

國立成功大學  
工業與資訊管理學系碩士在職專班  
碩士論文

考慮風力影響之單台無人船最短覆蓋路徑規劃問題

The shortest coverage path planning problem of a single  
unmanned ship considering the influence of wind

研 究 生：陳蓁慧

指 導 教 授：王逸琳 博士

中 華 民 國 一 百 一 十 年 九 月

國立成功大學

碩士在職專班論文

考慮風力影響之單台無人船最短覆蓋路徑規劃問題  
The shortest coverage path planning problem  
of a single unmanned ship considering the  
influence of wind

研究生：陳蓁慧

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳

吳政翰

阮婕如

指導教授：王逸琳

系(所)主管：翁榮宗

(單位主管是否簽章授權由各院、系(所、學位學程)自訂)

中華民國 110 年 9 月 9 日

# 摘要

近年來因感測裝置與自動導航技術已趨成熟，各種新興具智慧的無人載具已能取代部分人力去執行以往必須倚賴人力才能完成的某些任務。然而目前的人工智慧或大數據分析等技術僅能協助無人載具在行進途中不致擦撞毀損，至於更上層的路徑規劃決策則仍需倚賴作業研究理論才能處理。本研究即探討單台無人船的巡邏路徑規劃問題，在一給定的水域範圍內，先等距定位出平面縱橫雙向的整齊格點，假設無人船行經某格點時其偵測範圍可覆蓋該格點之鄰近範圍，給定水域中某些必須被偵測範圍覆蓋的重點巡邏區域，本研究探討如何規劃單台無人船巡邏路線以使所有重點區域可在最短時間內被偵測範圍覆蓋到；考量船隻在水面的行進速度會受風力影響，且風的速度與方向可能又會依時而變。假設該水域中各格點未來數期間內的風力與風向皆已被準確預測，本研究擬將這些已知的風力與風向變化列入考量，規劃單台無人船巡邏路徑的最佳路線。

本研究與文獻中的最短覆蓋路徑規劃問題(Shortest Coverage Path Problem, SCPP)最為相關，主要差異之處在於本研究進一步考量環境外力（譬如風力）對行進速度的影響，以及使用單台無人船來處理 SCPP，第一階段規劃求解單期風力之單台無人船路徑；第二階段進一步處理多期風力之單台無人船路徑規劃，但限制其路線不能有任何重複經過的路段；而第三階段則再進一步探討可重複經過任何路段的最佳路線規劃問題。我們將先提出各問題之整數規劃模型，再設計二種貪婪演算法機制(Greedy Algorithm)和四種模擬退火法(Simulated Annealing)機制以在更短時間內求解。

在數值測試部分，我們建構出 3 類不同大小的隨機水域網路，再分別設定 3 種不同的重點區域個數，各種設定組合個別產生 10 個、總共 90 個隨機測試案例。數值測試結果顯示，不考慮路段可重複之整數規劃模型(M1)兼具效能與效率，而 G1 則適合用來產生優質初始解，再以 S2 或 S4 可在短時間內收斂到品質不錯的近似最佳解。

關鍵詞：無人船；最短覆蓋路徑；整數規劃；貪婪演算法；模擬退火法

# Abstract

## **The shortest coverage path planning problem of a single unmanned ship considering the influence of wind**

Chen-Hui Chen

I-Lin Wang

Department of Industrial and Information Management,  
National Cheng Kung University

### **SUMMARY**

This research discusses the problem of patrol path planning for an unmanned ship. Within a given water area, we first locate grid points in the vertical and horizontal directions at equal distances and assume an unmanned ship can cover some grid points within its detection range. As a result, a target patrol area can be covered as long as an unmanned ship moves to its neighborhood. We develop mathematical programming models and algorithms to plan an unmanned ship patrol route to cover all target patrol areas in the shortest time. We consider the ship's speed on the water surface to be affected by the wind, whose speed and direction may change over time. Assuming that the predicted wind speed and direction at each grid point in the water area are sufficiently accurate, we consider these factors in planning the best patrol route for an unmanned ship.

Our problem can be considered an extension of the Shortest Coverage Path Problem (SCPP) in the literature. Two integer programming models, M1 and M2 over a time-space network, are proposed. Repeated node or arc traversal is not allowed in M1 but can appear in M2. Although M1 is faster than M2, both are very time-consuming. Therefore, we design efficient algorithms including two Greedy Algorithms, G1 and G2, and four Simulated Annealing mechanisms, S1, S2, S3, and S4. In our computational experiments, we have generated ninety random networks of 10 categories, where each category has a different network size and quantity in target patrol areas. The test results indicate M1 has the best trade-off performance, G1 is best for an initial solution, which can be further improved by S2 (or S4).

Keywords: Unmanned Ship; Shortest Coverage Path; Integer Programming; Greedy Algorithm; Simulated Annealing

## INTRODUCTION

In recent years, the unmanned vehicles have been widely used to perform patrolling tasks or inspection tasks. These tasks all have several things in common: operated in a designated area, affected by the external environment such as wind, needed to cover target areas, and conducted in the neighborhood of target areas. This research will discuss the shortest coverage path planning. When performing observation or inspection tasks in a certain area, we consider the influence of the external environment on the ship speed, such as: wind, wind direction, or ocean current, to route an unmanned ship to finish the tasks as soon as possible.

## MATERIALS AND METHODS

We first focus on integer programming models for this problem. First, we investigate the routing on an ideal no-wind area in a single-period planning horizon; then, we extend the duration to multiple periods and further consider the time-dependent arc lengths caused by wind in different periods. To simplify the problem, we first propose an integer programming (IP) formulation M1 with the simple-path assumption. Then, we relax the simple-path assumption to form another IP formulation M2, which allows more general routings that may pass through a node more than once. M1 consumes much time, and M2 takes even more time than M1, which leads to the need to develop more efficient solution methods. To this end, two shortest-path-based greedy algorithms G1 and G2 are designed. However, both G1 and G2 lack some iterative mechanism to improve their results iteratively. Therefore, we propose four heuristics S1, S2, S3, and S4, based on the simulated annealing mechanism. In particular, these four heuristics can start from a solution by greedy algorithms and further converge the solution.

## RESULT AND DISCUSSION

We have conducted extensive computational experiments on 90 randomly generated networks that cover nine groups of different sizes and target patrol area distributions, where 10 random networks are generated in each group. Theoretically, M2 can calculate optimal solutions, but it takes too much time. M1 can calculate optimal solutions for most cases, and with small optimality gaps for some cases in a much shorter time than M2. G1 is usually better than G2 in both solution time and quality. The performances of S1, S2, and S4 are all very good, with an average Gap below 1%, but S2 and S4 are more robust.

## CONCLUSION

This research takes time-dependent travel time in a water area into consideration to calculate an optimal patrol route for an unmanned ship to cover the target patrol areas with the shortest time. We develop IP formulations on a time-space network, greedy algorithms based on shortest paths to calculate a good solution quickly, and heuristics based on a simulated annealing mechanism. Our major contributions have three folds: (1) innovative IP modeling techniques to deal with time-dependent travel time; (2) efficient greedy mechanisms by integrating ideas used for solving the knapsack problem and the shortest paths; and (3) new simulated annealing mechanisms to further converge a feasible solution to a better one. For future research, we suggest the extension from a single ship to multiple ships. In addition, how to take the power (gas or electricity) renew scheduling into consideration can also be very challenging.



## 誌謝

在經歷漫長碩士階段的求學生涯，很開心終於能完成碩士班學業，過去從大學畢業後由求學階段走入職場上，在經過了幾年在職場上的磨練，其實學習到很多在校學習過程中不一樣的事物，但有些時候其實在同一環境待久了，工作方式的想法和模式被固定了，反而被侷限了想像的空間，因此很開心能有這機會能重回校園，對於曾經所學的專業知識，有另一番更深入的了解其知識的用途，而對於過去從未接觸的學科，進而增加了我學習的視野和新的技能，相信對未來的我一定會有所幫助。

此篇論文的完成，首先要非常感謝我的指導教授王逸琳老師，因過去從未有作業研究相關學習經歷，有很多不懂的地方，而老師總是非常有耐心和細心的指導並帶著我一步一步了解，讓我從中受益良多，也要感謝口試委員吳政翰教授和阮婕如教授在這特殊的防疫期間能來擔任此口試委員，並在口試前細心閱讀我的論文，口試時給予寶貴的建議，讓此篇碩士論文能更加完善。也要感謝林瑩婷同學在繁忙的課業外能協助我在數學模型上的解惑和幫助，終於將碩士論文完成。

在研究所的生活裡，很感謝老師們的細心教導，讓我學習到寶貴的知識，也很開心能成為我們這一班的一份子，雖然大家來自不同的地方，但彼此都能互相關心、互相幫助，完成了很多不可能的任務和課業，這相處的過程也將成為我人生中最美好的回憶，但因為工作職務關係常常需能及時排除系統上問題到了後期遇到公司這期間導入新 ERP 系統使得工作生活更加忙碌不堪，工作佔滿了我大部份的生活也佔據了睡眠，因此對指導教授王逸琳老師有滿滿的感激和感謝，謝謝老師。

陳蓁慧 謹誌

成功大學工業與資訊管理學系

中華民國一百一十年九月

# 目錄

摘要 .....	I
Abstract .....	II
誌謝 .....	V
目錄 .....	VI
表目錄 .....	VIII
圖目錄 .....	IX
第一章 緒論 .....	1
1.1 研究背景與動機 .....	1
1.2 研究目的 .....	4
1.3 研究範圍 .....	4
1.4 研究架構 .....	5
第二章 文獻探討 .....	7
2.1 USV 與 UAV 相關之覆蓋路徑應用文獻 .....	7
2.2 最短覆蓋路徑之相關文獻 .....	8
2.2.1 最短覆蓋路徑問題(SCPP) .....	8
2.2.2 覆蓋銷售員問題(CSP) .....	11
2.2.3 覆蓋旅遊問題 (CTP) .....	14
2.3 小結 .....	16
第三章 最短覆蓋路徑規劃問題之整數規劃模型 .....	17
3.1 問題描述 .....	17
3.2 單期風力單台無人船整數規劃模型 .....	17
3.3 多期風力單台無人船整數規劃模型 M1 .....	19
3.4 以 M1 為基礎但路線可重複之路徑規劃數學模式 M2 .....	23
3.5 小結 .....	28
第四章 求解演算法 .....	29
4.1 求解演算法 G1 .....	29
4.2 求解演算法 G2 .....	31
4.3 求解演算法 S1 .....	32
4.4 求解演算法 S2 .....	36
4.5 求解演算法 S3 .....	38
4.6 求解演算法 S4 .....	41
4.7 小結 .....	42
第五章 例題測試與分析 .....	44



5.1 測試案例描述 .....	44
5.2 測試結果分析 .....	45
5.2.1 小型網路 .....	46
5.2.2 中型網路 .....	48
5.2.3 大型網路 .....	51
第六章 結論與建議 .....	56
6.1 結論 .....	56
6.2 未來研究與建議 .....	57
參考文獻 .....	59
附錄 A .....	64
多期風力 K 台無人船設計其整數規劃模型和目標函式與限制式 .....	64



## 表目錄

表 2.1 USV 的應用 .....	7
表 2.2 UAV 的應用 .....	8
表 3.1 案例 A 說明整數規劃模型和演算法最佳解 .....	25
表 3.2 案例 A 整數規劃模型可重覆經過和不可重覆經過與演算法最佳解路徑比較 .....	29
表 5.1 產生測試網路圖定義 .....	46
表 5.2 求解時間與 Gap 比較表-小型網路 .....	47
表 5.3 求解時間與 Gap 比較表-中型網路 .....	50
表 5.4 求解時間與 Gap 比較表-大型網路 .....	53
表 5.5 M1 和 M2 求解時間與 Gap 比較表-大型網路 .....	53



## 圖目錄

圖 1.1 (a)偵察或巡邏水域; (b)等距定位出平面縱橫雙向的整齊格點圖 .....	2
圖 1.2 (a)指定偵察或巡邏的點(b)需要巡邏的點與定義湖面格點對照位置 .....	2
圖 1.3 不同的路徑規劃 .....	3
圖 1.4 在不同時段受風力影響程度不同，路徑的規劃也會不同 .....	3
圖 1.5 多台無人船路徑規劃 .....	4
圖 2.1 最短覆蓋路徑問題路徑規劃 .....	10
圖 2.2 可覆蓋各個需求點之節點與節線集合 .....	10
圖 2.3 覆蓋銷售員問題路徑規劃 .....	13
圖 2.4 CSP 可覆蓋各個需求點之節點和路徑規劃 .....	13
圖 2.5 CTP 可覆蓋各個需求點之節點和路徑規劃 .....	15
圖 3.1 單期風力單台無人船路徑規劃 .....	19
圖 3.2 多期風力單台無人船路徑規劃 .....	20
圖 3.3 藍色箭頭: 整數規劃模型執行路徑; 橘色箭頭: 演算法執行路徑 .....	25
圖 3.4 將每個節點增加其對應的虛擬點 .....	26
圖 4.1 G1 演算法步驟說明 .....	31
圖 4.2 G1 演算流程圖 .....	32
圖 4.3 求解演算法 G2 步驟說明 .....	33
圖 4.4 求解演算法 S1 流程圖 .....	35
圖 4.5 演算法 G1 計算而得之路線 .....	36
圖 4.6 演算法 S1-隨機互換兩個節點的位置 .....	36
圖 4.7 演算法 S1 計算而得之路徑 .....	37
圖 4.8 求解演算法 S2 流程圖 .....	38
圖 4.9 演算法 G1 計算而得之路線 .....	39
圖 4.10 演算法 S2 計算而得之路徑 .....	39
圖 4.11 求解演算法 S3 流程圖 .....	41
圖 4.12 演算法 G1 計算而得之路線 .....	42
圖 4.13 演算法 S2 計算而得之路徑 .....	42
圖 4.14 求解演算法 S4 流程圖 .....	43
圖 5.1 大、中、小型的網路圖範例 .....	46
圖 5.2 小型網路-Gap 盒鬚圖 .....	48
圖 5.3 小型網路-求解時間(s)盒鬚圖 .....	48
圖 5.4 中型網路-Gap 盒鬚圖 .....	50
圖 5.5 中型網路-求解時間(s)盒鬚圖 .....	51
圖 5.6 中型網路-測試組別 7-7-8 演算法改善程度 .....	51
圖 5.7 中型網路-測試組別 7-7-16 演算法改善程度 .....	52

圖 5.8 大型網路-Gap 盒鬚圖 .....	54
圖 5.9 大型網路-求解時間(s)盒鬚圖 .....	54
圖 5.10 大型網路-測試組別 10-10-11 演算法改善程度 .....	55
圖 A-1 多期風力 K 台無人船路徑規劃 .....	65



# 第一章 緒論

近年來因科技的進步，對於各種新興與智慧的無人載具已成為未來發展和研究的趨勢，無人載具廣泛的應用下，常常被應用在各種混亂環境中執行任務或檢測任務，像勘災、巡邏、或執行空中農藥噴灑等，它們都有幾個共同點，像是運行在某指定的區域、受外在環境的影響、無人載具都具有覆蓋或檢測的範圍、必須走過整個區域其路徑規劃的問題。

## 1.1 研究背景與動機

隨著各種智慧無人載具及機器人的蓬勃發展，資訊、通信與網路技術的整合，無人載具的執行效率提高和資源成本下降，由早期以軍事用途為主的應用，進而延伸至其它相關的產業，包括商業、農業、國防領域、建築/採礦/工程、能源/公共設施巡檢，以及在自然災害發生時提供監測…等，將幫助我們避免在高風險的環境中執行造成的人為傷害，且不用考量時間的限制，如需在深夜時段執行任務。都能透過無人載具的幫助來降低人力成本，因此未來無人載具發展潛力可期。

在無人載具的應用中，不管是無人水面載具(Unmanned Surface Vehicle, USV)或無人飛行載具(Unmanned Aerial Vehicle, UAV)等都是常應用於各種混亂環境中執行任務且自動執行觀察或檢測任務，像勘災、農業監控、環境監測、偵察和巡邏，這些它們都有共同的特性就是要在一個區域的空間裡面、受各種外在環境因素影響、有些地方或幾個地點必須去或到達。假設今天我們有一個任務是要偵察或巡邏某個水域(圖 1.1 (a))其當日的天氣狀況為無風，我們先將水域做等距定位出平面縱橫雙向的整齊格點(圖 1.1 (b))，每格點可知道能被哪些其它格點所覆蓋到，事先指定有幾個地點位置是一定要去偵察或巡邏的點(即必須被覆蓋的點)(圖 1.2)，假設巡邏任務的起始與終點不一定相同，如果要覆蓋到這些指定的地點，可能會有數種不同的路徑選擇，而不

同的路線會有其對應的花費時間、距離長短、耗用電力或能源多寡。以圖 1.3 為例，其左圖的路徑長度將比右圖的路徑距離短。因船速會受不同方向之風力影響，而風速與風向可能又會依時間而有所變化，進而影響行徑時間，導致同一路線在不同時刻可能會耗費不同時間與能源。如圖 1.4 所示，同一時段若風力和風向有變化，將對應出不同之最佳路徑規劃。

