完工時間 T_{min} 後，將此完工時間作為搶修完工時間上限，即各節點打通時間上限，並最小化時間點中最大之工作隊數需求量，如式(3.11)。

$$\text{Minimize } \text{Max}_{t=1,\dots,T_{min}} \left\{ \sum_{\substack{(i,j) \in A \\ i \in N, j \in N/S}} \sum_{m=1}^{M_{(i,j)}} r_{(i,j),m,k} \sum_{s=\max\{t-P_{(i,j),m}+1,0\}}^t x_{(i,j),m,s} \right\} \quad (3.11)$$

3.6 模式延伸

本節將考慮其他不同因素下之災後道路網緊急搶修問題，例如考量工作隊移動時間或只修復特定災點之修復計劃，再由原模式延伸出相對應之數學模式。

3.6.1 延伸一：修復特定災點

在現實搶修計劃中，有時可能因資源不足等其他考量，而只優先考慮急迫性較高的災點，而部分較不急迫的災點則允許暫時不打通。首先，定義 N^{urgent} 為較急迫的節點集合且 $N^{urgent} \in N$ ，並將式(3.7)改為式(3.12)及(3.13)，表對於較急迫的節點必須有一作業去打通，而較不急迫的節點則允許不被打通。

$$\sum_{i \in N} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} = 1 \quad \forall j \in N^{urgent} \quad (3.12)$$

$$\sum_{i \in N} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} \leq 1 \quad \forall j \in N / \{N^{urgent} \cup S\} \quad (3.13)$$

3.6.2 延伸二：考量工作隊移動時間

3.5 節所述之整數規劃模式主要假設工作隊的移動時間遠小於各道路修復時間，因此可將移動時間視為零；但當災難範圍很大時，在整體修復計劃時間軸下，工作隊的移動時間將占很大的比例，對最後完工時間的影響不能忽視，進而須考慮移動時間對整體完工時間之影響。而 3.5 節所提供之模式，因無決定資源的如何移動，因而無法考量資源運輸時間。

Krüger and Scholl (2010) 提出考量資源運輸時間之 RCPSP 之數學模式，將資源視作商品，並以單一商品網路流問題的概念來建構模式，將資源運輸時間定義為：資源從作業 i 的位置移置作業 j 的位置所耗費之時間成本，此時間成本為給定之一常數值。故本節以該模式為基礎，針對道路搶修問題，建構一考慮固定資源運輸時間之整數規劃模式，並將此模式稱作 GRB2。

在介紹模式前，需先在原網路中加入一虛擬節點及多條虛擬節線，如圖 3.10 中，在原網路加入一虛擬節點 f ，且對於所有節點皆有一條通往 f 的虛擬有向節線，其目的使最後所有資源流量接流向此虛擬節點，以滿足模式中之流量守恒限制。此外，模式無法求解工作隊合作的情況，亦即需假設各作業只能指派一組工作隊來進行搶修。

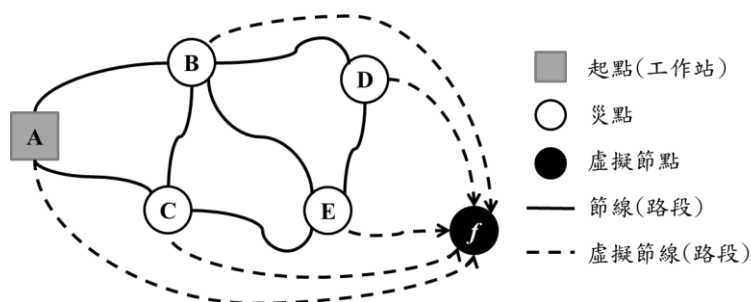


圖 3.10 虛擬節點與節線示意圖

符號定義

模式額外定義的符號如下：

集合

f 工作隊的匯集節點

參數

$d_{i,j}$ 從節點 i 到節點 j 所需運輸時間

$r_{(i,j),k}$ 工作隊 k 修復路段 (i,j) 所需的修復時間

決策變數

$$f_{(i,j),k,t} = \begin{cases} 1, & \text{工作隊 } k \text{ 於時間 } t \text{ 完成修復路段 } (i,j) \\ 0, & \text{O.W.} \end{cases}$$

$$z_{h,(i,j),k} = \begin{cases} 1, & \text{工作隊 } k \text{ 從災點 } h \text{ 到災點 } i, \text{ 並修復路段 } (i,j) \\ 0, & \text{O.W.} \end{cases}$$

$x_{h,(i,j),k}$ 表從災點 h 到節點 i 修復路段 (i,j) 之工作隊 k 數量

其中，決策變數 $z_{h,(i,j),k}$ 及 $x_{h,(i,j),k}$ 說明如下：

假設有一修復路徑如圖 3.11(a)，圖中實線表示修復路徑，虛線表示工作隊的移動方向，虛線段(B, A)表示有工作隊從節點 B 至節點 A 來進行節線(A, C)的修復作業，且產生一時間成本 3；另外，亦有不會產生時間成本的情況，如圖 3.11(b)，此路徑並不會產生運送時間成本，因工作隊每完成一個修復作業，只需從該節點繼續下一個修復作業。

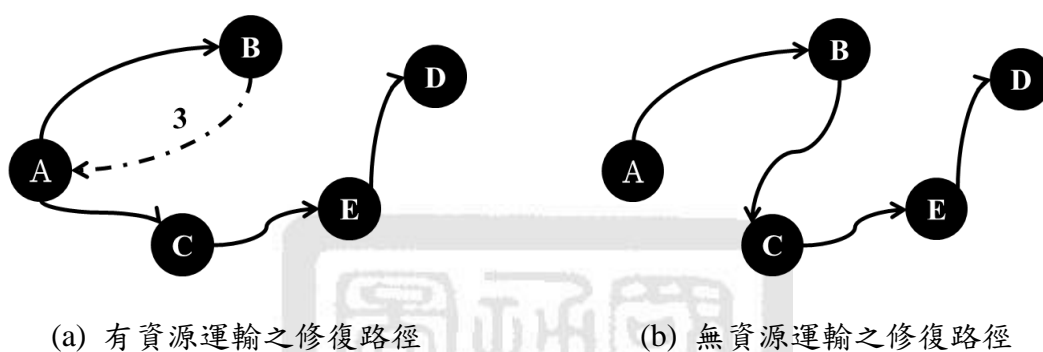


圖 3.11 資源運輸示意圖

整數規劃模式

以 Krüger and Scholl (2010) 所提出之可考量資源運輸時間之 RCPSP 數學模式，針對延伸問題建構一整數規劃模式，稱作 GRB2，問題之目標式相同於式(3.1)及(3.2)，限制式則如下：

限制式

式(3.14)-(3.15)，定義每一個災點的打通時間。

$$F_j = \sum_{(i,j) \in A} \sum_{k \in R} \sum_{t=0}^T (t + p_{(i,j),k}) f_{(i,j),k,t} \quad \forall j \in N \quad (3.14)$$

$$F_j = 0 \quad \forall j \in S \quad (3.15)$$

式(3.16)，限制在整體修復時程下，並不會有修復方向修往起點的修復作業。

$$\sum_{\substack{(i,j) \in A \\ i \in N/S, j \in S}} \sum_{k \in R} \sum_{t=0}^T f_{(i,j),k,t} = 0 \quad (3.16)$$

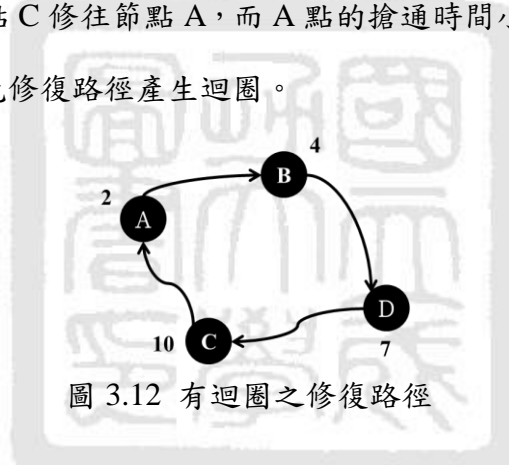
式(3.17)，限制所有非起點之節點，只有一修復方向通往該節點之作業，以此避免不必要之搶修作業。

$$\sum_{\substack{(i,j) \in A \\ i \in N}} \sum_{k \in R} \sum_{t=0}^T f_{(i,j),k,t} = 1 \quad \forall j \in N/S \quad (3.17)$$

式(3.18)-(3.20)，決定各災點的先後順序；當路段 (i,j) 決定要修復，則災點 j 的打通時間必大於災點 i 的打通時間加上路段 (i,j) 的修復時間，如式(3.18)。

$$F_j - F_i \geq \sum_{k \in R} p_{(i,j),k} \sum_{t=0}^T f_{(i,j),k,t} - \left(1 - \sum_{k \in R} \sum_{t=0}^T f_{(i,j),k,t} \right) B \quad \forall (i,j) \in A \quad (3.18)$$

如同式(3.6)，此式亦可避免修復路徑產生迴圈，如圖 3.12，節點上為其搶通時間，其中節線(C,A)表從節點 C 修往節點 A，而 A 點的搶通時間小於 C 點的搶通時間，違反式(3.18)，藉此可避免修復路徑產生迴圈。



若工作隊 k 欲從災點 h 到災點 i 去修復路段 (i,j) ，則災點 j 的打通時間必比災點 h 的打通時間加上工作隊 k 從災點 h 運送到災點 i 的運送時間再加上路段 (i,j) 的修復時間來的晚，同樣也比災點 i 的打通時間加上工作隊 k 從災點 h 運送到災點 i 的運送時間再加上路段 (i,j) 的修復時間來的晚，如式(3.19)及(3.20)。

$$F_j - F_h \geq d_{h,i} + p_{(i,j),k} - (1 - z_{h,(i,j),k}) B \quad \forall h \in N, (i,j) \in A, k \in R \quad (3.19)$$

$$F_j - F_i \geq d_{h,i} + p_{(i,j),k} - (1 - z_{h,(i,j),k}) B \quad \forall h \in N, (i,j) \in A, k \in R \quad (3.20)$$

式(3.21)，若路段 (i,j) 並沒有要修復，則此路段不允許工作隊在移動時經過。

$$\sum_{h \in N} z_{h,(i,j),k} \leq \left(\sum_{t=0}^T f_{(i,j),k,t} \right) B \quad \forall (i,j) \in A, k \in R \quad (3.21)$$

式(3.22)，若工作隊無移動的動作，則其資源流量必為 0。

$$x_{h,(i,j),k} \leq z_{h,(i,j),k} B \quad \forall h \in N, (i,j) \in A, k \in R \quad (3.22)$$

式(3.23)-(3.26)，為資源流量守恒限制；式(3.23)-(3.24)，若路段 (i,j) 有修復，則前往災點 i 去修復路段 (i,j) 的工作隊 k 數量必等於其所需隊數，且修復完後從災點 j 前往其他節點的工作隊 k 數量，亦等同其到達的數量。

$$r_{(i,j),k} \sum_{t=0}^T f_{(i,j),k,t} = \sum_{h \in N} x_{h,(i,j),k} \quad \forall (i,j) \in A, k \in R \quad (3.23)$$

$$\sum_{(i,h) \in A} r_{(i,h),k} \sum_{t=0}^T f_{(i,h),k,t} = \sum_{(i',j') \in A} x_{h,(i',j'),k} \quad \forall h \in N, k \in R \quad (3.24)$$

式(3.25)-(3.26)限制從起點出發的工作隊 k 總數及最後流往匯集點的工作隊 k 總數，皆等同於工作隊 k 之上限。

$$\sum_{\substack{(i,j) \in A \\ i \in S, j \in Y_S}} x_{h,(i,j),r} = \bar{r}_k \quad \forall h \in S, k \in R \quad (3.25)$$

$$\sum_{h \in N} x_{h,(h,f),k} = \bar{r}_k \quad \forall k \in R \quad (3.26)$$

式(3.27)-(3.30)，定義變數類型及其值域範圍。

$$f_{(i,j),k,t} \in \{0,1\} \quad \forall (i,j) \in A, k \in R, t = 0, \dots, T \quad (3.27)$$

$$F_j \geq 0 \quad \forall j \in N \quad (3.28)$$

$$z_{h,(i,j),k} \in \{0,1\} \quad \forall h \in N, (i,j) \in A, k \in R \quad (3.29)$$

$$0 \leq x_{h,(i,j),k} \leq \bar{r}_k \quad \forall h \in N, (i,j) \in A, k \in R \quad (3.30)$$

除此之外，此模式亦可求解無資源運輸時間之問題，只需將所有 $d_{i,j}$ 設為 0， $\forall i, j \in N$ ，便可求解。

3.7 小結

此章節所建構的數學模式 GRB1 是以 RCPSP 的觀點去建構而得，在過去文獻中較無以此作為出發點來求解道路網重建問題。而道路網重建問題在某些條件下是等同於 RCPSP，例如當網路結構為樹時，但在一般網路結構中，本問題比 RCPSP 更複雜，因 RCPSP 中，所需完成的作業及作業間的順序限制皆為已知，但在本研究探討的道路網重建問題，需先決定搶修路徑才可知道需要進行的搶修作業為何及其之間的順序限制，故在模式建構中，除了需考量資源限制與作業順序限制外，還需保證搶修路徑為一展開樹，使得問題較 RCPSP 更加複雜。而從過去探討 RCPSP 之文獻中發現，當作業數越多且可更新資源種類越多時，以數學模式求解會耗費過多的時間成本，同樣在緊急道路重建問題上，隨著網路規模越大，數學模式之規模變得非常龐大，直接求解將耗費數日，故本研究對此問題將發展演算法來求解，希望能在有限的時間內求得一近似最佳解，下章節將說明演算法的內容與步驟。除此之外，本章節亦提出了兩種延伸問題，一為只搶修特定節點，另一則將資源運輸時間納入考量，但在後續演算法求解，只專注在原問題的求解，此二個延伸問題則可作為未來研究方向。

