

國立交通大學

資訊學院 資訊學程

碩士論文

以路徑選擇策略為基礎的電信系統故障派工研究

A route-based algorithm for task assignment in
Telecommunications system breakdown

研究生:李坤達 Student : Kun-Da Lee

指導教授:陳登吉 教授Advisor : Dr. Deng-Jyi Chen

中華民國 一百零一年七月

以路徑選擇策略為基礎的電信系統故障派工研究

A route-based algorithm for task assignment in
Telecommunications system breakdown

研究生:李坤達

Student: Kun-Da Lee

指導教授:陳登吉

Advisor: Dr. Deng-Jyi Chen



in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

July 2012

Hsinchu, Taiwan, Republic of China

中華民國 一百零一年七月

以路徑選擇策略為基礎的電信系統故障派工研究

學生:李坤達

指導教授:陳登吉 博士

國立交通大學 資訊學院 資訊學程 碩士班

摘要

有鑑於數位化時代的來臨，人們日常生活的範圍不脫電信產業。電信系統設施的維修與養護亦相對重要；而電信設備損壞的情形，以颱風、地震等大型災害發生後最為嚴重屆時常伴隨著維修人員與資源短缺的情形發生。故如何依需求資訊、維修資源，規劃維修路徑以提昇維修效能、降低電信系統故障對人們生活的影響，即成為一重要的課題。

本研究之電信系統故障派工 (Task Assignment) 包含資源分配(Resource Allocation) 與路徑選擇(Path Selection) 的問題，資源分配(Resource Allocation) 是指維修設備(材料) 與維修人員的配置，好的路徑選擇(Path Selection) 是行車距離最短，壅塞程度最小，而有效率的派工方法是讓維修效率最高的修護人員在最短的時間內到達叫修地點。一個有效率的派工方法對業者做人員調動，資源分配可以有很大的幫助

符合電信系統派工問題限制下，為了達到每一次的派工都能讓維修效率最高的人員在最短的時間內，到達故障地點。我們先用MFST(Minimal File Spanning Tree, MFST) 演算法找出所有符合條件限制的派工路徑, 利用不同維修點有不同的維修效率，來計算一項工作分派的等待時間，找出一個延遲時間最小與維修效率最高的派工路徑。

關鍵字: 工作分配問題(Task Assignment Problem)、資源分配(Resource Allocation)、路徑選擇(Path Selection)、MFST(Minimal File Spanning Tree)

A route-based algorithm for task assignment in

Telecommunications system breakdown

Student: Kun-Da Lee

Advisor: Dr. Deng-Jyi Chen

Degree Program of Computer Science

National Chiao Tung University

Abstract

The digital era, telecommunication is in our every aspect of life. Accordingly, the repair and maintenance of the telecommunication systems and facilities are also important. The most serious damages are always associated with the shortage of maintenance personnel and resource when the large disasters such as typhoons and earthquakes occur. Therefore, how to improve maintenance efficiency and reduce the failure of telecommunication systems by using demand for information, maintenance resources and planning maintenance path in order to decrease the effects on society has become an important issue.

In this study, task assignment of telecommunication system breakdown contains resource allocation and path selection. Resource allocation of resources refers to allocation of the maintenance equipments (materials) and maintenance personnel. A good the path selection is the shortest traveling distance and the minimal congestion. Moreover, the most efficient dispatching method is to arrive the the repair site at the shortest time. An efficient dispatching method is beneficial for resource allocation.

Found to comply with the limits for the telecommunications system task assignment, in order to achieve every dispatching can make the repair efficiency highest in the shortest possible time, to reach the fault location. We use the Minimal File Spanning Tree(MFST) algorithm to identify all dispatching path complying with the constraints. The use of different maintenance points have different maintenance efficiency, to calculate the waiting time for an assignment, the highest dispatching path to identify a minimum delay time and maintenance efficiency.

Keyword: Task Assignment Problem、Resource Allocation、Path Selection、Minimal File Spanning Tree

誌謝

能夠完成這份論文，首先對指導教授陳登吉老師至上最誠摯的謝意，感謝老師的教誨與提攜，將懵懂無知的我，帶入這塊研究的領域，這二年來，不論在研究上或是生活上都給了我很大的鼓勵與指引，更時常與我們分享人生大道裡與生活經驗，讓我學到了很多書本上學不到的東西。特別感謝孔崇旭老師耐心與不厭其煩的教導，還犧牲假日的休息時間額外的給我指導，很感謝有這麼多好老師的幫助，讓我可以順利的完成這兩年的課業及論文的研究，也讓我受益良多。老師，謝謝您。

同時，也感謝在交大陪伴我、互相砥礪的同學、學長姐、學弟妹，尤其是李鎮宇學長不厭其煩的指導我，讓我學到正確的研究方式。還有許多鼓勵、幫助我的朋友們。在課業上，有你們不遺餘力的幫助；遇到困難時，也總是有你們安慰與鼓勵。因為有你們，豐富了我的生活。認識你們，是我的榮幸。

最後，感謝我的家人們，因為有你們的支持與栽培，成就了今天的我。謝謝你們讓我在沒有後顧之憂的環境下完成了碩士學位，未來我將會更加努力，以期有好的作為。

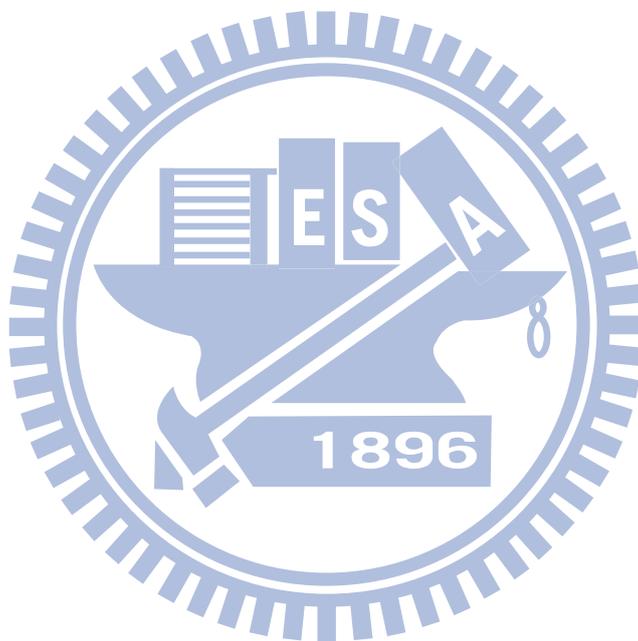
目錄

摘要	i
誌謝	iii
目錄	iv
圖目錄	v
一、緒論	1
1.1 研究背景	1
1.2 研究動機	2
1.3 研究目的	2
1.4 研究流程	3
二、文獻探討	5
2.1.工作分派問題(task assignment problem)	5
2.2. Minimal File Spanning Tree	10
2.2.1. MFST 演算法	10
2.2.2. FREA (Fast Reliability Evaluation Algorithm)演算法	11
2.2.3.圖形化簡實例說明:	13
2.2.4. Minimal File Spanning Tree 實例說明	15
三、模式構建及求解方法	17
3.1.問題描述與限制	17
3.2.參數及目標函數定義	18
3.3.演算方法說明	19
3.4. Time Complexity	20
四、實例展示	21
4.1.模型建置流程圖	21
4.2.實際案例執行說明	30
4.3.程式展示	34
4.3.1.初始畫面	34
4.3.2.定義各項資源	35
4.3.3.即時調整路徑權重	38
4.3.4.執行派工任務	39
4.3.5.執行結果圖形輸出	40
五、結論	42
5.1.結論	43
5.2.未來研究方向	43
參考文獻或資料	44

圖目錄

圖 1:研究流程圖	4
圖 2:TAP 問題分類圖	5
圖 3: Generation of replicated trees in MFST algorithm.....	10
圖 4: Simple distributed computing system	12
圖 5: Example of degree-1 reduction	13
圖 6: Example of degree-2 reduction	14
圖 7: Simple distributed computing system	15
圖 8: Example of degree reduction.....	15
圖 9: Example of DCS and all MFST's for P1 under consideration.....	16
圖 10: Example of DCS and all MFST's for P 1 under consideration.....	16
圖 11: 模型建置流程圖	21
圖 12:實際應用例子圖 1.....	22
圖 13: 實際應用例子圖 2.....	22
圖 14:抽象化後之應用例子圖	23
圖 15:定義路徑權重之例子圖	24
圖 16:加入節能資源之例子圖	26
圖 17:加入節點資源之例子圖	28
圖 18: 不同節點間有不同的工作的執行效率	29
圖 19:應用例子圖	30
圖 20:MFST 化簡圖.....	31
圖 21:MFST 化簡後之 Sub-graphs	33
圖 22:建議派工的路線圖.....	33
圖 23:初始畫面	34
圖 24:自行設地圖-台中地區	35
圖 25 自行設地圖-台北地區	35
圖 26 建立派工地點	36
圖 27 建立 2 點連線圖	36
圖 28 資源修改圖	37
圖 29:更新路徑權重圖	38
圖 30:執行派工任務圖-1.....	39
圖 31 執行派工任務圖-2.....	39
圖 32:輸出結果圖-1.....	40
圖 33 輸出結果圖-2.....	40
圖 34 實際案例執行畫面	41
圖 35 :實際案例執行派工任務	41

圖 36:執行派工結果-1.....42
圖 37:執行派工結果-2.....42



表目錄

表格 1:TAP 相關論文比較表	8
表格 3:定義節點名稱	23
表格 4:路徑權重表	24
表格 5:原 MFST 符號定義表	25
表格 6:電信系統故障派工符號表	25
表格 7:資源定義表	27
表格 8:即時路徑更新表	32



一、緒論

1.1 研究背景

有鑑於數位化時代的來臨，人們日常生活的範圍不脫電信產業，所以電信系統設施的維修與養護亦相對重要；而電信設備損壞的情形，以颱風、地震等大型災害發生後最為嚴重，屆時常伴隨著維修人員與資源短缺的情形發生。故如何依需求資訊、維修資源，規劃維修路徑以提昇維修效能、降低電信系統故障對人們生活的影響，即成為一個重要的課題。

在傳統語音營收成長受限下，電信業者已經積極在進行業務轉型，目前主要電信業者都已逐漸自單純提供服務平台的業者轉型成為綜合網網路業者，電信業者所提供的服務已不在只是傳統的固網、數據與行動業務，而是結合資通訊 (Information and communication technologies, ICT) 技術，提供更多元的服務。例如以往維修人員只需負責與電信相關的業務就好，例如固網的修復，電話總機系統、行動機地等等，近年來則隨著業務種類的增加，負責的項目也愈來愈多，例如 IPTV、智慧電力系統、太陽光電系統、智能交通系統(Intelligent Transportation Systems, ITS)，路口監控系統等等資訊與通訊結的業務。

隨著業務的競爭，成像是以本因素的考量，維護人員已無法像以往一樣，只負責單一業務的維護，而是負責維護的工作種類愈來愈多。業務種類的增加使得維修所需的資源(維修所需料件、技能等等..)種類愈來愈多，如何做好資源配置(Resource Allocation)與人員的配置對電信業者就愈來愈重要，

客戶滿意成為企業競爭優勢因素之一，如何降低查修的服務失誤，讓每一次的查修派工都能讓最有經驗的維修人員在最短的時間內到達叫修地點，對電信業者就愈加重要，尤其在業務量與種類不變的變多，且人員不斷的縮減下，有效率的派工方法對電信業者就十分的重要。

1.2 研究動機

電信系統故障派工包含資源分配(Resource Allocation)與路徑選擇(Path Selection)的問題，一個有效率的派工方法，對業者做人員調動，資源分配可以有很大的幫助。資源分配(Resource Allocation)是維修設備(材料)與維修人員的配置，路徑選擇(Path Selection)問題包含行車距離最，壅塞程度，而有效率的派工方法是讓最有維修效率的修護人員在最短的時間內，到達發生故障問題地點。電信系統派工問題較複雜，要達到最佳化派工較為不易，其問題限制有：

1. 處理相同故障不同人有不同的維修效率(proficiency)[1]:每個機房的人員有不同的技能，處理的障礙也不同，且每個工地點的人員對每一項故障有不同的處理執行效率。
2. 多個啟始點(Multi-Origin)的選擇沒有確的出發點，可以從甲機房出發去做修復工作，也可以從乙機房出發，只要可以完成派工所需要的全部工作即可。
3. 具優先順序的必經點問題(Designated points Program) [2]:維修人員接到派工任務後，需先判斷是否有足夠的維修材料，若無則需先到料庫拿取派工任務所需的維修材料，派工路線就需先到料庫，也不能是先到叫修地點才到回到料庫。例如要如要修復油封工廠的太陽光電系統需先去料庫拿材料，才到油封工廠。
4. 一次派工分派包含多個(種)故障申告[3][4]: 同一批派工有多個故障點與多個種類的故障類型(一個 task 有很多的 job)，例如一項派工包含:修理醫院的 PBX、電力監控系統與修理橡膠工廠的太陽光電系統。而不是像一般的工作分派都是單一種類的工作分派，如修理交通號等等。

1.3 研究目的

1. 派工的初始條件:
 - 已知所有不同種類及數量的叫修任務
 - 不同的路徑距離及路況條件
 - 分散的維修資源
 - 分散的人力
2. 據上述條件，做最有效率的工作分派建議，目的詳述如下:

需符合電信系統派工問題限制，以達到每一次的派工都能讓維修效率最高的人員在最短的時間內，到達故障地點。

 - (1) 利用 MFST 演算法找出所有符合條件限制的派工路徑。
 - (2) 利用不同維修點有不同的維修效率，來計算一項工作分派的等待時間，找出一個延遲時間最小與維修效率最高的派工路徑。

1.4 研究流程

本研究之研究流程說明如下，流程圖如圖 1.1 所示：

1. 針對研究目的及背景加以了解，進而界定問題：
對於本研究背景與動機做一全面性了解，以建立明確的研究方向，釐清問題之界限；參考電信障派工的流程，訂定研究所需的假設與限制，做為本研究的基礎。
2. 文獻探討：
本研究文獻回顧包括工作分配(Task Assignment Program, TAP)問題、最小檔案擴張樹(Minimal File Spanning Tree, MFST)等。首先回顧 Task Assignment Program 的種類，了解不同的 Task Assignment Program 所要解決的問題重點與解決方法。接著探討 Minimal File Spanning Tree，尋找與電信系統故障派工相關的特性。
3. 模型設計：
依電信系統故障派工的限制，設計出符合實際派工之模型。
程式設計與撰寫：
4. 本研究將以物件導向設計 (Object-oriented design; OOD) 程式語言 C++，進行程式的設計與撰寫，因此於確立演算策略概念之後，將針對研究的需求設計物件，以利程式撰寫之實作。
5. 結論與建議：
最後，歸結以上各項目，提出本研究的結論與建議。

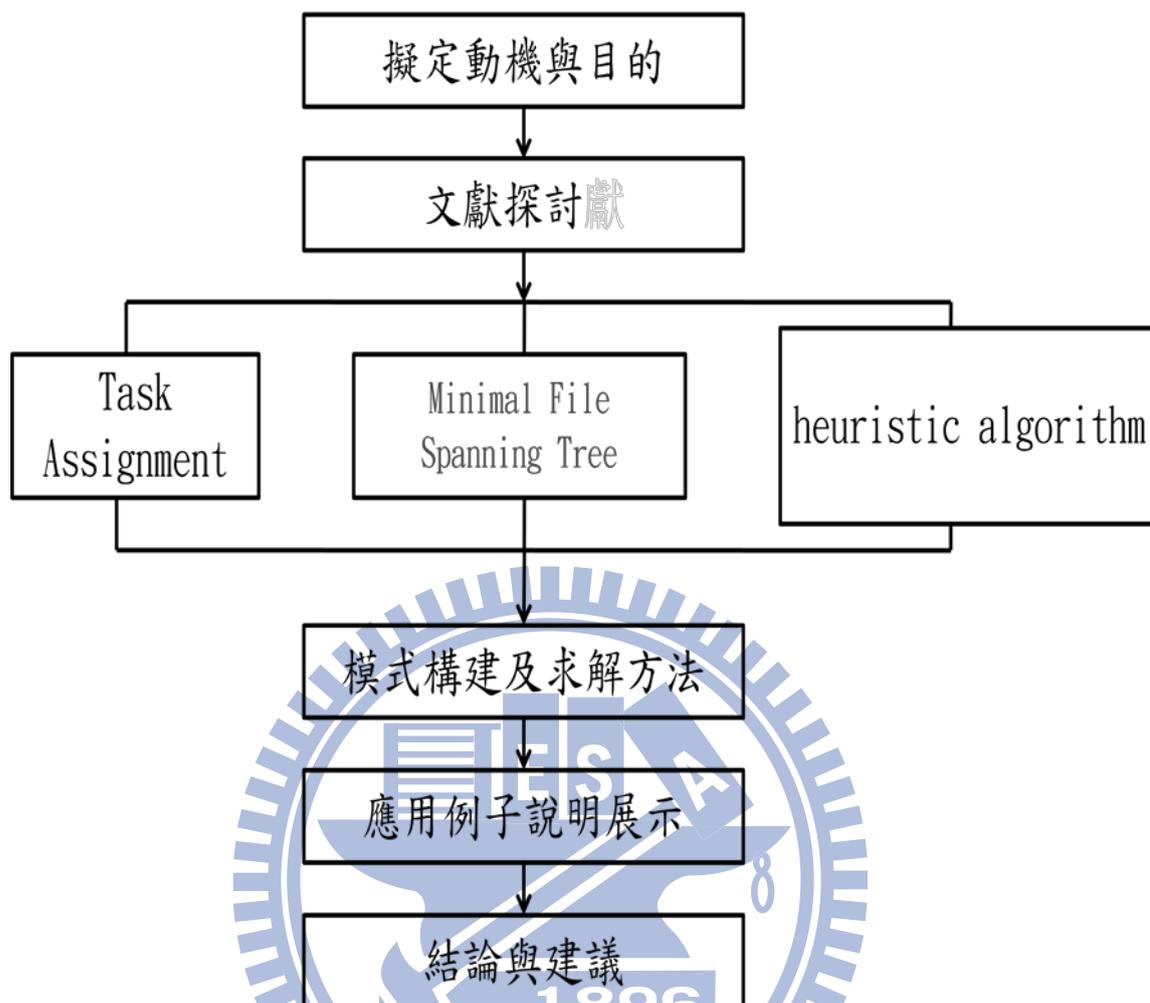


圖 1:研究流程圖

二、文獻探討

2.1. 工作分派問題(task assignment problem)