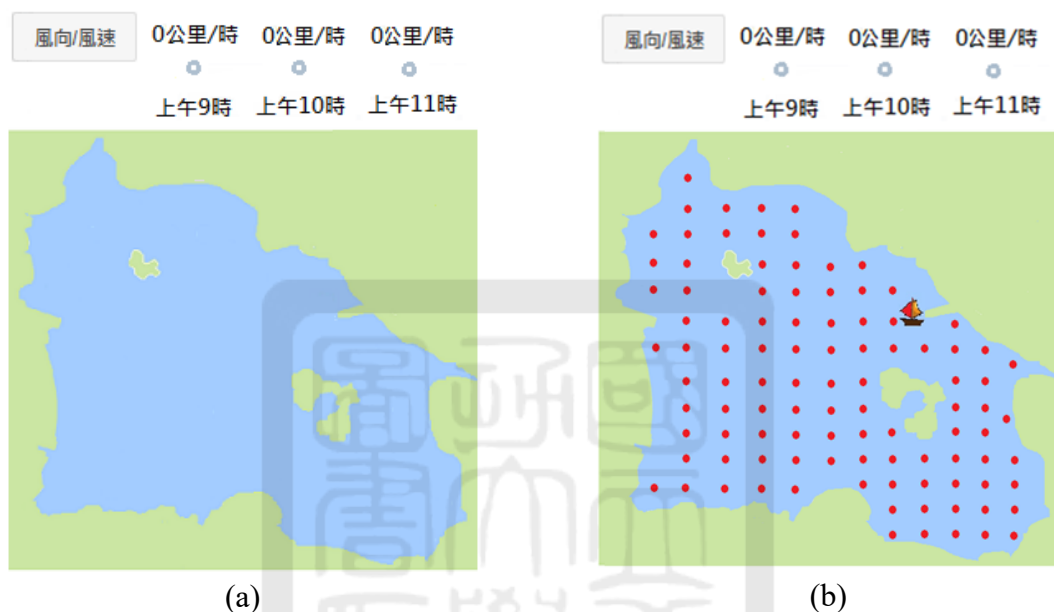


圖 1.1 (a)偵察或巡邏水域; (b)等距定位出平面縱橫雙向的整齊格點圖

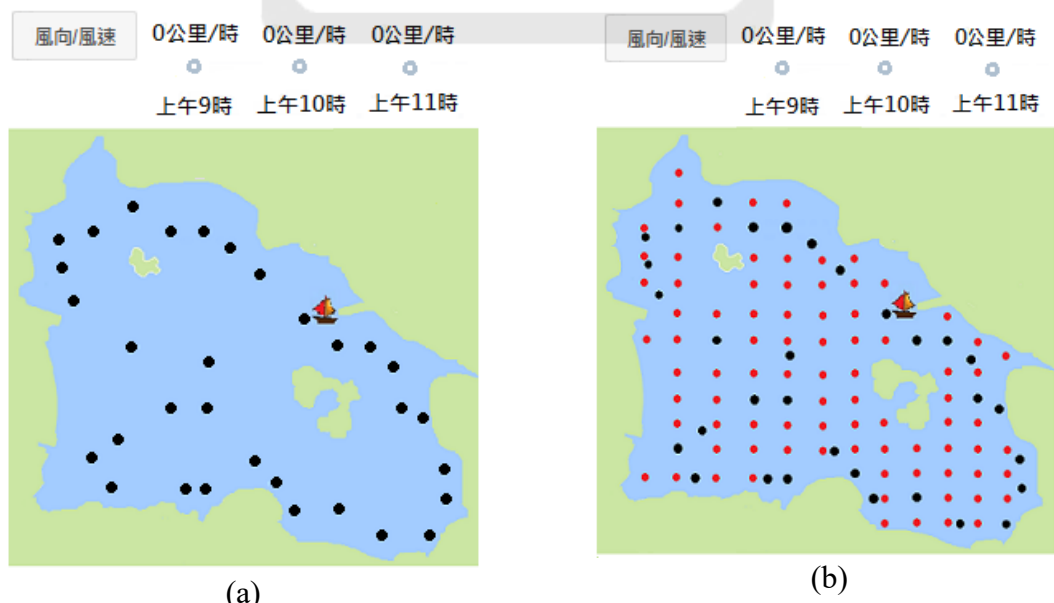


圖 1.2 (a)指定偵察或巡邏的點(b)需要巡邏的點與定義湖面格點對照位置

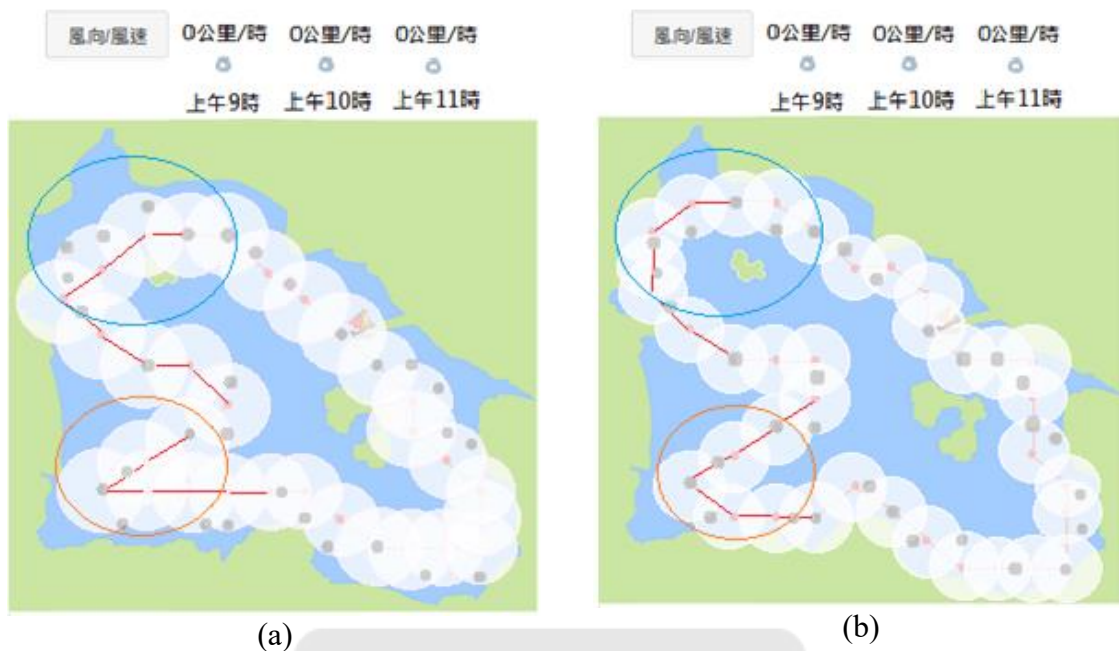


圖 1.3 不同的路徑規劃

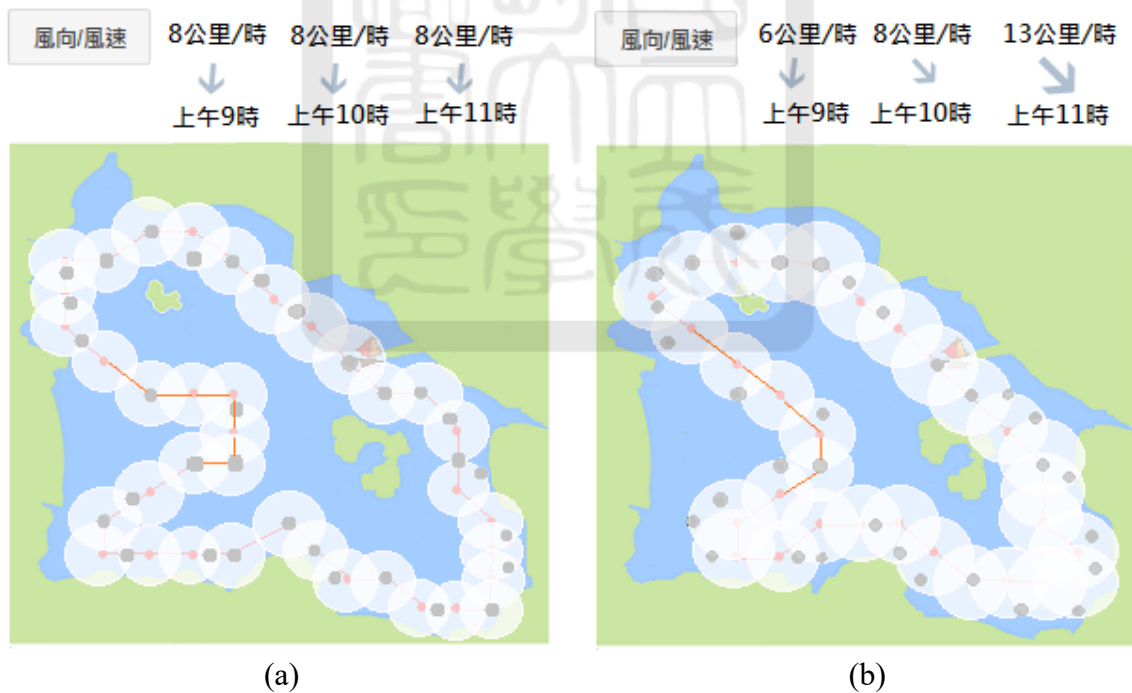


圖 1.4 在不同時段受風力影響程度不同，路徑的規劃也會不同

前面假設都是只有一台無人船的狀況，如果有多台船一起出任務偵察或巡邏時，能更快完成任務，但路徑的選擇變化將更多種，路線選擇亦更有變化(圖 1.5)，使路徑

規劃問題將更複雜。然而，欲求解多台船之路線規劃仍需以單台船之路線規劃為基礎。

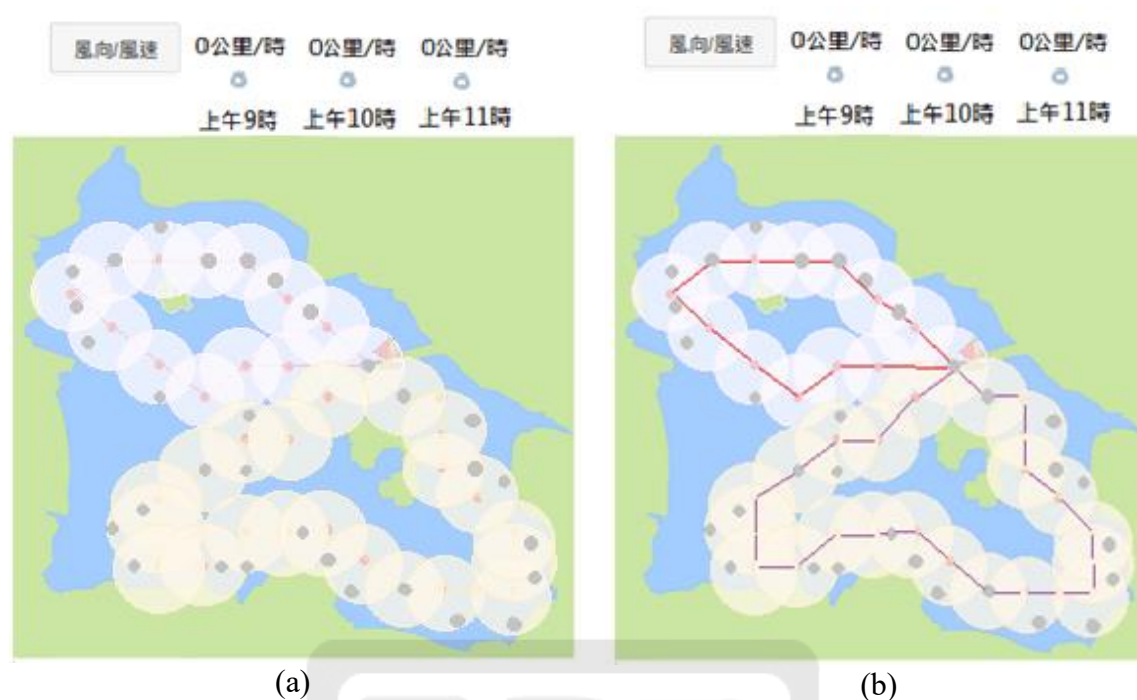


圖 1.5 多台無人船路徑規劃

## 1.2 研究目的

無人載具常被應用在各種混亂環境中執行巡邏任務，本研究將探討最短覆蓋路徑規劃問題，在某一區域執行觀察或檢測任務時，在考慮外在環境的影響船速下，例如：風力、風向、洋流；要如何安排單台的無人船在執行巡邏任務時的路徑規劃，可以達到總體花費最短的時間。若有本研究之基礎後，將來亦可延伸探討其它類似之路徑規劃問題，譬如如何在最節省電力的方式或如何在有限的電力時間內完成的最多覆蓋範圍等問題。

## 1.3 研究範圍

本研究依問題難度依序分為如下三個階段：

1. 單期風力之單台無人船路徑規劃。
2. 多期風力之單台無人船路徑規劃。



### 3. 可重複經過任何路段的最佳路線規劃問題。

並依序提出各問題之整數規劃模型。其中第一階段的問題最為基礎，可用來當第二階段之各期基本模型；第二與第三階段之主要差別在於路徑之規劃是否允許有重複經過的路段，其中第二階段使用一般較常見的路段不重複經過的假設，該假設大部分情境是合理的，然而倘若某些路段的行徑時間在未來期間因風力影響而變短了，將會造成重複經過部分先前已經過的路段反而更加有利，因此才有第三階段的問題。然而為了處理第三階段的路線可重複之彈性，其對應之整數規劃模型之規模將大幅增加，因此我們認為第二階段的整數規劃模型還是有其存在價值，譬如可用它來當成求解第三階段問題的初始解。因為此類的整數規劃模型通常求解耗時甚久，本研究將針對多期風力之單台無人船路徑規劃問題提出更有效率的求解演算法。

## 1.4 研究架構

本研究共分為六個章節，各章節的架構簡述如下：

第一章：主要為確定研究主題，描述研究背景和動機、研究目的、研究範圍和研究架構。

第二章：收集並回顧相關文獻，以整理既有的理論與相關文獻研究。

第三章：介紹三種不同情況之整數規劃模型：單期風力單台無人船、多期風力單台無人船 M1、以 M1 為基礎但路線可重複之路徑規劃數學模式 M2。

第四章：設計比整數規劃模型更有效率的求解演算法，本研究設計二種貪婪演算法機制(Greedy Algorithm)G1、G2，以及四種模擬退火法(Simulated Annealing)機制 S1、S2、S3、S4，以在短時間內求得近似最佳解。

第五章：主要為實驗結果分析，根據第三章的整數規劃模型和第四章的求解演算法所撰寫的程式來進行實驗與分析，來了解所建構的方法在求解的效率與績效。

第六章：進行結論與建議，主要針對本研究的實驗結果作為最終的結論以及對未來研

究方向提出建議，作為後續深入研究之參考。



## 第二章 文獻探討

本研究主要目的是針對單台無人載具在一個範圍的空間內且必須到達數個指定的地點，並規劃出最短覆蓋路徑，因此文獻探討主要分為二個部分，首先簡介無人水面載具(USV)以及無人飛行載具(UAV)相關之覆蓋路徑文獻，再針對最短覆蓋路徑問題的相關文獻進行探討。

### 2.1 USV 與 UAV 相關之覆蓋路徑應用文獻

隨著全球對於無人水面載具(USV)和無人飛行載具(UAV)的發展應用日益增加，在相關商業、科學和軍事用途都有相關的應用，本研究將只針對相關的覆蓋路徑的應用介紹。

USV 被定義為在各種混亂環境中執行任務的無人駕駛載具，而不需要任何人為干預，並且基本上表現出非常高的非線性動力學(Breivik, 2010)，如科學研究、環境任務、海洋資源勘探、軍事用途和其它應用開發廣泛的具體應用，詳如表 2.1 所示。

表 2.1 USV 的應用

類型	USV 具體應用
科學研究	測深測量(Roberts & Sutton, 2006)；海洋生物影像以及生態系統的遷移和變化(Goudey et al., 1998)
環境任務	環境監測，抽樣和評估(Caccia et al., 2005; Naeem et al., 2008; Rasal, 2013; Svec et al., 2014b)；災難（如海嘯，颶風，潛艇火山爆發）輔助預測和管理以及緊急應對(Murphy et al., 2008)
海洋資源勘探	石油，天然氣和礦山勘探(Pastore & Djapic, 2010; Roberts & Sutton, 2006)；海上平台/管道建設和維護( Bertram, 2008; Breivik et al., 2008)
軍事用途	港口和沿海監視，偵察和巡邏(Caccia et al., 2007; Kucik, 2004; Pastore & Djapic, 2010; Svec & Gupta, 2012)；搜索和救援(Murphy et al., 2008; Roberts & Sutton, 2006)

【資料來源：Liu et al. (2016)】

UAV 可以透過自動執行觀察或檢測任務遙控，主要用於測繪應用，環境變化監測，防災應對，資源勘查等。與其它飛行器相比衛星遙感技術，UAV 在拍攝航空照片時有兩個優點：成本低，流動性高(Arfaoui, 2017)。

表 2.2 UAV 的應用

類型	UAV 具體應用
監控和安全應用	龍捲風事件風暴的損害評估(DeBusk, 2010);監視行動還包括檢查和監測邊界和海岸線。一個小型UAV(簡稱sUAV)可以沿著一個地區部署，為海運提供有用的數據攔截/攔截行動(Healey et al., 2007);各種環境觀測資料(Bland et al., 2007)
林業和農業	檢測森林火災(Casbeer et al., 2006);監視非法森林開採(Huang et al., 2010);在許多熱帶國家/地區獲得無法進入或具有挑戰性的地區(Huang et al., 2010);管理農業生產和物資保護(Horcher & Visser 2004);估計作物生長和食物質量(Herwitz, 2004);執行空中農藥噴灑，以減少人為干預(Miller, 2005);葡萄栽培(Turner et al., 2011)
城市規劃，建築，土木工程與採礦	建築檢測(He et al., 2012);交通監控調查(Puri, 2005)

【資料來源：Arfaoui (2017)】

## 2.2 最短覆蓋路徑之相關文獻

最短覆蓋路徑問題(Shortest Coverage Path Problem, SCPP)主要是尋找從起始點到終點最短一條覆蓋的路徑，路徑經過的地方必須覆蓋所有該被覆蓋的所有節點，藉由相關文獻整理，將針對如下幾種相關覆蓋路徑規劃問題進行探討。

### 2.2.1 最短覆蓋路徑問題(SCPP)

在一個網路裡，先定義行經每個節點或節線是否能覆蓋（即 cover）哪些需求點（不見得剛好座落在網路的節點或節線上），通常可用需求點是否在某節點或節線的「覆蓋距離」內來判斷。SCPP 就是要計算一個最短（或最小成本）的路徑，而該路

徑又必須能覆蓋所有必須被覆蓋的需求點；最早由 Current et al. (1984) 提出，同時整合設施區位選址與最短路徑之問題，並將最短路徑覆蓋問題應用在網路設計之問題。最短路徑覆蓋問題將決策問題定義為預先選定起迄點，然後找尋最短路徑連接起迄點，並且使連接的路線能夠服務到所有的需求點。

Current et al. (1985) 將覆蓋(Coverage) 概念作延伸，從基本的設施區位選址問題與道路網路設計問題，提出覆蓋範圍最大之最短路徑 (Maximum Covering Shortest Path, MCSP) 與服務最多人口之最短路徑 (Maximum Population Shortest Path, MPSP) 兩種問題，並設計兩問題之網路規劃模式，模式中同時將服務範圍最大化以及路徑最短之兩個目標做整合，發展多目標規劃模式，利用權重法求解一假想簡例尋求非劣解集。

Current and Pirkul (1994) 持續對最短覆蓋路徑問題進行研究，著重於有效率的求解方法，因此提出啟發式演算法以及精確求解過程處理最短路徑覆蓋問題，其方法可以更有效地應用在大規模的實例問題上。

Boffey and Narula (1998) 以路徑覆蓋問題為基礎，內容包含服務人口最大化之最短路徑、服務範圍距離以及多路徑等問題，進而發展出多路徑覆蓋問題(Multi-path Covering-routing Problems)，文中介紹2-MPSP (Maximal Population Shortest Path, MPSP) 為尋找兩條服務範圍最大化之最短路徑，設計2-MPSP 規劃模型，透過權重法作目標權重之調整，利用拉氏鬆弛法(Lagrangian Relaxation)求解。

Niblett and Church (2016) 針對Current et al. (1984)所提出的最短覆蓋路徑問題已假定最短覆蓋路徑是不能循環或折返的部分，提出在不同的網路佈局，循環路徑設計可能比不循環路徑規劃更優，這取決於網路佈局的情況，這對於開發路線設計模型這項工作有很多的助益。

舉例來說，如圖2.1所示，其最短覆蓋路徑包含了節點和節線的覆蓋範圍，假設有A, B, C三個需求點(圖2.2)， $Z_0$ 表示為起始點所以節點0為起始點，依需求點A，可覆蓋A的節點有0、1、4，可覆蓋A的節線範圍(1,4)、(4,1)、(1,0)、(0,1)；而需求點B可覆

蓋的節點有3、5、8，可覆蓋的節線範圍(5,3)、(3,5)、(5,8)、(8,5)；需求點C可覆蓋的節點有4、7、8，可覆蓋的節線範圍(4,7)、(7,4)、(7,8)、(8,7) 可以服務到，其路線規劃為0→1→4→7→8→5→0。

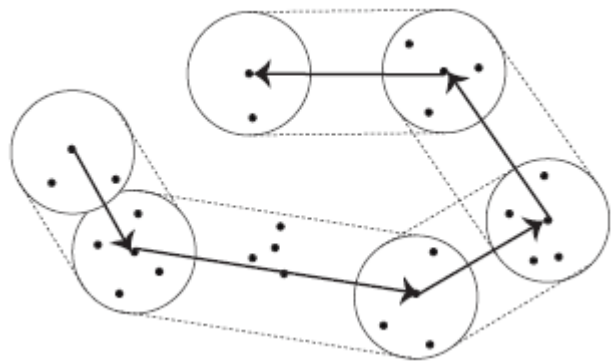


圖 2.1 最短覆蓋路徑問題路徑規劃

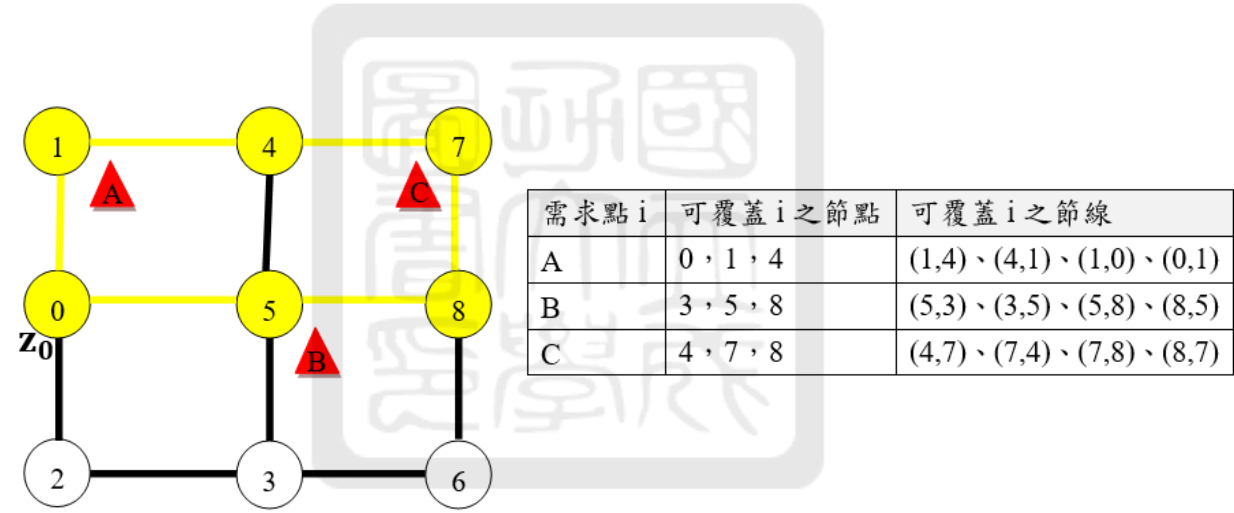


圖2.2 可覆蓋各個需求點之節點與節線集合

提供最短覆蓋路徑的整數規劃模型如下：

**符號定義**

- $N$ ：網路的節點之集合，網路編號  $i, j = 1 \dots |N|$
- $A$ ：網路經過的路徑的集合， $(i, j) \in A$
- $R$ ：必須覆蓋的區域所構成的集合， $r = 1 \dots |R|$