第四章

求解方法

本研究所建構之道路網搶修作業排程模式甚為龐大，舉例來說：在三組工作隊且每個路段的修復模式最多三個的假設下，設定完工時間上限 $T=72$ ，GRB1 的模式規模如表 4.1，可發現當節點數大於 65 個，節線數大於 120 條時，變數個數及限制式數都超過 3 萬，對如此龐大的規模，若直接求解將耗時多日，但本研究問題為一即時性之緊急排程規劃問題，時間性甚為重要，對此，本章將說明如何優化數學模式的求解效率，並發展一分段式啟發演算法進行求解，此演算法可適當的將問題分解來縮小問題規模，以便更有效率的求解模式。除此之外，本研究亦運用粒子群最佳化演算法 (Particle swarm optimization, PSO) 進行求解。以下將說明各求解方法：

表 4.1 GRB1 之模式規模

網路圖規模		變數個數	限制式數
節點數	節線數		
10	15	4244	4473
17	28	8339	8575
26	45	12874	13119
37	66	18287	18543
50	91	26476	26745
65	120	34229	34513
82	153	43590	43891
101	190	54267	54587

表 4.2 GRB1 之模式大小估算

變數個數	$O(A \times m \times T + N)$
限制式數	$O(2 N + 1 + A \times m \times T + N + R \times T)$

4.1 模式初始解與上下界

本研究是利用最佳化軟體 Gurobi 進行模式求解，且為了快速求得模式結果，藉由設定模式的初始解或設定上下界，希望能在有限時間內求得實務上可行的規劃結果，以下將說明模式初始解及其上下界設定：

4.1.1 啟發式初始解及上界設定

由最佳解性質可知最佳的搶修路徑結構必為一展開樹，故藉由在網路中找到一展開樹，並在對應之 PERT 網路圖，求解 MRCPSP 問題來得到一初始解，但一個網路中之展開樹有非常多個，故本研究尋找展開樹的策略如下：

【步驟一】 節線之權重設定：各節線之權重為其修復模式中所需修復時間最短之修復時間。定義 $w_{(i,j)} = \min_{m=1,\dots,M_{(i,j)}} \{p_{(i,j),m}\}$ 為各節線 (i,j) 之權重，

$Mode_{(i,j)} = \arg \min_{m=1,\dots,M_{(i,j)}} \{p_{(i,j),m}\}$ 為該節線的修復模式中，耗費最少時間之模式編號。

【步驟二】 尋找網路中之 MST：在步驟一所得之權重下之找尋 MST，並以此 MST 作為初始求解之展開樹，接著進行步驟三。

【步驟三】 求解 RCPSP 模式：在展開樹上對應之 PERT 網路圖，求解 RCPSP 模式，且各節線的修復模式為 $Mode_{(i,j)}$ ，並以所得之最佳解作為初始解。

以上述步驟找到之初始解，取名為 *MST-heuristic*。以現有找尋 MST 的演算法，可很快的找到網路中之 MST，例如：Prim's 演算法及 Kruskal's 演算法，且在此展開樹求解 RCPSP 所得之解，除了可作為模式初始解外，所求得之目標值亦可作為問題之上界及用於預估完工時間上限 T ，但預估之 T 的品質卻無法保證，因此預估值與實際完工時間可能尚距一大段誤差，對此在 4.2 小節將介紹另一預估完工時間上限 T 之方法。

4.1.2 下界設定

在下界設定，從定理 2 可知，若在工作隊組數無限多時，最佳搶修路徑為起點對其他節點的 SPT，可得各節點最快的打通時間，若問題目標為 **SUM** 則將打通時間加總，問題目標為 **MAX** 則取所有最晚打通的節點之打通時間，故由此可做為原問題最佳解目標值之一下界。

4.2 完工時間 T 之預估

從表 4.2 發現，由於網路規模大小是固定的，影響模式的規模大小最大的因素是給予之時間上限 T ，因此若能在求解前估計出一好的搶修計劃完工時間上限，便可大幅降低模式之規模。雖然 *MST-heuristic* 可以用來預估 T ，但無法確保其品質，為了能得到更好的預估值，本研究以二分逼近法來預估 T ，步驟如下：

【步驟一】 設定 T 之上下界：以 *MST-heuristic* 之完工時間為上界 T_{upper} 與 SPT 之完

工時間為下界 T_{lower} ，並令 $T = \left\lfloor \frac{T_{upper} + T_{lower}}{2} \right\rfloor$ 。

【步驟二】 以目前之 T 求解模式：

若模式無解，則更新 $T_{lower} = T$ 及 $T = \left\lfloor \frac{T_{upper} + T}{2} \right\rfloor$ ；

若模式有解，則更新 $T_{upper} = T$ 及 $T = \left\lfloor \frac{T_{lower} + T}{2} \right\rfloor$ 。

【步驟三】 終止條件：若 $T_{upper} - T_{lower} \leq 2$ ，則終止。

其流程如下頁圖 4.1 所示，雖然相較於 *MST-heuristic* 之解，以此二分逼近法預估出之完工時間 T 較能貼近實際之完工時間，但並無法依此認定 *MST-heuristic* 為較差的方式，因 *MST-heuristic* 之功用在於能提供模式一初始解，並藉此加速模式的求解效率，但以二分逼近法預估 T ，卻無法得到與之相對應的搶修路徑與作業排程，故在使用上各有其優缺點。

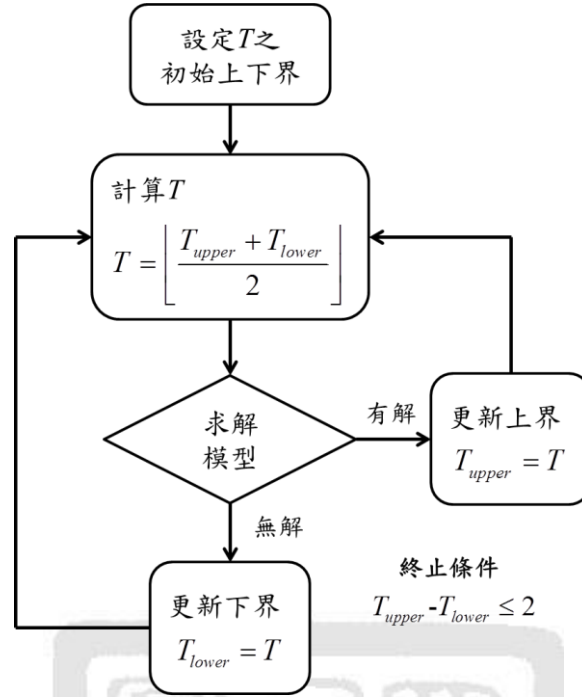


圖 4.1 完工時間 T 之預估流程

4.3 分段式啟發演算法 (Sequential segment heuristic, SSH)

除了設定模式之上下界與初始解外，爲了快速求得規劃結果，本研究設計一分段式啟發演算法 (Sequential segment heuristic, 簡稱為 SSH)，此演算法透過貪婪式演算法的觀念，由近而遠先求解距起點較近災點的搶通方式，以其為基礎再逐漸擴張搶通範圍直到全部災點皆搶通為止，以下將說明分段式演算法之步驟與內容，並舉一範例說明之。

假設有一網路 $G=(N, A)$ ，在 k 組能力不同的工作隊下進行搶修，其中工作隊起始位置為 $s, s \in N$ 。接著，對每一個節點 i 及每一組工作隊 k 各別加入一釋放時間 (release time)，定義為 a_i 及 b_k ，並初始化為 0。定義 $P_{MAX} = \max_{(i,j) \in A} \left\{ \max_{m=1, \dots, M_{(i,j)}} \{p_{(i,j),m}\} \right\}$ ，表所有路段中最長搶修時間， $cycle$ 為迭代次數上限，且 $cycle < \frac{T}{P_{MAX}}$ 。接著，從時間點 $t=0$ 開始，每次求解從開始時間點往後 $\frac{T}{cycle}$ 個單位時間之時間區段，令 I_j 為第 j 次迭代求

解的時間區段， $N_j^{Connected}$ 為第 j 次迭代打通的節點集合， $N^{Connected}$ 為已打通的所有節點集合， t_j^* 為第 j 次迭代求解的時間區段開始時間點，並從 $I_1 = \left[t_1^*, t_1^* + \frac{T}{cycle} \right]$ 開始以下

步驟：

4.3.1 步驟流程

【初始化】 初始化參數

$$N^{Connected} = \{s\} ; a_i = 0, \forall i \in N ; b_k = 0, \forall k \in R ; t_0^* = 0 .$$

【步驟一】 找尋最短路徑樹

找尋從起點 s 到各災點的最短路徑，並記錄每個節點最快打通時間，去除在該時間區段下無法到達的災點，得到一縮減網路圖。

【步驟二】 求解鬆弛部分限制式之 GRB1 模式

首先，允許節點在目前的迭代中，可暫時不被打通，鬆弛限制式(3.7)的為式(4.1)，並重新定義中各節點的打通時間為式(4.2)、增加作業順序限制式(4.3)以及針對釋放時間的限制式(4.4)、(4.5)，如下：

$$\sum_{\substack{(i,j) \in A \\ i \in N}} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} \leq 1 \quad \forall j \in N/S \quad (4.1)$$

$$F_j = \sum_{\substack{(i,j) \in A \\ i \in N, i \neq j}} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T (t + p_{(i,j),m}) x_{(i,j),m,t} + B \left(1 - \sum_{\substack{(i,j) \in A \\ i \in N, i \neq j}} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} \right) \quad \forall j \in N/S \quad (4.2)$$

$$\sum_{(i,j) \in A} \sum_{m=1}^{M(i,j)} \sum_{t=0}^T x_{(i,j),m,t} \leq B \left(\sum_{(h,i) \in A} \sum_{m=1}^{M(j,h)} \sum_{t=0}^T x_{(h,i),m,t} \right) \quad \forall i \in N \quad (4.3)$$

$$\sum_{(i,j) \in A} \sum_{m=1}^{M(i,j)} \sum_{t=0}^{\min\{a_i-1, T+1\}} x_{(i,j),m,t} = 0 \quad \forall i \in N \quad (4.4)$$

$$\sum_{\substack{(i,j) \in A \\ i \in N, j \in N/S}} \sum_{m=1}^{M(i,j)} r_{(i,j),m,k} \sum_{s=\max\{t-p_{(i,j),m}+1, 0\}}^t x_{(i,j),m,s} \leq 0 \quad \forall k \in R, t=1, 2, \dots, b_k-1 \quad (4.5)$$

式(4.1)鬆弛原本節點皆須被打通的限制，並允許在目前的迭代中，部分節點可暫不被打通。式(4.2)則重新定義節點的打通時間，表若該節點沒有打通，其打通時間為一極大值。式(4.3)限制若節點 i 未被打通，則不會有任何作業的修復方向是從節點 i 出發。式(4.4)-(4.5)為針對釋放時間的限制。其中，式(4.4)限制從節點 i 出發的作業需在節點 i 之釋放時間 a_i 後才可開始進行；式(4.5)限制工作隊 k 需在其釋放時間 b_k 後才可被指派。

求解時，不管問題目標為 **MAX** 或 **SUM** 時，在步驟三設定之目標式皆採 **SUM**（最小化各節點打通時間總和），因在此鬆弛模式中，對未被打通的節點會給予一極大值，若求解時模式目標設為 **MAX** 會使得每次迭代過程中只要網路中有一個節點未被修復，結果之目標值便為極大值；再來，此步驟主要放鬆節點必須被打通的限制，造成每次迭代的排程結果傾向不進行任何修復作業。

【步驟三】 更新網路

記錄在步驟二所得之搶修路徑、路段的維修方式，並將打通的節點加至 $N_j^{Connected}$ ，並更新 $N^{Connected} = N^{Connected} \cup N_j^{Connected}$ ，若所有節點皆已打通，則跳至步驟五；若尚未全部打通，則繼續步驟四，並將 $N^{Connected}$ 內之節點與初始起點 s 合併為一新節點 s' ，依據原網路圖建立新節線 (s', j) ，若節點 j 存在一條以上之節線 (s', j) ，則可將之合併為一，其修復方式亦可合併。

【步驟四】 更新各節點與各組工作隊的釋放時間 (release time)

記錄步驟二所得結果中各工作隊的開始閒置時間，更新 t_{i+1}^* 為其中最早的閒置時間，且為下一次迭代的開始時間，而各節點與工作隊的釋放時間，依求解結果計算如下：

1. 若有節點被打通

$$b_k = \text{工作隊的開始閒置時間} - t_{i+1}^*, k \in R$$

$$a_i = \min\{\text{節點 } i \text{ 的打通時間} - t_{i+1}^*, 0\}, i \in N_j^{\text{Connected}}$$

$$t_i^N = 0, i \notin N_j^{\text{Connected}}$$

2. 若無節點被打通

$$b_k = b_k - 1, k \in R$$

$$a_i = \min\{0, a_i - 1\}, i \in N$$

【步驟五】 檢查整體時間軸

檢查是否有作業之開始時間可往前移動，且不違反各時間點之資源上限限制及作業間的時序限制，若有，則將作業開始時間往前調整，若無，則演算法結束。

分段式啟發演算法求解流程，整理如圖 4.2 所示。

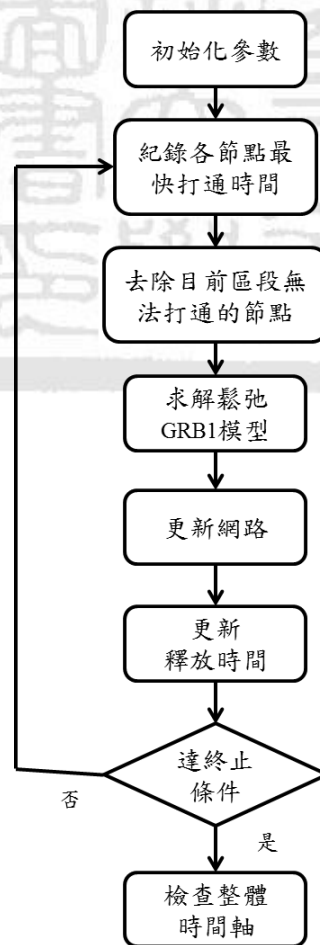


圖 4.2 分段式啟發演算法求解流程

4.3.2 範例說明

假設有一 17 個節點，28 條節線的網路圖 G ，如圖 4.3，網路內所有節線狀況皆無法通行，除了起始點（節點 1）外，其他節點皆屬災點。若今有三組工作隊，分別為工作隊 A、B 及 C，此三組工作隊皆位於起始點等待指派，而在不考慮工作隊合作修復下，各節線在不同工作隊下的所需的搶修時間如表 4.3，並由表 4.3 可知 $P_{MAX}=3$ 。