傳統的 TAP 問題中，大多都是針對系統成本最小化 (Cost-minimization task assignment problem, CMin-TAP) 或是針對系統可靠度最大化 (Reliability-maximization task assignment problem, RMax-TAP) 來做探討，較少有論文有提出針對最小化系統成本和最大化系統可靠度一起來做探討 (Cost-Reliability Analysis task assignment problem, CRA-TAP)。在本研究中會針對不同類型的 TAP 問題來一一做探討。

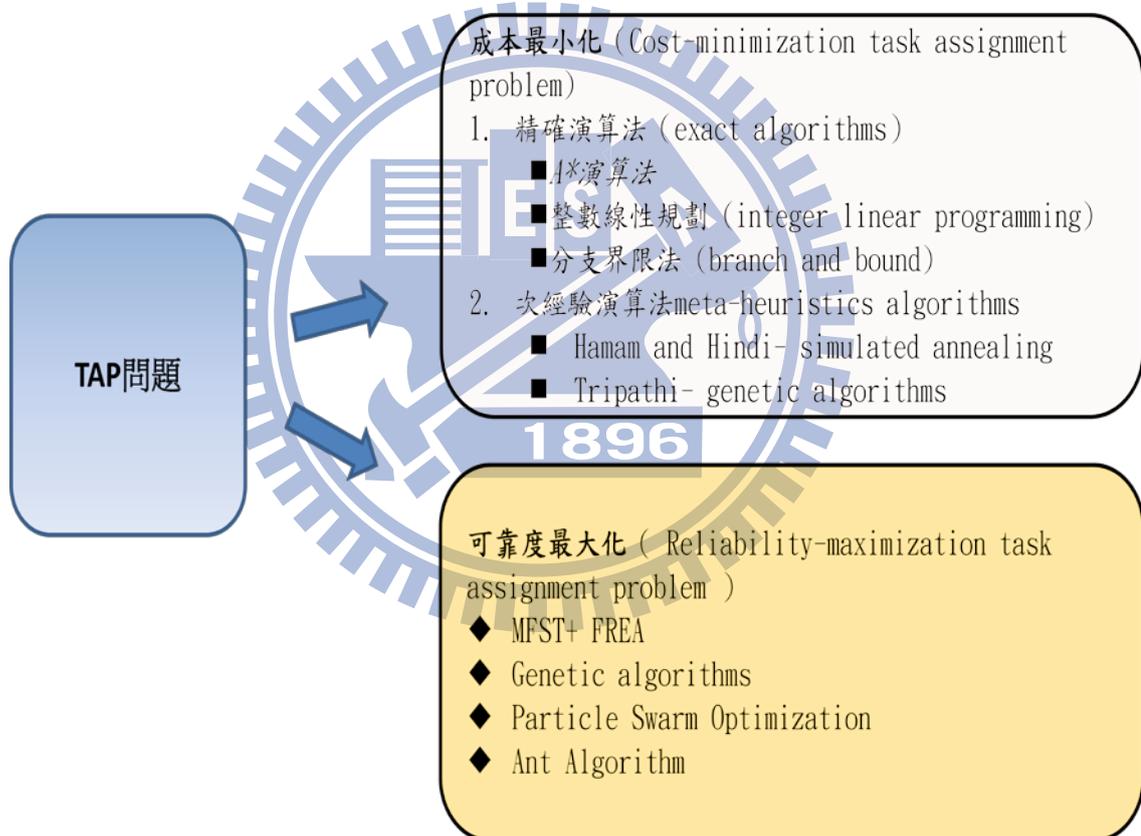


圖 2:TAP 問題分類圖

2.1.1. 成本最小化之 TAP

過去提出來解決 CMin-TAP 的方法可以大致分為精確演算法 (exact algorithms) [5-7] 與次經驗法則演算法[8-10]兩大類。精確演算法是利用數學規劃(mathematical programming)尋找問題的絕對最佳解，例如 Kafil and Ahmad [5]提出用 A*演算法可以保證求取問題絕對最佳解，使用 A*演算法可以減少搜尋的空間並且較快速的找到 TAP 的最佳解。但是仍然不能使用在大型的問題上面，因為會花費太多的時間。Ernst et al. [6] 則是利用整數線性規劃 (integer linear programming) 來求解問題，但是同樣的使用於大型 TAP 問題上，也無法在合理的時間內獲得最佳解。Hagin et al. [7] 則是利用分支界限法 (branch and bound) 來解決 TAP。雖然使用精確演算法可以找到 TAP 的最佳解，但是當問題變大的時候，無法在有效時間內找到最佳解。所以使用次經驗法則演算法的目的就是希望可以在有限的時間內找到趨近於最佳解的答案，它的特點是利用一些簡單的經驗法則，從一個隨機的初始解出發，反覆推求下一個更好的答案。目前已經提出解決 TAP 的次經驗法則演算法包括 Hamam and Hindi [8]提出利用模擬退火演算法(simulated annealing, SA)去解決 TAP 問題，在這篇論文當中有整理出幾個常見的 TAP 數學模式，並且針對不同的限制條件也有相關的說明。Tripathi et al.[9] 提出使用基因演算法 (genetic algorithms, GA) 去解決多個模組的 TAP 問題，所謂多個模組的意思就是說一個模組可能代表的就是一個應用程式，而每個應用程式則包含一個或數個的子工作，此篇論文採用 GA 去找出這些模組的最佳工作分配。Hou et al. [9] 提出利用 GA 去解決 TAP 問題，此論文最大的特點就是他的工作之間是有先後次序的，一定要先完成某些工作才可以執行下一個工作。此篇論文最大的挑戰就是在進行 GA 時如何保持其合理的解答。Bicking, F., Conrard, B[10] 利用分支界限法(branch and bound)求解多資源生產排程，目標是在最短時間內完成所有工作，需有明確的起始與結束點且所有工作的處理可靠度都一樣。張克誠[11]也是利用 branch and bound(分枝界限法)來做災後搶修設計工作人力指派最佳化之研究，在單一出發點且維修人員的交率都一樣與需在限定時間內完成任務的限制下，去如何達到最小的人力成本的派工。T Zheng[12]則是利用 Ant Colony Algorithm 去計算何時在最短時間內完成所有任務，但需符合只有一個出發點且所有機器人的處理能力都相同的限制。Zhigang Lu[13]也是利用 Ant Colony Algorithm 在電力系統查修派工上，在滿足其時間與的任務沒有優先順序的特性下，找出最小的運輸成本，優點是決定變數的選擇，可以有較周詳的考量，也就是在決定變數值時，比較容易滿足限制式的要求，而不會有超出上下界的情況產生，所以可以產生合理解，缺點是容易形成陷入區域最佳解的缺點，而非整體最佳解。林瑜禎[43]利用分枝界限法進行做自來水管損壞派工的研究，在不考慮路徑問題及人員施工可靠度的情況下，去計算人力分配。利用分枝界限法的優點是可是找出精確解，缺點是由於此類方法屬於窮舉式求解，其求解過程既繁雜又耗時，解題規模易 受到限制，實務應用較不廣泛

2.1.2. 可靠度最大化之 TAP

可靠度最大化 (Reliability-maximization task assignment problem) 是透過最佳化工作分配方式，將工作分配到可以使系統可靠度最高的單位(人員)上去執行或是選擇可靠度最佳的路徑去做運算(工作分派)以達滿整體任務可靠度最高。

Shatz et al.[13] 是以執行時間長短來調整系統的相關可靠度，此篇論文當中的處理器及網路連線可靠度都不是一個固定的數字，而是會透過工作執行時間的長短來計算其可靠度。在此篇論文當中主要是利用啟發式演算法去求取 TAP 問題的近似最佳解。Wang and Shatz [14] 也是採用隨著處理器和通訊網路的執行時間的不同，而有不同的可靠度模式。並且提出使用 A*演算法去求取 TAP 問題的解答。Lin and Chen [15]則是採用尋找 MFST(minimum file spanning tree)並且使用 FREA(fast reliability evaluation algorithm ; FREA)來求取系統可靠度。Chin-Ching Chiu [4] 1.使用一個基於蟻行可靠度導向工作分派方法，在計算在分散式工作指派可靠度問題。所提出的演算法是利用蟻行演算選取一個具有系統最大或趨近最大可靠度之程式及檔案指派之集合，其任務沒有優點順序的問題。R.Seethalakshmi,S.Savichandran [17]在論文當中則假設每個處理器有相同的失敗率以及每個通訊網路線的失敗率也是相同的，再根據此失敗率去計算整個系統的可靠。所有的處理器的處理能力都一樣，不能變動且是單一起點、任務沒有優點順序的問題。



2.1.3. TAP 相關論文比較表

相關論文 比較項目	Hui Cheng, Xuen in chu	Bicki ng, F., Conra rd	T Zhen	Zhiga ng Lu	Chin-Ching Chiu	張克誠	蔣宗哲	林瑜禎
處理相同故障不同人有不同執行效率 (proficiency)	0	X	X	X	X	0	X	X
多個啟始點 (Multi-Origin) 的選擇	X	X	X	X	0	X	X	X
具優先順序的必經點問題 (Designated points Program)	X	X	X	X	X	X	X	X
一次派工分派包含多個(種)故障申告	X	X	X	X	0	X	X	0
具時間限制	X	0	0	X	X	0	0	0

表格 1:TAP 相關論文比較表

2.1.4. TAP 相關論文求解方法簡介

2.1.4.1. 分支界限法

分支界限法是一種列舉(enumeration)的方法，它使用了樹狀結構(tree structure)來表達問題中全部合理解的解空間，一開始的解只有樹根(root)最後探測到最底層的樹葉(leaf)可以發展成完整的解。此種方法也能夠使用在混合整數規劃問題上，其為一種系統化的解法，以一般線性規劃之單形法解得最佳解後，將非整數值之決策變數分割成為最接近的兩個整數，分列條件，加入原問題中，形成兩個子問題(或分枝)分別求解，如此便可求得目標函數值的上限或下限，從其中尋得最佳解。

2.1.4.2. 基因演算法

最早是 1967 年由 Bagley 在論文中提到有關模擬遺傳基因的運算法則，1975 年由 John Holland 發展出基因演算法的理論。其主要精神源於 1859 年達爾文的『物種演化』書中「物競天擇、適者生存」的觀念。在基因演算法中模擬生物的競爭模式，並且生物會根據時間的演化去適應當時的環境，適應力高的生物所留下來的後代會比適應力低的生物留下的後代多。基因演算法通常被應用在搜尋問題上，基因演算法一開始會產生初始人口，通常產生初始人口是透過亂數來產生。接下來會計算每個個體的適合度，根據每個個體的適合度選擇較好的進入母體，再來就是進行交配及突變的運算來產生下一代的人口，如此一代一代的演化下去，直到達到終止條件為止，最後，再將所找過的最好的解答輸出。

2.1.4.3. Ant Colony Algorithm

1992 由 Marco Dorigo 在其博士論文中所提出來 蟻群優化演算法是模仿螞蟻覓食行為的演算法。螞蟻覓食的時候，若找到食物，在搬運食物回程的途中會分泌一種特殊的賀爾蒙，以告訴其他螞蟻可以循著該路徑去搬運食物。由於這些賀爾蒙會隨時間的經過而蒸發，但若有其他螞蟻也循該路徑回來則會繼續分泌賀爾蒙補強之。因此，這些賀爾蒙成為良好的食物指標，讓螞蟻得以同心協力搬回最多的食物。當一問題具有最短路線特性之問題即可利用蟻行演算法進行搜尋最佳方案。

2.2. Minimal File Spanning Tree

2.2.1. MFST 演算法

Kumar [18]於 1985 年，首先將網路可靠度 (Network Reliability)的問題轉移到分散式系統上，並提和 Ragharendra 提出 MFST 演算法。當找出圖形中的所有最小檔案擴展樹後，再將此所有最小檔案擴展樹利用 Terminal Reliability Evaluation Algorithm 去計算出分散式程式可靠度。此方法非常的直覺簡單，但當其找出圖形中的所有最小檔案擴展樹後，發現有相當多的重複樹，因此還須移除這些重複樹，故此步驟須執行二回。

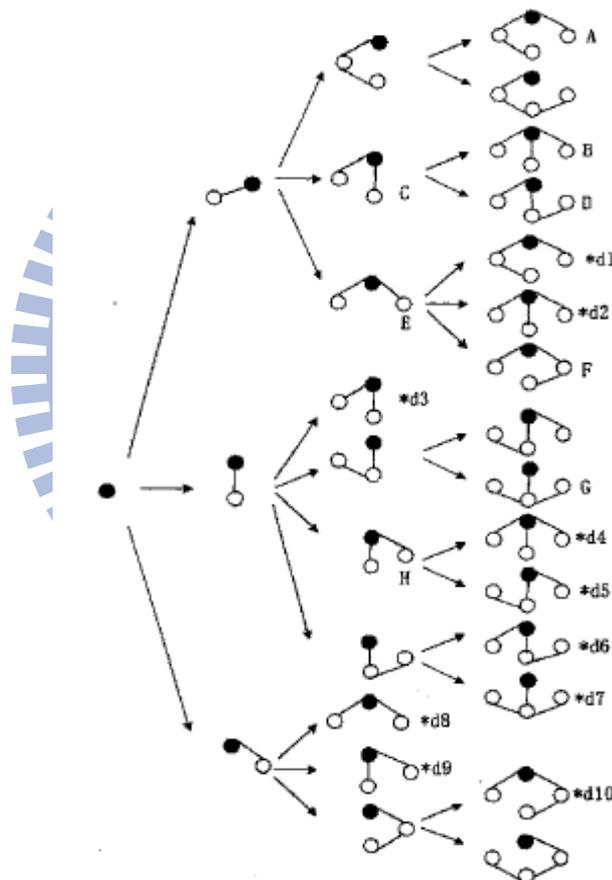


圖 3: Generation of replicated trees in MFST algorithm

1.Kumar 的演算法產生 MFST。首先找到尺寸(邊的數目)為 0 的樹，再找到尺寸為 1 的樹，直到找到尺寸為 $n-1$ 的樹。如圖 3。

2。演算法產生了相同的樹，如 A 和 d1, B, d2 和 d4, C 和 d3，因此需要額外調用過程 "CLEAN" 去掉重複的樹。

2.2.2. FREA (Fast Reliability Evaluation Algorithm)演算法

陳登吉教授[21]於 1994 年提出了 FREA 演算法此演算法結合上述的圖形化簡法則與一般化的 factoring theorem:此演算法先找出網路的所有最小檔案擴展樹 (Minimal File Spanning Tree)，再去計算出分散式程式可靠度 (Distributed Program Reliability)。

演算法簡要說明如下:

符號定義

X_i	Node representing a processing element i .
X_{ij}	Link between processing elements i and j .
F_i	Data file I .
P_i	Distributed program i .
PA_i	Set of programs that can be run at processing element x_i
FA_i	Set of data files available at processing element x_i
FN_i	Set of data files needed to execute P_i
PN	Set of programs to be executed.
FN	Set of data files needed to execute all programs in PN
FST	file spanning tree consisting of the root node (processing element that runs the program) and some other nodes which hold all the files needed for the program held in the root node under consideration
$MFST$	minimal FST containing no subset file spanning tree

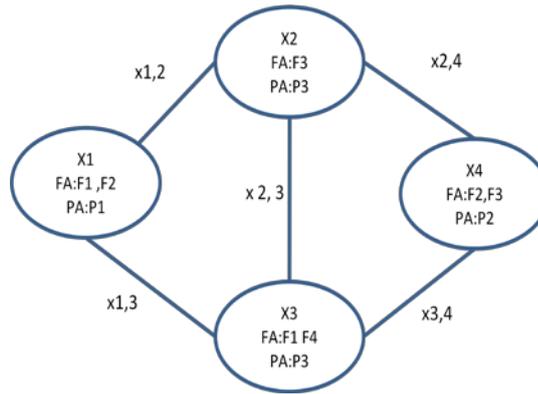


圖 4: Simple distributed computing system

$FA_3 = \{F1, F4\}$ 表示在節點 3 上有檔案 1 和檔案 4 可供訪問

$PA_2 = \{P2, P3\}$ 表示程序 P2, P3 可在節點 2 上運行

$FN_3 = \{F1, F2, F3, F4\}$ 表示執行程式 P3 需要訪問資料檔案 F1, F2, F3 和 F4; 缺一不可

FREA (Fast Reliability Evaluation Algorithm) 演算法說明:

此方法是利用 Factoring Theorem 及 Reliability Preserving Reduction 來化簡分散式資料庫系統的圖形，藉由化簡後的圖形，可快速計算出可靠度。圖形化簡包含 Degree-1 Reduction 與 Degree-2 Reduction，說明如下：

- Degree-1 Reduction: A node is referred to as a degree-1 node if it has only one incident edge. Degree-1 reduction removes degree-1 nodes that contain no needed data files and programs under consideration and their incident edges
- Degree-2 Reduction: Suppose node v is a reducible node, then one can apply series reduction on node v and move data files and programs within node v to a node u or w .