$b_{ir}$  :  $r$  是否在節點  $i$  所覆蓋的區域內，是則為1，反之為0

$C_{ij}$  : 節線  $(i, j)$  所花費的旅行成本

$X_{ij}$  : 路徑是否經過節線  $(i, j)$ ，是則為1，反之為0

$Z_i$  : 是否會經過  $i$ ，1 代表是，反之為0

$T_i$  : 抵達節點  $i$  的時刻

$M$  : 極大值

### 數學模式

$$\text{Minimize} \quad \sum_{(i,j) \in A} C_{ij} X_{ij} \quad (2.1.1)$$

Subject to

$$\sum_{(i,j) \in A} X_{ij} = Z_i \quad \forall i \in N \quad (2.1.2)$$

$$\sum_{(j,i) \in A} X_{ji} = \sum_{(i,j) \in A} X_{ij} \quad \forall i \in N \quad (2.1.3)$$

$$Z_0 = 1 \quad (2.1.4)$$

$$T_j \geq T_i + C_{ij} - M(1 - X_{ij}) \quad \forall i, j \in N \mid (i, j) \in A \quad j \neq 0 \quad (2.1.5)$$

$$\sum_{i \in N} b_{ir} Z_i \geq 1 \quad \forall r \in R \quad (2.1.6)$$

$$T_j \geq 0 \quad \forall j \in N \quad (2.1.7)$$

$$X_{ij} \in \{0,1\} \quad \forall (i, j) \in A$$

$$Z_i \in \{0,1\} \quad \forall i \in N$$

### 2.2.2 覆蓋銷售員問題(CSP)

「覆蓋銷售員問題」(Covering Salesman Problem, CSP)可被視為是一個特殊的

SCPP，該問題特別將必須被覆蓋的需求點集合定義為所有的節點，由於覆蓋距離不見得為0，因此CSP不見得要經過所有的節點。Current & Schilling (1989)將CSP介紹為「旅行銷售員問題」(Traveling Salesman Problem, TSP)的延伸。因為如果最大覆蓋距離為零的話(亦即行徑之節點僅能覆蓋自己)，則CSP即等同於TSP。由於TSP是NP-Hard(Garey & Johnson, 1979)，所以CSP和SCPP皆為NP-Hard。Current & Schilling (1989)提出了一種簡單啟發式演算法以求解CSP，首先求解一個集合覆蓋問題(Set Covering Problem)，然後使用集合覆蓋解決方案中的節點求解TCP。Golden et al. (2012)提出了一個延伸的CSP版本，要求每個節點 $i$ 至少必須被覆蓋 $k_i$ 次，而行徑到的每個節點相關成本，提出二種以鄰近搜索為基礎的啟發式演算法來求解，假設一個節點可以被行徑不止一次。Salari & Naji-Azimi (2012)提出了一個鄰近搜索為基礎的啟發式演算法，假設各節點可覆蓋其最近的7個節點，測試了多達783個節點的實例。

Shaelaie et al. (2014)提出了廣義的兩種替代數學公式以探討人道物流方面的議題，其節點包括倉庫、設施和客戶；目標則是在一小部分設施上進行可至少覆蓋 $P$ 客戶端的最短距離巡視路徑，並設計了兩個啟發式演算法：改良式基因模擬法、變動鄰域搜索法。Salari, et al. (2014)則針對CSP提出結合蟻群演算法和動態規劃之混合演算法。

如圖2.3所示，CSP必須覆蓋所有的節點，但不見得要經過所有的節點。假設有0~8的需求點(圖2.4)， $Z_0$ 表示為起始點，依第0個需求點為例，該點為起始點且只能被第0個節點服務到；第1個需求點亦只能被第1個節點服務到；而第2個需求點則能被第1、2個節點服務到；第3個需求點只能被第3個節點服務到；第4個需求點只能被第4個節點服務到；第5個需求點只有第4、5個節點可以服務到；第6個需求點只有第6個節點可以服務到；第7個需求點只有第7個節點可以服務到；第8個需求點只有第7、8個節點可以服務到，其路線規劃為 $0 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow 0$ ，如圖2.4。



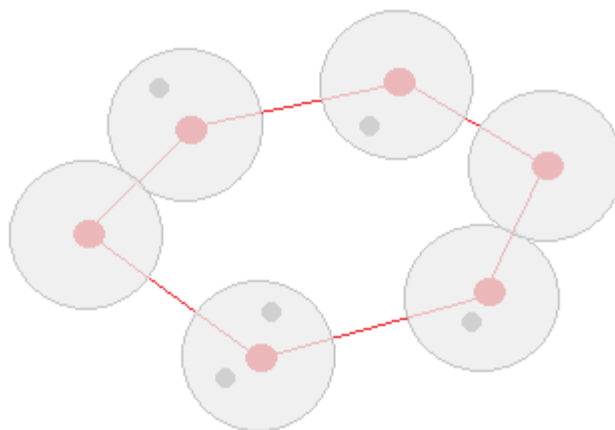


圖 2.3 覆蓋銷售員問題路徑規劃

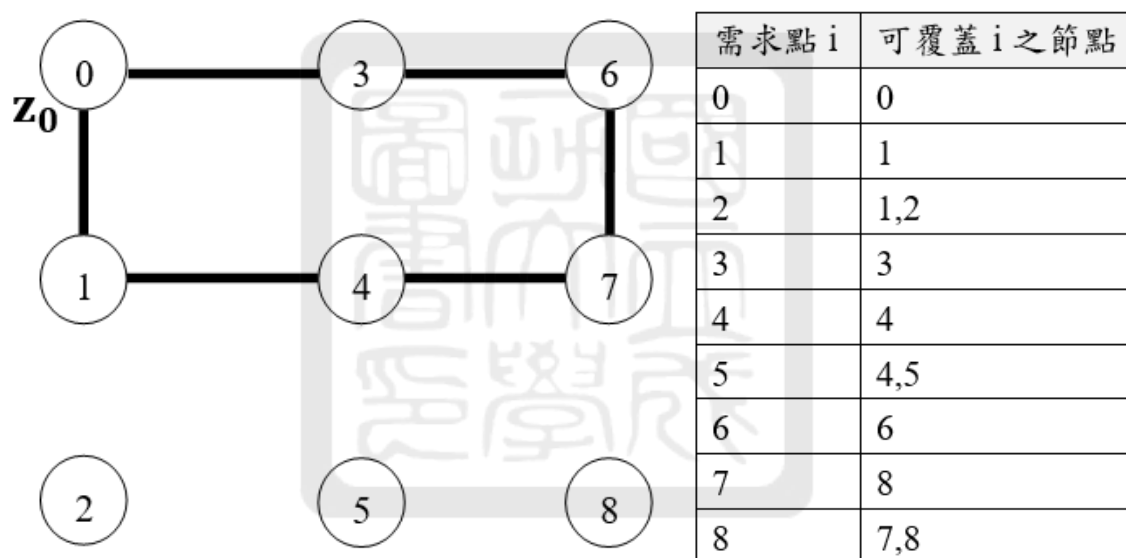


圖 2.4 CSP 可覆蓋各個需求點之節點和路徑規劃

提供CSP的整數規劃模型如下：

**符號定義**

$N$ ：網路的節點之集合，網路編號  $i, j = 1, \dots, |N|$

$A$ ：網路經過的路徑的集合， $(i, j) \in A$

$R$ ：必須覆蓋的區域所構成的集合， $r = 1, \dots, |R|$

$b_{ir}$  :  $r$  是否在節點  $i$  所覆蓋的區域內，是則為1，反之為0

$C_{ij}$  : 節線  $(i, j)$  所花費的旅行成本

$X_{ij}$  : 路徑是否經過節線  $(i, j)$ ，是則為1，反之為0

$Z_i$  : 是否會經過  $i$ ，1 代表是，反之為0

$T_i$  : 抵達節點  $i$  的時刻

$M$  : 極大值

### 數學模式

$$\text{Minimize} \quad \sum_{(i,j) \in A} C_{ij} X_{ij} \quad (2.2.1)$$

Subject to

$$\sum_{(i,j) \in A} X_{ij} = Z_i \quad \forall i \in N \quad (2.2.2)$$

$$\sum_{(j,i) \in A} X_{ji} = \sum_{(i,j) \in A} X_{ij} \quad \forall i \in N \quad (2.2.3)$$

$$Z_0 = 1 \quad (2.2.4)$$

$$T_j \geq T_i + C_{ij} - M(1 - X_{ij}) \quad \forall i, j \in N \mid (i, j) \in A \quad j \neq 0. \quad (2.2.5)$$

$$\sum_{i \in N} b_{ir} Z_i \geq 1 \quad \forall r \in N \quad (2.2.6)$$

$$T_j \geq 0 \quad \forall j \in N \quad (2.2.7)$$

$$X_{ij} \in \{0,1\} \quad \forall (i, j) \in A$$

$$Z_i \in \{0,1\} \quad \forall i \in N$$

### 2.2.3 覆蓋旅遊問題 (CTP)

覆蓋旅遊問題(Covering Tour Problem，CTP) (Gendreau et al., 1997)和 CSP 有相同

的目標：尋找出一個覆蓋路徑的最小總長度，但是 CSP 的路線必須繞一圈（即有相同的起訖點）且一定要經過一組必須被行經的節點集合。因此，如果必須行經的節點集合是空集合（亦即不需一定要經過除起訖點外之某些特別節點），則 CTP 將變成 CSP；如果必須行經的節點集合包含網路的所有節點的集合，那麼 CTP 就會被轉換為 TSP。Tan & Hill (2013) 提出應用於 CTP 的多個無人駕駛地面車輛 (UGV) 的路徑產生方式，用於監控和保護重要設施；其它應用包括求解回收玻璃的區域位置路徑問題 (Rahim et al., 2014)、災害人道救援物流 (Hodgson et al., 1998; Naji-Azimi et al., 2012)、城市和地理邊界的巡邏 (Oliveira et al., 2015; Saricicek & Akkus, 2015)；Hachicha et al. (2000) 研究 m-CTP，並提出一個數學模式和三個啟發式演算法，應用在移動醫療保健服務隊的移動路徑設計。

因此CTP必須覆蓋所有的節點，且必須經過某些特定的節點。以圖2.5為例，假設有0~8的需求點，可覆蓋各需求點之節點集合如圖右側列表，其中節點2必須被經過，則一個可行之路線規劃為0→3→6→7→4→5→2→1→0。

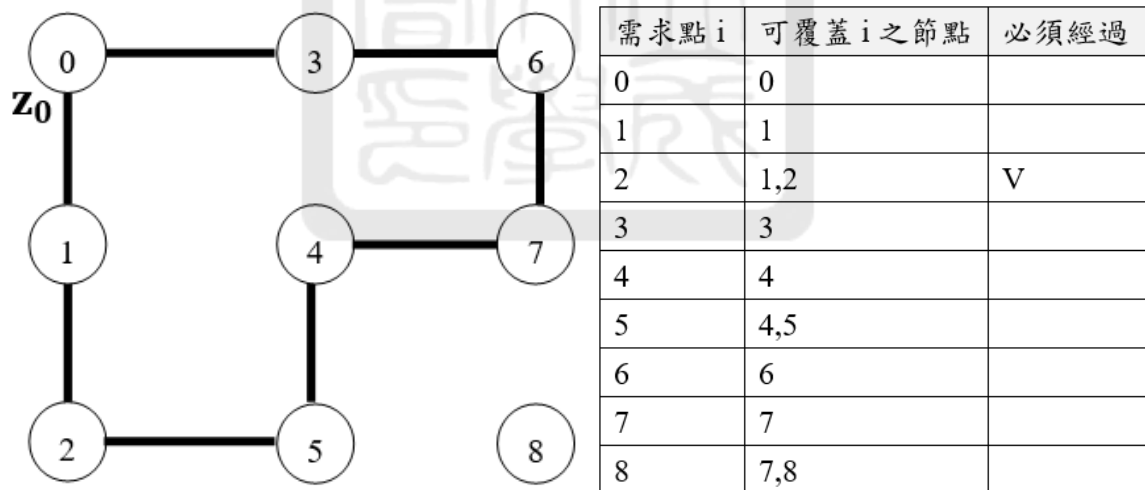


圖 2.5 CTP 可覆蓋各個需求點之節點和路徑規劃

提供CTP的整數規劃模型如下：

**符號定義**

同CSP

數學模式

$$\text{Minimize} \quad \sum_{(i,j) \in A} C_{ij} X_{ij} \quad (2.3.1)$$

Subject to

$$\sum_{(i,j) \in A} X_{ij} = Z_i \quad \forall i \in N \quad (2.3.2)$$

$$\sum_{(j,i) \in A} X_{ji} = \sum_{(i,j) \in A} X_{ij} \quad \forall i \in N \quad (2.3.3)$$

$$Z_0 = 1 \quad (2.3.4)$$

$$T_j \geq T_i + C_{ij} - M(1 - X_{ij}) \quad \forall i, j \in N \mid (i, j) \in A \quad j \neq 0 \quad (2.3.5)$$

$$\sum_{i \in N} b_{ir} Z_i \geq 1 \quad \forall r \in N \quad (2.3.6)$$

$$T_j \geq 0 \quad \forall j \in N \quad (2.3.7)$$

$$Z_i = 1 \quad \forall i \in \bar{N} \quad (2.3.8)$$

$$X_{ij} \in \{0,1\} \quad \forall (i,j) \in A$$

$$Z_i \in \{0,1\} \quad \forall i \in N$$

## 2.3 小結

本章首先說明無人載具 USV 和 UAV 與覆蓋路徑相關的應用，接著介紹與最短覆蓋路徑相關的 SCPP、CSP、CTP 等相關的研究和文獻回顧，在相關的研究和文獻中了解 SCPP 為尋找從起始點到終點最短覆蓋的路徑，路徑經過的地方必須覆蓋所有該被覆蓋的所有節點其包含節點和節線的覆蓋範圍；CTP 則以必須覆蓋所有的節點，但不見得要經過所有的節點的路徑規畫問題；CTP 必須覆蓋所有的節點，且必須經過某些特定的節點。進而得知本研究的問題比較接近 SCPP，因此會採用 SCPP 來解決在考慮風力影響下，如何規劃單台無人船在整個空間的行徑路線的問題。

### 第三章 最短覆蓋路徑規劃問題之整數規劃模型

本章節將說明本研究所需使用的方法。第一節將描述問題與說明本研究的基本假設；第二節說明單期風力單台無人船整數規劃模型，第三節說明多期風力單台無人船整數規劃模型 M1，第四節說明以 M1 為基礎但路線可重複之路徑規劃數學模式 M2。

#### 3.1 問題描述

本研究將針對無人船在搜尋一個空間範圍內，必須覆蓋數個指定的地點，並考量外在環境風力在不同時段的影響下，規劃出最短覆蓋路徑；我們將透過數學規劃的方法，以整數規劃模型求解出最短覆蓋路徑。

本研究的基本假設有下列幾項：

- 在選定的水域內，已知必須被覆蓋的節點或節線集合
- 定義指定的節點為起點
- 定義每個節點和節線之覆蓋範圍
- 定義每段節線之行經時間
- 定義每段節線之能源耗損量
- 依每個時段的風力或風向，假設其對節線之行經時間與能源耗損之影響為已知

#### 3.2 單期風力單台無人船整數規劃模型

本節內容將先針對單期風力單台無人船以整數規劃模型求解出最短覆蓋路徑，如圖 3.1 所示，單期風力意指無論在哪個時間點，每段節線之行經時間都不會改變的情況下，在此水域內已知起點為節點 1，必須被搜尋的地點 A 可被節點 6 和節線範圍 (3,6)、(6,3)、(5,6)、(6,5)、(6,9)、(9,6) 所覆蓋，必須被搜尋的地點 B 可被節點 7 和節線範圍 (4,7)、(7,4)、(7,8)、(8,7) 所覆蓋，每段節線之行經時間為已知，其單期風力單台無人船整數規劃模型的目標函式與限制式，如公式 3.1.1 至公式 3.1.7 所示。

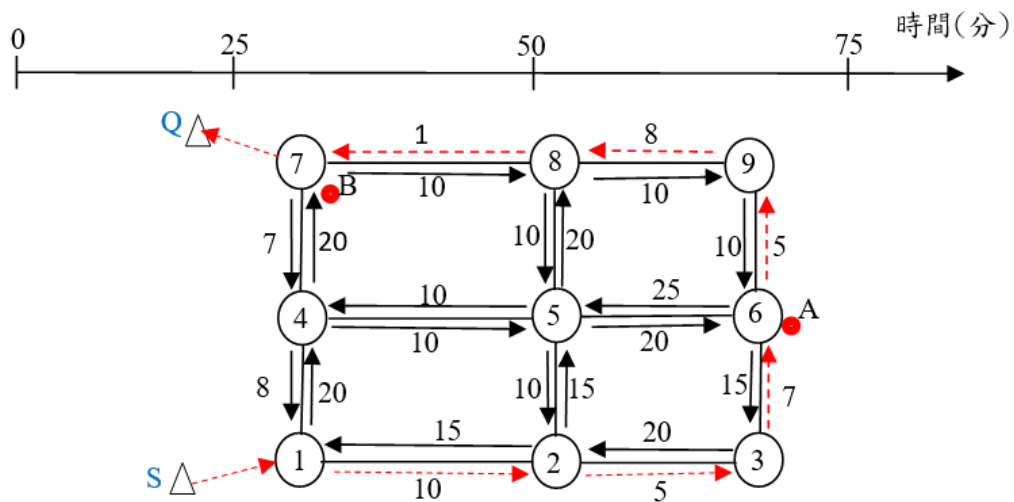


圖 3.1 單期風力單台無人船路徑規劃

符號定義

同CSP

數學模式

$$\text{Minimize} \quad \sum_{(i,j) \in A} C_{ij} X_{ij} \quad (3.1.1)$$

Subject to

$$\sum_{(i,j) \in A} X_{ij} = Z_i \quad \forall i \in N \quad (3.1.2)$$

$$\sum_{(j,i) \in A} X_{ji} = \sum_{(i,j) \in A} X_{ij} \quad \forall i \in N \quad (3.1.3)$$

$$Z_0 = 1 \quad (3.1.4)$$

$$T_j \geq T_i + C_{ij} - M(1 - X_{ij}) \quad \forall i, j \in N \mid (i, j) \in A \quad j \neq 0 \quad (3.1.5)$$

$$\sum_{i \in N} b_{ir} Z_i \geq 1 \quad \forall r \in R \quad (3.1.6)$$

$$T_j \geq 1 \quad \forall j \in N \quad (3.1.7)$$

$$X_{ij} \in \{0,1\} \quad \forall (i, j) \in A$$

$$Z_i \in \{0,1\} \quad \forall i \in N$$

其中公式 3.1.1 為整數規劃模型的目標函數，指必須完成搜尋的地點總路徑花費最小成本；限制式公式 3.1.2 表示針對每一個節點 $i$ 而言，其行經之次數必須等於 $Z_i$ ，若其值為 1 則表示經過，反之為 0；公式 3.1.3 表示迴圈之流量守恒限制式，亦即進入每一個節點 $i$ 的總次數等於其流出的總次數；公式 3.1.4 表示 $Z_0$ 為起點需經過；公式 3.1.5 使用 $T_i$ 來避免發生獨立子迴圈 subtour（若存在 subtour，將此公式沿 subtour 的所有節點走一圈，即可發現該公式將會變成不可行）；公式 3.1.6 表示指定必須要搜尋的地點都必定要被覆蓋到。

### 3.3 多期風力單台無人船整數規劃模型 M1

本節將介紹如何以整數規劃模型求解出「多期風力單台無人船之最短覆蓋路徑」，首先可從圖 3.2 所示，在不同的時段因受外在風力的影響下，其同一路徑所花費的時間會不同，然而在行經過程中如果時間已到達下一時段時間，下一個開始節點需要用當下新的時段區間來計算最短覆蓋路徑，其目標函式與限制式，如公式 3.2.1 至公式 3.2.18 所示。

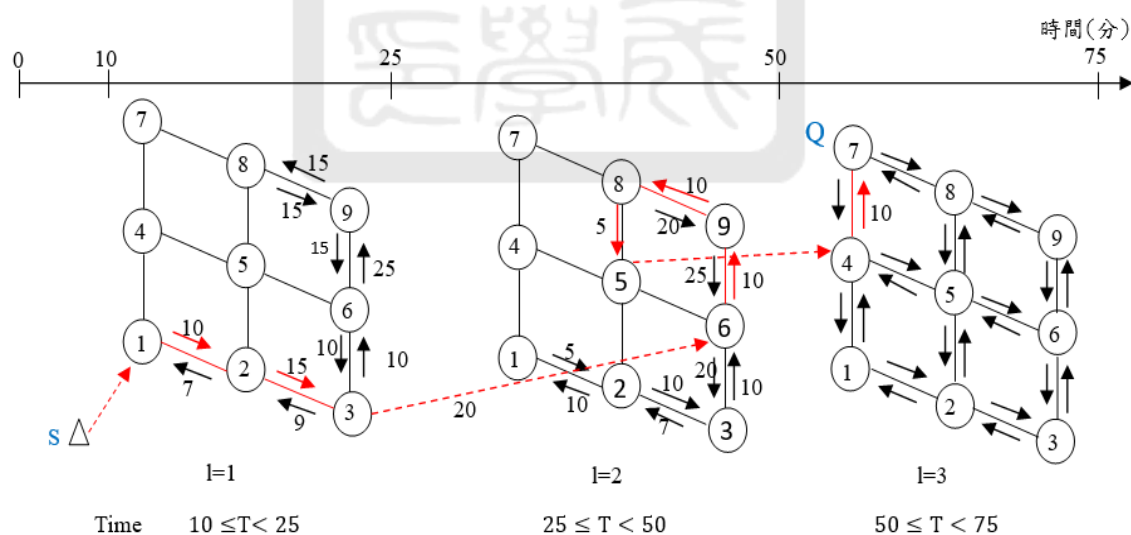


圖 3.2 多期風力單台無人船路徑規劃

#### 符號定義

$N$  : 網路的節點之集合  $i, j = 1 \dots |N|$

$A$  : 網路經過的路徑的集合,  $(i, j) \in A$

$L$  : 虛擬層的集合  $l = 1 \dots |L|$

$S$  : 虛擬的起點

$U$  : 起點的節點集合

$Q$  : 虛擬的目的地

$V$  : 目的地的節點集合

$T_{\max}^l$  : 在第  $l$  層的最大時限

$X_{ij}^l$  : 在同一層第  $l$  層的節線  $(i, j)$  是否被經過, 1 代表是, 反之為 0

$Y_{ij}^l$  : 在跨層第  $l$  層的節線  $i$  到第  $l-1$  層的  $j$  是否被經過, 1 代表是, 反之為 0

$C_{ij}^l$  : 在第  $l$  層的節線  $(i, j)$  的行經路徑時間成本

$Z_i^l$  : 在第  $l$  層是否會經過  $i$ , 1 代表是, 反之為 0

$T_i^l$  : 在第  $l$  層抵達節點  $i$  的時刻

$\varepsilon$  : 極小值

其餘同 CSP

## 數學模式 M1

$$\text{Minimize } \sum_{l=1}^L \sum_{(i,j) \in A} C_{ij}^l X_{ij}^l + \sum_{l=1}^{L-1} \sum_{(i,j) \in A} C_{ij}^l Y_{ij}^l$$