欲求目標為 **SUM** 下之搶修路徑與其作業指派，並設定完工時間上限 $T=30$ ，迭代次數

上限 $cycle=10$ ，故每次迭代求解的時間長度為 $\frac{T}{cycle} = \frac{30}{10} = 3$ 單位。

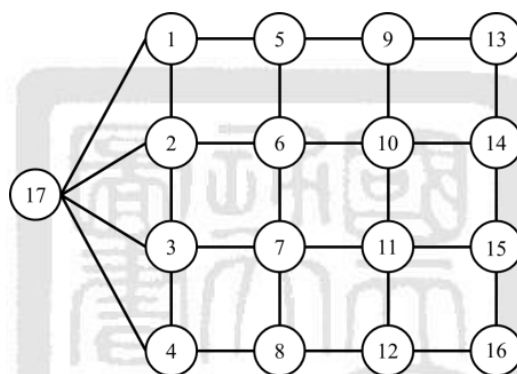


圖 4.3 範例網路圖

表 4.3 各路段之在不同工作隊下的修復時間

節線	在各工作隊下的修復時間			節線	在各工作隊下的修復時間		
	A	B	C		A	B	C
(1, 2)	1	2	2	(6, 10)	1	3	2
(1, 5)	3	2	1	(7, 8)	3	2	3
(1, 17)	3	1	1	(7, 11)	2	3	1
(2, 3)	1	2	3	(8, 12)	2	2	3
(2, 6)	3	3	2	(9, 10)	1	3	3
(2, 17)	2	1	1	(9, 13)	2	3	2
(3, 4)	1	2	3	(10, 11)	3	1	3
(3, 7)	3	2	2	(10, 14)	1	1	2
(3, 17)	1	3	2	(11, 12)	2	3	1
(4, 8)	2	2	2	(11, 15)	3	3	3
(4, 17)	2	2	1	(12, 16)	3	3	1
(5, 6)	3	2	3	(13, 14)	3	2	2
(5, 9)	1	3	2	(14, 15)	3	3	1
(6, 7)	3	1	2	(15, 16)	1	3	2

【初始步驟】 $N^{Connected}=\{1\}$ 。

時間區段 $I_1=[0, 3]$

【步驟一】 找尋從起點 s 到其他各點的最短路徑樹，如圖 4.4(a)，節點旁為該點之最快打通時間，去除在 I_1 內不可能打通的節點（節點 7、8、11、12、13、14、15 及 16），去除後所得之縮減網路，如圖 4.4(b)。

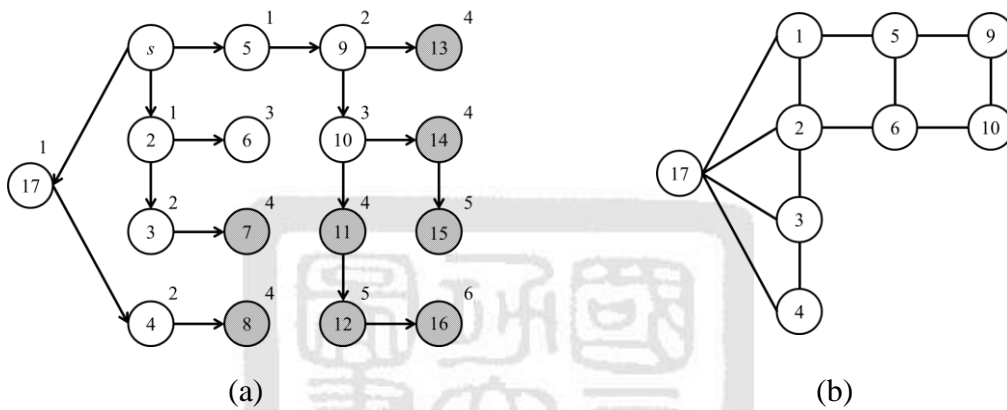


圖 4.4 各節點之最快修復路徑（網路之 SPT）及縮減後之網路

【步驟二】 求解鬆弛部分限制式之 GRB1 模式：本研究係利用最佳化求解軟體 Gurobi 進行數學模式的求解。其求解結果如圖 4.5。

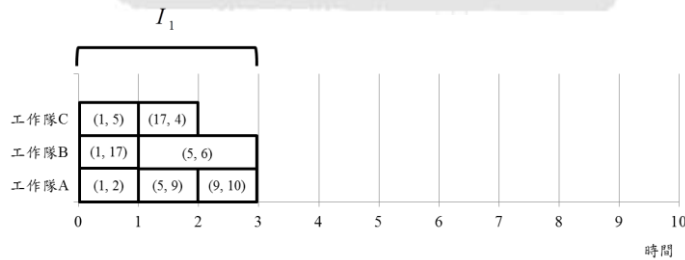


圖 4.5 時間區段 I_1 內之搶修路徑及作業排程

【步驟三】 依步驟二的求解結果，可知 $N_1^{Connected}=\{2, 4, 5, 6, 9, 10\}$ ，並更新 $N^{Connected}=\{1, 2, 4, 5, 6, 9, 10\}$ 。接著，將 $N^{Connected}$ 內之節點與起始點合併為一新節點 s' ，並依原網路 G 合併重複之節線，如圖 4.6，有兩條 $(s', 3)$ ，將兩條節線之修復模式合併，若修復模式亦有重複，則選擇修復時間較小者。

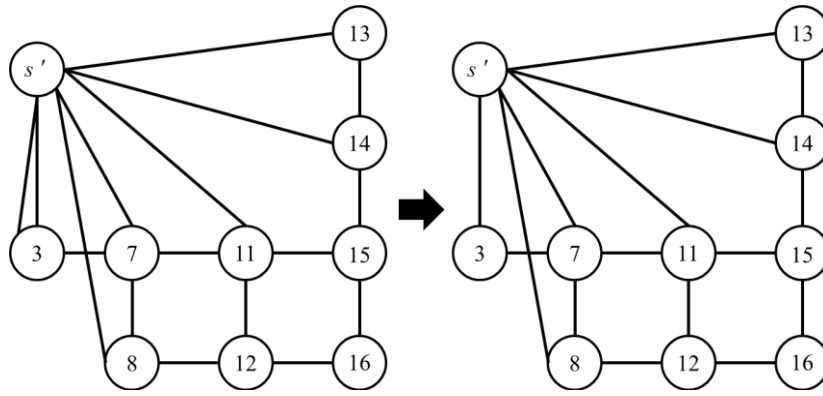


圖 4.6 投入時間區段 I_2 之網路

【步驟四】 依步驟二的求解結果，可看出工作隊 C 最早開始閒置，其開始時間為 2 ($t^*=2$)，故下一階段求解，各節點與工作隊的釋放時間為 $a_i=\{0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0\}$ 及 $b_k=\{0, 0, 1\}$ ，接著，從 $t^*=2$ 開始求解。

時間區段 $I_2=[2, 5]$

重複步驟一及步驟二，其求解結果如圖 4.7， $N_2^{Connected}=\{3, 7, 8, 11, 12, 14\}$ ， $N^{Connected}=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 17\}$ 。接著，計算各節點與工作隊的下一階段的釋放時間，並從 $t^*=5$ 開始求解下一階段。

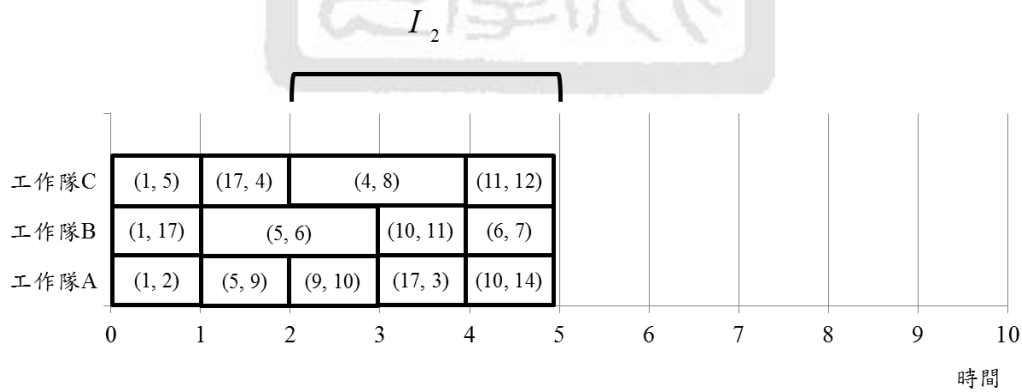


圖 4.7 時間區段 I_2 內之作業排程

時間區段 $I_3=[5, 8]$

重複步驟一及步驟二，其求解結果如圖 4.8，可發現在時間區段 I_3 便已打通網路所有節點，演算法中止。最後所得各工作隊之搶修路徑如圖 4.9，其目標值為 60。

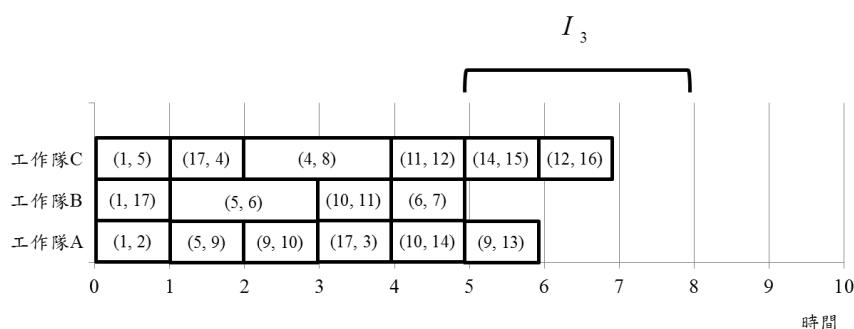


圖 4.8 時間區段 I_3 內及最後整體作業排程

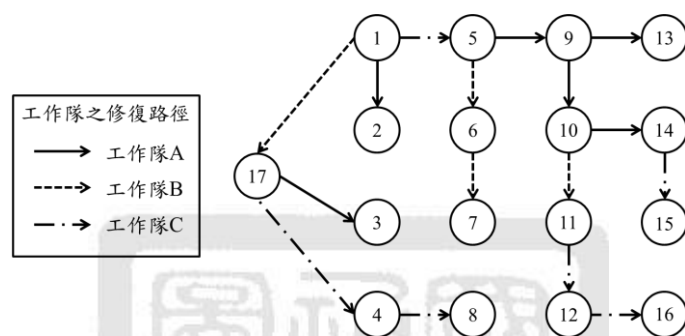


圖 4.9 各工作隊之搶修路徑

4.3.3 演算法修正

此分段式啟發演算法，是將原問題之時間軸分為數個較小之時間區段後，各別求解各時間區段之子問題，雖可大幅降低求解所耗費之時間，但並不能確保每個時間區段內之最佳解為整體最佳解。因在求解各區段時，皆只考慮該時間區段而已，並不會考慮到後續區段，以上一小節所舉範例所得之結果為例，其所得之解並非為最佳解，最佳搶修路徑與作業排程應如圖 4.10 及圖 4.11，目標值為 59。

因此，在每次迭代中求解時，增長的時間區段長度，但只保留可在當前區段完成之作業，舉例來說，假設第 j 次迭代的時間區段為 $I_j = \left[t, t + \frac{T}{cycle} \right]$ ，在步驟二求解時，

將 I_j 延長 αP_{MAX} 時間單位，即求解 $I'_j = \left[t, t + \frac{T}{cycle} + \alpha P_{MAX} \right]$ ，但在步驟三只保留可在 I_j

內完成的作業，故修正步驟一及步驟三如下：

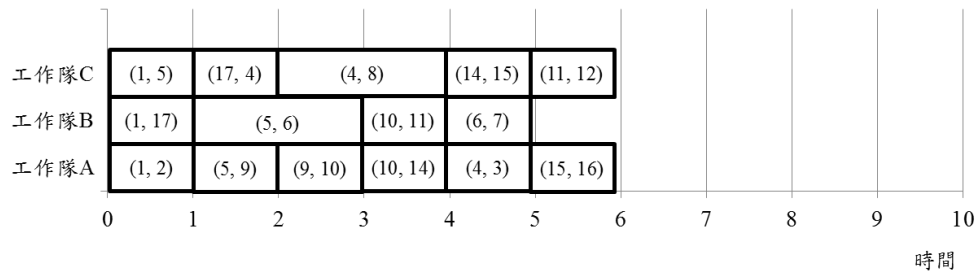


圖 4.10 最佳搶修作業排程

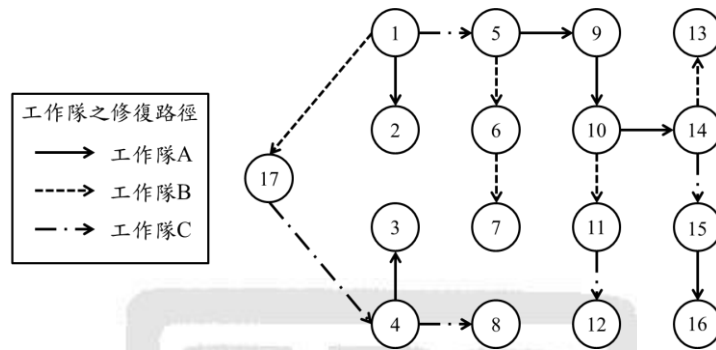


圖 4.11 各工作隊之最佳搶修路徑

【修正步驟一】 找尋從起點到各災點的最短路徑，並去除無法在目前時間區段 I_j' 打通之節點，得到一新縮減網路圖。

【修正步驟三】 記錄在時間區段 I_j' 中之搶修路徑、選擇路段的維修方式後，再縮減時間區段 I_j' 為 I_j ，並去除不可在 I_j 內完成的作業。

在修正分段式啟發演算法中，求解各時間區段時，先將區段範圍放寬為 I_j' ，目的使求解時，可將後續部分時間區段納入考量，但最後只保留原時間區段 I_j 下能完成的作業，藉此讓各時間區段求解時較能考慮到後續時間區段。