2.2.3. 圖形化簡實例說明:

Degree-1 :對於 P1, 節點 x6 不含執行 P1 所需的 F1,F2,F3 與 F4,且節點的 degree 為 1，因此，進行 Degree-1，可以直接刪除 X6 和它相連的邊 X36

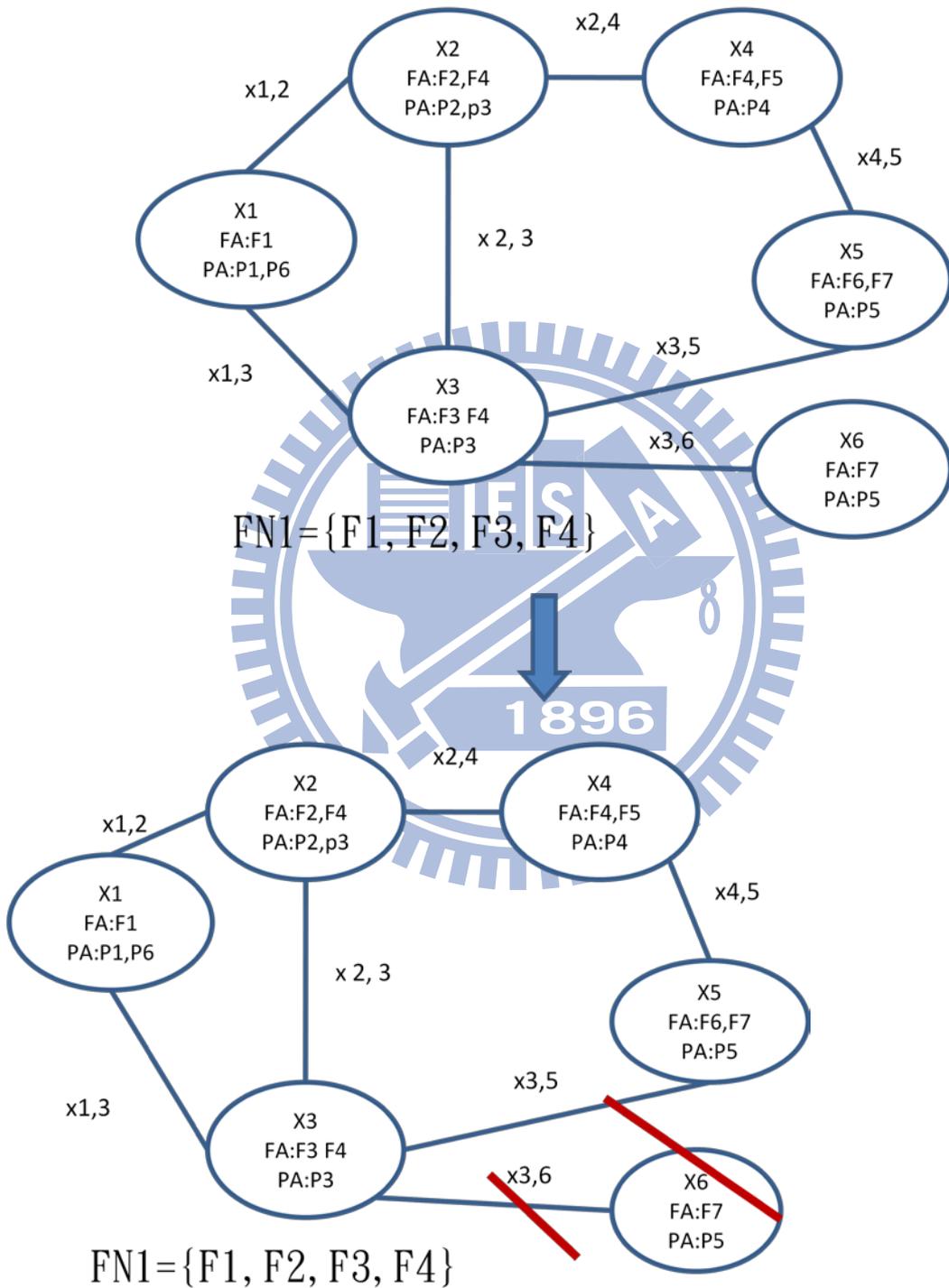


圖 5: Example of degree-1 reduction

Degree-2 Reduction: X5 在節點上不含有它執行時所需的資料檔案 F1, F2, F3 和 F4，且結點 X5，的度數為 2，其中 X3, X4，它們之間沒有邊連接其容量: $C_{34} = \min\{C_3, C_4\}$ 可靠度: 若每條連結的可靠度都為 95%，即 錯誤為 5%。X3, X4 的錯誤率為: $\lambda = \lambda_3 + \lambda_4 - \lambda_3 * \lambda_4 = 5\% + 5\% - 5\% * 5\% = 9.75\%$ X3, X4 的可靠度為 90.25

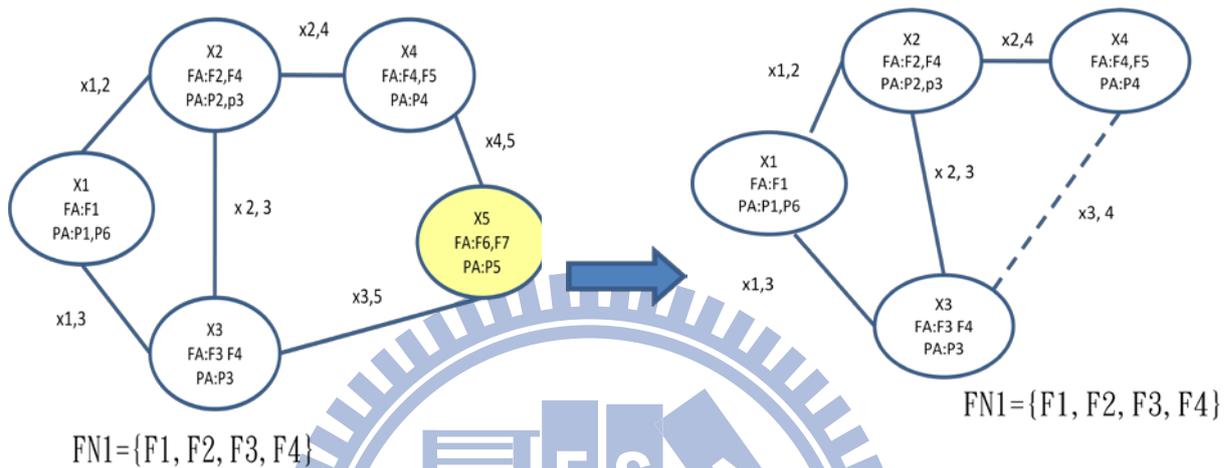


圖 6: Example of degree-2 reduction

2.2.4. Minimal File Spanning Tree 實例說明

1. 一分散式環境網路圖如下，現執行 $P1=\{F1,F2,F3\}$ 即檔通需從 P1 出發，所需的資源有 F1,F2,F3。

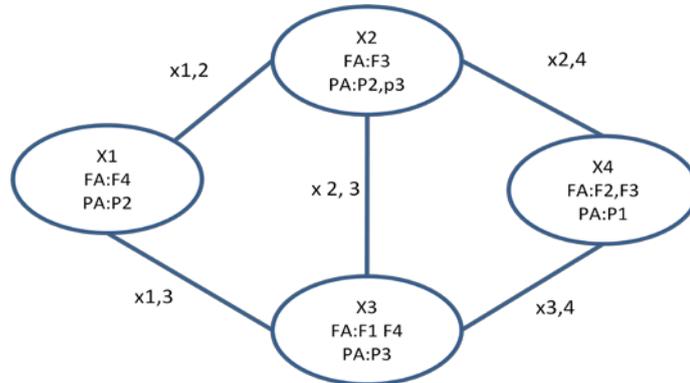


圖 7: Simple distributed computing system

2. 執行 Degree Reduction: 因為節點 X1 上不含本次執行所需的程序(P)與資源 (F)，所以將其 Reduction 掉。

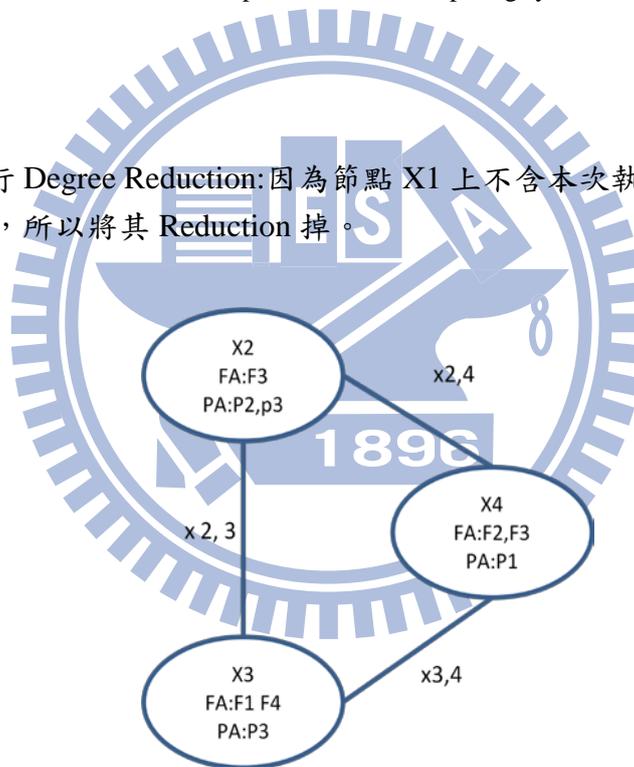


圖 8: Example of degree reduction

3. 找出所有 MFST:對程式 p1 而言，有二個最小檔案生成樹 (minimal files spanning trees, MFSTs) 即 $x_3, x_4, x_{3,4}$ 與 $x_2, x_3, x_4, x_{23}, x_{24}$ 。

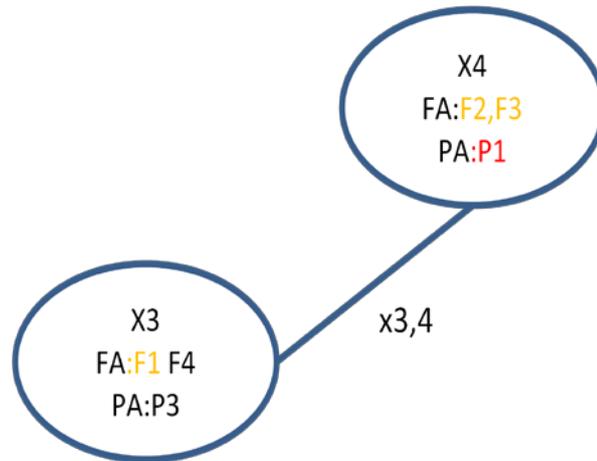


圖 9: Example of DCS and all MFST's for P1 under consideration

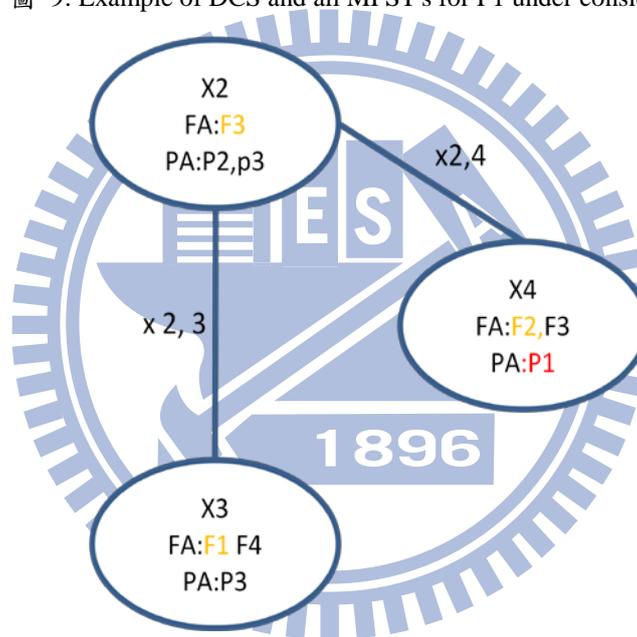


圖 10: Example of DCS and all MFST's for P 1 under consideration

三、模式構建及求解方法

3.1. 問題描述與限制

3.1.1. 派工的初始條件:

本研究以電信系統故障派工為例

1. 已知所有不同種類及數量的叫修任務
2. 不同的路徑距離及路況條件
3. 分散的維修資源
4. 分散的人力

3.1.2. 派工的條件限制:

1. 處理相同故障不同人有不同的維修效率(proficiency)[1]:每個機房的人員有不同的技能，處理的障礙也不同，且每個工地點的人員對每一項故障有不同的處理執行效率。
2. 多個啟始點(Multi-Origin)的選擇沒有確的出發點，可以從甲機房出發去做修復工作，也可以從乙機房出發，只要可以完成派工所需要的全部工作即可。
3. 具優先順序的必經點問題(Designated points Program) [2]:維修人員接到派工任務後，需先判斷是否有足夠的維修材料，若無則需先到料庫拿取派工任務所需的維修材料，派工路線就需先到料庫，也不能是先到叫修地點才到回到料庫。例如要如要修復油封工廠的太陽光電系統需先去料庫拿材料，才到油封工廠。
4. 一次派工分派包含多個(種)故障申告[3][4]: 同一批派工有多個故障點與多個種類的故障類型(一個 task 有很多的 job)，例如一項派工包含:修理醫院的 PBX、電力監控系統與修理橡膠工廠的太陽光電系統。而不是像一般的工作分派都是單一種類的工作分派，如修理交通號等等。

3.2. 參數及目標函數定義

3.2.1. 參數定義說明

Input: the original task assignment system graph $G=(V,L,S,R,Ta,\mu)$

V :派工圖上的節點集合.

L :節點間的行車距離

S :行車路況

R :執行工作所需的資源集合

Ta :可執行工作的集合

μ :執行工作的執行效率

3.2.2. 目標函數

目標函數:在符合電信系統派工問題限制下，選出一條最佳的派工路徑: TAP_p

object: min $TAP_p \cdot P = \sum_{(i,j) \in V} \left(\frac{L_{ij}}{S_{ij}} + \frac{T_j}{\mu_j} \right)$

■ $Path = \{(i,j) | (i,j) \in V | i \neq j\}$

■ μ : Employee_i 對 Task_j 的執行效率(Proficiency of the employees on the Task)

Task = $\{J_k | K \geq 1\}$

μ : 0~1

3.3. 演算方法說明

- 利用 MFST 演算法找出所有符合條件限制的派工路徑。
- 利用不同維修點有不同的維修效率，來計算一項工作分派的等待時間，找出一個延遲時間最小與維修效率最高的派工路徑。

說明如下：

符號定義：

D = An undirected TAP graph with vertex (node) set V , edge set E , and data file set F .

Without loss of generality, we identify a program with a special type of data file, i.e.

$\text{program} \in F$.

SS : A subset of E that represents the edges merged during the process of finding all MFSTs

$D - e$: The graph D with edge e deleted

$D + e$: The graph D with edge $e = (u, v)$ contracted so that node u and v are merged into a single node. This new merged node contains all data files and programs that were in nodes u and v .

H : Subset of files of F , i.e. $H \in F$, where H contains the programs to be executed and all data files needed for the execution of these programs.