(3.2.1)

Subject to

- For layer 1 ( $l = 1$ ):

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l \quad \forall i \in N - U - V \quad l = 1 \quad (3.2.2)$$



$$Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l \quad \forall i \in V \quad l=1 \quad (3.2.3)$$

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l + Y_{si}^0 \quad \forall i \in U \quad l=1 \quad (3.2.4)$$

- For other layer ( $l = 2 \sim L$ , and  $Y_{ij}^L = 0$ ):

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} (X_{ki}^l + Y_{ki}^{l-1}) \quad \forall i \in N-V \quad l=2,3,4,\dots,L. \quad (3.2.5)$$

$$Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} (X_{ki}^l + Y_{ki}^{l-1}) \quad \forall i \in V \quad l=2,3,4,\dots,L \quad (3.2.6)$$

- For source node :

$$\sum_{(S,i) \in A} Y_{Si}^0 = 1 \quad (3.2.7)$$

- For destination :

$$\sum_{l=1}^L \sum_{(i,Q) \in A} Y_{iQ}^l = 1 \quad (3.2.8)$$

- For Z variable ( $l=1 \sim L$ , and  $Y_{ij}^L = 0$ ):

$$Z_i^l = Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \quad \forall i \in N-V \quad l=1,2,3,\dots,L \quad (3.2.9)$$

$$Z_i^l = Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \quad \forall i \in V \quad l=1,2,3,\dots,L \quad (3.2.10)$$

$$MZ_i^l \geq \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \geq \varepsilon Z_i^l \quad \forall i \in N-V \quad l=1,2,3,\dots,L \quad (3.2.9-A)$$

$$MZ_i^l \geq Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \geq \varepsilon Z_i^l \quad \forall i \in N-V \quad l=1,2,3,\dots,L \quad (3.2.10-A)$$

$$Y_{ij}^L = 0 \quad \forall (i,j) \in A \quad (3.2.11)$$

(時間限制式)

- IF  $X_{ij}^l = 1$  then  $T_j^l = (T_i^l + C_{ij}^l)$

$$(T_i^l + C_{ij}^l) - M(1 - X_{ij}^l) \leq T_j^l \leq (T_i^l + C_{ij}^l) + M(1 - X_{ij}^l) \quad \forall l \in L \quad \forall i \in N \quad (3.2.12)$$

- IF  $Y_{ij}^l = 1$  then  $T_j^{l+1} = (T_i^l + C_{ij}^l)$

$$(T_i^l + C_{ij}^l) - M(1 - Y_{ij}^l) \leq T_j^{l+1} \leq (T_i^l + C_{ij}^l) + M(1 - Y_{ij}^l) \quad \forall l \in L - \{1\} \quad \forall i \in N \quad (3.2.13)$$

- IF  $Z_i^l = 1$  then  $T_{\max}^{l+1} \leq T_i^l \leq T_{\max}^l$   
 $T_{\max}^{l-1} \times Z_i^l \leq T_i^l \leq T_{\max}^l \times Z_i^l \quad \forall l \in L(l=1,2,3\ldots,L) \quad \forall i \in N$  (3.2.14)

- IF  $Y_{sj}^0 = 1$  then  $T_j^1 = 0$   
 $-M(1-Y_{sj}^0) \leq T_j^1 \leq M(1-Y_{sj}^0) \quad \forall j \in U$  (3.2.15)

(覆蓋限制式)

$$\sum_{l=1}^L \sum_{i \in N} b_{ir} Z_i^l \geq 1 \quad \forall r \in N \quad (3.2.16)$$

$$b_{ir} = \begin{cases} 1 : \text{if node } i \text{ covers area } r \\ 0 : \text{o.w.} \end{cases}$$

(其它限制式)

$$T_{\max}^0 = 0, T_{\max}^1 = 60, T_{\max}^2 = 120, T_{\max}^3 = 180 \quad (3.2.17)$$

$$X_{ij}^l = \{1, 0\}, Y_{ij}^l = \{1, 0\}, Z_i^l = \{1, 0\} \quad (3.2.18)$$

其中公式 3.2.1 為整數規劃模型的目標函數，旨在水域內搜尋必須巡邏的地點都完成所花費總路徑時間最小化；我們將限制式區分為第一層和其它層以個別說明之：其中公式 3.2.2 至 3.2.4 為第一層的限制式；依據流量守恒限制式，公式 3.2.2 表示當在第一層時，對於每個  $i$  到  $j$  的節點必須排除起點  $U$  和終點  $V$  所有的集合的節點，因第一層中沒有跨層流入節點，所以在同一層從  $i$  流出加上跨層從  $i$  流出的等於其同一層流入的總次數；因我們不知道開始起點是那個點，所以我們使用虛擬起點  $S$ ；因不知道最後會停在那一層，所以我們使用虛擬目的地  $Q$ 。公式 3.2.3 表示當在第一層節點  $i$  為終點時，同一層從  $i$  出去加上跨層從  $i$  流出的總次數等於其同一層流入的總次數加上  $i$  為終點跨層虛擬  $Q$  的總次數；公式 3.2.4 表示如果節點  $i$  等於起點  $Q$  時，即跨層從  $i$  流出等於同一層流入  $i$  的總次數加上跨層從  $i$  流出的總次數等於其同一層流入的總次數加上跨層虛擬起點  $S$  流入的總次數；而公式 3.2.5 至 3.2.6 為其它並包含最後一層的限制式，依據流量守恒限制式，公式 3.2.5 表示當在非第一層時對於每個  $i$  到  $j$  的節點必須排除終點  $V$  的節點，亦即在同一層從  $i$  出去加上跨層從  $i$  流出的等於同一層從  $i$  流入加上跨層從  $i$  流入的總次數，公式 3.2.6 表示如果節點  $i$  等於目的地  $V$  時，即跨層從  $Q$

流出等於同一層流入 $i$ 加上跨層流入 $i$ 的總次數；對於第一層中的起始點而言，限制式 3.2.7 強制該虛擬起始點連接可能連接的節點，作為真實起始點；同理，對於目的地的而言，限制式 3.2.8 限制從每一個層節點 $i$ 到目的地 $Q$ 只能有一次；公式 3.2.9 及 3.2.10 透過 $Z_i'$ 以確認在每一層中是否有經過該節點 $i$ ；而公式 3.2.9-A 與 3.2.10-A 則是應用於可以重複經過該節點 $i$ 的情況；公式 3.2.12 至 3.2.15 為時間限制式：公式 3.2.12 表示在同一層中到達節點 $i$ 的時間加上該層的行徑花費的時間必須等於到達節點 $j$ 的時間；公式 3.2.13 表示在跨層中到達節點 $i$ 的時間加上跨層的行徑花費的時間必須等於到達節點 $j$ 的時間；公式 3.2.14 確保節點到達時刻必須在該層的時間間隔內，這三個限制式可以避免發生獨立子迴圈 subtour（若存在 subtour，將此公式沿 subtour 的所有節點走一圈，即可發現該公式將會變成不可行）；當累計行徑時間長度超過該層的最大時間長度時，路徑必須連到下一層，公式 3.2.15 為時間約束，須確保在第一層時節點 $j$ 當作起點需等於開始的時刻；公式 3.2.16 為覆蓋限制式表示 $R$ 是必須巡邏的區域 $r$ 所構成之集合，因此指定必須要搜尋的地點都必須要被覆蓋到；公式 3.2.17 設定時間區間；公式 3.2.18 為變數之定義域。

## 3.4 以 M1 為基礎但路線可重複之路徑規劃數學模式 M2

在透過整數規劃模型和演算法求解多期風力單台無人船的最短路徑的過程中，發現如圖 3.4 將水域做等距定位出平面縱橫雙向的整齊藍色格點，紅色點為需要巡邏的地點，使用整數規劃模型求解最短行經路徑從  $0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 21 \rightarrow 16 \rightarrow 11 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 3 \rightarrow 4$  花費最小時間成本為 46.5(如表 3.1)；然而，若使用演算法求解，反而發現最短行經路徑從  $0 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 15 \rightarrow 16 \rightarrow 11 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 3 \rightarrow 4$ ，花費最小時間成本為 46.2，竟然比整數規劃模型之解更佳。再進一步畫圖確認，可由圖 3.4 看出藍色箭頭表示整數規劃模型執行路徑，橘色箭頭表示演算法執行路徑，可發現主要是因為演算法從  $15 \rightarrow 20 \rightarrow 15 \rightarrow 16$  可以重覆經過節點 15 兩次，但整數規劃模型限制式並無法重覆經過該節點，使得演算法這段的行徑花費的成本反而多節省 0.3。因此我們得知整數

規劃模型必須被調整成可考量路段重複被經過的可能性，以求得最佳解。

表 3.1 案例 A 說明整數規劃模型和演算法最佳解

求解方法	路徑	最小時間成本
整數規劃模型	0→5→10→15→20→21→16→11→6→7→8→3→4	46.5
演算法	0→5→10→15→20→15→16→11→6→7→8→3→4	46.2

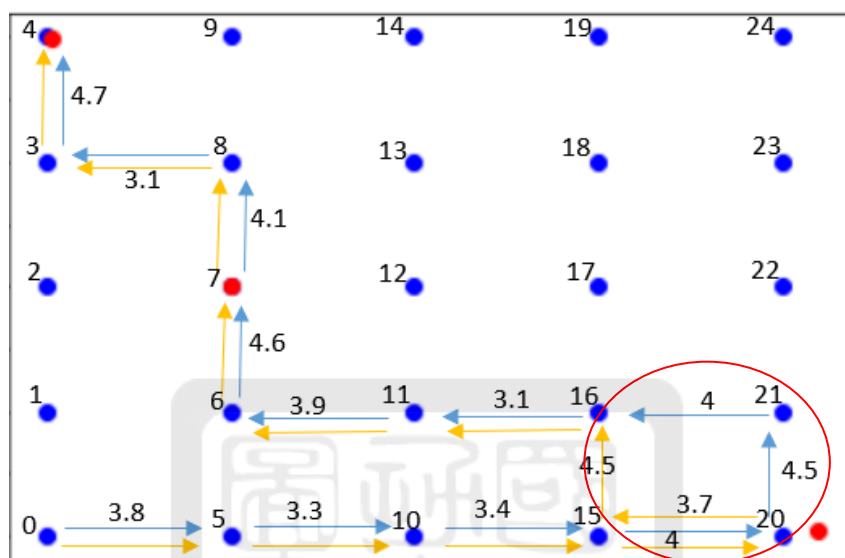


圖 3.3 藍色箭頭：整數規劃模型執行路徑；橘色箭頭：演算法執行路徑

為能讓整數規劃模型路徑達到可以重覆經過，將每個點各自依據可重複經過的次數增加對應數量的虛擬點並建立新的網路圖，起點為編號 0 的點須產生 1 個虛擬點，而編號為 1、2、3 則各自須產生兩個虛擬點(如圖 3.5)，原編號 0、1、2、3 的 4 個點各自增加其對應的虛擬點編號 4、5、6、7、8、9、10，並將原  $T$  變數改以  $\pi$  變數代替及增加  $\pi$  變數相關限制式，調整可重覆經過的整數規劃模型的目標函式與限制式，如公式 3.4.1 至公式 3.4.21 所示。

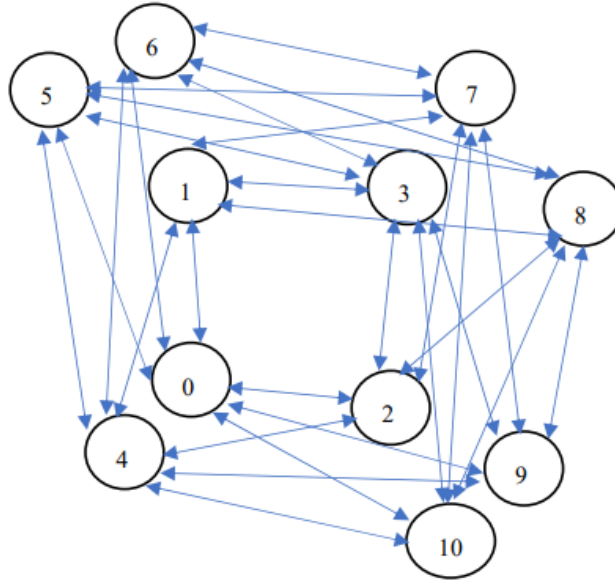


圖 3.4 將每個節點增加其對應的虛擬點

#### 符號定義

$N$  : 網路的節點之集合  $i, j = 1, \dots, |N|$

$A$  : 網路經過的路徑的集合,  $(i, j) \in A$

$L$  : 虛擬層的集合  $l = 1, \dots, |L|$

$S$  : 虛擬的起點

$U$  : 起點的節點集合

$Q$  : 虛擬的目的地

$V$  : 目的地的節點集合

$T_{\max}^l$  : 在第  $l$  層的最大時限

$X_{ij}^l$  : 在同一層第  $l$  層的節線  $(i, j)$  是否被經過, 1 代表是, 反之為 0

$Y_{ij}^l$  : 在跨層第  $l$  層的節線  $i$  到第  $l-1$  層的  $j$  是否被經過, 1 代表是, 反之為 0

$C_{ij}^l$  : 在第  $l$  層的節線  $(i, j)$  的行經路徑時間成本

$Z_i^l$  : 在第  $l$  層是否會經過  $i$ , 1 代表是, 反之為 0

$\pi_i^l$  : 在第  $l$  層的抵達節點  $i$  的時刻

其餘同CSP

## 數學模式 M2

$$\text{Minimize } \sum_{l=1}^L \sum_{(i,j) \in A} C_{ij}^l X_{ij}^l + \sum_{l=1}^{L-1} \sum_{(i,j) \in A} C_{ij}^l Y_{ij}^l \quad (3.4.1)$$

Subject to

- For layer 1 ( $l = 1$ ):

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l \quad \forall i \in N - U - V \quad l = 1 \quad (3.4.2)$$

$$Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l \quad \forall i \in V \quad l = 1 \quad (3.4.3)$$

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} X_{ki}^l + Y_{si}^0 \quad \forall i \in U \quad l = 1 \quad (3.4.4)$$

- For other layer ( $l = 2 \sim L$ , and  $Y_{ij}^L = 0$ ):

$$\sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} (X_{ki}^l + Y_{ki}^{l-1}) \quad \forall i \in N - V \quad l = 2, 3, 4, \dots, L \quad (3.4.5)$$

$$Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) = \sum_{(k,i) \in A} (X_{ki}^l + Y_{ki}^{l-1}) \quad \forall i \in V \quad l = 2, 3, 4, \dots, L \quad (3.4.6)$$

- For source node :

$$\sum_{(S,i) \in A} Y_{Si}^0 = 1$$

(3.4.7)

- For destination :

$$\sum_{l=1}^L \sum_{(i,Q) \in A} Y_{iQ}^l = 1 \quad (3.4.8)$$

- For Z variable ( $l=1 \sim L$ , and  $Y_{ij}^L = 0$ ):

$$Z_i^l = \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \quad \forall i \in N - V \quad l = 1, 2, 3, \dots, L \quad (3.4.9)$$

$$Z_i^l = Y_{iQ}^l + \sum_{(i,j) \in A} (X_{ij}^l + Y_{ij}^l) \quad \forall i \in V \quad l = 1, 2, 3, \dots, L \quad (3.4.10)$$

$$Y_{ij}^L = 0 \quad \forall (i, j) \in A \quad (3.4.11)$$

- For  $\pi$  variable(  $l=1\sim L$ , and  $\pi_s^0 = 0$ ):

$$C_{ij}^l - M(1 - X_{ij}^l) \leq \pi_j^l - \pi_i^l \leq C_{ij}^l + M(1 - X_{ij}^l) \quad \forall l \in L \quad \forall (i, j) \in A \quad (3.4.12)$$

$$-M(1 - Y_{sj}^0) \leq \pi_j^1 - \pi_s^0 \leq M(1 - Y_{sj}^0) \quad \forall l \in L \quad \forall j \in U \quad (3.4.13)$$

$$\pi_i^1 \geq 0 \quad \forall i \in N \quad \forall l \in L \quad (3.4.14)$$

(時間限制式)

- IF  $X_{ij}^l = 1$  then  $\pi_j^l = (\pi_i^l + C_{ij}^l)$

$$(\pi_i^l + C_{ij}^l) - M(1 - X_{ij}^l) \leq \pi_j^l \leq (\pi_i^l + C_{ij}^l) + M(1 - X_{ij}^l) \quad \forall l \in L \quad \forall i \in N \quad (3.4.15)$$

- IF  $Y_{ij}^l = 1$  then  $\pi_j^{l+1} = (\pi_i^l + C_{ij}^l)$

$$(\pi_i^l + C_{ij}^l) - M(1 - Y_{ij}^l) \leq \pi_j^{l+1} \leq (\pi_i^l + C_{ij}^l) + M(1 - Y_{ij}^l) \quad \forall l \in L - \{1\} \quad \forall i \in N \quad (3.4.16)$$

- IF  $Z_i^l = 1$  then  $T_{\max}^{l+1} \leq T_i^l \leq T_{\max}^l$

$$T_{\max}^{l-1} \times Z_i^l \leq \pi_i^l \leq T_{\max}^l \times Z_i^l \quad \forall l \in L(l=1,2,3\ldots,L) \quad \forall i \in N \quad (3.4.17)$$

- IF  $Y_{sj}^0 = 1$  then  $\pi_j^1 = 0$

$$-M(1 - Y_{sj}^0) \leq \pi_j^1 \leq M(1 - Y_{sj}^0) \quad \forall j \in U \quad (3.4.18)$$

(覆蓋限制式)

$$\sum_{l=1}^L \sum_{i \in N} b_{ir} Z_i^l \geq 1 \quad \forall r \in N \quad (3.4.19)$$

$$b_{ir} = \begin{cases} 1 & \text{if node } i \text{ covers area } r \\ 0 & \text{o.w.} \end{cases}$$

(其它限制式)

$$T_{\max}^0 = 0, T_{\max}^1 = 60, T_{\max}^2 = 120, T_{\max}^3 = 180 \quad (3.4.20)$$

$$X_{ij}^l = \{1, 0\}, Y_{ij}^1 = \{1, 0\}, Z_i^l = \{1, 0\} \quad (3.4.21)$$

調整後如表 3.2 所示，改善了原整數規劃模型無法重覆經過的問題，並得到一個最佳解，但比較求解所執行耗用時間由原整數規劃模型 0.56 秒改善至可重覆經過整數規劃模型增加到需要 12.7 秒，比原整數規劃模型花更久的時間，約慢 23 倍時間才

能取得最佳解。

表 3.2 案例 A 整數規劃模型可重覆經過和不可重覆經過與演算法最佳解路徑比較

求解方法	路徑	最小成本	耗用時間(s)
可重覆經過整數規劃模型	0→5→10→15→20→21→16→11→6→7→8→3→4	46.2	12.7
不能重覆經過整數規劃模型	0→5→10→15→20→21→16→11→6→7→8→3→4	46.5	0.56
演算法	0→5→10→15→20→15→16→11→6→7→8→3→4	46.2	0.003

### 3.5 小結

本章介紹三種不同情況之整數規劃模型：單期風力單台無人船、多期風力單台無人船 M1、以 M1 為基礎但路線可重複之路徑規劃數學模式 M2。後續將使用 Gurobi 軟體將本研究提出的基本假設條件以隨機的方式產生相關的測試網路圖，以求解無人船在各水域網路圖進行巡邏時的最短路徑規劃，然而使用整數規劃模型求解時，求解的執行時間會因設定水域網路圖大小或其它設定而增加求解時間，因此為能提升無人船巡邏最短路徑規劃的求解效率，將會在設計出更有效率的求解演算法作為後續實驗測試的比較。



## 第四章 求解演算法

本章將針對多期風力單台無人船整數規劃模型，設計出更有效率的求解演算法，使用兩種演算法分別為貪婪演算法(Greedy Algorithm)和模擬退火法(Simulated Annealing)來設計出更有效率的無人船巡邏最短路徑規劃的求解。本章節將說明本研究提出的六種求解方法，使用貪婪演算法有兩種，分別以代號 G1、G2 表示，模擬退火法有四種，分別以代號 S1、S2、S3、S4 表示，並說明其求解設計概念。

### 4.1 求解演算法 G1

G1 求解演算法的設計概念是先以經典的「戴克斯特拉演算法」(Dijkstra Algorithm)，計算出從某個點到其它全部點的最短路徑，再結合貪婪演算法，每一步選擇中都採取在當前狀態下最好或最佳的路徑規劃。假設有一網路圖，有很多段的時段區間，不同的時間區間每個節點與節點之間行經時間不同，以某一節點來當作無人船行駛的起點，從起點出發到所有其它節點的最短路徑中，分別計算起點到其它節點最短路徑依當前的時間區間所定義行經該路徑總共花費的時間和計算每一個從起點出發到其它點的最短路徑中有覆蓋到必需要巡邏點的個數，再以最短路徑總共花費的時間除以覆蓋到必需要巡邏點的個數取得其最小值即為 CP (Cost-Performance) 值最好，這裡我們選擇 CP 值最好的最短路徑的目的地，作為下一輪的起點，如果行進的過程中有跨不同時間區間，只需使用當起點的時間區間做為行進該路徑的時間，直到所有需要搜尋的點都已搜尋完成，如圖 4.1 G1 演算法步驟說明。