4.4 粒子群最佳化演算法 (PSO)

在第三章所提出之數學模式，為一困難的整數規劃模式，會隨著網路圖規模的擴大（包括節點數、節線數、時間軸長度與各路段修復模式數），而花費更長之求解時間。因此，本研究利用常見的粒子群最佳化演算法，針對路網重建問題設計 PSO_{IP} ，

希望能加速問題之求解效率。

首先定義 $p_{(i,j)}^m(r)$ 為第 r 迭代第 m 個粒子位置中所含的節線 (i, j) 之權重值, $v_{(i,j)}^m(r)$ 為第 r 迭代第 m 個粒子下節線 (i, j) 的速度。以圖 4.3 之網路圖為例, 若有 10 個粒子, 其粒子示意圖如下頁圖 4.12 所示。

粒子1 (m=1)	$p_{(1,2)}^1(r)$	$p_{(1,5)}^1(r)$...	$p_{(14,15)}^1(r)$	$p_{(15,16)}^1(r)$
粒子2 (m=2)	$p_{(1,2)}^2(r)$	$p_{(1,5)}^2(r)$...	$p_{(14,15)}^2(r)$	$p_{(15,16)}^2(r)$
⋮					
粒子10 (m=10)	$p_{(1,2)}^{10}(r)$	$p_{(1,5)}^{10}(r)$...	$p_{(14,15)}^{10}(r)$	$p_{(15,16)}^{10}(r)$

圖 4.12 粒子示意圖

4.4.1 PSO_{IP} 步驟流程

PSO_{IP} 之求解流程, 整理如圖 4.13。以下將說明 PSO_{IP} 個步驟的詳細內容：

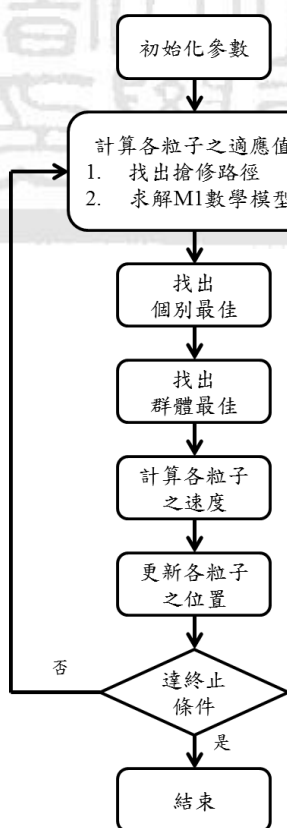


圖 4.13 PSO_{IP} 演算法流程圖

【步驟一】 初始化參數：每個粒子的位置代表網路中各節線被選為搶修路徑的權重，權重越大的表示越容易被選為路徑之一。PSO_{IP} 一開始需先產生初始粒子之位置與速度，即 $p_{(i,j)}^m(0)$ 與 $v_{(i,j)}^m(0)$ ，分別表示初始第 m 個粒子下，節線 (i, j) 之權重與速度參數。速度參數 $v_{(i,j)}^m(0)$ 為介於 $[-1, 1]$ 之間的隨機亂數，而粒子位置 $p_{(i,j)}^m(0)$ 產生方式有兩種：(1) 直接隨機產生介於 $[0, 1]$ 之間的亂數；(2) 依據 MST-heuristic 將其搶修路徑之權重設為 1，其餘設為 0。

【步驟二】 計算粒子之適應值：步驟二主要分為兩個階段：

第一階段：選擇搶修路徑。為針對每一粒子中各節線的權重，從起點出發，在與其相連且尚未被選取的節點裡，選擇權重最大的節線，並將新的節點加入，直到增加 $n-1$ 條節線為止（假設共有 n 個節點）。

第二階段：在第一階段所獲得之搶修路徑，求解 GRB1 模式，所得解之目標值即為該粒子之適應值，適應值越小代表該粒子的表現越好。

【步驟三】 找出個別最佳解：計算完各粒子之適應值後，找出每一個粒子 m 到目前第 r 迭代為止搜尋過程中的最佳解 $Pbest_i^m(r)$ 。

【步驟四】 找出群體最佳解：根據每一個粒子新的最佳解 $Pbest_i^m(r)$ ，更新所有粒子到目前為止搜尋的群體最佳解 $Gbest_i(r)$ 。

【步驟五】 更新各粒子之速度與位置：以式(4.4)及(4.5)更新例子之速度與位置，其中 c_1 與 c_2 表示加速度常數（acceleration constant）， $Rand_1$ 與 $Rand_2$ 則為介於 $[-1, 1]$ 之間的隨機亂數。

$$v_i^m(r+1) = v_i^m(r) + c_1 Rand_1 (Pbest_i^m(r) - p_i^m(r)) + c_2 Rand_2 (Gbest_i(r) - p_i^m(r)) \quad (4.4)$$

$$p_i^m(r+1) = p_i^m(r) + v_i^m(r+1) \quad (4.5)$$

【步驟六】 終止條件：判斷是否達到終止條件，終止條件可分為兩種，一種是達到

迭代次數上限即停止；另一種達到設定的門檻值 δ ，也就是當所有粒子群之適應值與目前最好的適應值之間的差距低於門檻值時即停止演算法，其計算方式如式 (4.6)， RPD 為相對誤差百分比 (Relative percentage deviation)，該值愈小愈好，若每個粒子之 RPD 皆小於 δ ，則停止演算法。

$$RPD = \frac{fitness - best\ fitness}{best\ fitness} \times 100\% \quad (4.6)$$

雖然在 PSO_{IP} 之步驟二中，是將網路圖縮小為一展開樹去進行求解，但當網路之節點數太大時，其求解效率不佳，故以下將修正步驟二改以一啟發式演算法替代之，並將修正後之 PSO_{IP} 取名為 PSO_H 。

4.4.2 PSO_H 步驟流程

首先，在 PSO_H 所使用的每個粒子位置為 $p_{k,(i,j)}^m(r)$ ，表示第 r 迭代第 m 個粒子下工作隊 k 對節線 (i,j) 的權重值； $v_{k,(i,j)}^m(r)$ 則為第 r 迭代第 m 個粒子下工作隊 k 對節線 (i,j) 的速度。同樣，以圖 4.3 之網路圖為例，若有 10 個粒子，3 組工作隊，其粒子示意圖如圖 4.14 所示。

		(1,2)	(1,5)	(14,15)	(1,2)
粒子1 (m=1)	工作隊1	$p_{1,(1,2)}^1(r)$	$p_{1,(1,5)}^1(r)$...	$p_{1,(14,15)}^1(r)$	$p_{1,(15,16)}^1(r)$
	工作隊2	$p_{2,(1,2)}^1(r)$	$p_{2,(1,5)}^1(r)$...	$p_{2,(14,15)}^1(r)$	$p_{2,(15,16)}^1(r)$
	工作隊3	$p_{3,(1,2)}^1(r)$	$p_{3,(1,5)}^1(r)$...	$p_{3,(14,15)}^1(r)$	$p_{3,(15,16)}^1(r)$
粒子2 (m=2)		$p_{1,(1,2)}^2(r)$	$p_{1,(1,5)}^2(r)$...	$p_{1,(14,15)}^2(r)$	$p_{1,(15,16)}^2(r)$
		$p_{2,(1,2)}^2(r)$	$p_{2,(1,5)}^2(r)$...	$p_{2,(14,15)}^2(r)$	$p_{2,(15,16)}^2(r)$
		$p_{3,(1,2)}^2(r)$	$p_{3,(1,5)}^2(r)$...	$p_{3,(14,15)}^2(r)$	$p_{3,(15,16)}^2(r)$
...						
粒子10 (m=10)		$p_{1,(1,2)}^{10}(r)$	$p_{1,(1,5)}^{10}(r)$...	$p_{1,(14,15)}^{10}(r)$	$p_{1,(15,16)}^{10}(r)$
		$p_{2,(1,2)}^{10}(r)$	$p_{2,(1,5)}^{10}(r)$...	$p_{2,(14,15)}^{10}(r)$	$p_{2,(15,16)}^{10}(r)$
		$p_{3,(1,2)}^{10}(r)$	$p_{3,(1,5)}^{10}(r)$...	$p_{3,(14,15)}^{10}(r)$	$p_{3,(15,16)}^{10}(r)$

圖 4.14 粒子示意圖

PSO_H 的流程如同 PSO_{IP} ，只有在步驟二計算各個粒子適應值的方式不同，改以一演算法（而非使用 Gurobi 求解 GRB1）計算，以下將針對此步驟做說明：

【修正步驟二】 計算粒子之適應值：從時間點 0 開始，若目前工作隊 k 閒置，依照粒子中工作隊 k 在各路段的權重，指派該工作隊 k ，從目前已打通之區域相連且尚未被選取的節點裡，選擇權重最大的節線來進行搶修，並將新的節點加入，若無工作隊閒置則查看下一時間點，直到增加 $n-1$ 條節線為止（假設共有 n 個節點）。

4.5 小結

本研究主要應用最佳化軟體 Gurobi 中所提供之分枝界限法來求解模式，但隨著網路圖規模越大，需耗費龐大的時間成本來求解，故在此章節裡，首先提供在問題之上下界及初始解 MST-heuristic，在執行分枝界限法時，可去除許多不必尋找的分枝，以期降低求解所耗費之時間；接著，提出一段式啟發演算法，在演算法中，主要應用貪婪式演算法的概念，將時間軸分為多個區段後，依序對各區段進行求解，在每次迭代中，去除打通的節點達到縮小網路的效果，接著固定所得之解，進行下一次迭代，但此演算法之缺點在於求解各區段時，並不會考慮到後續的作業排程，雖在各區段為最佳，但整體來講未必是最佳，故對演算法提出一修正方式：在求解各區段時，拉長原區段長度，但只保留能在原區段進行的作業，使得在進行求解時，能將後續之時間區段納入考慮；此外，本研究亦探討粒子分群演算法求解本問題之效能，並設計 PSO_{IP} 及 PSO_H 兩種方式， PSO_{IP} 中之粒子位置代表各節線的權重，藉由這些權重找到一搶修路徑後，在此搶修路徑上求解 RCPS 或利用 GRB1 模式求解得到各粒子之適應值；不過當網路太大時，在求解模式亦須花費許多時間，故 PSO_H 修正計算適應值之方式，改以一貪婪式演算法計算。第五章將測試本研究之數學模式及演算法在不同網路結構下的求解表現，並與過去文獻中的方法進行比較。

第五章

數值分析

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，故本章將以三種不同網路結構為測試對象，分別為路徑網路、樹狀網路及格狀網路。本研究以 Microsoft Visual Studio 2012 為程式語言介面，以 C++ 程式語言撰寫程式，並配合 Gurobi 6.0.4 版求解數學模式；同時在個人電腦執行，測試平台為 Microsoft Windows 7 作業系統，搭配之處理器及記憶體為 Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz 與 16GB RAM。最後，本研究將針對相關參數進行敏感度分析。

5.1 輸入資料

5.1.1 測試網路

在現實生活中，山區之道路網，通常屬於樹狀或格狀網路，故本研究所測試之網路圖將以樹狀與格狀網路為主，而路徑網路，為網路中只有一條路徑之網路結構，常見如隧道等，亦是緊急道路搶修中的網路種類之一，故也將之作為測試對象，此三種網路結構示意圖，如圖 5.1 所示。

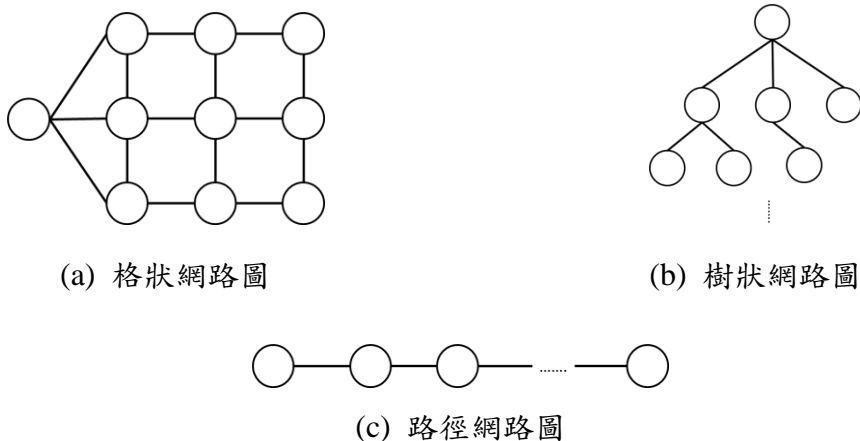


圖 5.1 網路結構示意圖

分別針對這三種不同結構之網路圖測試小、中及大規模，且每個規模再分別隨機產生 10 個例子，其詳細資訊如表 5.1。其中，格狀與樹狀網路，只測試單一起點的例子，因本研究假設資源運輸時間為 0，意即工作隊能在已打通的節點中任兩點之間立即移動，故將多起點以一個虛擬起點來代替，而路徑網路則考慮兩個起點。