FA_i : Set of data files available at processing element x_i

PA_i : Set of programs that can be run at processing element x_i

Input: the original TAP graph $G = (V, L, S, R, T, \alpha, \mu)$

Output: a MFST and a sub-graph

Step 1. Initialize all data structures

Step 2. 做 degree Reduction

Repeat // reduce the original TAP graph D use FREA

perform the degree-1 reductions

perform the degree-2 reduction

until no reductions can be made

Let D' be the TAP graph after the reduction step

Step 3 找出所有的 FST (File Spanning Tree) // Find FSTs

FOUND $\leftarrow 0$

$SS \leftarrow 0$

找出所有有程序(PN)的節點

判斷資源是否在必經點上

是， PN_i 節點與必經點先做 merge

FIND FST(D' ; SS)

// call FIND FST to find FSTs //

```

for all s; t ∈ FOUND do
  if t ∩ s D s then remove t from FOUND
  else if t ∩ s D t then remove s from FOUND  endif
  endif //對所有之前的找到的 MFST 來說，如果這次找到的 FST 是他
        們的子集代表這次的 FST 是一個最佳的解，故把之前的 tree 從解集中移除

FIND FST(D',SS)
begin
if there are no FSTs in D then return endif // failure case
if there exists one node n such that FAn ⊃ H then FOUND ← FOUND ∪ SS
return // success case //

endif
for all Li ∈ the set of edges incident on the nodes containing the programs to be executed do
  FIND FST(D+Li,SS ∪ { Li })
  D ← D - Li
remove the irrelevant components from D
if there are no FSTs in D then return endif
end FIND FST //程式會遞迴呼叫直到下列條件被滿足(是否經搜集到所有需要的檔案))
Step 5 計算每一 MFST 的權重 // Calculation all MFST each link weight
Step 6 選出權重最小的 MFST // Select the weight of the smallest MFST

```

3.4. Time Complexity 1896

本演算法是以應用 kumar[18][19]和 Chen[21]的演算法，他們演算法的 Time Complexity 都是 $O(2^m)$ in the worst case, where m denotes the number of links in the graph.所以本演算法的 worst case 的 Time Complexity 也是 $O(2^m)$ 。

四、實例展示

本次展次以電信業者之彰化營運處為例，該營運處有 3 個派修機房有維修人員進駐運，進行各項派工任務的；有 2 個料庫存放各種不同維修所需的材料，維修客戶為該營運處大型企業客戶。

4.1. 模型建置流程圖

模型建置流程步驟如下：

- Step 1：將應用對象抽象化，轉換成網路架圖
- Step 2：定義圖形節點
- Step 3：定義節點間的路線
- Step 4: 定義圖形節點中的資源
- Step 5：定義執行工作
- Step 6：定義不同地點的對工作的執而效率

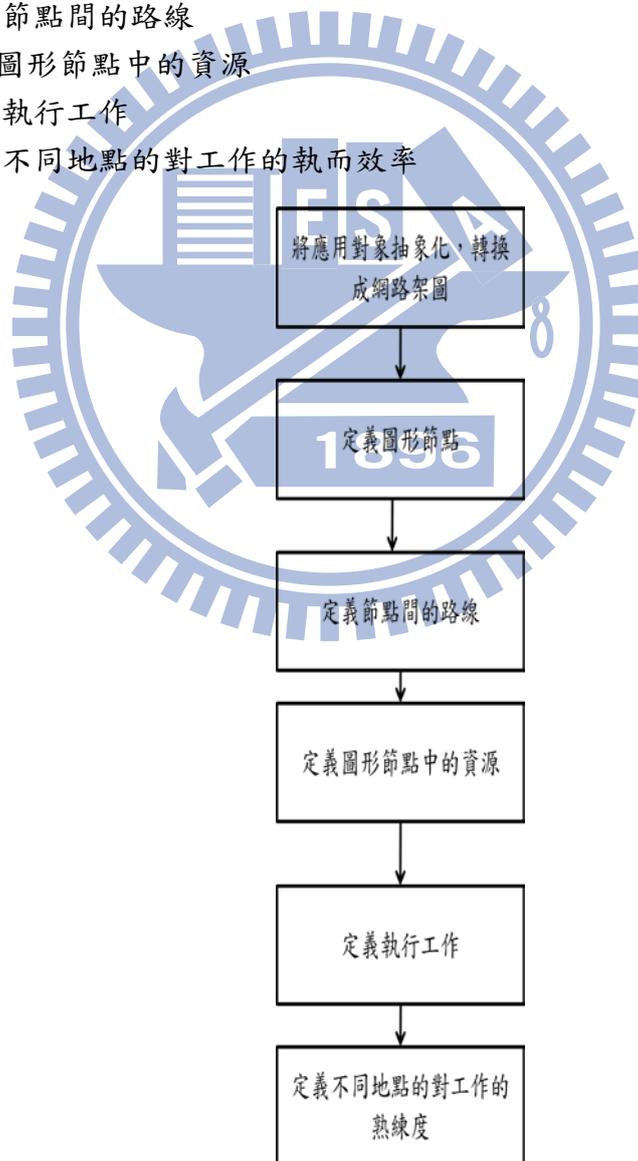


圖 11: 模型建置流程圖

詳細說明如下:

4.1.1. Step 1: 將應用對象抽象化，轉換成網路架圖

選一電信業者之營運處，在地圖上將實際機房、料庫及客戶所在地點抽象化，轉換成網路圖，如圖 12。



圖 12: 實際應用例子圖 1



圖 13: 實際應用例子圖 2

4.1.2. Step 2：定義圖形節點
 定義每個節點的實際地點名稱

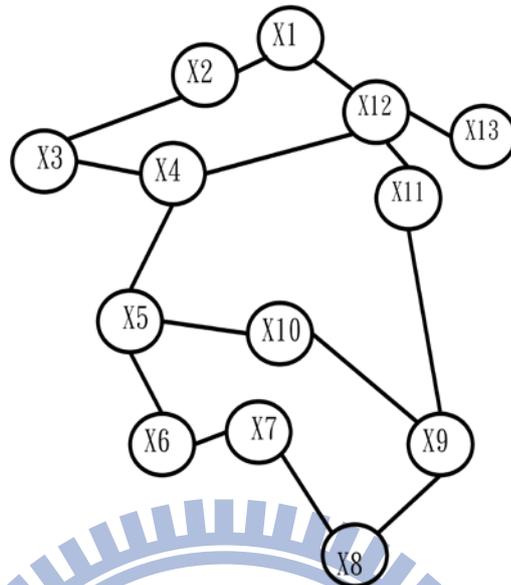


圖 14:抽象化後之應用例子圖

節點	目標名稱	節點	目標名稱
X1	機械公司	X8	員林機房
X2	車燈工廠	X9	料庫 2
X3	鹿港機房	X10	油封工廠
X4	料庫 1	X11	醫院
X5	製鞋工廠	X12	彰化機房
X6	泡麵工廠	X13	補習班
x7	橡膠工廠		

表格 2:定義節點名稱

4.1.3. Step 3：定義節點間的路線

每個節點間的路徑為 2 點間的行車距離預估，如員林機房到橡膠工廠的最短行車路徑即表示為 x7,8，行車距離為 30KM。

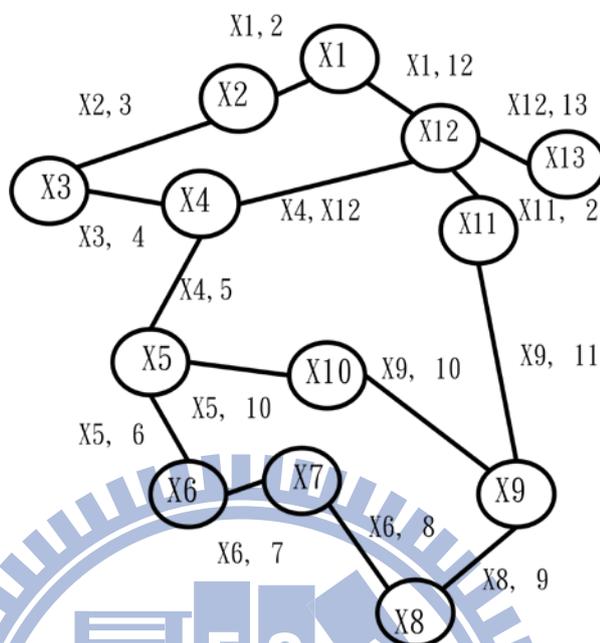


圖 15:定義路徑權重之例子圖

路線	路離(KM)	路線	路離(KM)
X1,2	20	X7,8	30
X1,12	30	X5,10	30
X2,3	30	X9,10	20
X3,4	20	X8,9	20
X4,12	40	X9,11	40
X4,5	30	X11,12	10
X5,6	30	X11,13	10
X,6,7	20		

表格 3:路徑權重表

4.1.4. Step 4: 定義圖形節點中的資源(1/2)

利用原 MFST 演算法所定義的符號，轉換為本研究所適合的參數。

符號	定義
X_i	Node representing a processing element i .
X_{ij}	Link between processing elements i and j .
F_i	Data file I .
P_i	Distributed program i .
PA_i	Set of programs that can be run at processing element x_i
FA_i	Set of data files available at processing element x_i
FN_i	Set of data files needed to execute P_i
PN	Set of programs to be executed.
FN	Set of data files needed to execute all programs in PN

表格 4:原 MFST 符號定義表

符號	定義
X_i	機房、故障派工地點
X_{ij}	2 地點之間的路徑
R_i	執行工作所需的資源 → 網路線材(電話總機、4 芯電話線、光纖..)、Its 車機、太陽光電統變流器、攝影機、DVR....
Ta_i	執行電信修復工作 (修復電話總機系統、IP-PBX 系統、FOT 修復....)
TaA_i	需執行的工作的集合
RA_i	執行工作可用程式集
RN_i	執行工作所需的資源集合
TaN	執行工作的集合
RN	資源的集合

表格 5:電信系統故障派工符號表

Step 4: 定義圖形節點中的資源(2/2)

- 如節點 X4 中的資源有 R1,R3,R4,R5,R9，即表示在料庫 1(節點 x4)擁有 R3(電力監控設備)、R4(太陽光電變流器)、R5(L2 switch)與 R9(熱泵系統)等資源。
- 節點 X3 中有 TaA:Ta2,Ta3 表示在鹿港機房(X3)，可以從這個機房出去做維修工作的種類有 Ta2 與 Ta3，若派工的需求是 Ta1 的話，就不能從這個機房派出維修人員。

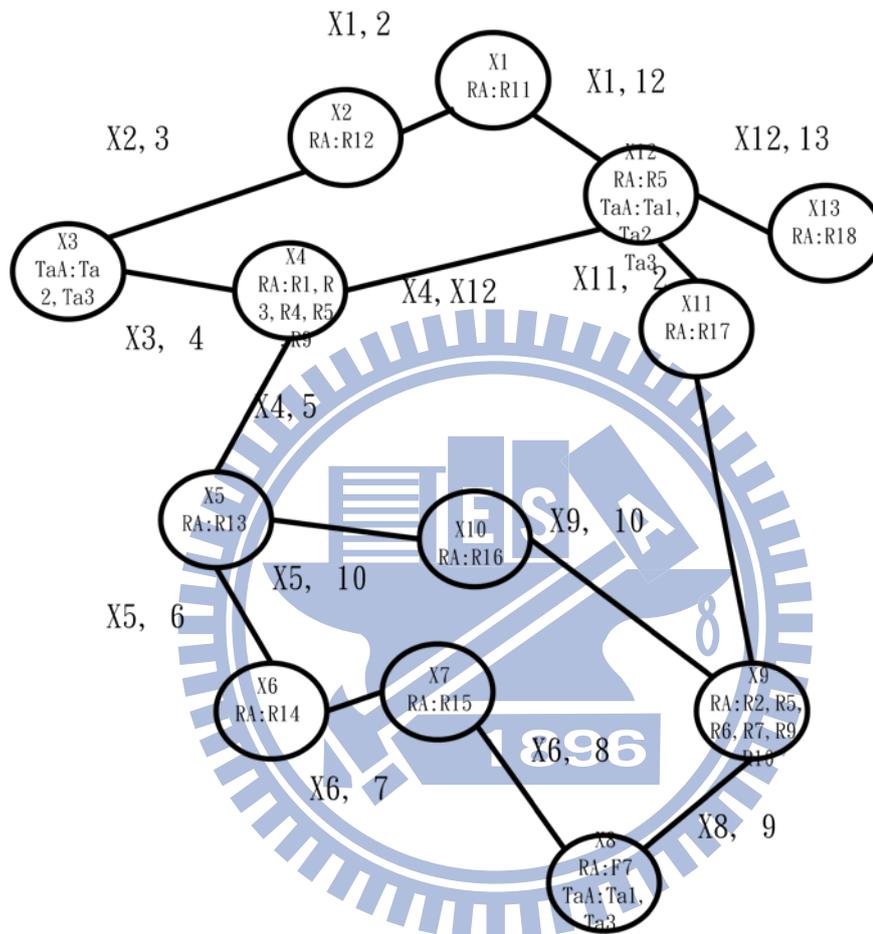


圖 16:加入節能資源之例子圖

- 可隨著任務需求的增減，更改資源數量、名種。

資源代號	資源項目
R1	攝影機
R2	FOT(Fiber Optics Transmitter)
R3	電力監控設備
R4	太陽光電變流器
R5	I2 switch
R6	DVR(digital video recorder)
R7	電話總機(PBX)
R8	IP-PBX
R9	熱泵系統
R10	LED 路燈
R11	機械公司維修告警
R12	車燈工廠維修告警
R13	製鞋工廠維修告警
R14	泡麵工廠維修告警
R15	橡膠工廠維修告警
R16	油封工廠維修告警
R17	醫院工廠維修告警
R18	補習班維修告警