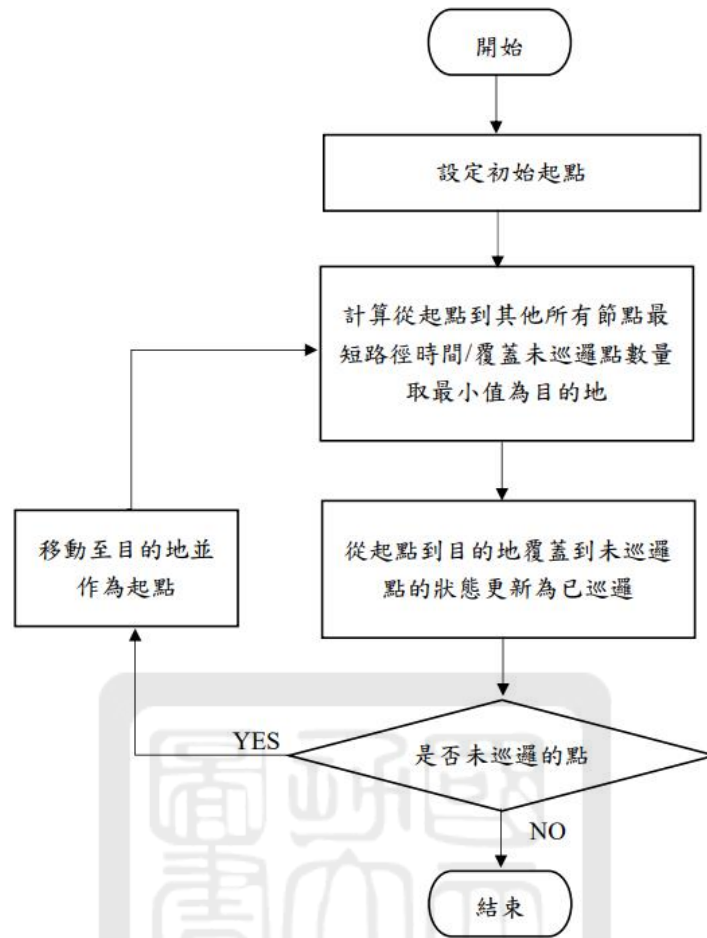


圖 4.1 G1 演算法步驟說明

假設範例說明如下圖 4.2(a)所示，紅色點 A、B、C 為需巡邏的點，這三點分別可被節點 1、4、6 和路徑(1, 4)、(4, 7)、(7, 6) 所覆蓋，假設無人船起點為節點 0 出發，計算從起點節點 0 到其它點的最短路徑時間和可達成巡邏的點數量，取得 CP 值最好的是從起點 0 到節點 4，最短路徑 0→1→4 花費的時間 7.4 可巡邏 2 個點，其 CP 值為 3.7，就沿著路徑走到節點 4 如圖 4.2(b)，節點 4 成為下一個起點如圖 4.2(c)，以起點節點 4 到其它點的最短路徑時間和可達成巡邏的點數量，取得 CP 值最好的是從起點 4 到節點 6，最短路徑 4→7→6 花費的時間 8.2，就沿著路徑走到節點 6 如圖 4.2(d)，最後無需要巡邏的點，即完成無人船巡邏任務。



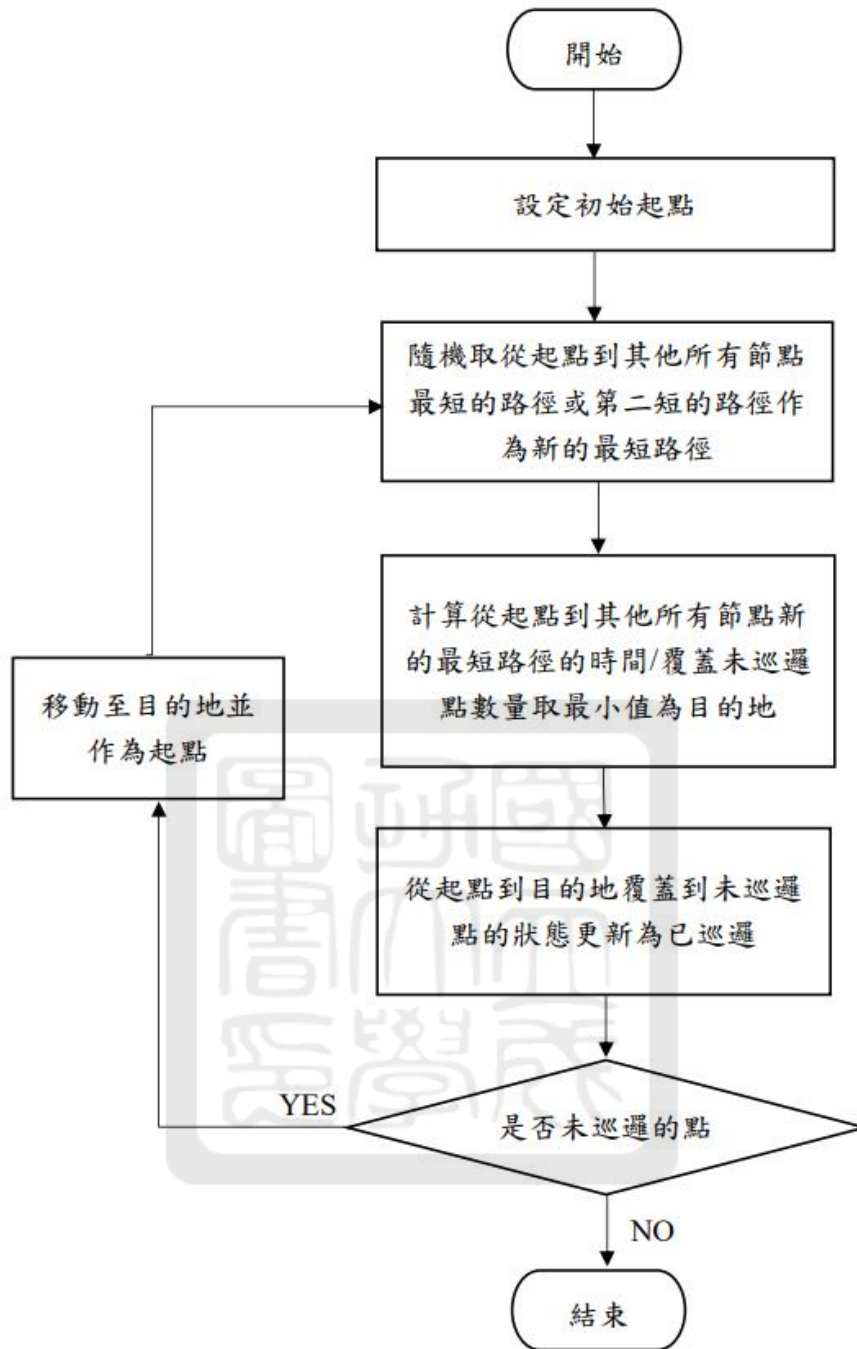


圖 4.3 求解演算法 G2 步驟說明

### 4.3 求解演算法 S1

本研究使用模擬退火法產生鄰近求解的方法是指隨機的互換兩個節點的位置。首先定義初始溫度為  $T$ ，使用上述 G1 求解演算法得出最短路徑節點當作初始解  $X_0$ ，

接著隨機的互換兩個節點，並使用點至點之間的最短路徑獲取一個新的解；再取得新的解與初始解做比較，如果新的解比初始解花費的時間來的少，則接受新的解，並將初始解更新為新的解，如果新的解比初始解花費的時間來的多，則是有機率接受比較差的新的解，再依設定迭代次數重複執行上述步驟後並依溫度參數將初始溫度設定降溫，如果溫度有低於初始溫度設定，則離開此迴圈，直到最後會產生一個近似最佳解，如圖 4.4 S1 演算法步驟說明，其設定參數值如下：

初始溫度：100

溫度控制參數：0.8

迭代次數：8

終止條件：0.00000001



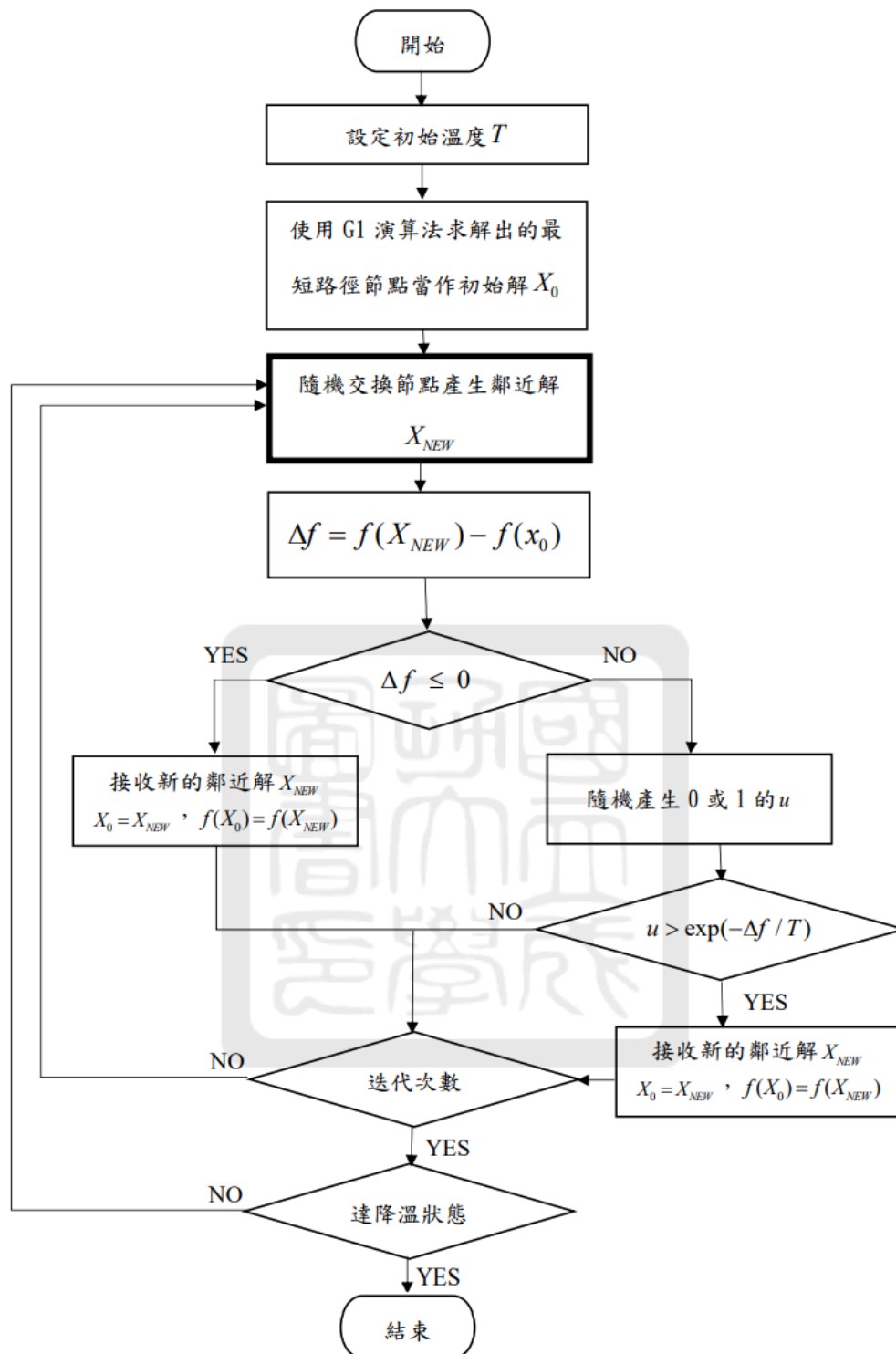


圖 4.4 求解演算法 S1 流程圖

以範例說明，假設有一網路圖如圖 4.5，必須搜尋的點有 A、B、C3 個點，A 點分別可被編號 4 和路徑(3, 4)、(4, 3)、(4, 9)、(9, 4)所覆蓋，B 點分別可被編號 7 和路

徑(2, 7)、(7, 2)、(7, 8)、(8, 7)、(7, 12)、(12, 7)、(7, 6)、(6, 7)所覆蓋，C 點分別可被編號 20 和路徑(20, 21)、(21, 20)、(15, 20)、(20, 15)所覆蓋，使用演算法 G1 計算取得初始解，行經路徑依 0→1→2→7→8→3→4→9→8→7→12→17→16→21→20，耗用的時間為 54.5。透過演算法 S1 將使用初始解取得必須到達的點其順序分別是：7、4、20，如圖 4.6 以隨機挑選兩個點互換後，則順序分別是：7、20、4，從起點 0 開始出發，0 至 7 最短行經路徑 0→1→2→7，7 至 20 最短行經路徑 7→12→17→16→21→20，20 至 4 最短行經路徑 20→21→22→23→18→13→8→3→4 如圖 4.7，其耗用的時間:61.5。

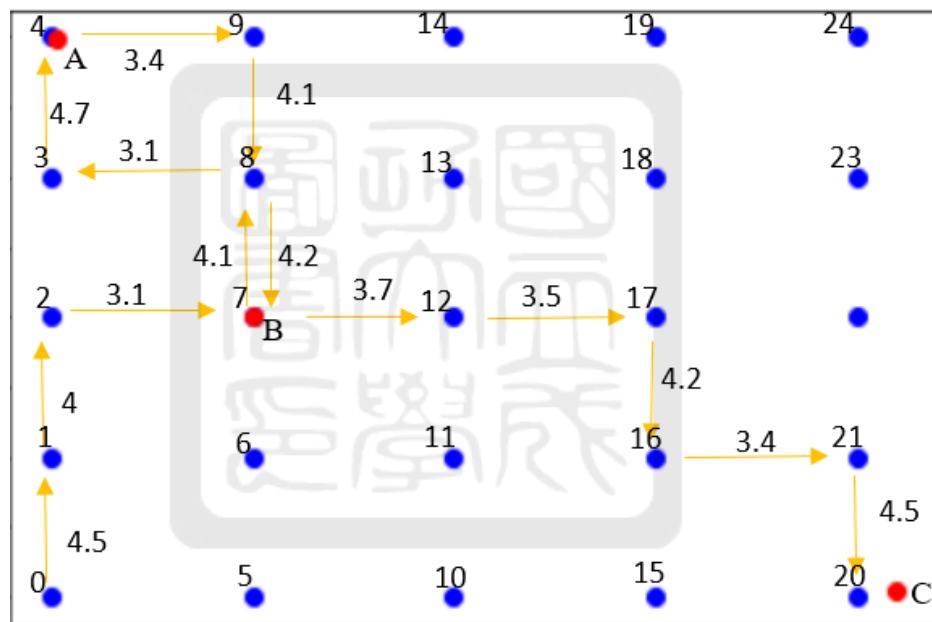


圖 4.5 演算法 G1 計算而得之路線

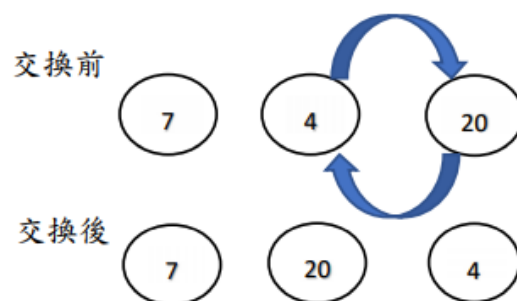


圖 4.6 演算法 S1-隨機互換兩個節點的位置

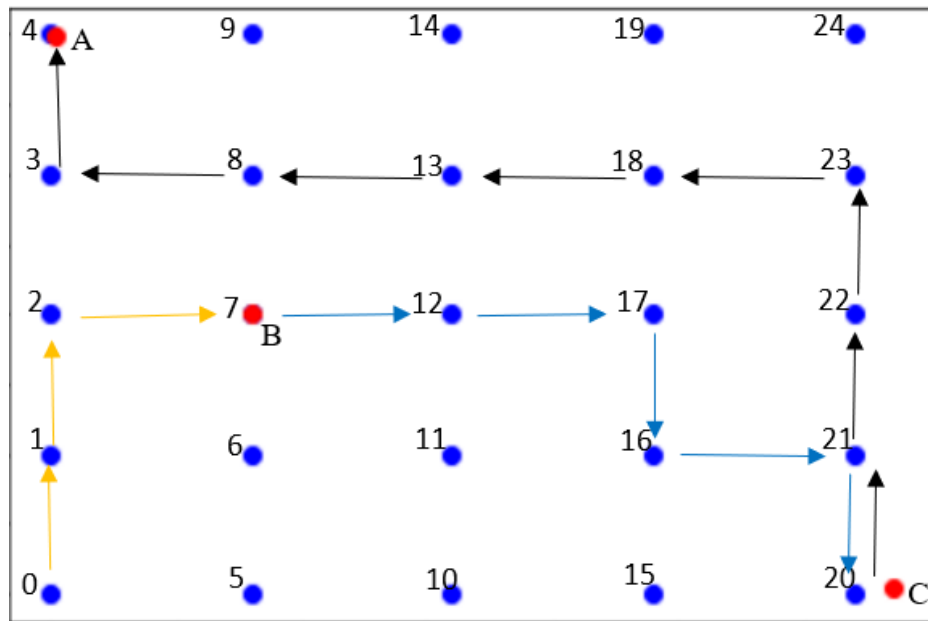


圖 4.7 演算法 S1 計算而得之路徑

## 4.4 求解演算法 S2

使用模擬退火法產生鄰近求解的第二種方法是指隨機挑選某個點將安插在另一個隨機的點後面，再使用點至點最短路徑取得新的規劃路徑解，如圖 4.8 S2 演算法步驟說明。

以範例說明，假設有一網路圖(圖 4.9)必須搜尋的點有 3 個分別為 A、B、C，A 點分別可被編號 4 和路徑(3, 4)、(4, 3)、(4, 9)、(9, 4)所覆蓋，B 點分別可被編號 7 和路徑(2, 7)、(7, 2)、(7, 8)、(8, 7)、(7, 12)、(12, 7)、(7, 6)、(6, 7)所覆蓋，C 點分別可被編號 20 和路徑(20, 21)、(21, 20)、(15, 20)、(20, 15)所覆蓋，使用演算法 G1 計算而得之路線作為初始解的行經路徑，依序 0→1→2→7→8→3→4→9→8→7→12→17→16→21→20，耗用的時間為 54.5。透過演算法 S2 將使用初始解取得必須到達的點，其順序分別是：7、4、20，以隨機挑選某個點將安插在另一個隨機的點後面，其必須到達的點依順序分別是：4→7→20，如圖 4.10 從起點 0 開始出發，0 至 4 最短行經路徑 0→1→2→3→4，4 至 7 最短行經路徑 4→9→8→7，7 至 20 最短行經路徑 7→12→17→16→21→20，耗用的時間:48.8。