表 5.1 測試網路之規模資訊

網路結構		規模				
		節點數	節線數	起點數	工作隊數	路段所需維修時間
Grid-I	格狀	10	15	1	3	[1, 3]
Grid-II		26	45	1	3	[1, 3]
Grid-III		50	91	1	3	[1, 3]
Tree-I	樹狀	10	9	1	3	[1, 3]
Tree-II		26	25	1	3	[1, 3]
Tree-III		50	49	1	3	[1, 3]
Path-I	路徑	10	9	2	3	[1, 3]
Path-II		26	25	2	3	[1, 3]
Path-III		50	49	2	3	[1, 3]

5.2 GRB1 模式測試

在 GRB1 模式測試的部分，主要測試模式在下述兩種實驗情境下的求解表現：

情境一：工作隊能力不同且不考慮合作，

情境二：工作隊能力不同且考慮合作；

並測試 *MST-heuristic* 作為初始解對模式求解表現的改善幅度，接著，與過去文獻 Averbakha and Pereira (2012) 所提之模式做比較。該篇文獻討論一組工作隊下之道路搶修問題，並提出一數學模式來進行求解。

5.2.1 實驗情境一：不考慮工作隊合作

此小節將測試在各路段修復模式的為工作隊能力不同且不考慮合作的情境下，

GRB1 模式對不同網路結構的求解表現。因工作隊能力不同，需將工作隊視為不同種的資源種類，在路段修復模式上的呈現方式為各路段交由不同工作隊修復下的修復時間，若假設有 k 組工作隊，則各路段可能的修復模式共有 $k-1$ 種。表 5.2 及 5.3 分別為針對問題目標式 **MAX** 及 **SUM** 的測試結果，其中，對每個網路圖皆測試 10 個例子，求解時間上限設定為 7200 秒，並以 *Number opt.* 代表在該例中求得最佳解的次數，*CPU Time* 為平均求解時間，若無法在求解時限內求得最佳解，則該例之求解時間不予納入平均。

表 5.2 GRB1 模式在實驗情境一下的求解表現 (MAX)

網路圖	測試個數	GRB1 數學模式		以 MST-heuristic 改善	
		<i>Number opt.</i>	<i>CPU Time (sec)</i>	<i>Number opt.</i>	<i>CPU Time (sec)</i>
Grid I	10	10	3.97	10	0.16
Grid II	10	10	332.14	10	17.22
Grid III	10	10	1316.76	9	848.86
Tree I	10	10	1.49	10	0.06
Tree II	10	10	24.94	10	6.00
Tree III	10	10	166.45	10	88.65
Path I	10	10	1.25	10	0.03
Path II	10	10	60.76	10	2.85
Path III	10	8	226.48	10	99.99

表 5.3 GRB1 模式在實驗情境一下的求解表現 (SUM)

網路圖	測試個數	GRB1 數學模式		以 MST-heuristic 改善	
		<i>Number opt.</i>	<i>CPU Time (sec)</i>	<i>Number opt.</i>	<i>CPU Time (sec)</i>
Grid I	10	10	3.33	10	0.35
Grid II	10	9	123.50	10	18.69
Grid III	10	8	1345.62	9	271.11
Tree I	10	10	1.62	10	0.05
Tree II	10	10	31.37	10	10.82
Tree III	10	10	247.71	10	97.01
Path I	10	10	1.49	10	0.04
Path II	10	10	41.29	10	3.57
Path III	10	8	102.19	10	356.64

由測試結果來看，在三種網路結構下，隨著網路規模增大，GRB1 模式的求解表現皆隨之變差。其中，樹狀網路與路徑網路，兩者網路本身便是最佳搶修路徑，但在樹狀網路下的求解表現卻較佳。

而以 MST-heuristic 改善模式的求解能力，改善後的比較整理如圖 5.2、5.3 及 5.4，圖中紅線為直接以 GRB1 模式求解的求解時間，藍線為以 MST-heuristic 改善後的求解時間，可以發現以 MST-heuristic 改善後，模式求解所耗費的時間有所減少，且當問題目標為 SUM 在網狀及樹狀網路下，求解時間皆減少到約原本的一半，而在 MAX - Path III 及 SUM - Grid II、Grid III、Path III 中，能在求解時限內找到最佳解的次數皆有增加，故 MST-heuristic 確實能改善模式的求解效率。

不過，在 MAX - Grid III 中卻發現，沒有利用 MST-heuristic 改善前，模式皆能找到最佳解，但以 MST-heuristic 改善後，卻有一個例子找不到最佳解，而在 MAX - Path III 中，改善後求解耗費的時間反而增加，原因在於以 MST-heuristic 預估之完工時間上限 T ，與實際之完工時間相差太大，模式多了許多不必要的變數及限制式，造成在求解時間上的浪費，故 MST-heuristic 雖可改善模式求解效率，但卻無法保證其品質。不過，在一些無法求得初始可行解的例子中，藉由 MST-heuristic 改善，GRB1 尚可找到其他更好的可行解，若當網路規模過於龐大到無法在短時間內求得最佳解的情況下，仍可藉由 MST-heuristic 來找到其他更好的可行解。

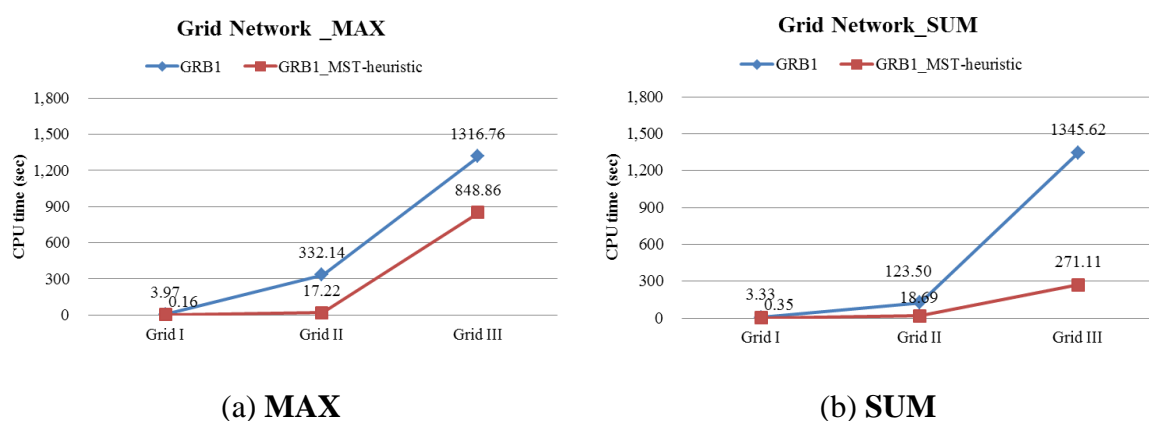


圖 5.2 網狀網路圖下 MST-heuristic 的改善比較

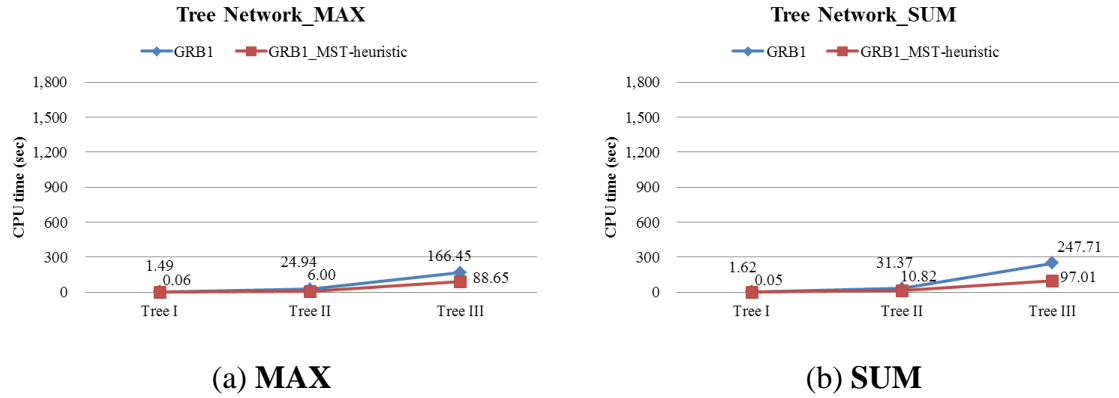


圖 5.3 樹狀網路圖下之 MST-heuristic 的改善比較

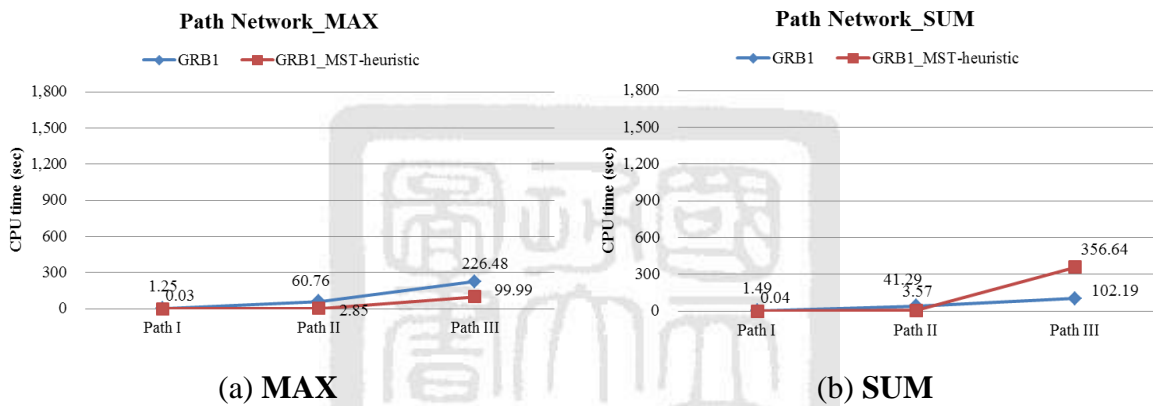


圖 5.4 路徑網路圖下之 MST-heuristic 的改善比較

5.2.2 實驗情境二：考慮工作隊合作

此小節將測試在各路段修復模式的為工作隊能力不同且考慮合作的情境下，GRB1 模式對不同網路結構的求解表現。在修復模式上的呈現，可表示為路段(i, j)在不同工作隊組合下，所需的修復時間。假設有 k 組工作隊，則各路段可能的修復模式共有 $2^k - 1$ 種。但因可能的模式有非常多，故在此實驗情境下，各路段(i, j)的修復模式，是從這 $2^k - 1$ 種可能的修復模式中以隨機之方式抽選 3 個。表 5.4 及 5.5 分別為針對問題目標式 **MAX** 及 **SUM** 的測試結果。同樣對每個網路圖皆測試 10 個例子，求解時間上限設定為 7200 秒，並以 *Number opt.* 代表在該例中求得最佳解的次數，*CPU Time* 為平均求解時間，若無法在求解時限內求得最佳解，則該例求解時間不予納入平均。

表 5.4 GRB1 模式在實驗情境二下的求解表現 (MAX)

網路圖	測試個數	GRB1 數學模式		以 MST-heuristic 改善	
		<i>Number opt.</i>	<i>CPU Time (sec)</i>	<i>Number opt.</i>	<i>CPU Time (sec)</i>
Grid I	10	10	7.67	10	0.43
Grid II	10	7	148.86	10	41.46
Grid III	10	0	-	0	-
Tree I	10	10	0.79	10	0.12
Tree II	10	10	435.84	10	19.44
Tree III	10	4	678.70	6	424.32
Path I	10	10	1.13	10	0.05
Path II	10	10	23.65	10	16.65
Path III	10	6	340.80	7	438.98

註：- 表該網路圖的 10 個例子，模式皆無法在求解時間上限（7200 秒）內求得最佳解。

表 5.5 GRB1 模式在實驗情境二下的求解表現 (SUM)

網路圖	測試個數	GRB1 數學模式		以 MST-heuristic 改善	
		<i>Number opt.</i>	<i>CPU Time (sec)</i>	<i>Number opt.</i>	<i>CPU Time (sec)</i>
Grid I	10	10	5.15	10	0.54
Grid II	10	10	309.22	10	40.95
Grid III	10	0	-	2	1315.35
Tree I	10	10	1.54	10	0.22
Tree II	10	9	16.24	9	16.30
Tree III	10	4	249.83	5	153.38
Path I	10	10	1.25	10	0.05
Path II	10	10	27.31	10	20.68
Path III	10	8	558.23	8	572.60

註：- 表該網路圖的 10 個例子，模式皆無法在求解時間上限（7200 秒）內求得最佳解。

從實驗結果來看，加入工作隊合作的可能性後，問題求解時間在各例中均有所增加，與實驗情境一相比，模式求解所耗費的時間皆明顯增加，在各網路圖找到最佳解的次數也有所減少，甚至在 Grid III 的網路中，10 個測試的例子皆無法在設定的求解時間上限中求得最佳解，且以 MST-heuristic 來改善後，也只有 2 個例子能找到最佳解。而 MST-heuristic 雖能夠改善模式的求解效率，但改善的幅度亦不如在實驗情境

一來的好。儘管如此，相較於過去文獻所提出之數學模式，GRB1 模式更能處理複雜度較高的網路重建問題，且 *MST-heuristic* 依舊有達到期望之改善的效果。下一小節將繼續測試 GRB1 與過去文獻提出之數學模式求解表現。

5.2.3 模式比較

在此小節主要測試對象為 Averbakha and Pereira (2012) 提出之混整數數學規劃模式，簡稱為 FNCP，不過此篇文獻只探討一組工作隊下的緊急道路重建問題，故以下測試皆為在一組工作隊下的情境，因只有一組工作隊，測試之網路只選用格狀網路（路徑與樹狀網路在觀察 2 中說明可以一演算法在 $O(n \log n)$ 下求得最佳解，故在此不做為測試對象）。以下為用最佳化軟體 Gurobi 之測試結果（求解時間上限設定為 1800 秒），其中 *B-Gap* 的計算方式如下：

$$B-Gap = \frac{UB - LB}{UB} \times 100\% \text{。} UB \text{ 及 } LB \text{ 為 Gurobi 找到之最佳上下界。}$$