表格 6:資源定義表

4.1.5. Step 5：定義執行工作

Ta1=修復設影機，Ta1={F1,F11} 機械工廠的攝影機損壞維修

Ta=電力設備的維修，Ta2={F3,F9,F16} 油封工廠的電力監控設備及熱泵系統損壞

Ta=電信系統維修,Ta3={F8,F11,F12}機械公司與車燈工廠的 IP-PBX 損壞

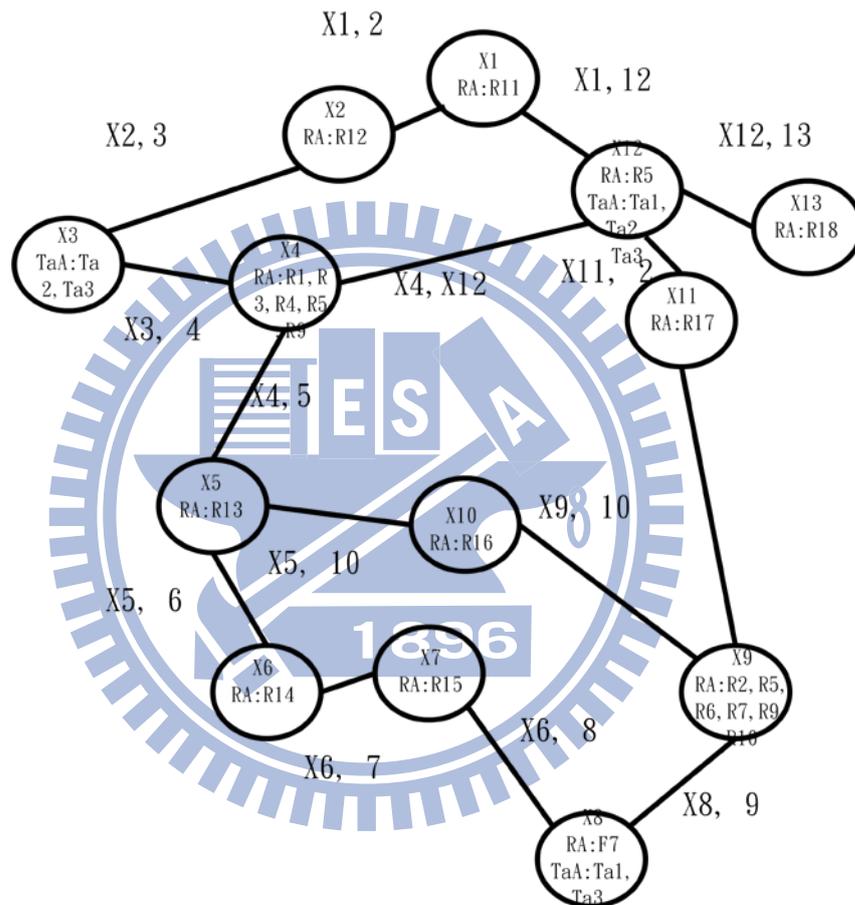


圖 17:加入節點資源之例子圖

4.1.6. Step 6：定義不同地點的對工作的執行效率

Ta2 在節點 X3 與 X12 都可出發去做維修，從節點 X3 出發維修的維修效率是 0.78 ，從節點 X12 出發的維修效率是 0.85 。

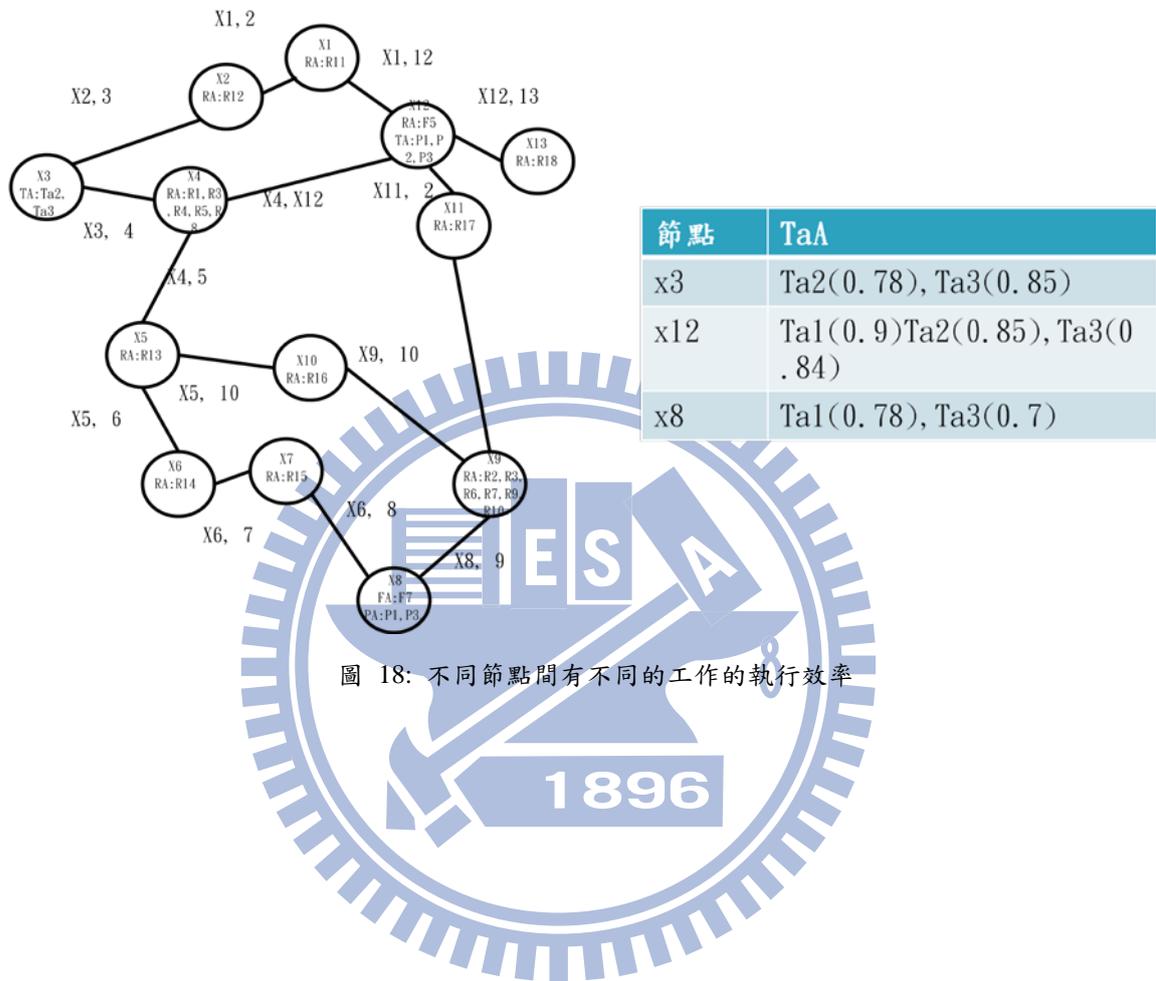


圖 18: 不同節點間有不同的工作的執行效率

4.2. 實際案例執行說明

- 4.2.1. 現有一故障發生，橡膠工廠的電力監控設備及太陽光電變流器損壞，透過資源及可執行工作表可得知，派工任務 $Ta2=\{R3,R4,R15\}$ 。即派工所需的工作技能 $Ta2$ ，所需的資源為 $R1,R4,R5$ 缺一點，找出一條符合上述要求的最佳派工路徑。
- 從圖中可發現，可執行工作 $Ta2$ 的節點有 $X3$ 與 $X12$ ，透過本研究所設計的研算法計算由那個地點出發去做修復較佳。

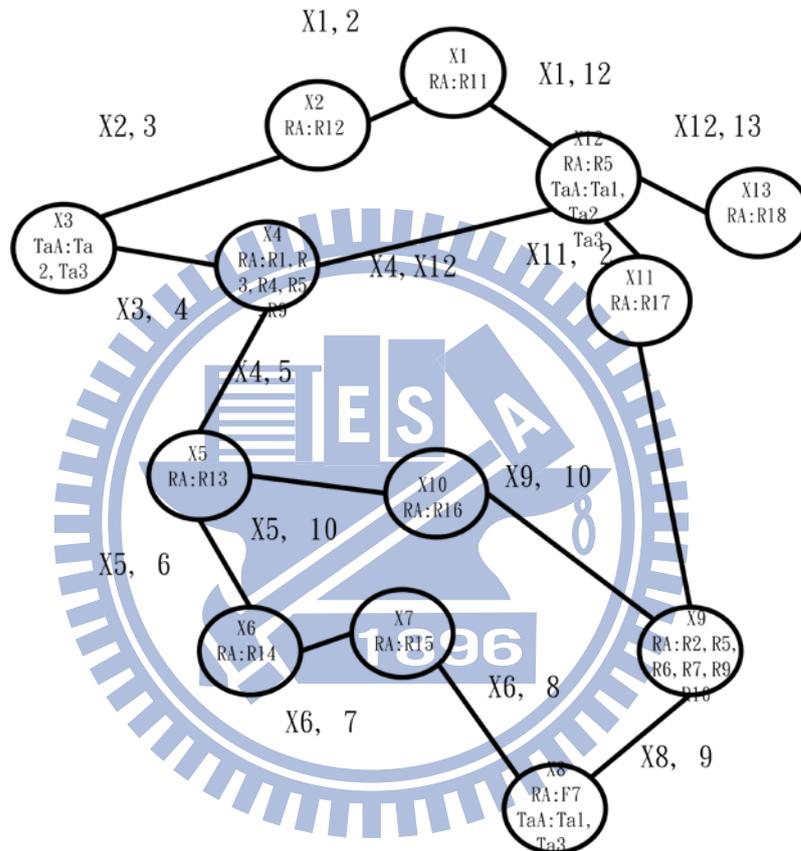


圖 19:應用例子圖

4.2.2. 利用 MFST 演算法找出符合有資源限制的 MFST

首先利用 MFST 演算法找出所有符合本次派工資源限的後有 MFST (Minimal File Spanning , Tree)

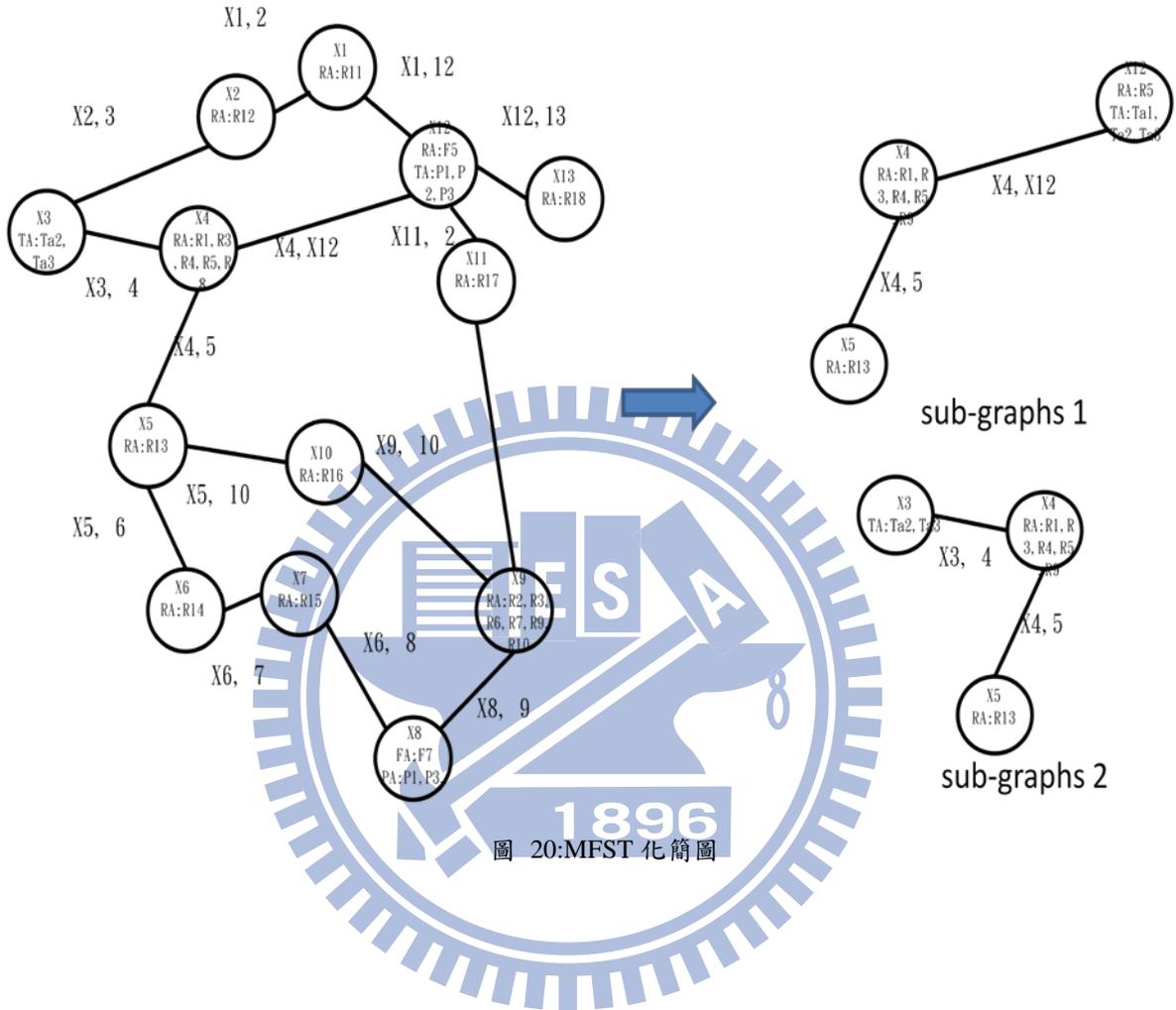


圖 20:MFST 化簡圖

4.2.3. 加入即時路況更新: 可即時更新實際路徑用路情況
 依實際路況可做即時路況更新，不同的時間點會有不同的路況

路線	路離 (KM)	路況 (KM/HR)	路徑 權重(T)	路線	路離 (KM)	路況 (KM/HR)	路徑 權重(T)
X1,2	20	30	0.67	X7,8	30	40	0.75
X1,12	30	40	0.75	X5,10	30	55	0.55
X2,3	30	30	1.00	X9,10	20	48	0.42
X3,4	20	25	0.80	X8,9	20	52	0.38
X4 ,12	40	41	0.98	X9 ,11	40	32	1.25
X4,5	30	55	0.55	X11,12	10	28	0.36
X5,6	30	35	0.86	X11,13	10	32	0.31
X6,7	20	43	0.47				

表格 7:即時路徑更新表

4.2.4. 計算每一 MFST(\sum 路徑權重/執行效率)的值，選出最小的一條路徑。

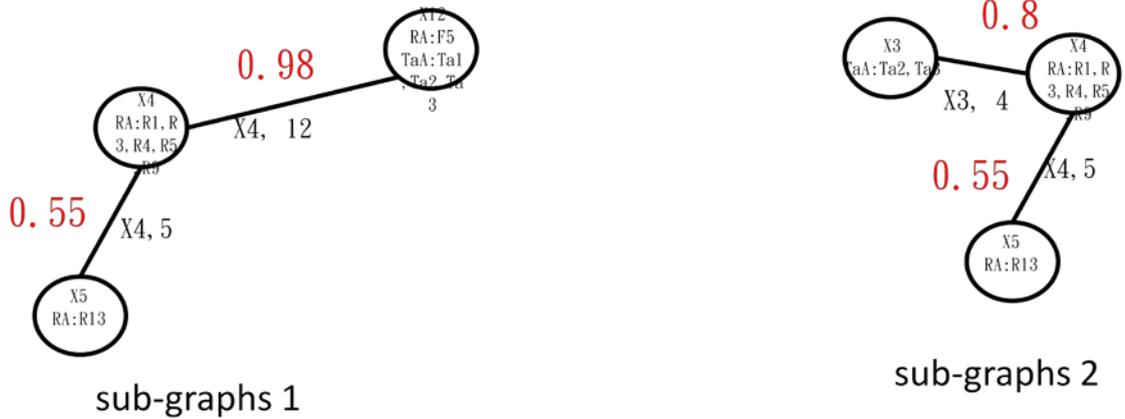


圖 21:MFST 化簡後之 Sub-graphs

- Sub-graphs1 $T=0.55+0.98=1.53$
- Sub-graphs2 $T=0.55+0.8=1.35$

4.2.5. 找出對 Task_j 執行效率 (μ) 最高的人員 Employee_i

- sub-graphs1 對 ta2 的執行效率 \rightarrow 節點(x12) Ta2(0.85)
- sub-graphs2 對 ta2 的執行效率 \rightarrow 節點(x3) Ta2(0.78)

4.2.6. 選出一條最佳的派工路徑 TAP_p, $P = \sum_{(i,j) \in V} (\frac{L_{ij}}{S_{ij}} + \frac{T_j}{\mu_j})$

- sub-graphs1: TAP_p=1.53+1.76=3.29
- sub-graphs2 TAP_p=1.35+1.28=2.63
- sub-graphs2 < sub-graphs1 故本次派工建議路線為(x3,4)(x4,5)

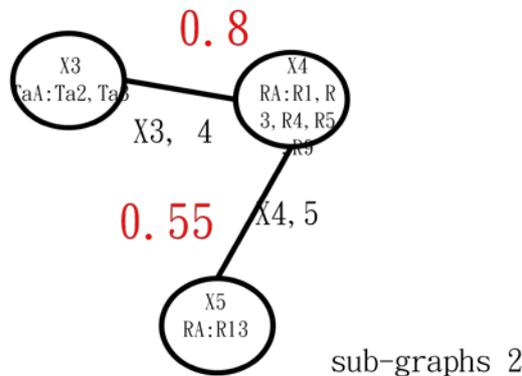


圖 22:建議派工的路線圖

4.3. 程式展示

4.3.1.1. 初始畫面

本程式的執行畫面分成 2 區，一區是資料輸入區，資料輸入區可定義每一節點的資源，可執行的工作及工作的執行效率等等。另一區為圖形化結果輸出區，將每一次的建議派工路徑以圖形化結果表示。