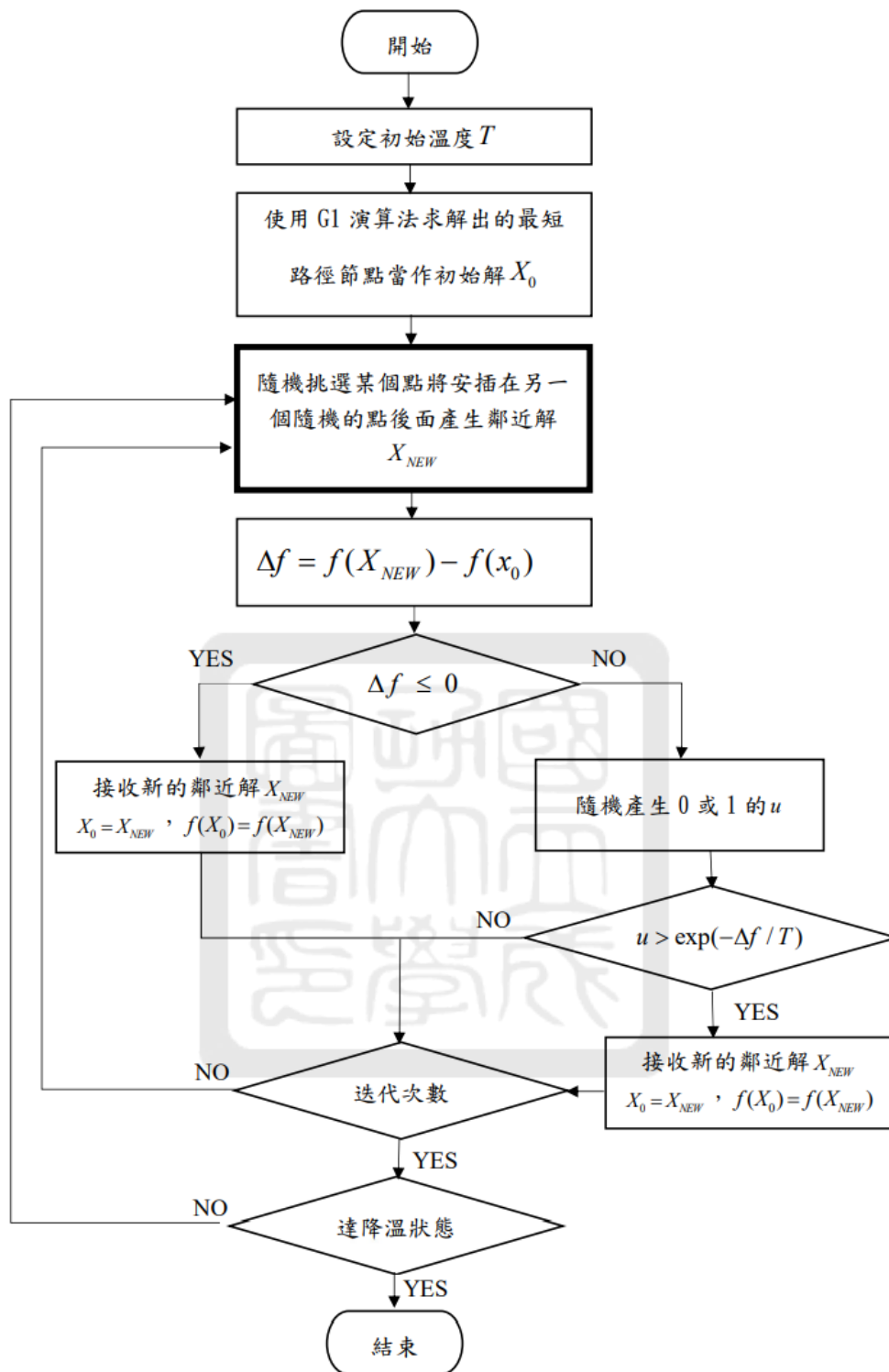


圖 4.8 求解演算法 S2 流程圖

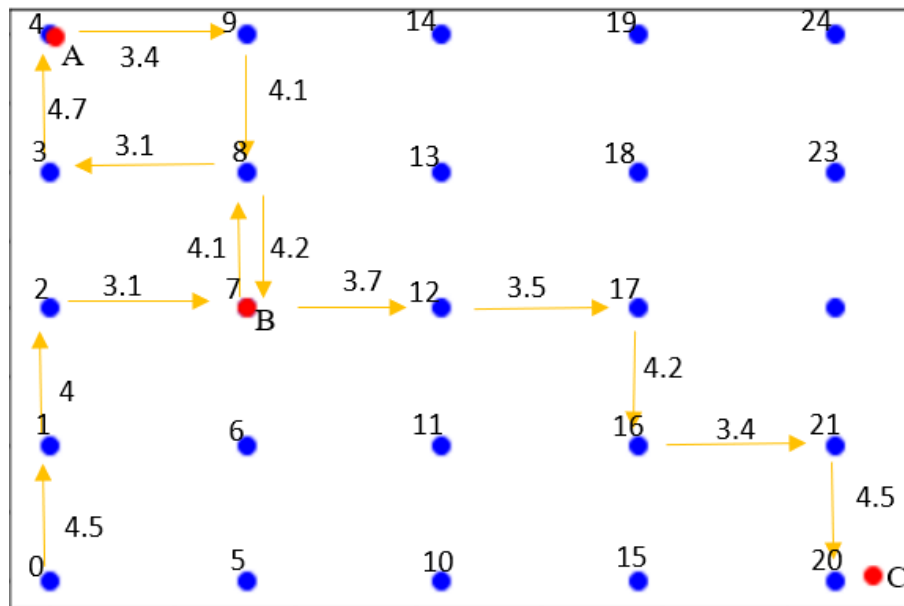


圖 4.9 演算法 G1 計算而得之路線

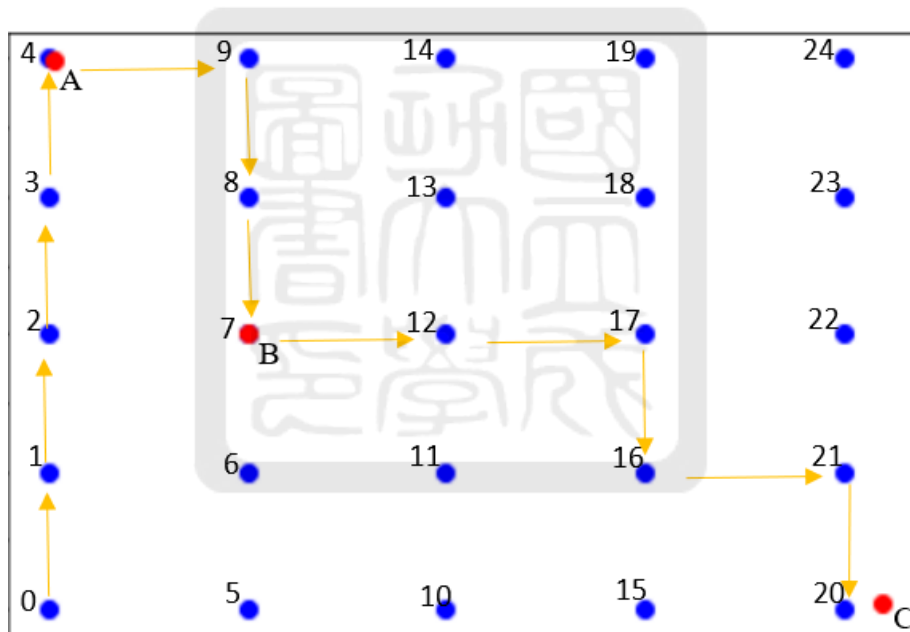


圖 4.10 演算法 S2 計算而得之路徑

## 4.5 求解演算法 S3

使用模擬退火法產生鄰近求解之第三種方法是指先找出點與點相近的關係並設為一組，在以兩兩一組的點互換直到換完後，再隨機的挑選某一組來進行兩個節點互換，接著使用節點至節點最短路徑取得新的解，如圖 4.11 S3 演算法步驟說明。

以範例說明，假設有一網路圖(圖 4.12)必須搜尋的點有 8 個分別為 A、B、C、D、E、F、G、H，A 點可被編號 7 和路徑(2, 7)、(7, 2)、(6, 7)、(7, 6)、(7, 8)、(8, 7)、(7, 12)、(12, 7)所覆蓋，B 點可被編號 11 和路徑(6, 11)、(11, 6)、(10, 11)、(11, 10)、(11, 16)、(16, 11)、(11, 12)、(12, 11)所覆蓋，C 點可被編號 13 和路徑(8, 13)、(13, 8)、(12, 13)、(13, 12)、(13, 14)、(14, 13)、(13, 18)、(18, 13)所覆蓋，D 點可被編號 15 和路徑(10, 15)、(15, 10)、(15, 16)、(16, 15)、(15, 20)、(20, 15)所覆蓋，E 點可被編號 16 和路徑(15, 16)、(16, 15)、(11, 16)、(16, 11)、(16, 17)、(17, 16)、(16, 21)、(21, 16)所覆蓋，F 點可被編號 18 和路徑(13, 18)、(18, 13)、(17, 18)、(18, 17)、(18, 19)、(19, 18)、(18, 23)、(23, 18)，G 點可被編號 22 和路徑(17, 22)、(22, 17)、(21, 22)、(22, 21)、(22, 23)、(23, 22)所覆蓋，H 點可被編號 23 和路徑(22, 23)、(23, 22)、(18, 23)、(23, 18)、(23, 24)、(24, 23)所覆蓋，其相鄰的為一組可區為 23→22、15→16、11→7、13→18 為 4 組，使用演算法 G1 計算而得之路線作為初始解的行經路徑依序為 0→5→10→15→16→11→12→17→18→23→22→17→18→13→8→7，耗用的時間為 54.5。

以求解演算法 S2 計算而得之路線(圖 4.13)，從起點 0 開始出發，將相鄰的一組互換 23→22 依行經路徑為是 0→5→10→15→16→11→16→21→22→23→18→13→18→13→8→7，其耗用的時間:50.5。

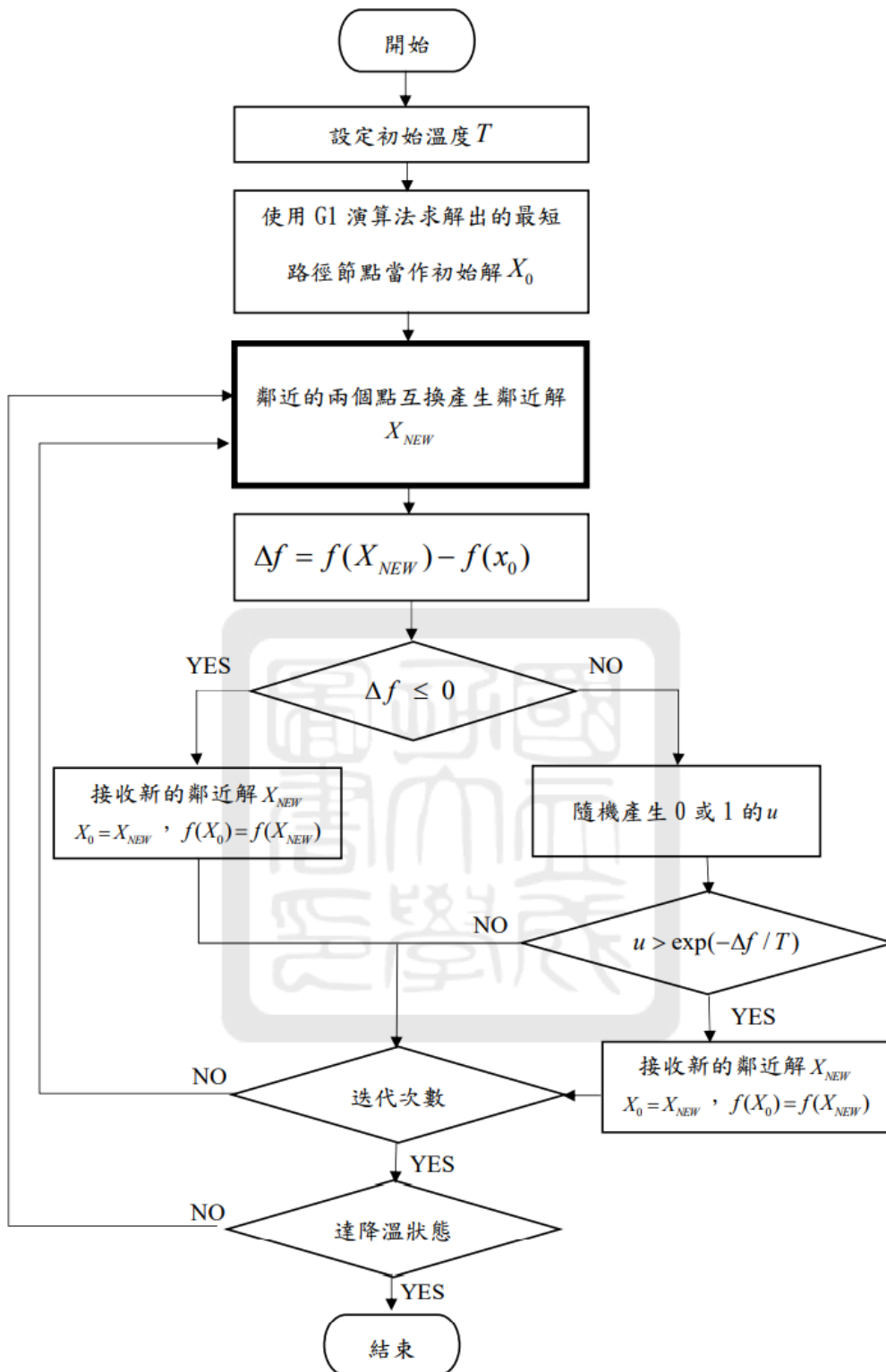


圖 4.11 求解演算法 S3 流程圖

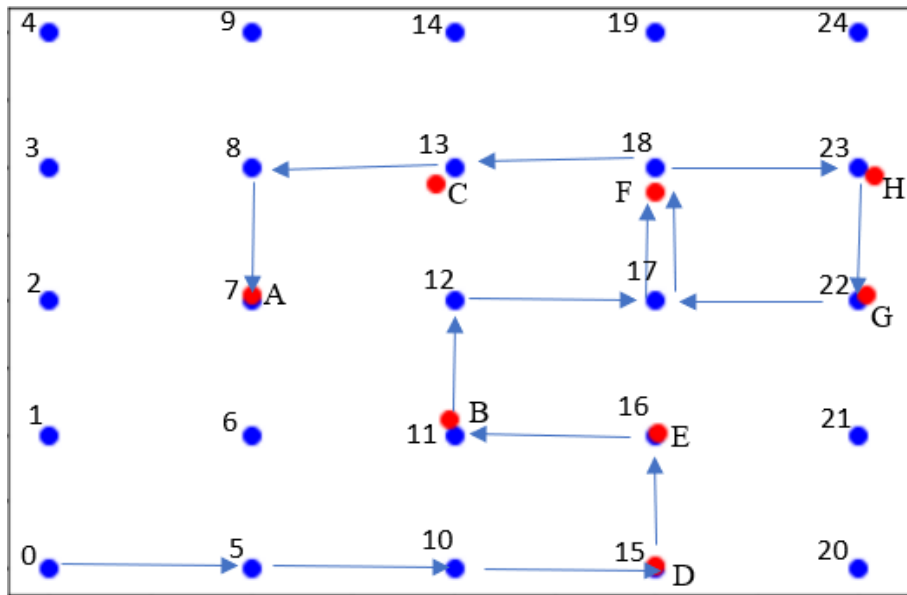


圖 4.12 演算法 G1 計算而得之路線

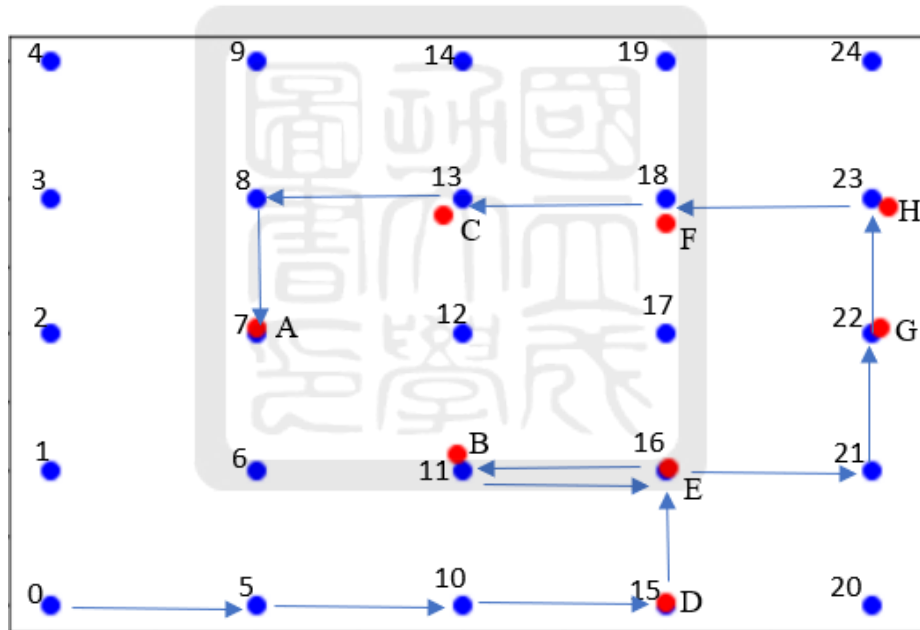


圖 4.13 演算法 S2 計算而得之路徑

## 4.6 求解演算法 S4

使用模擬退火法產生鄰近求解是指依比例分配求解演算法 S1 為 50%、S2 為 20%，S3 為 30% 混合三種上述的求解演算法，以演算法 G1 計算取得初始解行經路徑，再依比例分配使用演算法 S1、S2、S3 計算取得新的路線規劃，S4 演算法步驟說明如圖

4.14，因隨機產生的方式已在上述說明就不在此論述。

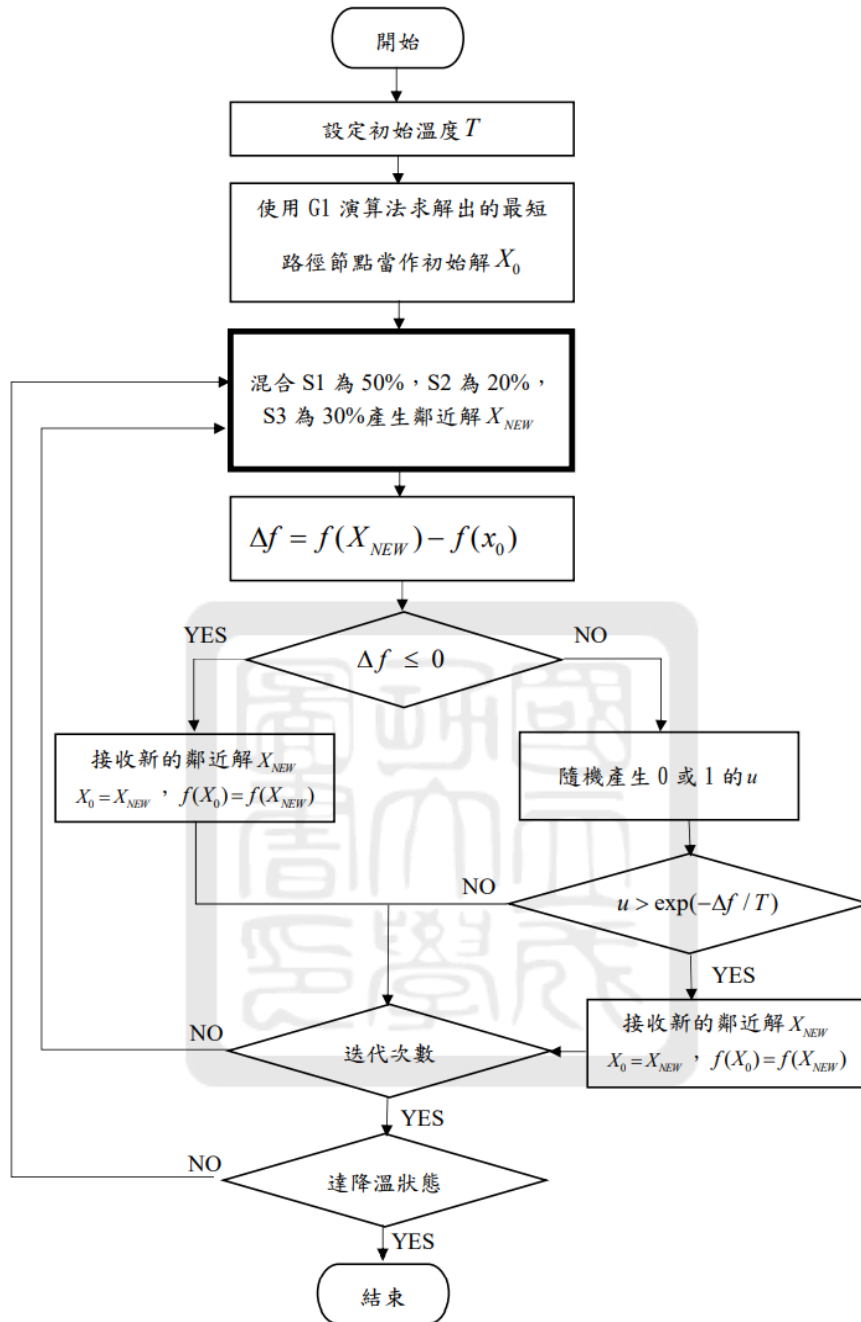


圖 4.14 求解演算法 S4 流程圖

## 4.7 小結

本章節設計二種演算法貪婪演算法(Greedy Algorithm)和四種模擬退火法

(Simulated Annealing) 方的求解方法，分別是使用貪婪演算法有二種為 G1 和 G2，使用模擬退火法有四種為 S1、S2、S3、S4，並說明其求解的概念，目的是為能設計出比多期風力單台無人船整數規劃模型更有效率的求解演算法，並會在下章節將六種求解方法與多期風力之單台無人船路徑規劃 M1 和以 M1 為基礎但路線可重複之路徑規劃數學模式 M2 進行數值比較。



## 第五章 例題測試與分析

本章將產生大、中、小型的三種網路圖，每組網路圖再各設定三種覆蓋的指定地點數目，經由設定不同的必須巡邏地點個數以及每個時段不同的風力影響，來測試本研究提出的多期風力單台無人船整數規劃模型和求解演算法在不同情境網路圖之求解表現，並說明測試結果。本研究之程式皆以 Python 3.6 版本的程式語言撰寫，並配合 Gurobi 9.1.1 版本求解整數規劃模型，測試平台使用 Windows 10 作業系統，搭配處理器 Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz 和記憶體 32GB 的個人電腦。

### 5.1 測試案例描述

本研究定義了三種網路圖分別有大、中、小型的網路圖，如圖 5.1(a)為小型的網路圖、圖 5.1(b)為中型的網路圖和圖 5.1(c)為大型網路圖，並定義小型網路圖節點個數有 25 個，中型網路圖節點的個數有 79 個，大型網路圖節點的個數有 100 個；以巡邏地點的個數，小型網路圖分別為 3、5、8 個，中型網路圖分別為 5、8、16 個，大型網路圖分別為 11、16、32 個；以每個 X 軸節點與節點之間的距離耗用時間(分鐘)，小型網路圖為 3~4 分鐘，中型網路圖為 2~2.7 分鐘，大型網路圖為 1.3~1.8 分鐘；以每個 Y 軸節點與節點之間的距離耗用時間(分鐘)，小型網路圖為 4~5 分鐘，中型網路圖為 2.7~3.6 分鐘，大型網路圖為 1.8~2.2 分鐘；每個時段的定義分別為 60 分鐘、120 分鐘，會先用求解演算法求得解，檢查如果超過時段的定義時間會在往後加 60 分鐘，為下一個時段區間，如表 5.1 產生測試網路圖的定義；每組設定將以隨機的方式產生巡邏地點的位置、每個時段的風力影響每段節線之行經所耗用的時間(分鐘)、已計算出每個節點和節線可覆蓋哪些巡邏地點，每組設定會在各產生 10 組的網路圖以進行測試分析。