表 5.6 GRB1 與 FNCP 求解表現比較 (MAX)

網路圖	測試個數	GRB1		FNCP	
		<i>Number opt.</i>	<i>B-Gap (%)</i>	<i>Number opt.</i>	<i>B-Gap (%)</i>
Grid I	10	10	0.00%	10	0.00%
Grid II	10	6	6.49%	10	0.00%
Grid III	10	2	27.87%	10	0.00%

表 5.7 GRB1 與 FNCP 求解表現比較 (SUM)

網路圖	測試個數	GRB1		FNCP	
		<i>Number opt.</i>	<i>B-Gap (%)</i>	<i>Number opt.</i>	<i>B-Gap (%)</i>
Grid I	10	10	0.00%	10	0.00%
Grid II	10	8	3.00%	8	0.37%
Grid III	10	3	5.17%	0	12.07%

由實驗結果可以發現，在問題目標為 **MAX** 下，FNCP 不管在求得最佳解次數上或在求解效率上，皆贏過 GRB1，但在問題目標為 **SUM** 時，反而在 Grid III 時，GRB1

的求解表現較 FNCP 來的好，其中 GRB1 在 10 個例子中有 3 個例子可以找到最佳解，但 FNCP 卻皆無法求得最佳解，此結果與 Averbakha and Pereira (2012) 之測試結果相同。故由此比較可發現，在問題目標為 **MAX** 時，FNCP 之求解表現較本研究之數學模式 GRB1 好；在問題為 **SUM** 時，隨著網路規模越大，雖然 GRB1 及 FNCP 的求解表現皆顯得越差，但 FNCP 皆無法找到最佳解，而 GRB1 在部分例子中卻可找到最佳解，且在 *B-Gap* 的表現確實比 FNCP 來的好。

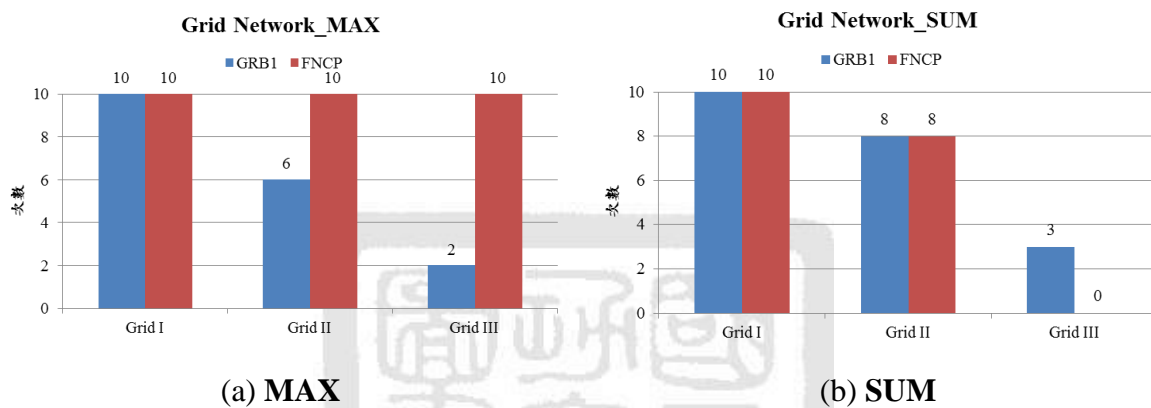


圖 5.5 GRB1 與 FNCP 之求得最佳解次數比較

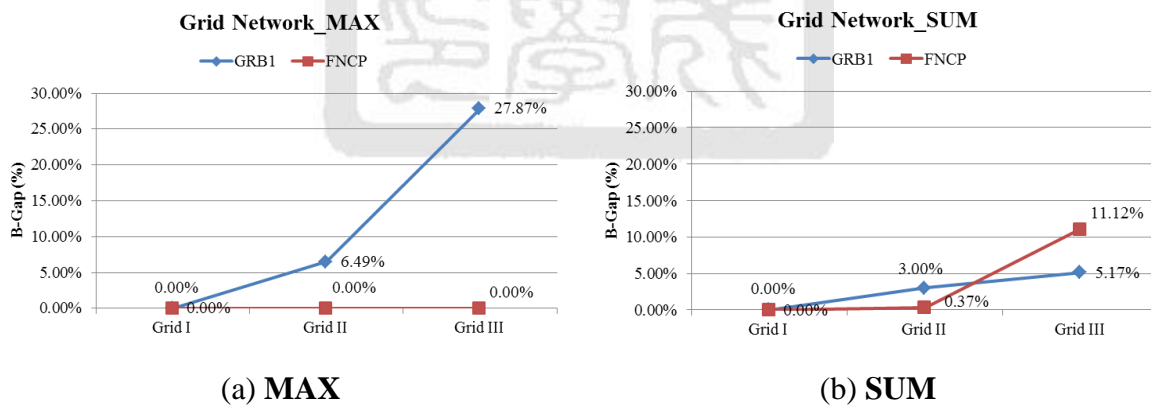


圖 5.6 GRB1 與 FNCP 之 *B-Gap* 比較

5.3 GRB2 模式測試

由於 GRB2 模式不能處理工作隊合作的情況，在此小節測試的實驗情境只測試

情境一：不考慮工作隊合作；

並分別測試在三種不同網路結構下的求解表現，而在各例子中兩點之間的移動時間是由 1 至 3 之間的隨機產生而得。求解時間上限設定為 7200 秒，表 5.8 為實驗結果，*Number opt.* 代表在該例中求得最佳解的次數，*CPU Time* 為平均求解時間，若無法在求解時限內求得最佳解，則該例之求解時間不予納入平均。

表 5.8 GRB2 模式在實驗情境一下的求解表現

網路圖	測試個數	MAX		SUM	
		<i>Number opt.</i>	<i>CPU Time (sec)</i>	<i>Number opt.</i>	<i>CPU Time (sec)</i>
Grid I	10	10	2.74	10	1.90
Grid II	10	0	-	0	-
Grid III	10	0	-	0	-
Tree I	10	10	172.17	10	237.74
Tree II	10	0	-	0	-
Tree III	10	0	-	0	-
Path I	10	10	0.16	10	0.18
Path II	10	10	188.58	10	796.45
Path III	10	1	5035.86	1	3209.50

註：- 表該網路圖的 10 個例子，模式皆無法在求解時間上限（7200 秒）內求得最佳解。

由實驗結果看出，GRB2 模式在三種網路結構中的求解表現以路徑網路最佳，而格狀與樹狀網路，當測試的網路圖規模超過 26 個節點後，GRB2 模式便無法在時限內求得最佳解。而樹狀與路徑網路，雖然本身即為最佳路徑，但 GRB2 在路徑網路的表現卻優於樹狀網路，此結果與 GRB1 模式測試結果相反，主要原因在於 GRB1 求解過程中，因假設工作隊在各點之間的移動時間為 0，使得工作隊在各點的移動方式較不會影響求解結果，但會影響無此假設的 GRB2 求解結果。接著，從網路的結構來看，相較於路徑網路，樹狀網路的分支較多，使得工作隊移動的次數較多，但在路徑網路中，工作隊在各點之間的移動次數則較少，工作隊通常在完成作業後，能指派的作業通常是搶修鄰近的路段，從排程結果可看出，各工作隊在結束作業後會傾向繼續搶修鄰近路段，但樹狀網路則相反，同理於格狀網路，除了分支更多外，各點的打通方式更有多種可能，故 GRB2 求解路徑網路的表現會優於格狀與樹狀網路。

5.4 分段式啟發演算法測試

同樣，此小節測試分段式啟發演算法 (Sequential segment heuristic, SSH) 在兩種實驗情境下求解不同網路結構的求解表現，並以 GRB1 模式所求的最佳解計算誤差，而各例在兩種實驗情境下的參數設定如表 5.9，由於演算法不管在問題目標為 **SUM** 或 **MAX** 時，求解鬆弛模式時（步驟三）的目標式皆設定為 **SUM**，再由排程結果去分別計算在問題目標為 **SUM** 及 **MAX** 時的目標值，故各例的實驗結果只會有一個 *CPU Time*，其結果與討論如下：

表 5.9 分段式啟發演算法參數設定

網路圖	P_{MAX}	情境一		情境二	
		T	$cycle$	T	$cycle$
Grid I	3	10	2	10	2
Grid II	3	20	4	20	4
Grid III	3	30	6	40	8
Tree I	3	10	2	10	2
Tree II	3	20	4	20	5
Tree III	3	30	6	50	10
Path I	3	20	4	20	4
Path II	3	30	6	30	6
Path III	3	50	10	50	10

5.3.1 實驗情境一：不考慮工作隊合作

表 5.10 為實驗結果，由結果可知演算法的求解時間相較於直接以 GRB1 模式求解時間短，且隨著網路規模越大而增加，但增加的幅度皆很小，凸顯將網路及時間軸縮減後，確實能有效減少求解所耗費的時間；而所得之解與最佳解的差距，因各次迭代求解的鬆弛 GRB1 模式，目標式採用 **SUM**，故在求解表現上，在 **SUM** 的誤差小於 **MAX** 的誤差；而以網路規模來看，在規模最小的例子中，*Gap* 皆小於 1%，但在中規模與大規模，除了格狀網路外，在樹狀與路徑網路的 *Gap* 皆表現不佳，原因在於小規模的例子中，最佳搶修排程的時程長度較短，使得所需迭代次數亦較少，故誤

差也較小，但隨著網路規模越大，迭代次數增加，誤差也隨之增加，凸顯該演算法在每次迭代求解時，不考慮後續區段的缺點，故以第四章所提之修正方法修正演算法之缺點，其結果如表 5.11。

表 5.10 分段式啟發演算法求解結果（實驗情境一）

網路圖	測試個數	CPU Time (sec)	MAX		SUM	
			Number opt.	Gap (%)	Number opt.	Gap (%)
Grid I	10	0.12	10	0.00%	10	0.25%
Grid II	10	2.06	7	2.91%	6	1.17%
Grid III	10	3.59	6	2.64%	5	0.25%
Tree I	10	0.05	9	1.43%	10	0.00%
Tree II	10	0.28	2	10.54%	1	2.89%
Tree III	10	1.02	1	4.27%	0	2.03%
Path I	10	0.06	10	0.00%	10	0.00%
Path II	10	0.17	3	6.64%	3	2.11%
Path III	10	0.32	1	8.62%	0	4.27%

表 5.11 修正後分段式啟發演算法求解結果（實驗情境一）

網路圖	測試個數	CPU Time (sec)	MAX		SUM	
			Number opt.	Gap (%)	Number opt.	Gap (%)
Grid I	10	0.65	10	0.00%	10	0.00%
Grid II	10	7.55	8	1.91%	9	0.07%
Grid III	10	13.87	8	1.08%	10	0.00%
Tree I	10	0.12	7	4.86%	10	0.00%
Tree II	10	2.07	6	3.69%	6	0.33%
Tree III	10	6.92	7	1.29%	5	0.38%
Path I	10	0.13	10	0.00%	10	0.00%
Path II	10	0.53	7	3.80%	4	0.59%
Path III	10	1.24	1	7.22%	0	3.53%

圖 5.7 為修正前後演算法與 GRB1 模式的求解時間比較，可明顯看出在網路圖規模為最大時，以演算法求解所耗費的求解時間是遠小於以 GRB1 求解的時間，尤其在網路規模最大的例子中，求解時間的差距最為明顯，而修正後得演算法求解時間雖有

所增加，但依舊是遠小於以 GRB1 模式求解所耗的時間，其中以格狀網路增加幅度最大，路徑網路增加幅度為最小，因路徑網路在每次迭代中所求解的網路規模最小，故以模式求解時，增加的耗費時間較不明顯，反之，在格狀網路的求解時間則增加較多。圖 5.8、圖 5.9 及圖 5.10 則為修正前後實驗結果的誤差比較整理，可發現修正後，演算法的誤差皆有降低，且各例子找到最佳解的次數皆有增加，說明此修正方法確實能夠改善演算法的求解能力。