資料輸入區可定義有：

- 節點名稱、節點所擁有資源(RA)、可執行的工作(TA)及 TA 的執行效率，與必經點。

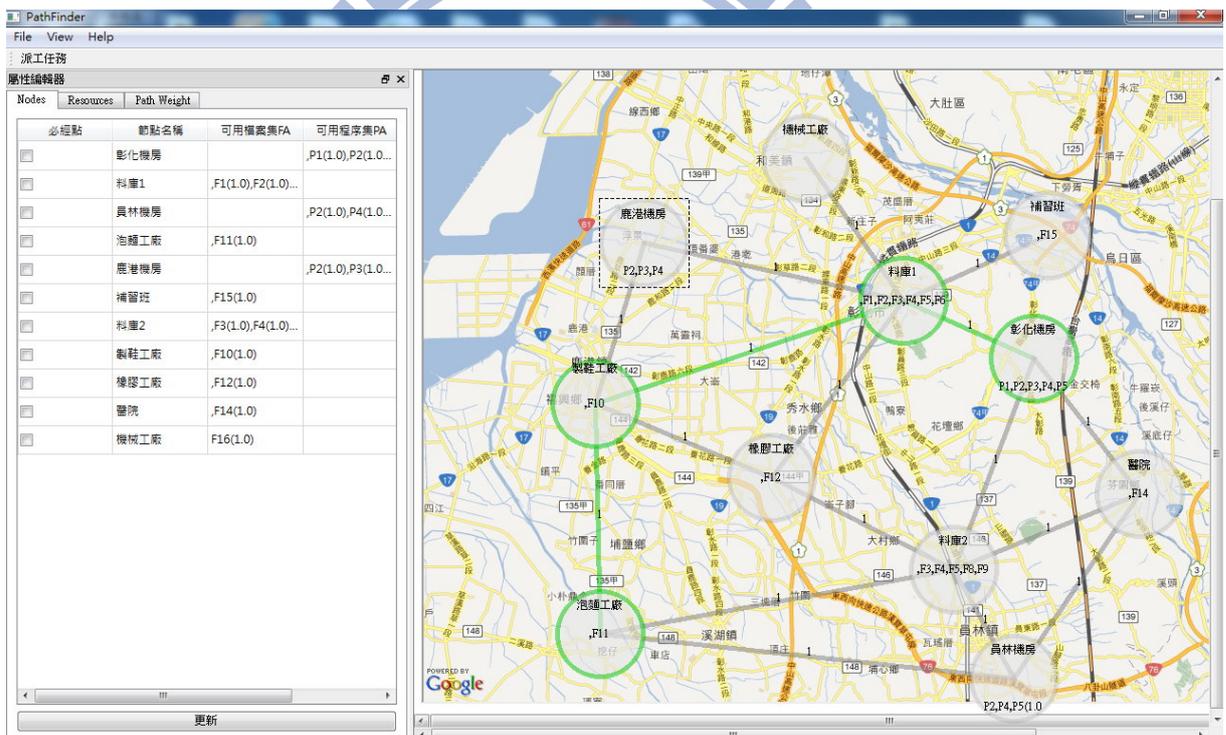


圖 23:初始畫面

4.3.1.2. 定義各項資源

(1) 可在執行畫面區自行托拉想設定的區域

透過 Google API 可自行設定要建立的派工地區，可設定為台中地區亦可透過托拉、放大與縮小的方式找到想設建立派工地圖的地圖。

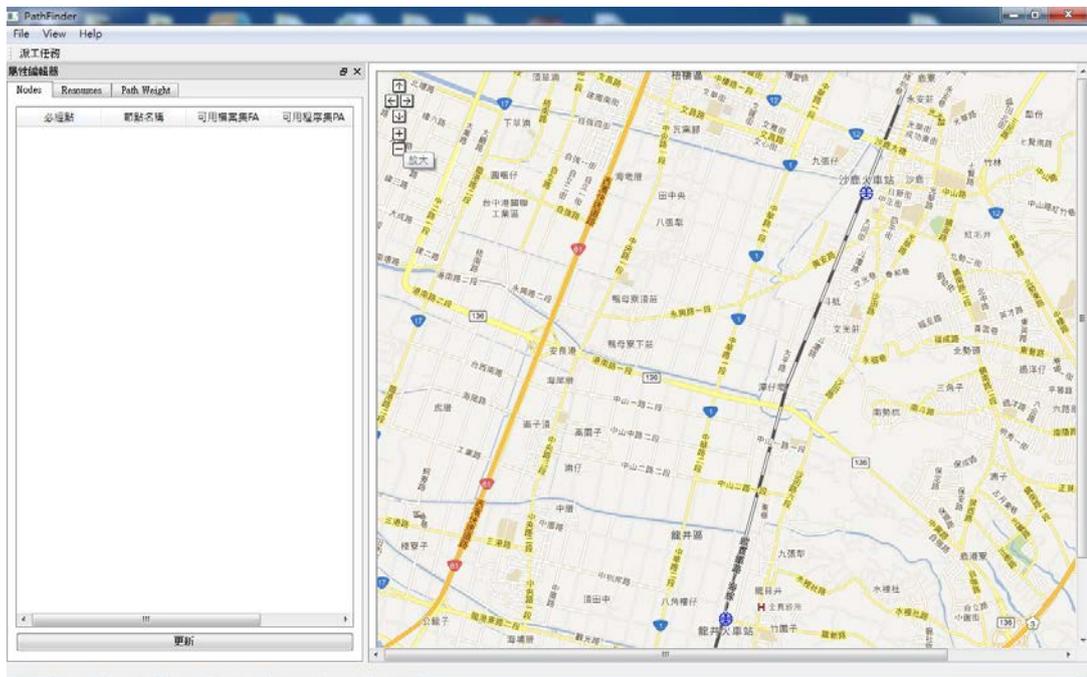


圖 24:自行設地圖-台中地區

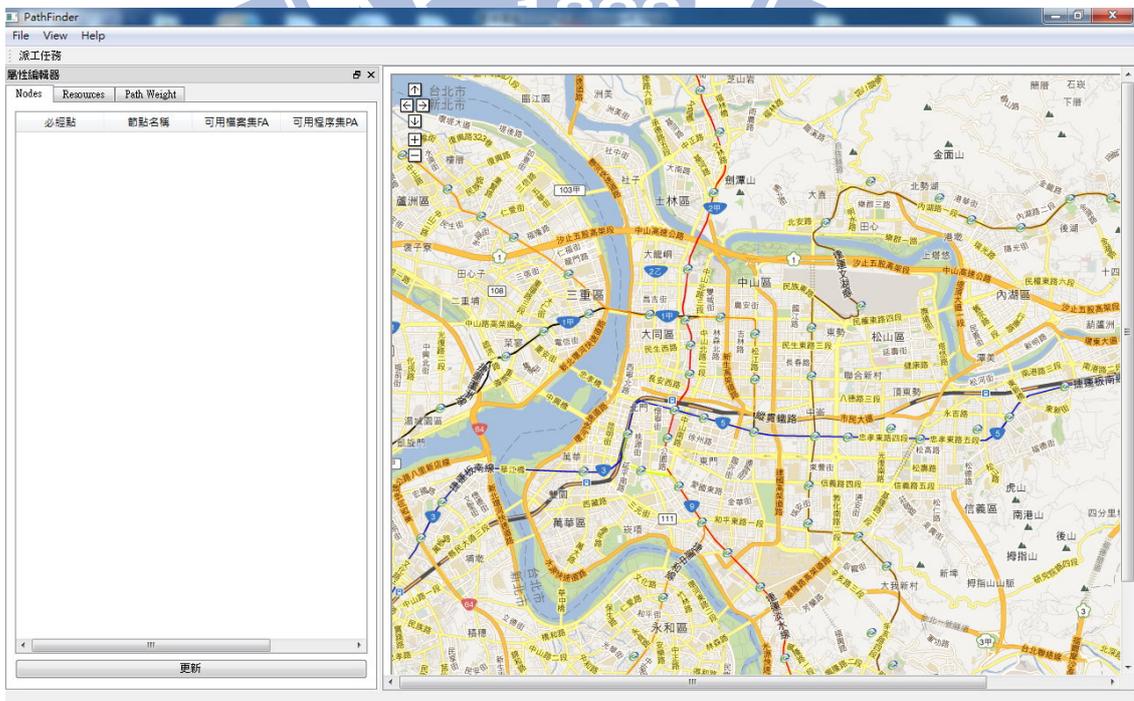


圖 25 自行設地圖-台北地區

(2) 建立派工機房及維修地點

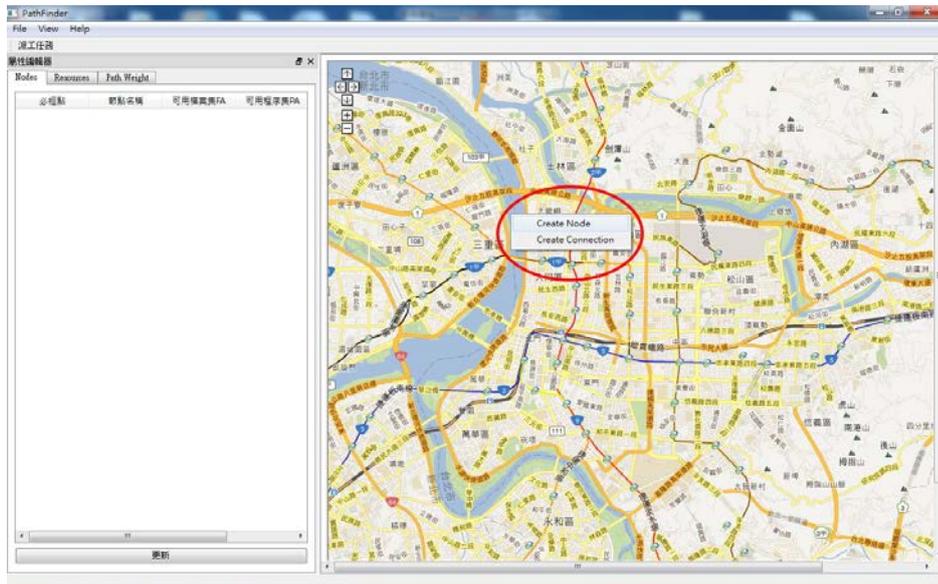


圖 26 建立派工地點

可自行輸入多個派工機房及維修地點，及設定路徑連結與否

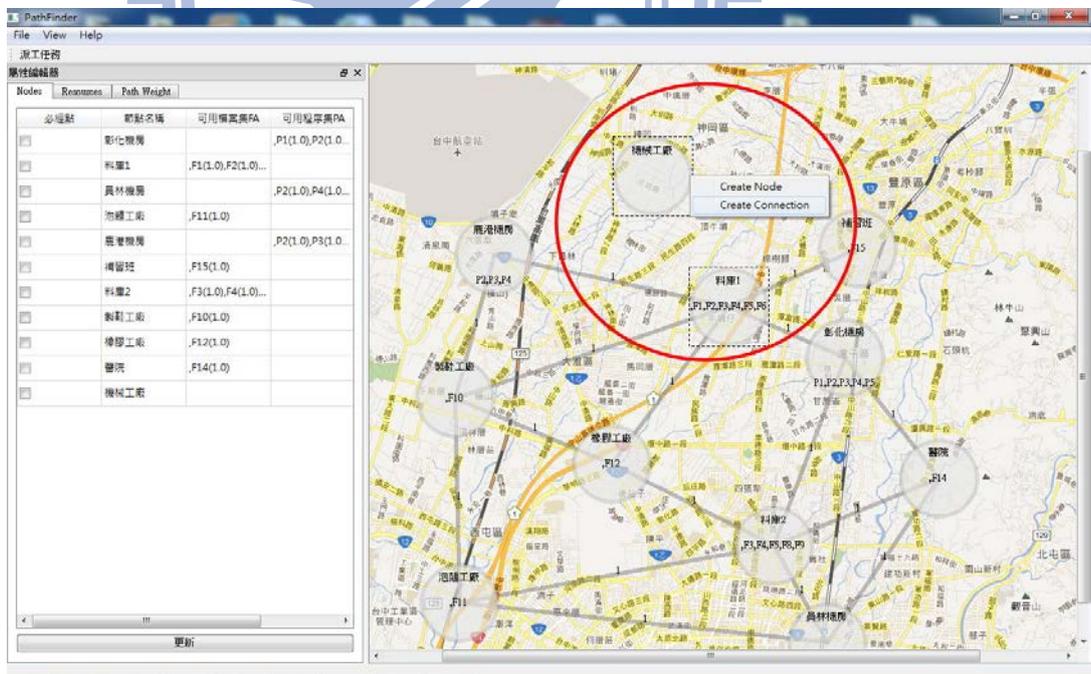
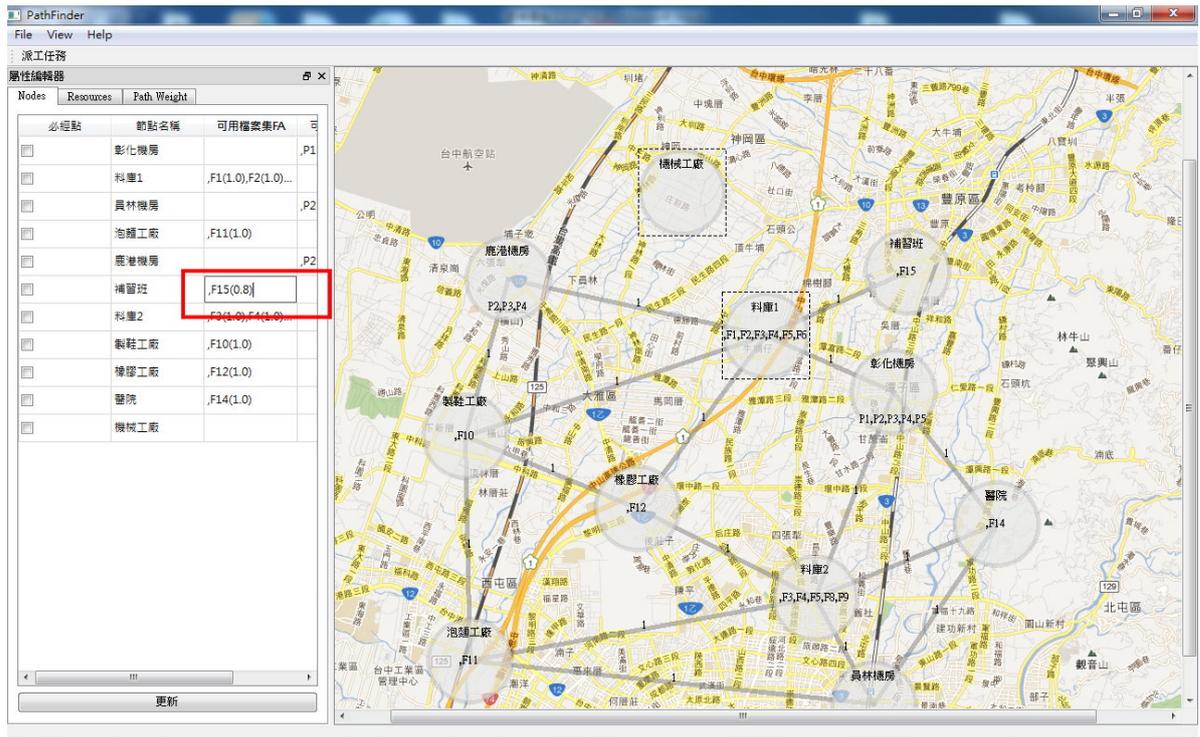


圖 27 建立 2 點連線圖

可修改每個障礙地點的故障的權重(0~1)



據實際派工的執行，可新增、刪除與修改各項資源名稱

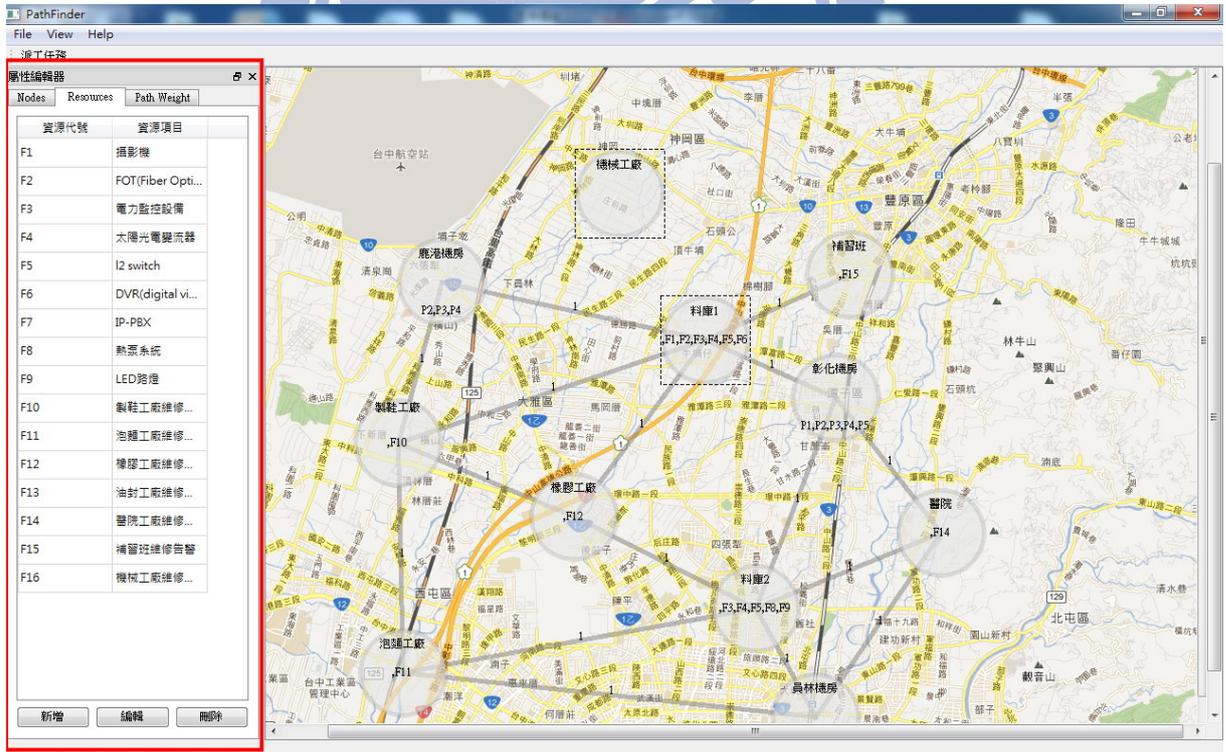


圖 28 資源修改圖

4.3.1.3. 即時調整路徑權重

隨著維修人員回報路況，可即時調整路況權重。

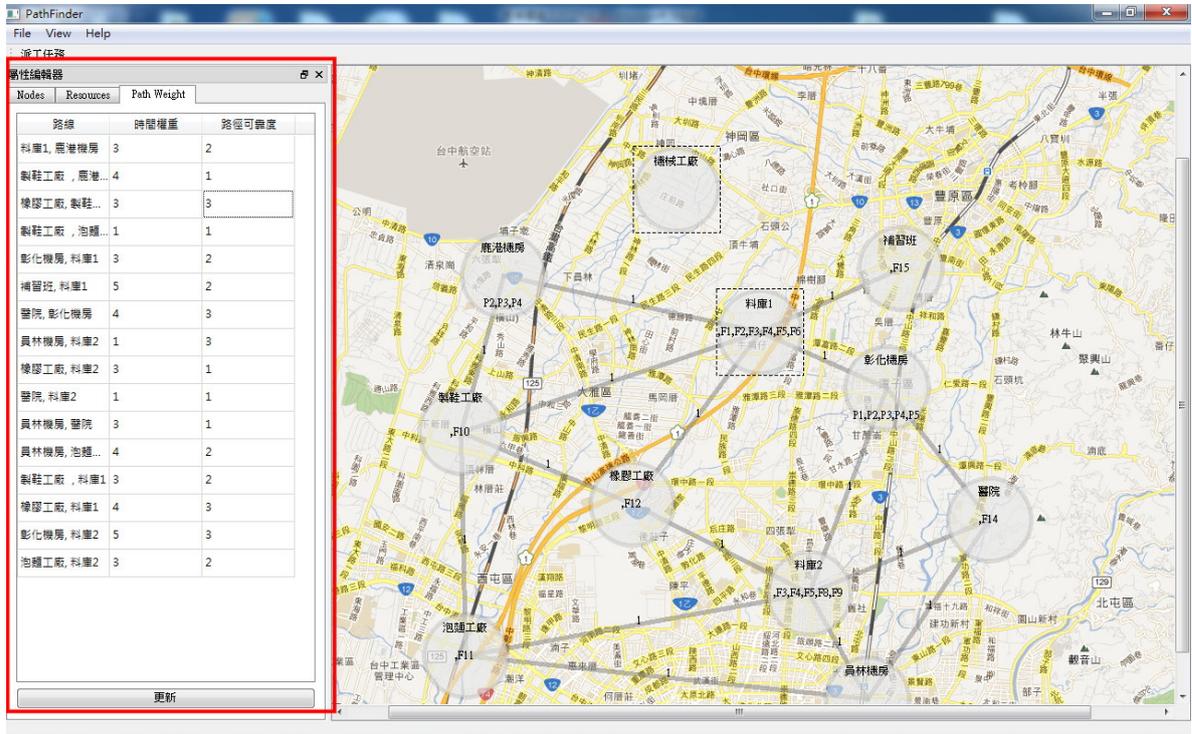


圖 29:更新路徑權重圖

4.3.1.4. 執行派工任務

- (1) 據實際派工需求輸入派工任務所需資源
- (2) 可執行 1 個種類或多個種類的派工任務:如 P1 為電信系統修復派工、P2 為電力系統修復派工。派工任務可為單純的修復電信系統 P1，在輸入損壞的地點與該項派工的權重，如下圖：