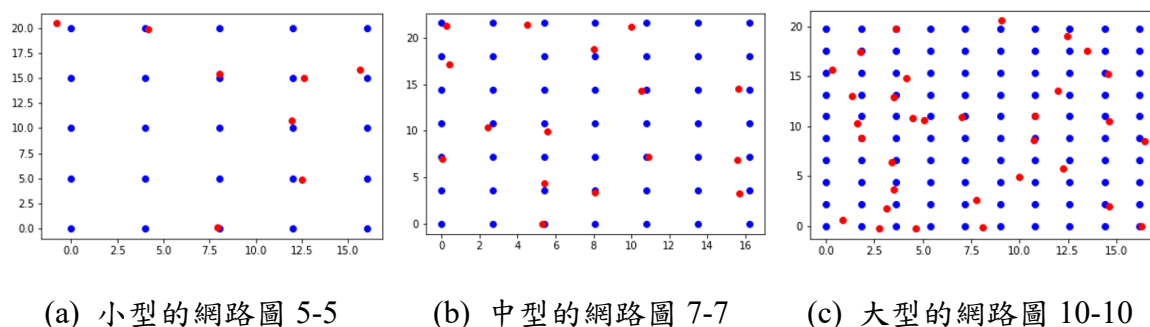


圖 5.1 大、中、小型的網路圖範例

表 5.1 產生測試網路圖定義

測試案例	節點	巡邏地點的個數	X 軸點至點之間耗用時間(分鐘)	Y 軸點至點之間耗用時間(分鐘)	時段(分鐘)
5-5-3	25	3	3~4	4~5	60, 120
5-5-5	25	5	3~4	4~5	60, 120
5-5-8	25	8	3~4	4~5	60, 120
7-7-5	49	5	2~2.7	2.7~3.6	60, 120
7-7-5	49	8	2~2.7	2.7~3.6	60, 120
7-7-5	49	16	2~2.7	2.7~3.6	60, 120
10-10-11	100	11	1.3~1.8	1.8~2.2	60, 120
10-10-16	100	16	1.3~1.8	1.8~2.2	60, 120
10-10-32	100	32	1.3~1.8	1.8~2.2	60, 120, 180

## 5.2 測試結果分析

本研究依據上述每一組設定值以隨機的方式各產生 10 組網路圖後，總共有 90 種不同的例子，將分別以多期風力單台無人船整數規劃模型 M1、多期風力單台無人船可重覆經過節點整數規劃模型 M2、使用貪婪演算法取最短的路徑的演算法 G1、使用貪婪演算法隨機取得最短和第二短的路徑的演算法 G2、使用模擬退火法隨機的互換二個點的演算法 S1、使用模擬退火法隨機挑選某個點將安插在另一個隨機的點後面的演算法 S2、使用模擬退火法兩兩一對相近的點互換 S3 和使用模擬退火法混合三種方法分配比例 S1 為 50%，S2 為 20%，S3 為 30%互換兩個點的演算法 S4，將這八種求解後將求解時間(單位：秒)與最佳解 Gap(單位：%)列出，若最佳解的求解時間超過

2 個小時(7200 秒)，模型將停止求解，若此時尚未出現最佳解，則演算法的 Gap 可能出現小於 0 的情況，接著我們將以三種大，中，小型的網路小節分別說明求解的結果。

### 5.2.1 小型網路

本小節針對小型網路 5x5 的水域範圍，並設定有三種不同的需巡邏地點的個數分別有 3、5、8 個，因此將測試組別命名為 5-5-3、5-5-5、5-5-8，各產生 10 組網路圖測試例子，如表 5.2 所示，將 M2 的求解時間(s)和最佳解與 G1、G2、S1、S2、S3、S4、M1 的求解時間(s)和最佳解 Gap(%)作列表比較，首先觀察表中求解時間(s)可以發現 G1 因求解相當迅速故求解時趨近於 0 秒，但從其它求解方法可發現隨著需巡邏點的增加不管是整數規劃模型或演算法，其需求解時間(s)皆呈現相對的遞增；而從最佳解的 Gap 比較下，可分為三個群組分別比較，首先是貪婪演算法這一組 G1 和 G2 比較，在測試組別: 5-5-3 和 5-5-5 時，G1 求解表現皆比 G2 來的好一點，差距 1%以內。在測試組別: 5-5-8 時可發現 G1 和 G2 求解表現皆一樣，但 G1 在求解時間(s)優於 G2。相較之下 G1 整體表現都略優於 G2，所以 G1 反而是不錯求解方法；模擬退火法這一組 S1、S2、S3、S4 比較，在測試組別: 5-5-3 時，S1、S2、S3、S4 同於最佳解，且求解時間(s)也都優於最佳解，其求解表現很好，在測試組別: 5-5-8 時，S1、S2、S4 同於最佳解，三個求解時間(s)也在 1 秒內，其求解表現也很好，在測試組別: 5-5-8 時只有 S4 同於最佳解，但比較 S1、S2 平均 Gap 也皆在 1%以下，因此 S1、S2、S4 都是不錯求解方法；整數規劃模型這一組 M1、M2 比較，M1 求解表現也不錯，比較平均 Gap 皆在 1%以下，其求解時間(s)也都優於 M2，所以 M1 也是不錯的求解方法。

表 5.2 求解時間與 Gap 比較表-小型網路

	G1		G2		S1		S2		S3		S4		M1		M2	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
<b>5-5-3</b>	0.00	4%	1.35	5%	0.31	0%	0.31	0%	0.32	0%	0.30	0%	0.38	0%	55.09	0%
<b>5-5-5</b>	0.00	10%	2.28	11%	0.47	0%	0.48	0%	0.52	7%	0.48	0%	0.39	1%	60.71	0%
<b>5-5-8</b>	0.01	23%	4.28	23%	0.92	1%	0.84	1%	1.01	17%	0.93	0%	0.64	0%	195.69	0%

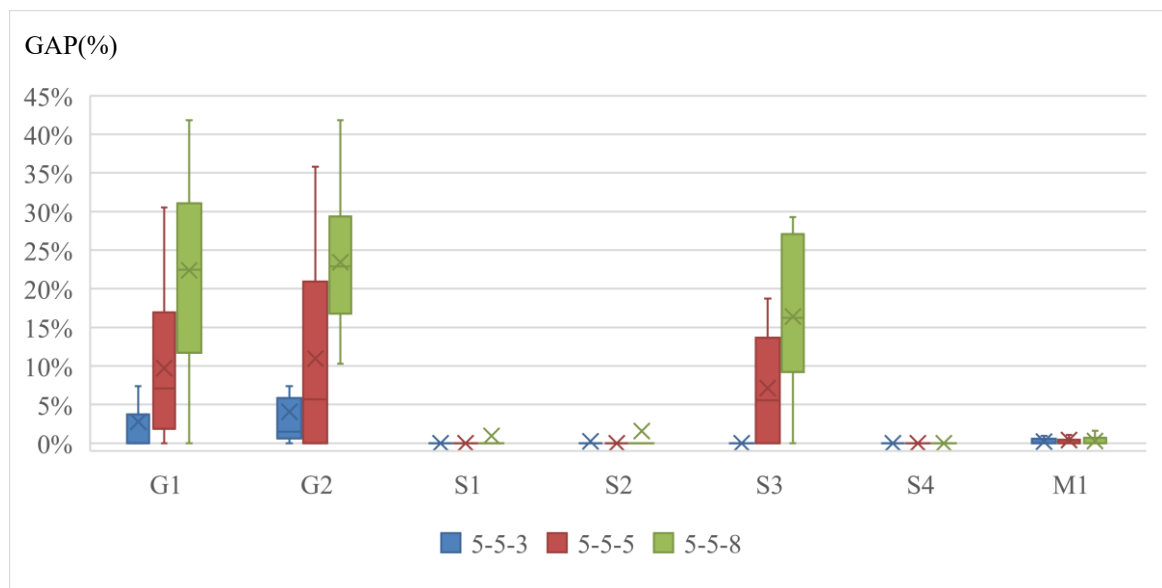


圖 5.2 小型網路-Gap 盒鬚圖

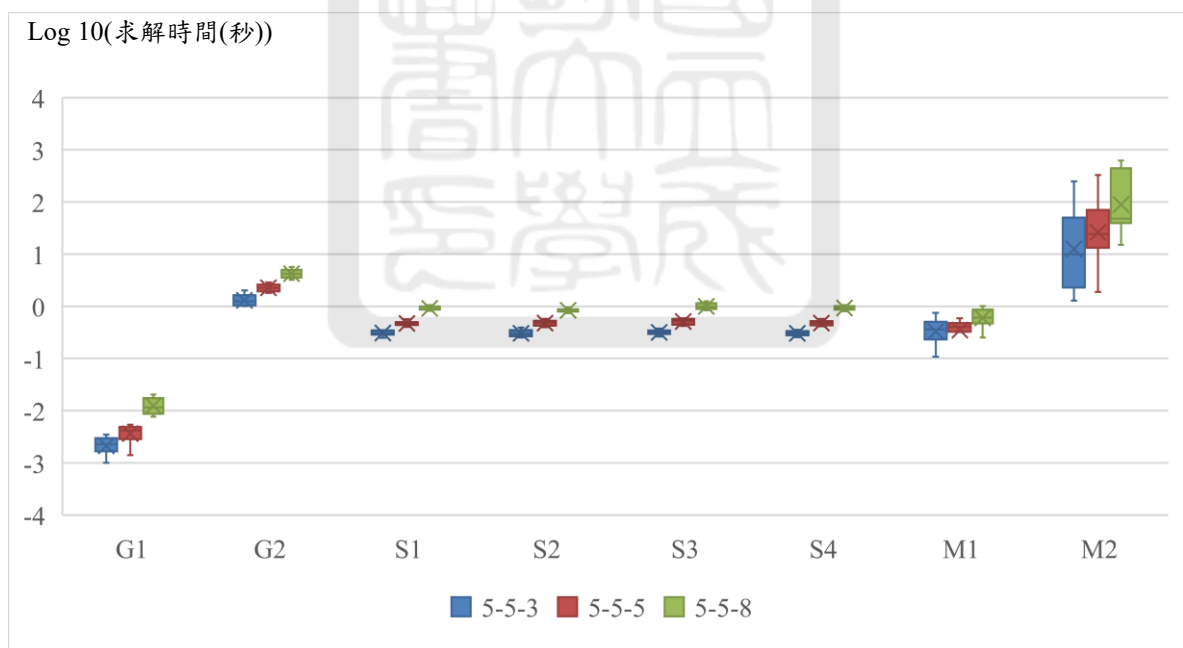


圖 5.3 小型網路-求解時間(s)盒鬚圖

在這裡我們也進一步透過盒鬚圖來觀察小型網路與求解方法的 Gap 差異，如圖 5.2 所示，測試組別：5-5-3 其巡邏點較少時其求解 Gap 差距並不大，而求解方法 S1、

S2、S4、M1 其 GAP 都接近於 0%，都是不錯的求解方法，但隨著巡邏點增加其 G1、G2 和 S3 的求解差異越來越大，從這可以發現求解方法 S1、S2、S4、M1 在不同情境的小型水域裡其安排需巡邏點的多或少，都能有不錯求解表現能力，而 G1、G2 和 S3 會因巡邏點增加相對在每個測試案例求解表現較不一；觀察小型網路與求解時間(s)盒鬚圖(如圖 5.3)，可發現 M2 為能可重覆經過節點取得最佳解，需要花費求解的時間最多，相對 G1 是所有求解方法中花費求解時間(s)最少的。

## 5.2.2 中型網路

本小節針對中型網路 7x7 水域範圍，並設定有三種不同的需巡邏地點的個數分別 5、8、16 個，因此將測試組別命名為 7-7-5、7-7-8、7-7-16，各產生 10 組網路圖測試例子，如表 5.3 所示，將 M2 的求解時間(s)和最佳解與演算法 G1、G2、S1、S2、S3、S4、M1 的求解時間(s)和最佳解 Gap(%)作列表比較，首先從表中觀察求解時間(s)可以看得出隨著需巡邏點的增加不管是整數規劃模型或數學式需求解時間(s)皆呈現相對的遞增，而從最佳解的 Gap 比較下，可分為三個群組分別比較首先是貪婪演算法這一組 G1 和 G2 比較，在測試組別：7-7-5、7-7-8、7-7-16 時，G1 求解表現比 G2 來得好一點，且 G1 求解相當迅速，其求解時間趨近於 0 秒；模擬退火法這一組 S1、S2、S3、S4 比較，在測試組別：7-7-5、7-7-8 時，S1、S2、S4 同於最佳解，且求解時間(s)也都優於最佳解，皆能在 5 秒內取得最佳解，其 S1、S2、S4 求解表現很好，在測試組別：7-7-16 時，S1、S2、S4 比較平均 Gap 皆在 5~7%，且求解時間(s)也都相當接近都在 6~7 秒內，都是不錯的求解方法，但 S2 平均 Gap 還是比其它 S1、S3、S4 好一點；整數規劃模型這一組 M1、M2 比較，M1 求解表現也不錯，比較平均 Gap 皆在 1%以下，其求解時間(s)也都優於 M2，是不錯的求解方法；在七種演算法中，在測試組別：7-7-5 時，這 4 個 S1、S2、S4 和 M1 的求解表現非常好，在測試組別：7-7-8 時，S1、S2、S4 的求解表現非常好，在測試組別：7-7-16 時，M1 在求解表現等同於最佳解，且求解時間(s)優於最佳解，其求解表現很好。

在透過盒鬚圖來進一步觀察中型網路與求解方法的 Gap 差異，如圖 5.4 所示，S1、S2、S4、M1 在不考慮水域的大小時，其巡邏點增到 8 個的時候，其求解 Gap 差距並不大，而求解方法其 GAP 都接近於 0%，都是不錯的求解方法，但隨著巡邏點增加到 16 個的時候，其 S1、S2 和 S3 的求解差異越來越大，相對在每個測試案例求解表現較不一；而觀察中型網路與求解時間(s)盒鬚圖如圖 5.5，可發現 M2 在處理中型網路的水域時，求解時間大部分都已超過設定的 2 個小時(7200 秒)求解效率較差，相對 M1 在求解時間(s)仍有不錯的表現，而 G1 是仍是所有求解方法中花費求解時間(s)最少的。

表 5.3 求解時間與 Gap 比較表-中型網路

	G1		G2		S1		S2		S3		S4		M1		M2	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
7-7-5	0.01	6%	5.41	10%	1.44	0%	1.44	0%	1.45	3%	1.45	0%	3.17	0%	3022.44	0%
7-7-8	0.02	11%	10.71	14%	2.95	0%	2.79	0%	3.08	7%	3.04	0%	6.72	1%	5439.47	0%
7-7-16	0.03	22%	27.75	26%	6.96	7%	6.90	5%	7.41	24%	6.75	6%	8.71	0%	6530.19	0%

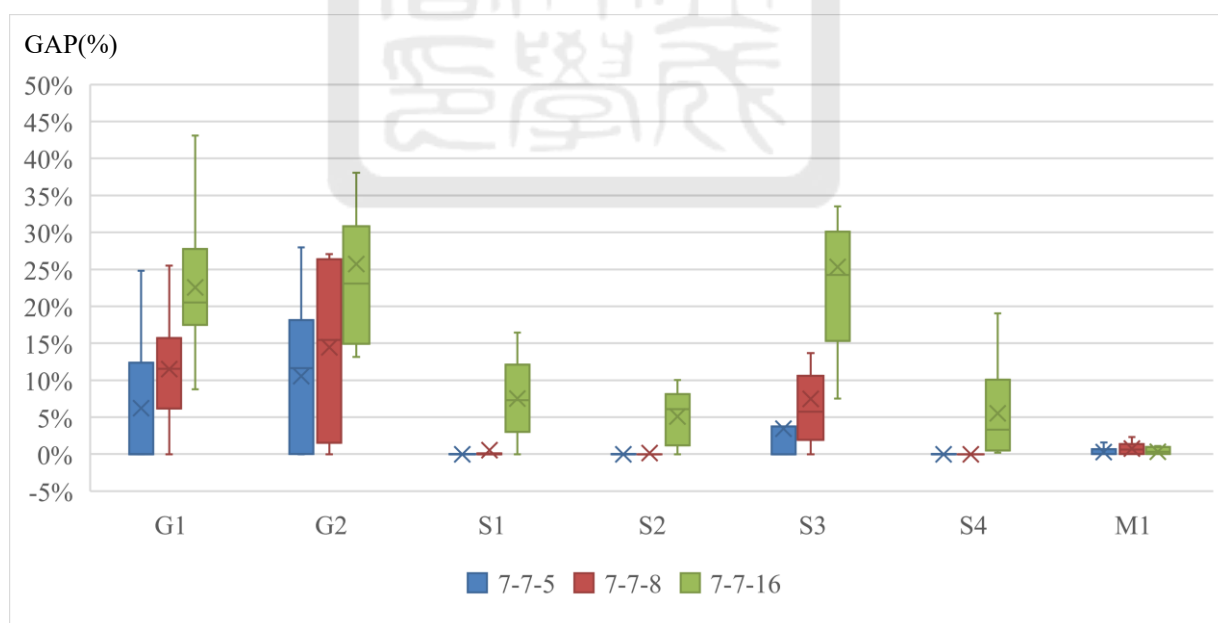


圖 5.4 中型網路-Gap 盒鬚圖

為了了解其演算法被改善程度使用組別 7-7-8 測試案例和組別 7-7-16 測試案例觀察如圖 5.6 和圖 5.7，可以發現 G2 其改善幅度相對較小，使得求解效果比其它相較較差，而使用模擬退火法 S1、S3、S4 都能比 G1 求解效果來得好，其改善幅度也有顯著的效果。