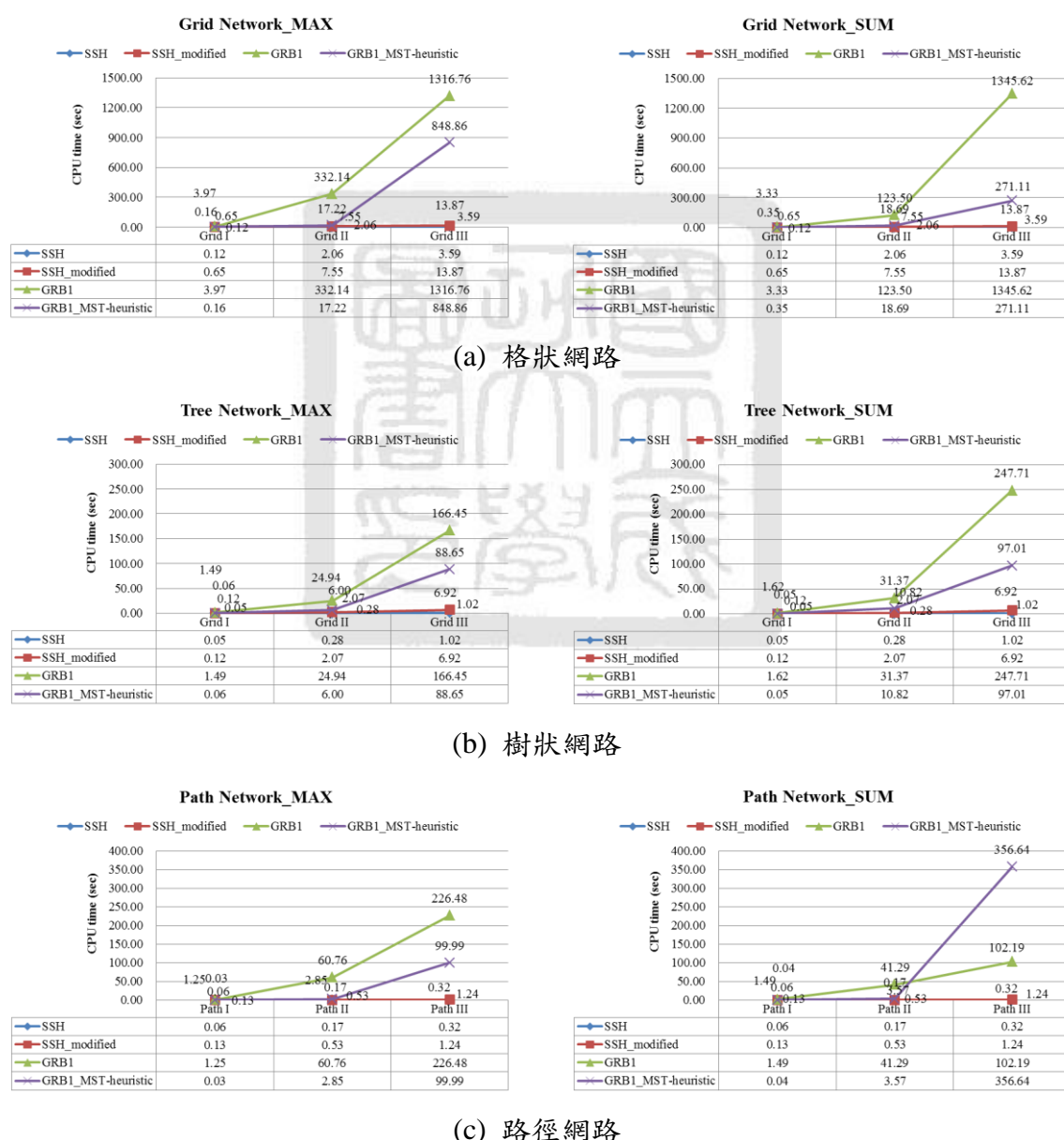


圖 5.7 修正前後的演算法與 GRB1 模式求解時間比較

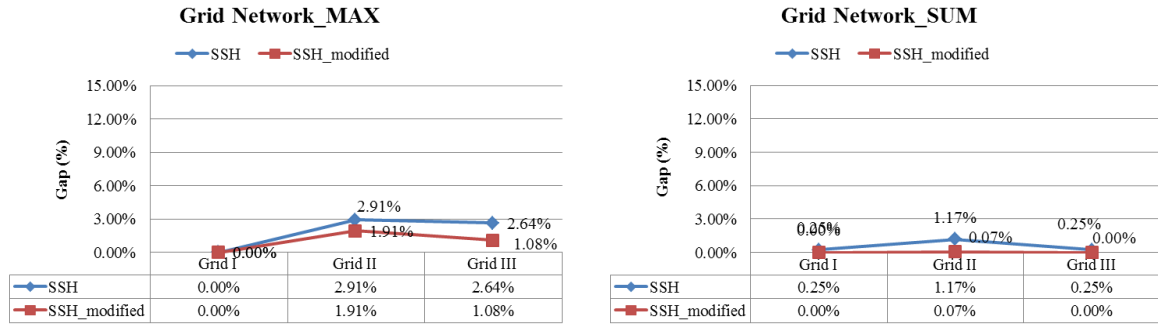


圖 5.8 修正前後的演算法求解誤差比較 (格狀網路)

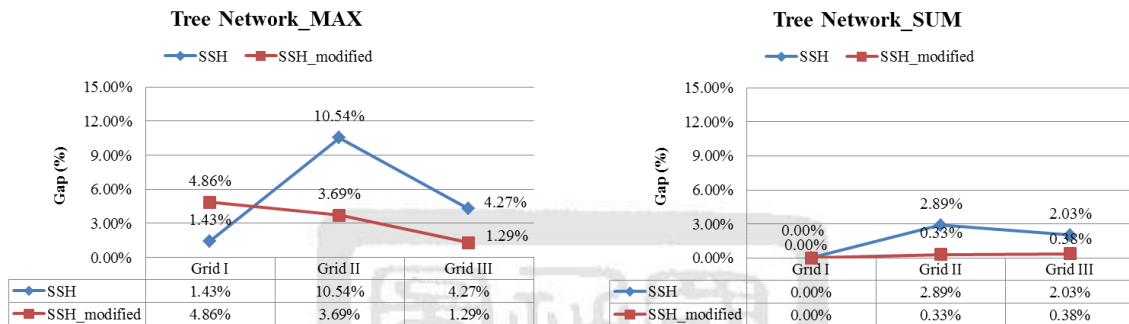


圖 5.9 修正前後的演算法求解誤差比較 (樹狀網路)

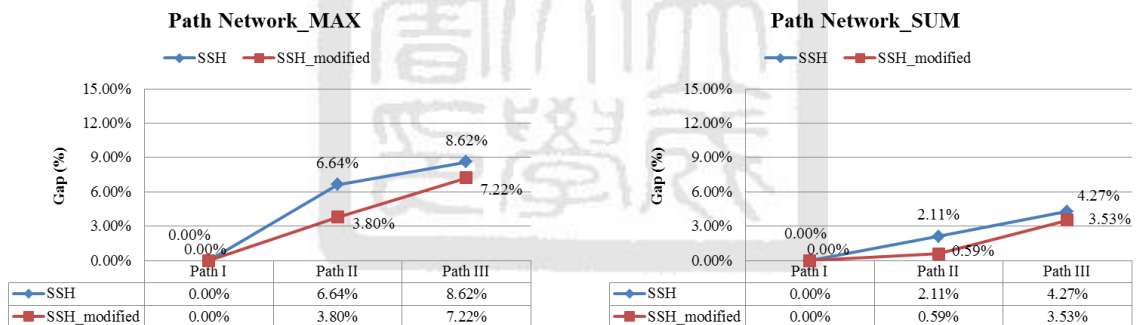


圖 5.10 修正前後的演算法求解誤差比較 (路徑網路)

接著，測試在路段修復時間增加下，對演算法求解表現的影響，測試的例子如表 5.12，差異在於，各路段的修復時間分別是由[1, 3]、[1, 5] 及 [1, 8] 三種區間大小隨機產生，測試的結果如表 5.13。由結果可看出，在目標式為 **SUM** 的表現依舊優於 **MAX**，且隨著可能的修復時間越大，演算法依舊可找到品質不錯的可行解，但在求解時間上的耗費則有明顯的增加。雖然無法保證此演算法能得到最佳解，但因災後道路重建問題本身在時間性上非常重要，而演算法則能在很短的時間內求得品質不錯的可行解，對此應在能容忍的範圍內。

表 5.12 測試之網路對象

網路圖		規模			
		節點數	節線數	起點數	路段所需 維修時間
Grid-I	格狀	10	15	1	[1, 3]
Grid-II		26	45	1	[1, 3]
Grid-III		50	91	1	[1, 3]
Grid-IV		10	15	1	[1, 5]
Grid-V		26	45	1	[1, 5]
Grid-VI		50	91	1	[1, 5]
Grid-VII		10	15	1	[1, 8]
Grid-VIII		26	45	1	[1, 8]

表 5.13 在不同路段修復時間下的求解結果

網路圖	測試 個數	CPU Time (sec)	MAX		SUM	
			Number opt.	Gap (%)	Number opt.	Gap (%)
Grid-I	10	0.65	10	0.00%	10	0.00%
Grid-II	10	7.55	8	1.91%	9	0.07%
Grid-III	10	13.87	8	1.08%	10	0.00%
Grid-IV	10	1.24	9	1.67%	10	0.00%
Grid-V	10	18.41	6	4.24%	9	0.35%
Grid-VI	10	71.86	2	8.14%	6	0.75%
Grid-VII	10	3.89	9	1.54%	10	0.00%
Grid-VIII	10	52.77	7	1.85%	10	0.00%

5.3.2 實驗情境二：考慮工作隊合作

此小節測試演算法在情境二的求解表現，表 5.14 為演算法作修正前之結果。其中網路圖 Grid III 的部分，因以 GRB1 模式無法在有限的時間內求得最佳解，使得演算法所得之解無從比較，故僅顯示演算法求解時間。

從實驗結果可看出，在增加問題複雜度後，演算法在各例的 *Gap* 變差，但演算法求解時間卻無顯著增加，且隨著網路規模擴大，求解時間的增長幅度依舊很小。以修正方式改善後，如表 5.15，各例之 *Gap* 皆有明顯的下降，說明演算法在此實驗情

境下，依舊能在很短的時間內求得品質不錯的可行解。

表 5.14 分段式啟發演算法求解結果（實驗情境二）

網路圖	測試 個數	CPU Time (sec)	MAX		SUM	
			Number opt.	Gap (%)	Number opt.	Gap (%)
Grid I	10	0.10	8	3.47%	7	0.96%
Grid II	10	1.48	1	7.84%	1	2.71%
Grid III	10	6.23	-	-	-	-
Tree I	10	0.07	7	7.94%	7	1.41%
Tree II	10	0.50	2	10.60%	1	3.91%
Tree III	10	2.43	0	5.66%	0	2.70%
Path I	10	0.08	10	0.00%	10	0.00%
Path II	10	0.61	1	15.32%	0	9.02%
Path III	10	1.81	0	16.25%	0	13.13%

註：- 表該例無最佳解可供比較

表 5.15 修正後分段式啟發演算法求解結果（實驗情境二）

網路圖	測試 個數	CPU Time (sec)	MAX		SUM	
			Number opt.	Gap (%)	Number opt.	Gap (%)
Grid I	10	0.64	10	0.00%	10	0.00%
Grid II	10	11.38	3	3.85%	3	0.82%
Grid III	10	44.99	-	-	-	-
Tree I	10	0.17	10	0.00%	10	0.00%
Tree II	10	3.87	2	6.09%	4	0.43%
Tree III	10	9.55	1	3.05%	0	0.69%
Path I	10	0.20	10	0.00%	10	0.00%
Path II	10	2.51	3	6.97%	0	3.24%
Path III	10	6.57	0	14.72%	0	5.75%

註：- 表該例無最佳解可供比較。

5.5 PSO 演算法測試

此小節測試 PSO 演算法分別在兩種實驗情境下求解不同網路結構的求解表現，並以 GRB1 模式所求的最佳解計算其誤差。由於在 PSO_{IP} 中各粒子適應值是藉由求解

RCPSP 問題來計算，但初步測試中發現，在 RCPSP 的求解時間導致 PSO_{IP} 收斂速度過於緩慢，故在此節之測試結果主要以測試 PSO_H 在實驗情境一為主，其中各例之粒子數設定為 100，迭代次數上限為 20000 次，表 5.16 為實驗結果。

由實驗結果顯示， PSO_H 在各例求解表現皆不理想，且在各網路圖中，只要網路規模達中規模以上 PSO_H 就無法收斂至最佳解，最大原因在於粒子適應值計算，由於目前無演算法可在多項式時間內求解 RCPSP 問題，若以數學規劃方式求解，計算各粒子適應值耗費之時間將拉長，導致整體 PSO 運作時間過長，希望 PSO_H 能夠藉由改以貪婪式演算法替代適應值計算方式，來改善此一問題。然而，此貪婪式演算法之求解誤差卻太大，無法達到設計 PSO_H 之預期效果，故在 PSO 改善部分，期望能發展一可在短時間內求得一品質不錯之可行解來改善 PSO_H 的求解表現。

表 5.16 PSO_H 之測驗結果

網路圖	測試個數	MAX			SUM		
		CPU Time (sec)	Number opt.	Gap (%)	CPU Time (sec)	Number opt.	Gap (%)
Grid I	10	20.66	8	3.67%	20.61	3	4.72%
Grid II	10	64.21	0	19.58%	64.45	0	9.02%
Grid III	10	138.18	0	29.79%	136.47	0	18.62%
Tree I	10	12.21	2	113.57%	12.48	0	88.80%
Tree II	10	36.09	0	239.93%	35.80	0	181.92%
Tree III	10	75.84	0	203.66%	75.38	0	180.32%
Path I	10	12.37	5	16.00%	12.32	2	16.08%
Path II	10	36.03	0	18.57%	35.88	0	11.61%
Path III	10	74.13	0	19.71%	73.88	0	11.86%

5.6 小結

在此章節中，首先各別測試了 GRB1 模式在兩種實驗情境下，求解三種不同網路結構的求解表現，並且皆個別測試三個規模大小，其結果顯示在三種網路結構下，隨著網路圖規模越大時，求解之表現越差，且部分例子無法求得最佳解，故以 MST-heuristic 作為 GRB1 模式的初始解，來優化模式的求解效率。從優化的測試結果

中，可發現 MST-heuristic 確實能夠有效降低 GRB1 模式的求解時間，並且增加求得最佳解的次數，但仍舊有少數的反例顯示 MST-heuristic 可能使求得最佳解的次數減少，此結果說明 MST-heuristic 雖能有效改善 GRB1 模式求解效率，但不能保證其作為初始解的品質。

接著，本研究將 GRB1 模式與文獻中 Averbakha and Pereira (2012) 提出之 FNCP 進行比較，由於 FNCP 是求解在一組工作隊下的網路重建問題，因此本研究分別測試 GRB1 與 FNCP 在一組工作隊下的求解表現。由結果顯示，當目標式為 MAX 時，GRB1 遠輸於 FNCP；但當目標式為 SUM 時，在求解小、中規模的網路時，GRB1 只略輸 FNCP；而求解大規模的網路（節點數為 50）時，FNCP 則是完全無法求得最佳解，然而 GRB1 仍舊能求解出部分例子的最佳解，此結果印證 Averbakha and Pereira (2012) 的測試結果：FNCP 無法求解節點數大於 30 以上的網路。