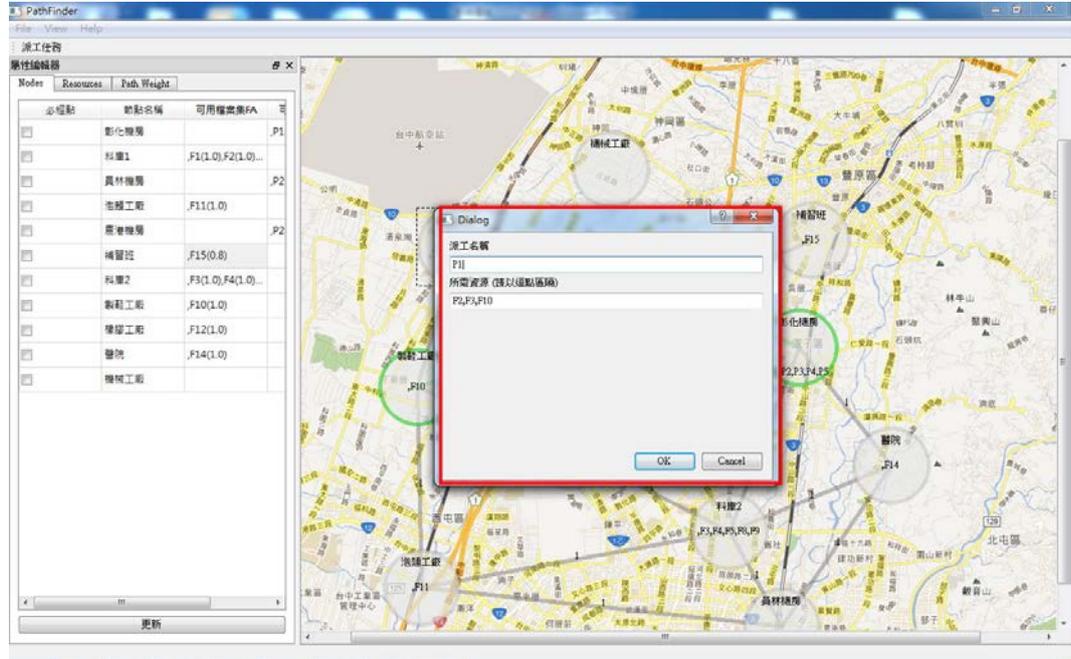


圖 30:執行派工任務圖-1

- (3) 一次派工，時的執行電信系統修復派工與電力系統修復派工，即一次派工需修復電信與電力系統;亦可輸入多個故障地點：

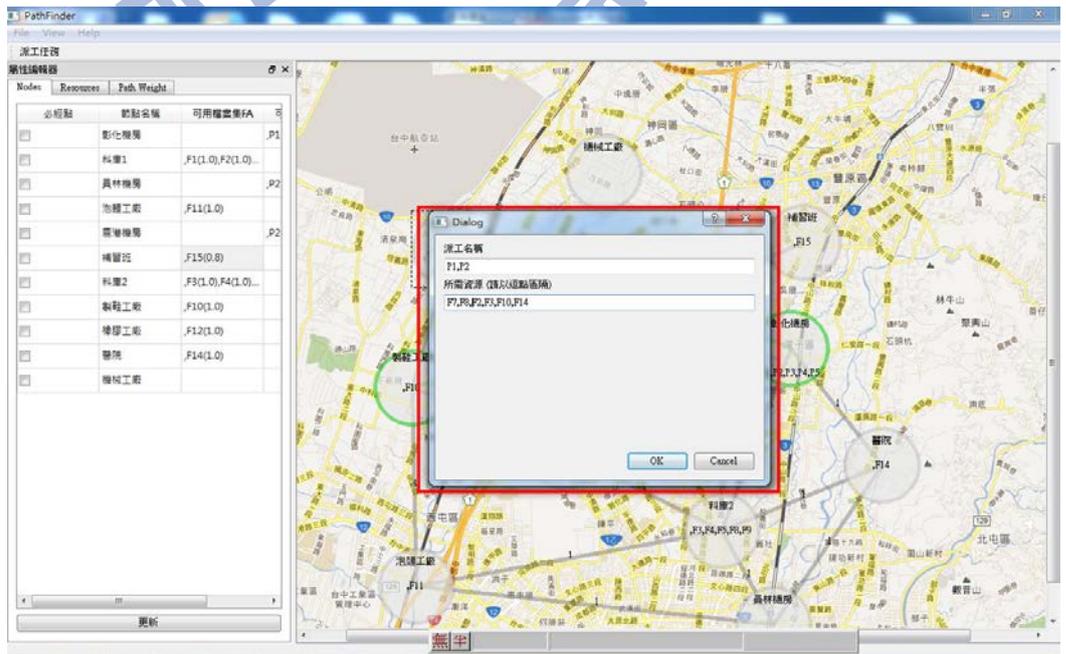


圖 31 執行派工任務圖-2

4.3.1.5. 執行結果圖形輸出

程式會在圖形化結果輸出區以圖形化的方式展示非出此次建議的派工路徑。另外會列出其他的符合條件的路徑。如下圖為執行。

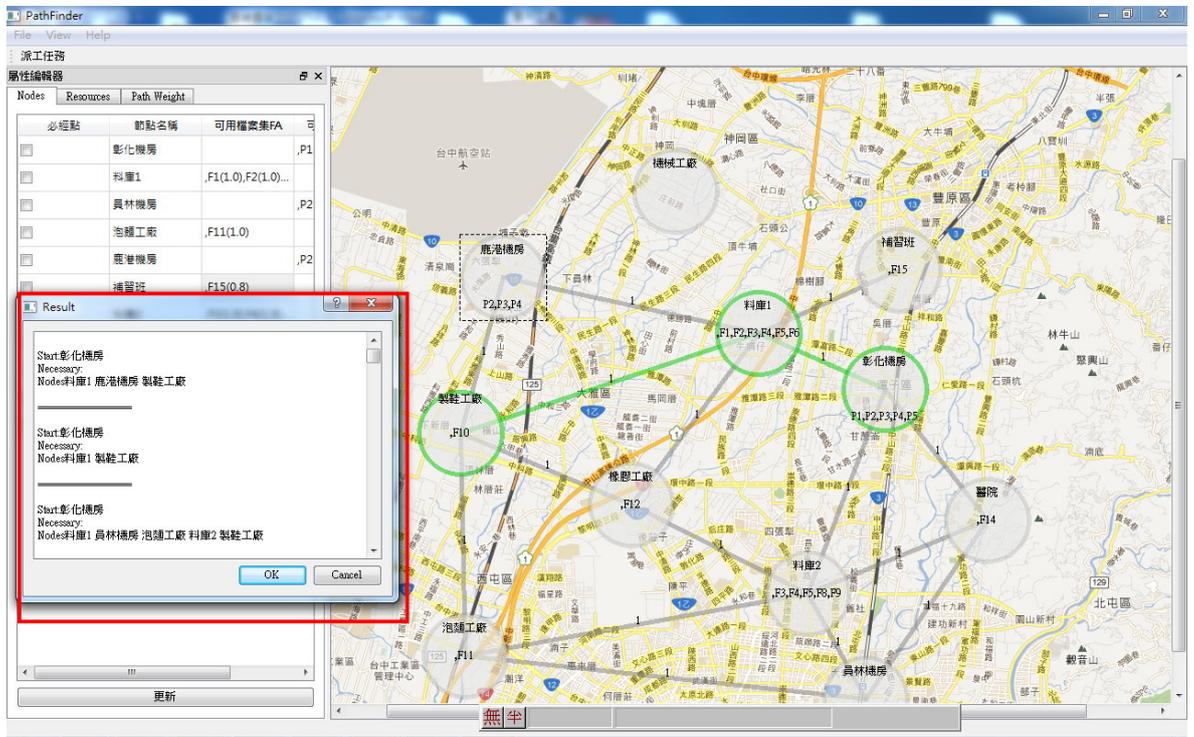


圖 32:輸出結果圖-1

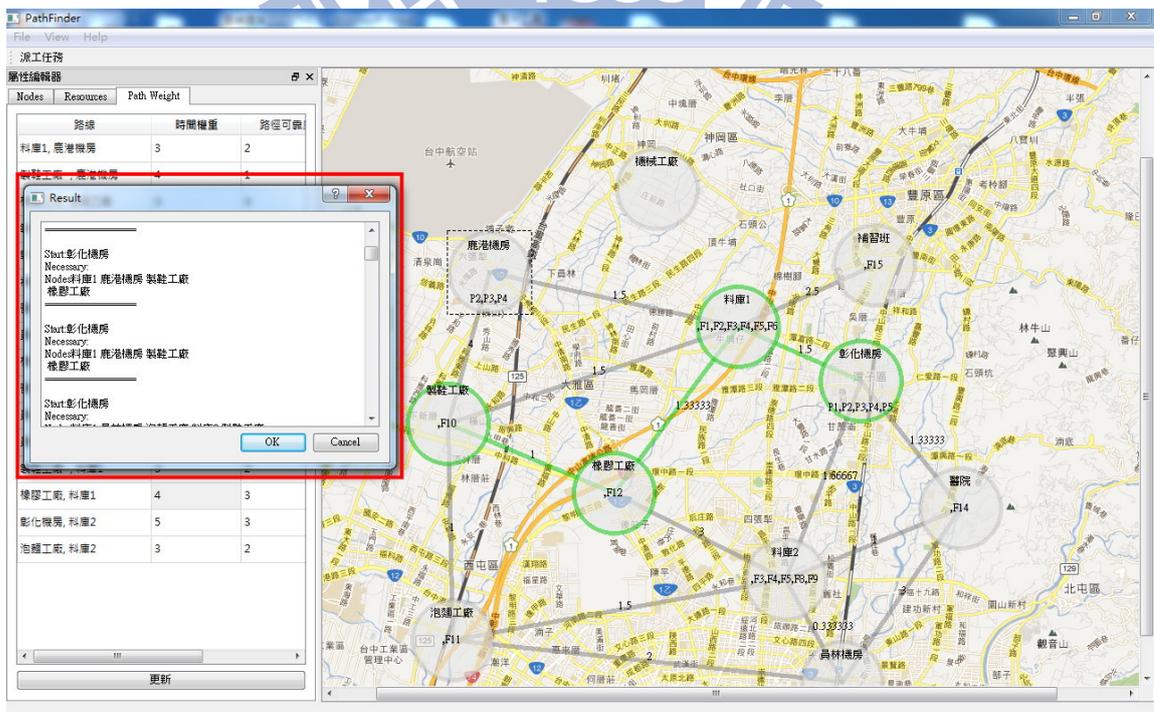


圖 33 輸出結果圖-2

4.3.2.1. 實際案例執行程式說明

將 4.2 實際案例各項數據輸入至程式，執行結果如下：

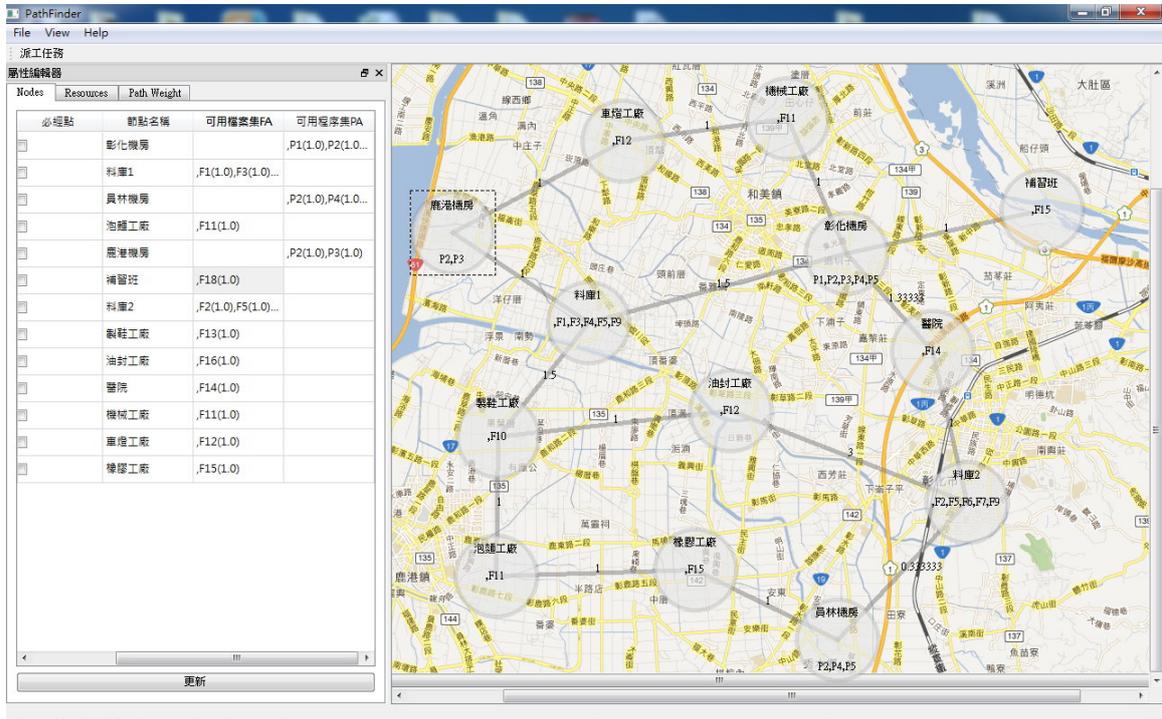


圖 34 實際案例執行畫面

執行 P2 所需資源為 F3, F4, F15 執行畫面如下：

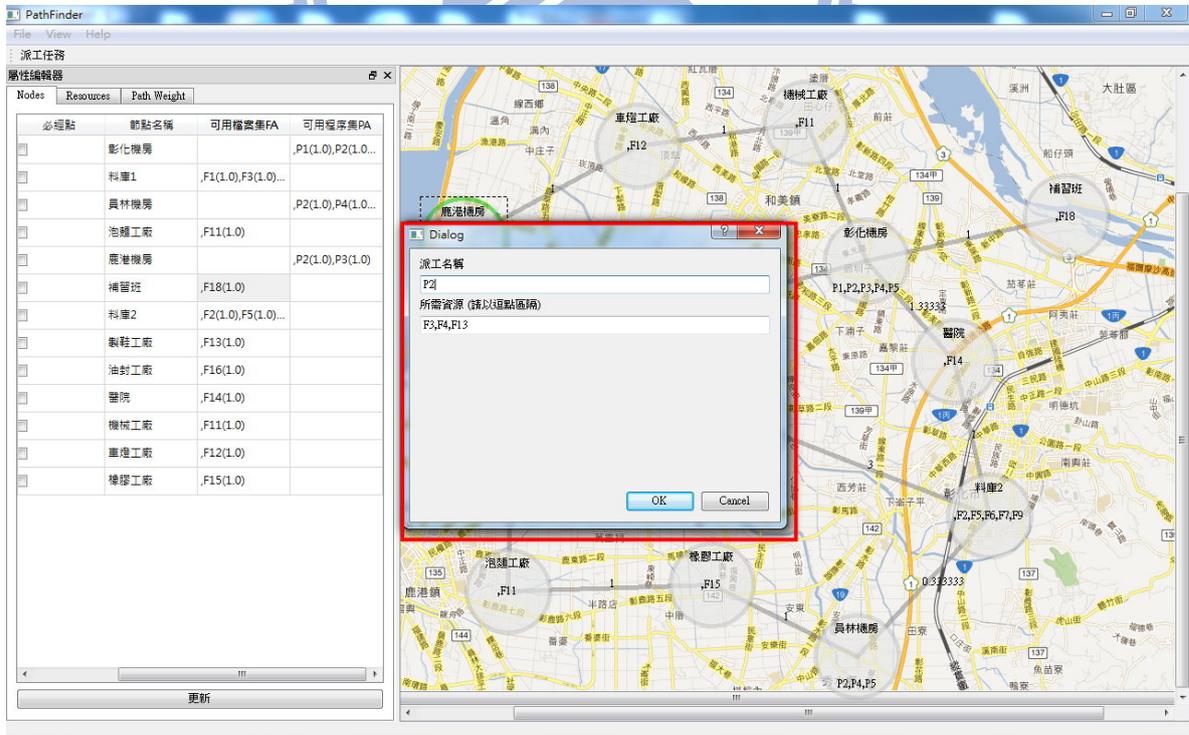


圖 35 :實際案例執行派工任務

執行結果：會將其他最佳解的可行路徑也都列出。

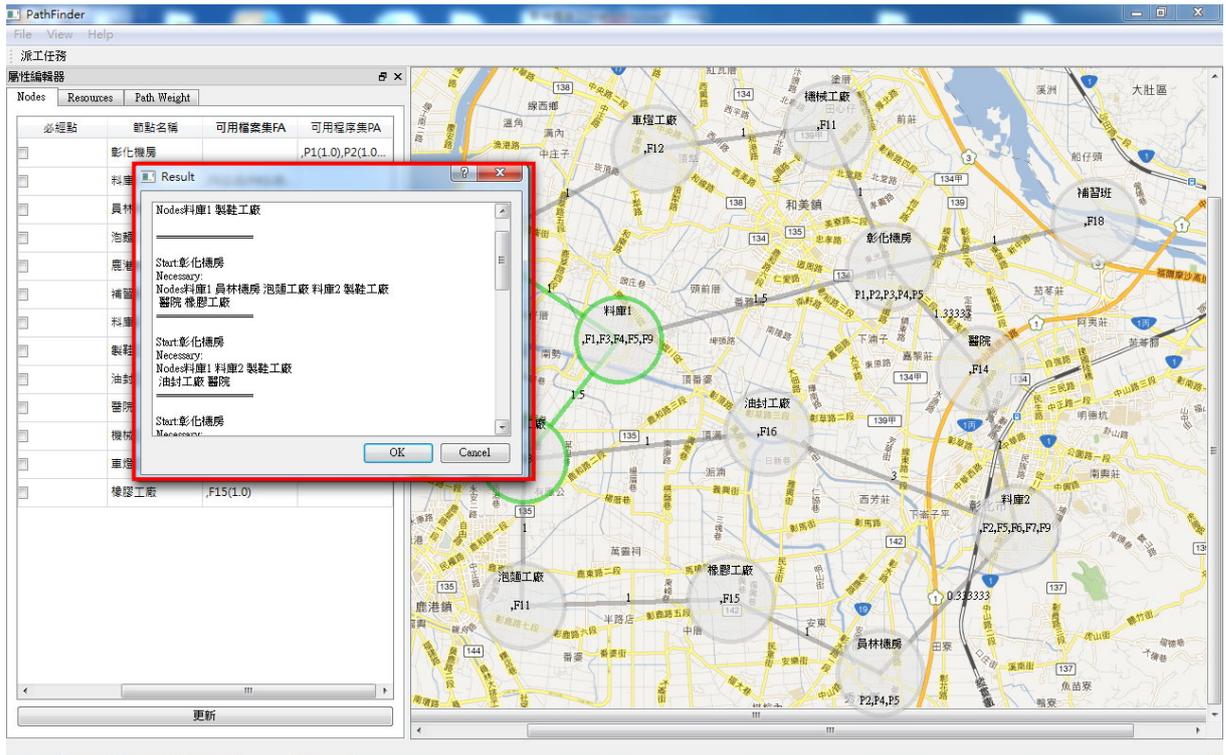


圖 36:執行派工結果-1

程式執行結果與 4.2 結果相同

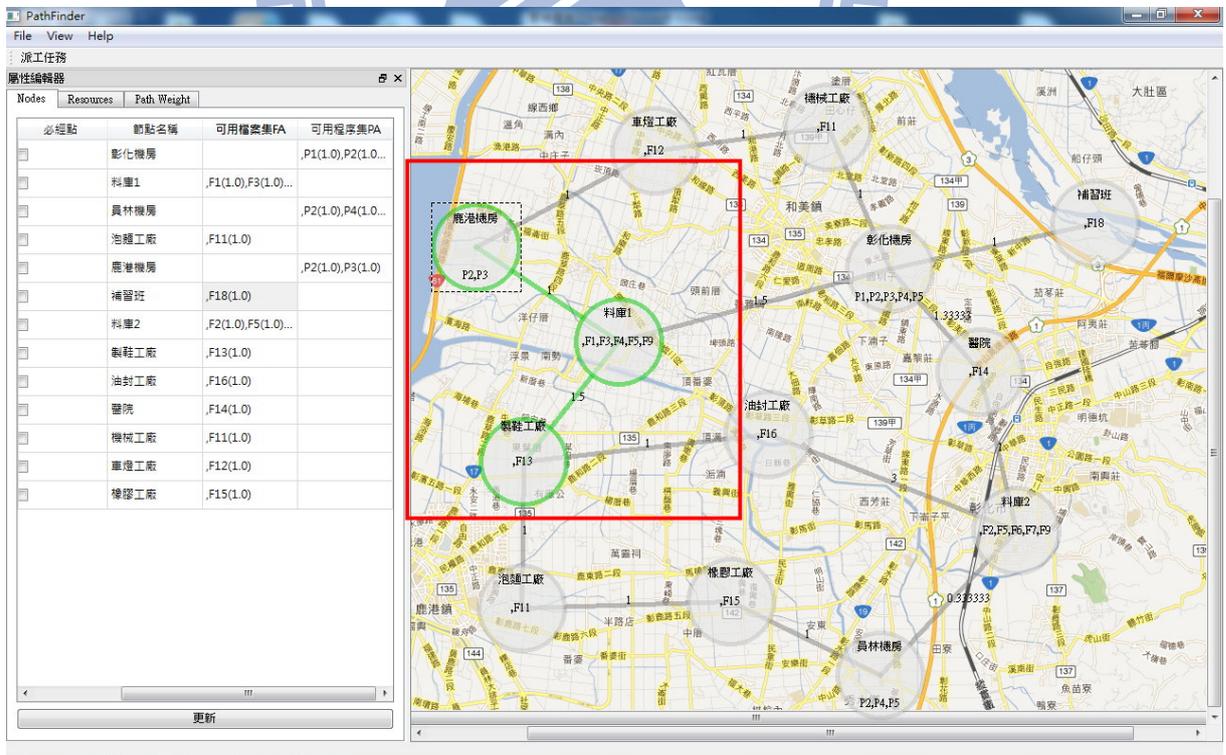


圖 37:執行派工結果-2

五、結論

5.1. 結論

1. 本研究的目的是提出一個有效率的電信系統故障派工方法，在符合派工問題與資源限制下，使每一次的派工都能讓維修效率最高的維護人員在最短的時間內，到達發生故障問題地點。
2. 我們所採用的方法為以 MFST 演算法為基礎來建立派工模型，並加入即時更新路徑權重以期能符合實際派工情況。
3. 在本篇研究當中，可以了解去解決更多類型的派工問題，未來的研究方向，我們期望可以去解決更多類型的派工問題，我們期望可以解決其他組合最佳化的問題，預期我們的演算法也能有同樣優良的表現。

5.2. 未來研究方向

1. 未來電信系統故障派工可透過 M 化派工平台及 LBS (Location Based Service) 定位技術，配合企業內部系統執行調派人員、料件與車輛，聰明調度資源，讓外勤人員及時掌握所需資訊與事件的地理位置。
2. 應用到警政、消防等領域:大型災害發生後常伴隨著維修人員與資源短缺的情形發生。如何有系統的做資源分配與規劃救災路線就十分重要。

參考文獻或資料

- [1] Task assignment with multiskilled employees and multiple modes for product development projects, *International Journal Of Advanced Manufacturing Technology* Volume: 61 Issue: 1-4 Pages: 391-403 DOI: 10.1007/s00170-011-3686-7 Published: JUL 2012
- [2] Xu, Q. and K.E. Xi-Zheng, 2009. Models and genetic algorithm for designated-points shortest path problem. *Syst. Eng. Electron.*, 31: 459-462.
- [3] G.J. Hwang, S.S. Tseng, "A heuristic task assignment algorithm to maximize reliability of adistributed system," *IEEE Transactions on Reliability*, vol R-42, 1993 Sep, pp 408-415.
- [4] Chin-Ching Chiu, Chung-Hsien Hsu, Yi-Shiung Yeh, "A genetic algorithm for reliability-oriented task assignment with k duplications in distributed systems", *Reliability IEEE Transactions on*, On page(s): 105 - 117 Volume: 55, Issue: 1, March 2006
- [5] M. Kafil, I. Ahmad, Optimal task assignment in heterogeneous distributed computingsystems, *IEEE Concurrency* 6: 42-52, 1998.
- [6] A. Ernst, H. Jiang, M. Krishnamoorthy, Mathematical programming approaches for solving task allocation problems, *Proc. of 16th National Conf. of Australian Society of Operations Research*: 1-11, 2001.
- [7] A. Hagin, G. Dermler, K. Rotherml, Problem formulation, model and algorithms formapping distributed multimedia applications to distributed computer systems, *Technical Report*, Universitat Stuttgart, Fakultat Informatik, 1996
- [8] Y. Hamam, K. S. Hindi, Assignment of program modules to processors: a simulatedannealing approach, *European Journal of Operational Research* 122: 509-513, 2000
- [9] A. K. Tripathi, B. K. Sarker, N. Kumar, A GA Based Multiple Task Allocation Considering Load, *International Journal of High Speed Computing*: 203-214, 2000
- [9] E. S. H. Hou, N. Ansari, H. Ren, A genetic algorithm for multiprocessor scheduling, *IEEE Trans. Parallel and Distributed Systems* 5: 113-120, 1994
- [10] Bicking, F., Conrard, B., Thiriet, J.-M., "Integration of dependability in a task allocation problem", *Instrumentation and Measurement, IEEE Transactions on*, On page(s): 1455 - 1463 Volume: 53, Issue: 6, Dec. 2007

- [11] 張克誠，災後搶修設計工作人力指派最佳化之研究-國立中央大學土木系碩士論文 2007
- [12] T Zheng, Optimal Ant Colony Algorithm based Multi-Robot Task Allocation and Processing Sequence Scheduling* Intelligent Control and Automation, 2008. *WCICA 2008. 7th World Congress on 25-27 June 2008*
- [13] Zhigang Lu ,An Improved ACO Algorithm for Service Restoration in Power Distribution Systems Power and Energy Engineering Conference, 2009. *APPEEC 2009. Asia-Pacific*
- [14] S. M. Shatz, J. P. Wang, M. Goto, Task allocation for maximizing reliability of distributed computing systems. *IEEE Trans. Computers*, vol.41, 1992, pp. 1156-1168. [13] J. P. Wang, S. M. Shatz, Task allocation for optimized system reliability. In Proc. 7th Symposium on Reliable Distributed Systems, 1988, pp. 82-90
- [15] M. S. Lin and D. J. Chen, Distributed program reliability analysis. In Proc. 3th Workshop on Distributed Computing Systems, 1992, pp.395-401
- [16] S. Hariri, C. S. Raghavendra, Distributed functions allocation for reliability and delay optimization. Proc. *IEEE/ACM 1986 Fall Joint Computer Conf*, 1986, pp. 344-352.
- [17] R. Seethalakshmi, S. Savichandran "reliability analysis of programs in the distributed computing environment", JATIA 2011