在 GRB2 模式的測試上，雖然 GRB2 可求解延伸問題：考量工作隊移動時間的網路重建問題，但現實中各點之間的移動時間應隨著搶修計劃的進行而有所變動，GRB2 在使用上則需限制點對點運輸時間為固定一常數，且由測試顯示其求解效率不佳，故在後續研究中，可除去運輸時間為固定常數之假設，使問題更具一般性，並對其發展有效的演算法來進行求解。

最後，本研究測試分段式啟發演算法的求解表現，並以 GRB1 模式求得的最佳解來計算演算法的誤差，由結果顯示分段式啟發演算法能夠在很短的時間內，便可求得一品質不錯的解，但隨著網路規模增大，誤差亦隨著增加，其原因在於分段式啟發演算法是藉由先縮小網路，在每次迭代中加入新的節點求解，因此求解時會忽略執行各作業對後續結果的影響；依據這個分析結果，本研究再藉由放寬每次迭代中求解的時間區段長度，但只保留能在原時間區段內執行完畢的作業，來改善此一原演算法的缺點，並由測試結果發現，此修正能有效降低演算法的誤差；雖然此修正演算法會增加一些求解時間，但其增加的幅度依舊遠小於直接以 GRB1 模式求解所耗費的時間。

第六章

結論與未來研究

天然災害的發生常造成生命與財產的重大損失，以往工程單位在實際搶修排程中，一般是由決策者依經驗進行，但依經驗指派缺乏全面系統性的客觀學理基礎，其決策就算可行，經常未必最佳；倘若決策失當，恐造成人命損失等難以彌補的後果。在過去針對災後管理的文獻中，多著重於救難物資配送問題，但要配送物資的前提是需要可通向各災點的道路網，然而對於路網重建的議題卻討論較少。緣此，本研究希望藉由探討災後緊急路網重建問題，能夠提供工務單位在有限時間內找出最佳效益的搶修計劃。根據模式架構、啟發式演算法求解結果，本章將歸納出結論與建議。

6.1 結論

本研究之目的在於求解有限資源下之網路修復問題，以求得最有效率之路段搶修順序及資源排程，且將路段具多種修復方式的可能性納入考慮。在過去文獻中，可發現多在已知欲修復之路段情況下，來進行作業的排程規劃，然而此種方法在第一章已藉由例子說明，在不考慮後續作業排程的下所做的搶修路徑決策，並非最佳的路徑決策，意即此二階段是互相影響的。有鑑於此，本研究將在尚未決定欲修復之路段情況下，進行搶修作業及資源的排程規劃。

在已知修復路徑的情況下，便可得知作業之間的順序，依此可先將修復路徑轉為 PERT 網路圖的型式，如此問題便轉化為常見的「具資源限制下之專案排程問題」(Resource Constrained Project Scheduling Problem, RCPSP)，且為 RCPSP 問題的一特例，因轉化的 PERT 網路圖為 out-tree 的結構，即各作業只會有一前置作業(predecessor)；但由於本研究欲在尚未決定修復之路徑情況下，進行排程規劃，無法在事前得知搶修作業間的作業順序，以直覺的作法是窮舉出一個網路中的所有展開

樹，來一一求解 RCPSP 問題找出最佳的路徑。但一個網路中的展開樹有非常多個，且 RCPSP 本身為一 NP-hard 的問題，且本研究再加入多種修復的可能性，更增添求解 RCPSP 的困難度。由此分析可知本研究問題求解十分困難，鑒於 RCPSP 已有較多文獻可參考，本研究以 RCPSP 觀點發展一整數規劃模式來進行求解。

過去文獻中，多假定各路段只能由一組工作隊來進行修復以及每一組工作隊的修復能力皆相同；但在實際的搶修計劃中，工作隊的修復能力可能會因配備不同而相異，且各路段亦可同時由兩組以上的工作隊合力修復，因此，在問題中加入路段具多種修復方式的可能性，能夠更加貼近實務，亦為本研究與過去文獻的不同之處。

在第三章中，本研究首先說明問題在不考慮多種修復方式下，如何針對單一與無限組工作隊等兩種特例，將之分別轉換成最小展開樹問題（Minimum spanning tree, MST）及最短路徑樹問題（Shortest path tree, SPT）求解，而針對一般性情況，即考慮多種修復方式以及多組工作隊，本研究則提出一整數規劃模式 GRB1。在模式測試上，模式雖可求得最佳解，但隨著網路規模變大，所需求解時間亦大幅提高；不過若與過去文獻 Averbakha et al. (2012) 提出之 FNCP 模式相比，結果顯示 GRB1 較能夠處理節點數高於 30 個以上的網路。

觀察影響模式大小的因素中，有網路節點數 $|N|$ 、節線數 $|A|$ 、工作隊數 $|R|$ 、各路段的修復方式 $M_{(i,j)}$ 及預估之時完工上限 T 等。但其中除了完工上限 T 以外，其他皆是固定無法變動的，因此能夠在事前預估一接近實際最後完工時間的上限 T ，便可有效降低模式的規模，對此本研究設計二分逼近法來推估此時間上限 T 。另外，為了快速求得模式結果，藉由設定模式的初始解或設定上下界亦是常用的方法。而從最佳解性質已知：最佳搶修路徑必為一展開樹，故以網路中任一展開樹求解 MRCPSP 問題，其結果除了可作為模式的初始解及時間上限的預估外，其目標值亦可作為目標值的上界；但網路中的展開樹產生方式有很多種，本研究主要採取的策略為：先以各節線最短的修復時間作為各節線的權重後，再以此權重找尋網路中的最小權重展開樹，其所得之結果為 MST-heuristic。數值測驗結果顯示 MST-heuristic 確實能夠有效縮短模式

的求解時間，但部分反例亦可能讓 *MST-heuristic* 的結果導致求解時間增長。因 *MST-heuristic* 可能會高估最後完工時間 T ，造成模式有多餘的變數及限制式，故無法保證 *MST-heuristic* 作為初始解的品質。儘管可再藉由其他方法來優化模式求效率，但隨著網路規模的增大，優化的成果有限，因此如何設計出具效率性之演算法將極具挑戰性。

在第四章中，本研究提出一分段式啟發演算法，此啟發式解法的概念，係透過貪婪式演算法的觀念，由近而遠先求解較近災點的搶通方式，以其為基礎再逐漸擴張搶通範圍直到全部災點皆搶通為止。由測試結果顯示，此演算法能夠在很短的時間內求得一品質不錯的可行解，甚至是最佳解。相對於問題本身具時間急迫性而言，此一結果應在可接受的範圍內。最後，本研究則探討粒子群演算法（Particle swarm optimization, PSO）求解本問題之效能，但由於問題之複雜性，使得 PSO 的收斂速度太過緩慢，相較於本研究所提出之分段式演算法而言，其求解的品質與穩定性皆顯得遜色許多。

整體而言，對於緊急網路重建問題，本研究加入多種修復方式的考量，相較於過去文獻更能貼近實務上的需求。在求解方法上，主要應用數學規劃及演算法的設計，而建構之數學模式與過去文獻的模式比較上，本研究的模式在處理大規模的網路圖的表現較顯優異，且透過本研究提出的分段式啟發演算法，能夠在短時間內求得一品質不錯的可行解，此方法雖然隨著問題規模變大，誤差略為提高，但因問題本身為一即時性之緊急排程規劃問題，時間性甚為重要，故其增加的誤差應屬可接受範圍，可以此彌補模式在求解大規模網路的不足。

綜合以上結論，本論文的具體貢獻歸納如下：

1. **加入路段具多種修復方式的可能性：**有別過去文獻多探討單一修復方式之路網重建問題，加入多種修復方式的考量，提出考慮多種修復方式之災後緊急路網重建最佳化問題，更加貼近實務，並說明如何針對單一與無限組工作隊等兩種特例，將之分別轉換成最小展開樹問題及最短路徑樹問題求解。。

2. **提出兩種整數規劃模式求解：**有別過去文獻，本研究以 RCPSP 的觀點，針對路網重建問題，建構一整數規劃模式 GRB1 進行求解；並延伸考量資源運輸移動時間段完工時間的影響，另以單一商品網路流問題的概念建構第二種整數規劃模式 GRB2，而 GRB2 亦能處理無資源運輸時間的情況。
3. **發展分段式啟發演算法：**由於此種最佳化問題，會隨著問題規模變大，求解時間亦大幅提升，故須仰賴演算法來行更有效率的求解。對此，本研究透過貪婪式演算法的觀念，設計分段式啟發演算法，由測試結果顯示分段式啟發演算法可在很短的時間內求得一品質不錯的可行解，且能處理大規模的網路圖。
4. **發展 PSO_{IP} 及 PSO_H 演算法：**本研究亦探討粒子群演算法求解本問題之效能。

6.2 未來研究

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. **加入干擾因素：**由於本研究是在假設災情資訊已完整蒐集，決策單位在整體考量下作出決策。但災害發生後，亦有可能發生二次災害等不確定因素，可能造成各災點搶修作業的延誤或是有新災點產生等。除此之外，尚有因災情資訊預估錯誤，造成災點搶修延誤等干擾情形，建議未來可進一步針對相關干擾因素進行探討。
2. **增加問題的即時性：**本研究目前模式主要是以災區所擁有之固定之搶修資源來進行規劃排程，且各組工作隊需待決策單位進行排程規劃結束後，才開始進行搶修作業。但在實際上，災害發生後，各地的工作隊便會即刻前往災點進行搶修，且其他地區亦會陸續加入支援，故建議未來能夠考慮此類狀況，針對已派出及後續支援之工作隊同步進行規劃排程，使模式更具即時性。
3. **考慮工作隊移動時間對完工時間的影響：**因本研究是假設資源運輸時間遠小於修復所需時間，而不將運輸時間納入搶修時程中，但實際上，受災區域廣大時，資源運輸所耗費時間對整體完工時間亦會有很大的影響，故如何將運輸時間結合至

網路修復問題中，亦為值得深入探討之研究議題。對此本研究雖已初步建構一整數規劃模式 GRB2，但在初步的求解測試結果中，GRB2 在求解節點數高於 26 個以上的網路圖的皆表現不佳，故如何繼續對此問題設計出具效率性之演算法亦極具挑戰性。

4. **結合賑災物資運輸問題：**災區的路完重建完成後，決策單位接著遇到的難題便是如何有效率地進行賑災物資運輸，以往此二項工作是分開且獨立執行的，忽略其間的相關性，換句話說，賑災物資配送若能清楚掌握道路搶通之資訊，而災後緊急搶修工程作業若能將賑災物資配送路線加以考量，則更可兼顧災後重建與災民的生活。



參考文獻

- Aksu, D.T. and L. Ozdamar, (2014), "A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation," *Transportation Research Part E: Logistics and Transportation Review*, **61**, 56-67.
- Altay, N. and W.G. Green III, (2006), "OR / MS research in disaster operations management," *European Journal of Operational Research*, **175(1)**, 475-493.
- Averbakha, I., (2012), "Emergency path restoration problems," *Discrete Optimization*, **9**, 58-64.
- Averbakha, I. and J. Pereira, (2012), "The flowtime network construction problem," *IIE Transactions*, **44(8)**, 681-694.
- Blazewicz, J., J.K. Lenstra and A.H.G. Rinnooy Kan, (1983), "Scheduling subject to resource constraints Classification and complexity," *Discrete Applied Mathematics*, **5(1)**, 11-24.
- Brucker, P., (2007), *Scheduling algorithm^{5th}*, Springer Press, New York.
- Coelho, J. and M. Vanhoucke, (2011), "Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers," *European Journal of Operational Research*, **213(1)**, 73-82.
- Debels, D. and M. Vanhoucke, (2005), "A bi-population based genetic algorithm for the resource-constrained project scheduling problem," *Lecture Notes in Computer Science*, **3484**, 378-387.
- Debels, D. and M. Vanhoucke, (2007), "A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem," *Operations Research*, **55(3)**, 457-469.
- Hartmann, S. and D. Briskorn, (2010), "A survey of variants and extensions of the

- resource-constrained project scheduling problem,” *European Journal of Operational Research*, **207(1)**, 1-14.
- Krüger, D. and A. Scholl, (2010), “Managing and modelling general resource transfers in (multi-)project scheduling,” *OR Spectrum*, **32(2)**, 369-394.
- Li, K.Y. and R.J. Willis, (1992), “An iterative scheduling technique for resource-constrained project scheduling,” *European Journal of Operational Research*, **56(3)**, 370-379.
- Liberatore, F., M.T. Ortuño, G. Tirado, B. Vitoriano and M.P. Scaparra, (2014), “A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in humanitarian logistics,” *Computers & Operations Research*, **42**, 3-13.
- Matisziw, T.C., A.T. Murray and T.H. Grubescic, (2010), “Strategic network restoration,” *Networks and Spatial Economics*, **10(3)**, 345-361.
- Meshkovskiy, K.A. and A.Y. Rokotyan, (1992), “Restoration of communications network connectivity following the failure of transmission junctions and lines,” *Telecommunications and Radio Engineering*, **47**, 1-5.
- Mingozzi, A., V. Maniezzo, S. Ricciardelli and L. Bianco, (1998), “An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation,” *Management Science*, **44(5)**, 714-729.
- Nurre, S.G., B. Cavdaroglu, J.E. Mitchell, T.C. Sharkey and W.A. Wallace, (2012), “Restoring infrastructure systems: An integrated network design and scheduling (INDS) problem,” *European Journal of Operational Research*, **223(3)**, 794-806.
- Talbot, F.B., (1982), “Resource-constrained project scheduling with time-resource tradeoffs - the nonpreemptive case,” *Management Science*, **28**, 1197-1210.
- Ven Peteghem, V. and M. Vanhoucke, (2010), “A genetic algorithm for the preemptive and

non-preemptive multi-mode resource-constrained project scheduling problem,”

European Journal of Operational Research, **201(2)**, 409-418.

Yan, S. and Y.L. Shih, (2009), “Optimal scheduling of emergency roadway repair and subsequent relief distribution,” *Computers & Operations Research*, **36(6)**, 2049-2065.