- [18] V. K. P. Kumar, S. Hariri and C. S. Raghavendra. Distributed program reliability analysis. *IEEE Transaction on Software Engineering*, 1986, 12 (1):42-50
- [19] S. Haririan dC .S .R aghavendra. SYREL: A Symbolic Reliability Algorithm based on Path and Cutset Methods , USC Tech .R ep., 19 84, 11 (3):52-64
- [20] A. K umar ,S .R aian dD .P. Agrawal .Reliability evaluation Algorithms for Distributed System s .P roc. IN FOCOM'88, 19 88:85-1860
- [21] D. J. Chen and M. S. Lin, "On Distributed Computing Systems Reliability Analysis Under Program Constraints," *IEEE Transaction on Computers*, Vol. 16, No. 1, January, 1994.
- [22] Luiz Chaimowicz, Mario F. M. Campos, and Vijay Kumar, "Dynamic Role Assignment for Cooperative Robots," In Proc. of the *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 293--298, Washington, DC, May 2002.", pp. 293--298, 2002.
- [23] T. Dahl, S., M. J. Mataric, and G. S. Sukhatme, "Multi-Robot Task-Allocation through Vacancy Chains," 2003.

- [24] Brian P Gerkey and Maja J Mataric, "Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures," Proc. of the *IEEE Intl. Conf. On Robotics and Automation (ICRA)*, 2003.
- [25] Z. J. Czech and Piotr Czarnas, "Parallel Simulated Annealing for the Vehicle Routing Problem With Time Windows," 2002.
- [26] K. H. Kim and K. C. Moon, "Berth scheduling by simulated annealing," *Transportation Research Part B: Methodological*, vol. 37, pp. 541-560, 2003. [10] S. Melouk, P. Damodaran, and P.Y. Chang, "Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing," *International Journal of Production Economics*, vol. In Press, Corrected Proof, 2003.
- [27] 鄭承泰，服務率變化下M/M/1 排隊系統之最佳存貨控制-國防管理學院碩士論文 2003
- [28] Naishuo Tian, Xinqiu Zhao, The M/M/1 Queue with Single Working Vacation *International Journal of Information and Management Sciences* Volume 19, Number 4, pp. 621-634, 2008
- [29] 劉郁青，含有可變動服務者及阻礙概念之機器修理問題-中興大學應用數學系碩士論文-2008
- [30] R. E. Merwin and M. Mirherkerk, "Derivation and use of survivability criterion for DDP systems," in Proc. 1980 Nat. Comput. Conj, May 1980, pp. 139-146.
- [31] K. K. Aggrawal and S. Rai, "Reliability evaluation in computer communication networks," *IEEE Trans. Reliability*, vol. R-30, pp. 32-35, Apr. 1981.
- [32] A. Grnarov and M. Gerla, "Multiterminal reliability analysis of distributed processing system," in Proc. 1981 Int. Conf Parallel Processing, Aug. 1986, pp. 79-86.
- [33] R. Kevin Wood, "Factoring algorithms for computing I; -terminal network reliability," *IEEE Trans. Reliability*, vol. R-35, pp. 269-278, Aug. 1986.
- [34] Yan, S. and Chen, H.L., "A Scheduling Model and a Solution Algorithm for Inter-city Bus Carriers," *Transportation Research*, Vol. 36A, pp. 805~825 (2002).
- [35] Méndez-Díaz, I., Zabala, P. and Lucena, A. (2008), "A new formulation for the traveling deliveryman problem," *Discrete Applied Mathematics*, Vol. 156, No. 17, pp. 3223-3237.
- [36] Gengqian Liu, Tiejun Li, Yuqing Peng, and Xiangdan Hou, The ant algorithm for solving robot path planning problem, *Proceedings of the International Conference on Information Technology and Applications (ICITA'05)* 2005.
- [37] Tan Guan-Zheng, He Huan and Sloman aron, Ant colony system algorithm for real-time globally optimal path planning of mobile robots, *ACTA AUTOMATICA SINICA*,

vol. 33, no. 3, pp. 279~285,

2007. (in Chinese with English abstract)

[38] .H. Thane and H. Hansson. Testing distributed real-time systems. *Microprocessors and Microsystems*, 24:463–478, 2001.

[39] Han-Pang Huang, and Shu-Yun Chung, *Dynamic visibility graph for path planning*, *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent and Systems*, pp. 2813-2818, 2004.

[40] Demot Madden, and Declan Delaney, *Visibility path-finding in relation to hybrid strategy-based models in distributed interactive applications*, *Distributed simulation and real-time application*, pp. 91-97, 2004.

[41] M. S. Lin. *The Reliability Analysis on Distributed Computing Systems*, PhD dissertation, National Chiao Tung University, Hsinchu, Taiwan, 1994.

[42] M. S. Lin, D. J. Chen, and M. S. Horng. The reliability analysis of distributed computing systems with imperfect nodes. *Computer Journal*, 42:129–141, 1999.

[43] 林瑜禎，自來水管線維修工程人員機具指派最佳化-國立大學土木工程學系碩士論文- 2011

[44] M. S. Lin, D. J. Chen, and M. S. Horng. A Heuristic Approach to Generating File Spanning Trees for Reliability Analysis of Distributed Computing Systems*, *Computers Math. Applic.* Vol. 34, No. 10, pp. 115-131, 1997