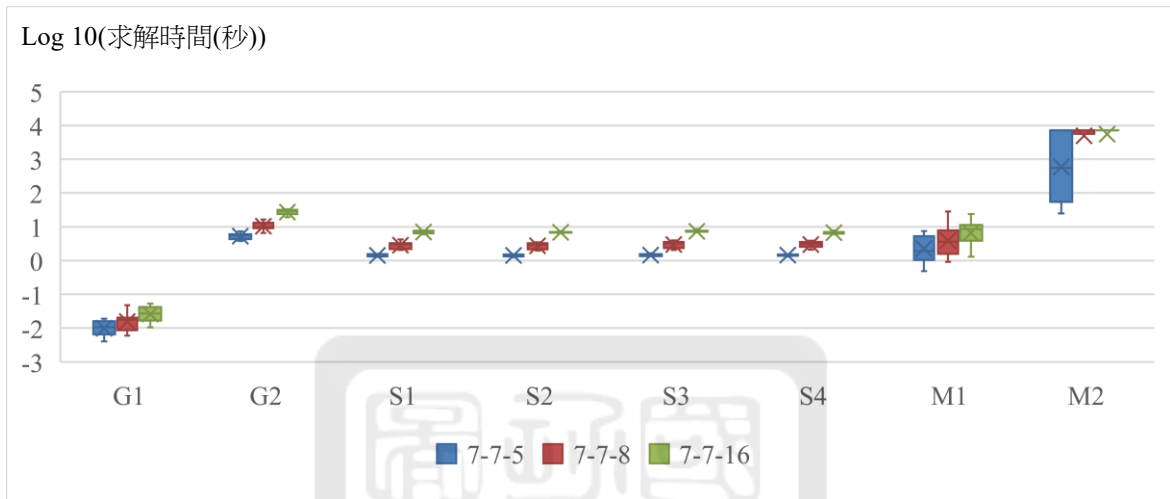


圖 5.5 中型網路-求解時間(s)盒鬚圖

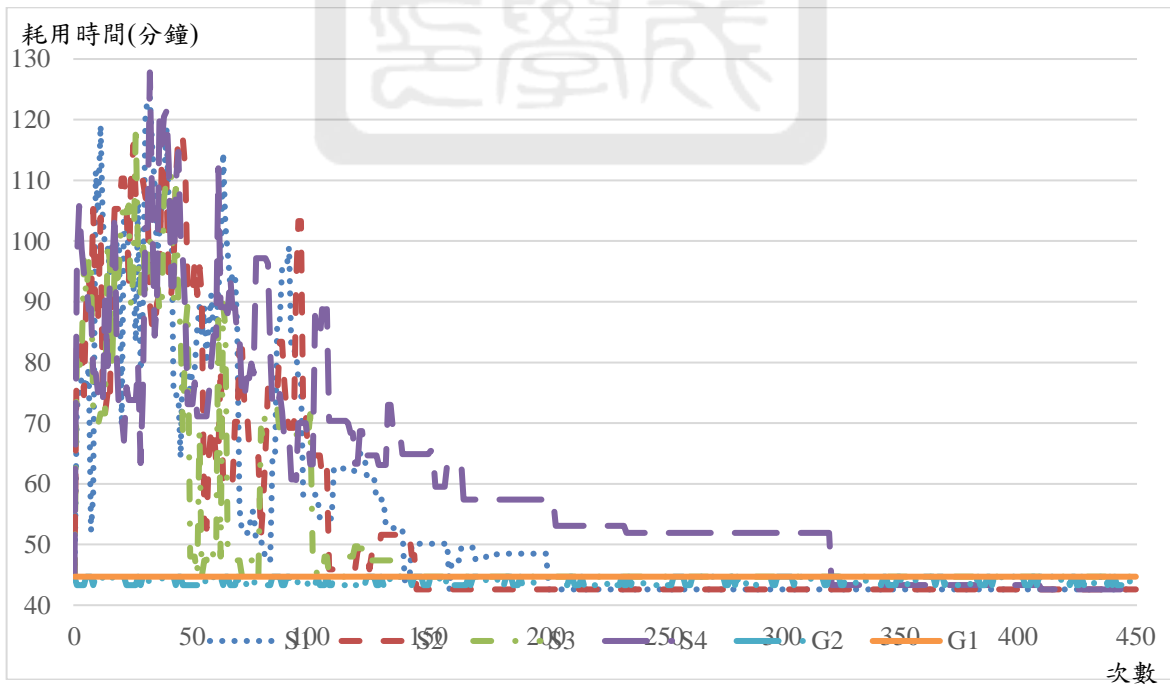


圖 5.6 中型網路-測試組別 7-7-8 演算法改善程度

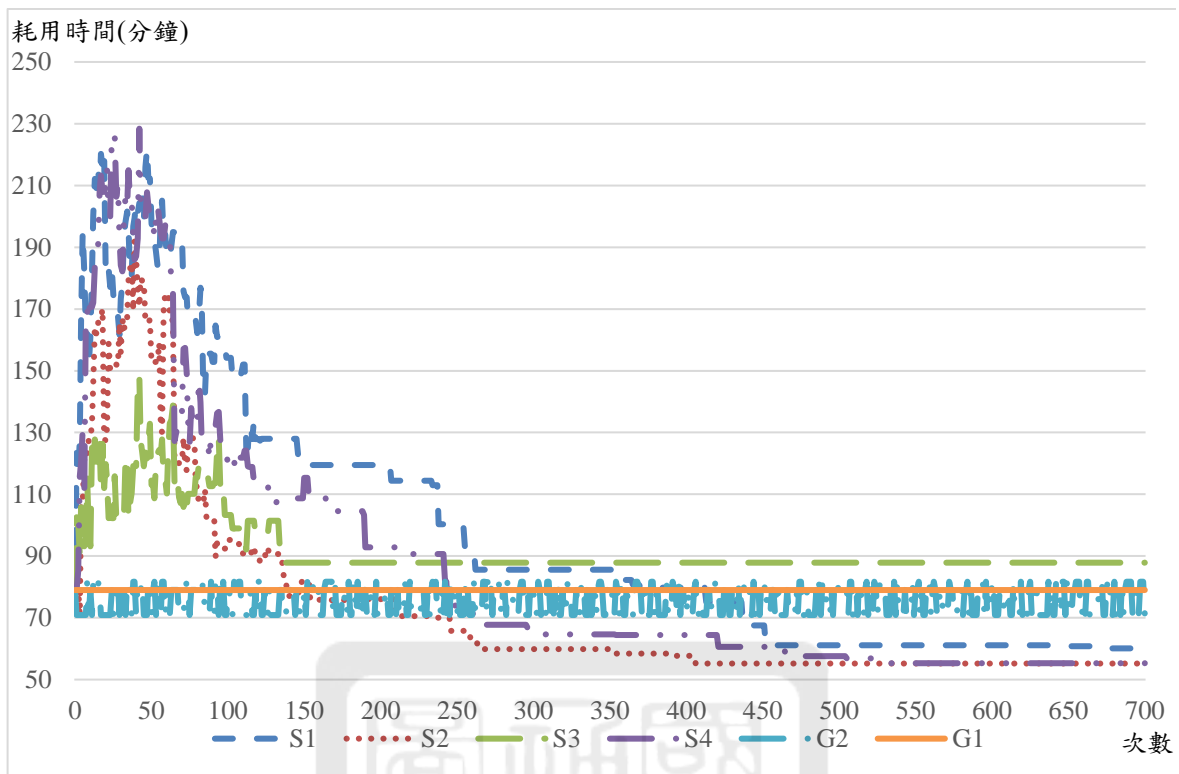


圖 5.7 中型網路-測試組別 7-7-16 演算法改善程度

### 5.2.3 大型網路

本小節針對大型網路 10×10 水域範圍，並設定有三種不同的需巡邏地點的個數分別 11、16、32 個，因此將測試組別命名為 10-10-11、10-10-16、10-10-32，各產生 10 組網路圖測試例子，表 5.4 將 M1 的求解時間(s)和最佳解與演算法 G1、G2、S1、S2、S3、S4 的求解時間(s)和最佳解 Gap(%)作列表比較，首先從表中觀察求解時間(s)可以看得出隨著需巡邏點的增加不管是整數規劃模型或數學式需求解時間(s)皆呈現相對的遞增，而從最佳解的 Gap 比較下，可分為兩個群組分別比較首先是貪婪演算法這一組 G1 和 G2 比較，在測試組別: 10-10-11、10-10-16 時，G1 求解表現比 G2 來得好，且 G1 求解相當迅速幾乎不到 1 秒。在測試組別: 10-10-32 時，G2 求解表現比 G1 好 1%，但 G1 求解速度依然不到 1 秒即可完成，G1 雖然在求解表現有誤差 1%，但因求

解速度快所以表現也不差表現依然不差，G2 則在隨著需巡邏點的增加表現算相當平均；模擬退火法這一組 S1、S2、S3、S4 比較，在測試組別 10-10-11 時，S2 求解表現較好且求解時間(s)也都優於最佳解，而 S1 和 S4 的表現也不差，平均 Gap 皆為 1%，在求解時間(s)比較  $S4 < S1 < S2$ ，所以 S1、S2、S4 都是不錯的求解方法，在測試組別: 10-10-16 時，S1 求解表現和求解時間(s)都優於其它三個，所以在這測試組別 S1 是不錯的求解方法，在測試組別: 10-10-32 時 S2 求解表現相較比其它 S1、S3、S4 來的好而求解時間(s)只比最快的 S1 差一點，所以在這測試組別 S2 反而是不錯的求解方法；整數規劃模型這一組 M1、M2 如表 5.5 所示，在測試組別: 10-10-11、10-10-16 時因 M2 求解時間大部分都超過設定的 2 個小時(7200 秒)，模型會停止求解並回饋求解結果，其求解表現和求解時間(s)都比 M1 來的差，在測試組別: 10-10-32 因 M2 在 10 組網路圖測試例子求解時間皆超過設定的 2 個小時(7200 秒)，只有一組有其求解結果且比 M1 求解表現來得好，其餘求解結果皆為 NA，所以相對而言 M1 在求解表現和求解時間(s)是不錯的求解方法。

表 5.4 求解時間與 Gap 比較表-大型網路

	G1		G2		S1		S2		S3		S4		M1	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
10-10-11	0.11	25%	65.01	27%	14.62	1%	14.95	0%	14.98	18%	13.95	1%	248.18	0%
10-10-16	0.18	11%	121.71	28%	18.96	7%	20.63	9%	19.84	13%	21.99	11%	282.45	0%
10-10-32	0.69	28%	331.65	27%	28.01	20%	35.09	12%	40.33	27%	41.09	20%	302.32	0%

表 5.5 M1 和 M2 求解時間與 Gap 比較表-大型網路

	M2		M1	
	Time(s)	Gap(%)	Time(s)	Gap(%)
10-10-11	6496.19	3%	248.18	0%
10-10-16	7200.62	2%	969.81	0%
10-10-32	7200.65	-1%	302.32	0%



在透過盒鬚圖來進一步觀察大型網路與求解方法的 Gap 差異，如圖 5.8 所示，S1、S2、S4 在大型網路的水域時巡邏點 11 個時其求解 Gap 差距並不大都接近於 0%，隨著巡邏點增加，其 S1、S2 和 S4 的求解差異越來越大，相對在每個測試案例求解表現較不一，S1 表現最為顯著；而觀察大型網路與求解時間(s)盒鬚圖(圖 5.9)，可發現演算法求解效率比整數規劃模型 M1、M2，在處理大型網路的水域時求解時間(s)表現來的好。

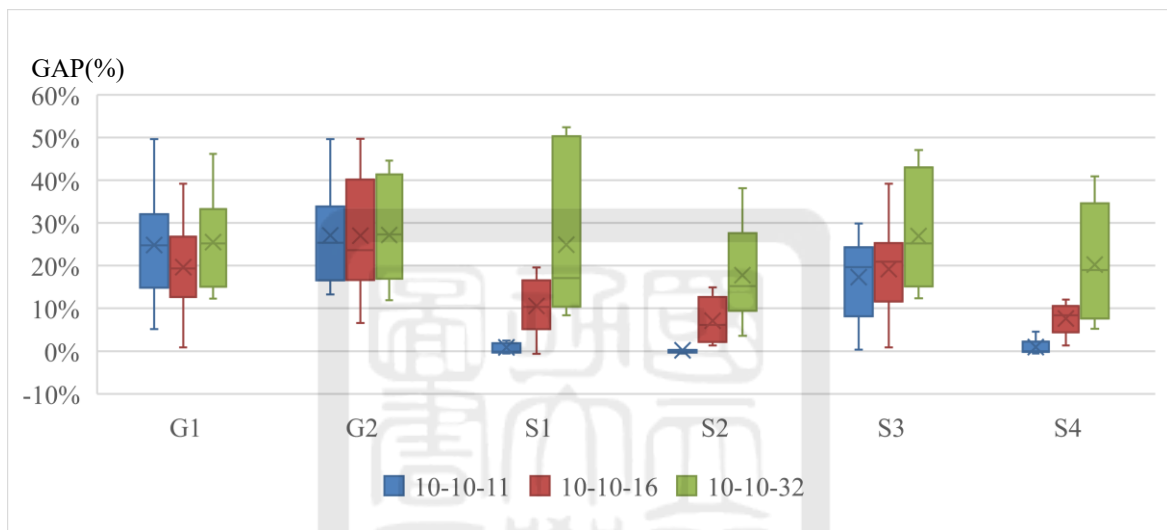


圖 5.8 大型網路-Gap 盒鬚圖

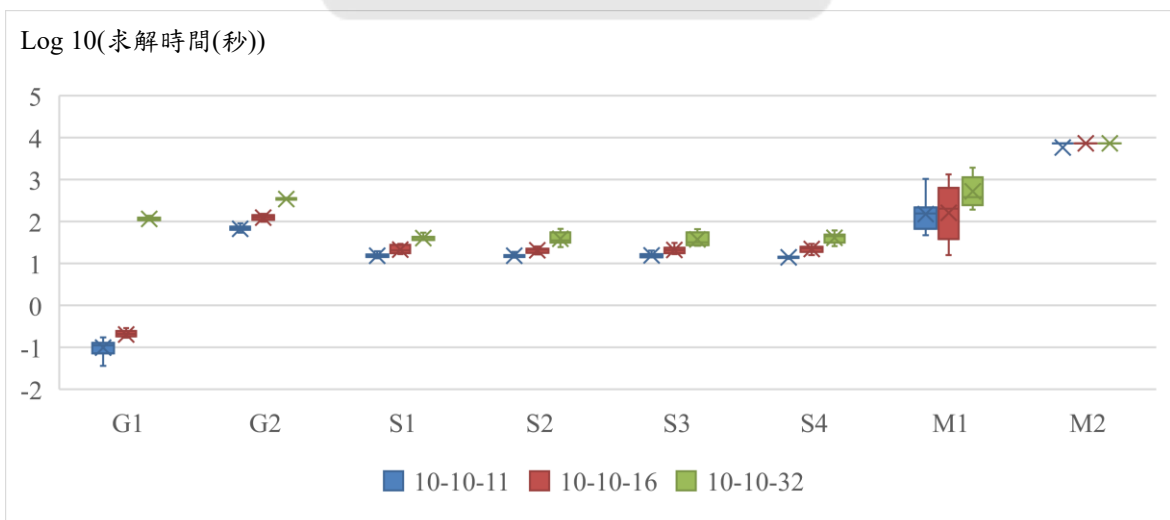


圖 5.9 大型網路-求解時間(s)盒鬚圖

為了了解其演算法被改善程度使用組別 10-10-11 測試案例觀察如圖 5.10，可以發現 G2 其改善幅度相對較小，使得求解效果比其它相較較差，S1、S2、S3、S4 使模擬退火法比 G1 求解效果來得好，其改善幅度效果優先順序分別  $S1 < S2 < S4 < S3$ 。

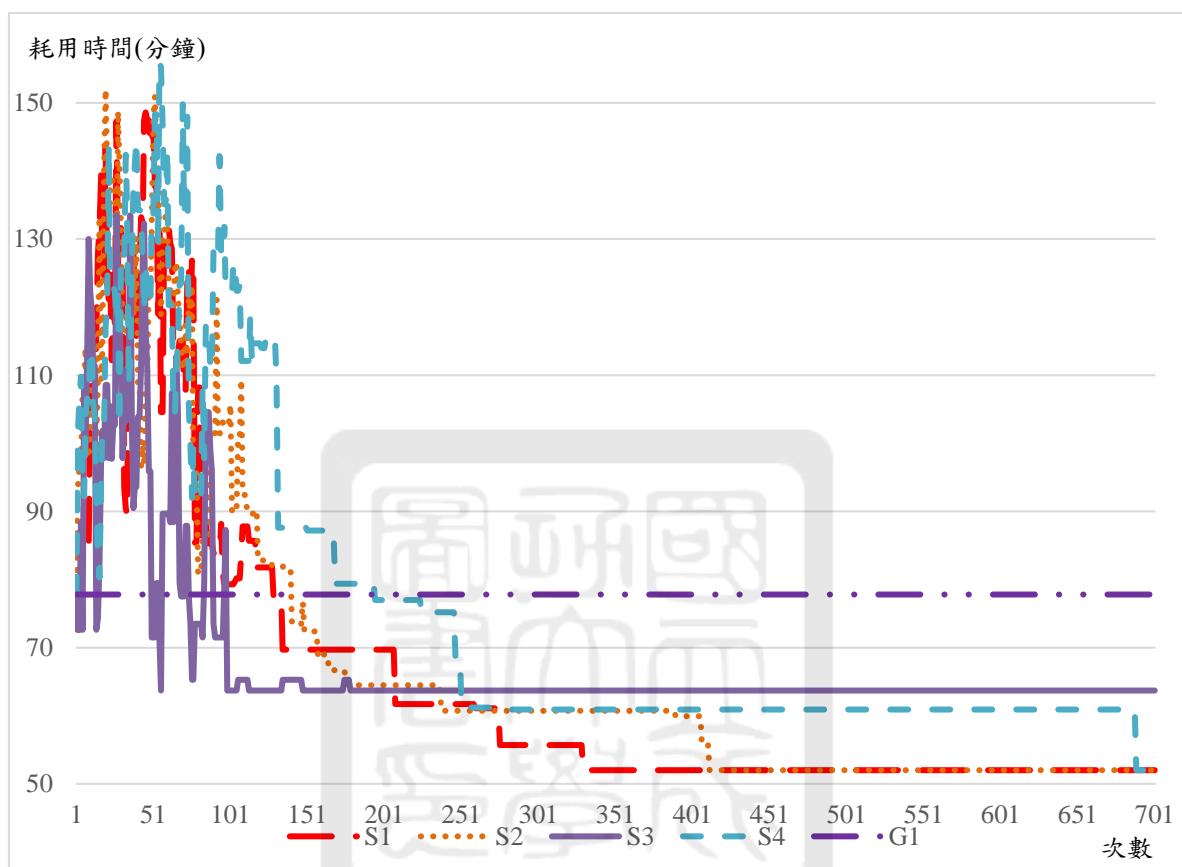
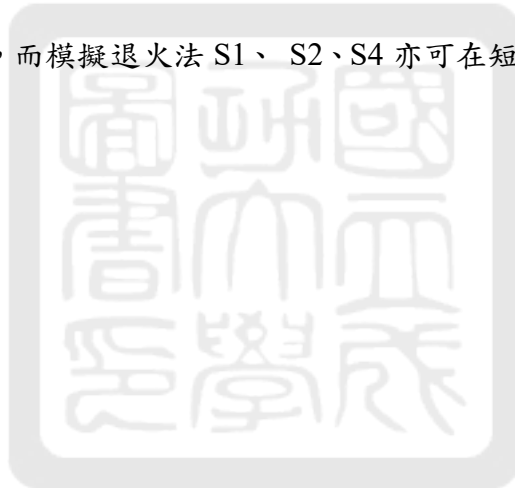


圖 5.10 大型網路-測試組別 10-10-11 演算法改善程度

### 5.3 小結

綜合上述測試不同大小的網路圖和其八個求解方法的比較的結果，以小型網路來說使用貪婪演算法這一組 G1 不管在求解表現和求解速度都優於 G2，會建議採用 G1 求解方法，以模擬退火法這一組其 S1、S2、S4 求解表現皆都很好，平均 Gap 都在 1% 以下，在求解速度皆都不分上下，如要建議使用可以選擇平均 Gap 都在 0% 的 S4 求解方法，整數規劃模型這一組，M1 求解表現都很好且求解速度都皆優於 M2，建議

可以採用 M1；以中型網路來說使用貪婪演算法這一組會推薦 G1 求解方法，因不管在求解表現和求解速度都仍然優於 G2，以模擬退火法這一組其 S1、S2、S4 求解表現也皆都很好，在求解速度皆都不分上下，但在巡邏點較多的時候 S2 表現仍優於其它模擬退火法，其建議可以採用 S2；整數規劃模型這一組，M1 求解表現都很好平均 Gap 都在 1%且求解速度都皆優於 M2，建議仍可以採用 M1；以大型網路來說使用貪婪演算法在求解表現上 G1 還是優於 G2 一點，且 G1 求解速度優於 G2，因此建議可以採用 G1；模擬退火法這一組 S1、S2 求解表現皆都不錯但在巡邏點較多的時候 S2 表現仍優於其它模擬退火法，其建議可以採用 S2；整數規劃模型這一組，M2 在大部分求解時間皆超過設定的 2 個小時(7200 秒)，其巡邏點較多時大都求解結果皆為 NA，所以反而會建議使用 M1 求解方法，綜合上述說明不考慮路段可重複之整數規劃模型 (M1)兼具效能與效率，而模擬退火法 S1、S2、S4 亦可在短時間內得到近似最佳解。



## 第六章 結論與建議

本研究主要探討無人船的巡邏最短路徑規劃問題，在某一水域執行巡邏任務時，在考慮例如風力、風向、洋流等外在環境對船速的影響下，執行任務時所規劃的路徑，以達到總體花費最短的時間。我們建構數個整數規劃模型和演算法，並進行不同的整數模型與演算法的執行效率測試比較。本章第一節將總結本研究，而第二節則說明未來研究方向的建議。

### 6.1 結論

本研究設定了三種不同區域範圍大小並分別設定受不同的外在環境影響下每個時段的船速行經所耗用的時間、到達的指定巡邏地點位置和數目，來測試並比較本研究提出的整數規劃模型和演算法之求解效能與效率。

根據本研究測試設定的三種水域網路的結果如下：

小型水域網路被設定為  $5 \times 5$  型式、共 25 個格點的平面網路，並針對預計將被覆蓋(偵測)目的地點數目設定 3、5、8 等 3 組目的地個數，再產生 10 組隨機網路圖測試。測試結果發現：兩種貪婪演算法機制中，G1 的求解效能與效率皆優於 G2，因此建議採用 G1；四種模擬退火法機制中，S1、S2、S4 皆有很好的求解效能（平均 Gap 都在 1% 以下），而其求解效率亦不分軒輊地不錯，如還是硬要自此三者中挑選，建議混合式的 S4 演算法機制。在兩種整數規劃模型中，M1 的求解效能與效率皆優於 M2，因此建議 M1。

中型水域網路被設定為  $7 \times 7$  型式、共 49 個格點的平面網路，並針對預計將被覆蓋(偵測)目的地點數目設定 5、8、16 等 3 組目的地個數，再產生 10 組隨機網路圖測試。測試結果發現：兩種貪婪演算法機制中仍然推薦求解效能與效率較優的 G1；四種模擬退火法機制中，S1、S2、S4 皆有很好的求解效能與效率，但在較多目的地個數的測資中 S2 的表現略優於 S1 與 S4，因此建議 S2 演算法機制。在兩種整數規劃模

型中，M1 的求解效能（平均 Gap 都在 1%以下）與效率皆優於 M2，因此建議 M1。

大型水域網路被設定為 10x10 型式、共 100 個格點的平面網路，並針對預計將被覆蓋(偵測)目的地點數目設定 11、16、23 等 3 組目的地個數，再產生 10 組隨機網路圖測試。測試結果發現：兩種貪婪演算法機制中，G1 在目的地個數較少的測資有較好的表現，而 G2 在目的地個數較多的測資有比 G1 稍微好一些的效能（約好 1%左右），然而就效率而言則是 G1 表現較好（皆可在 1 秒內完成），因此仍較推薦使用 G1；四種模擬退火法機制中，S1、S2、S4 仍是皆有很好的求解效能與效率，但在較多地個數的測資中 S2 的表現則仍略優於 S1 與 S4，因此建議 S2 演算法機制。在兩種整數規劃模型中，M2 求解時間大部分都超過設定的 2 小時(7200 秒)，且針對較多地個數的測資皆無法於 2 小時內求得可行解，因此建議使用 M1。

綜合上述小、中、大等數種大小測資的數值測試結果，在整數規劃模型部分我們建議使用 M1，而演算法部分我們建議使用 G1 產生初始解，而用 S2(或 S4)可被用來收斂 M1 或 G1 的求解結果。

## 6.2 未來研究與建議

即使本研究已提出兩種整數規劃模型(M1、M2)以及六種求解演算法機制(G1、G2；S1、S2、S3、S4)以處理考慮風力影響之多期單台無人船的巡邏路線規劃，應該還有一些可再探討的議題，以下臚列出數個未來可嘗試的研究方向建議：

1. 本研究僅就貪婪演算法(Greedy Algorithm)和模擬退火法(Simulated Annealing)設計不同的鄰近解的求解方法，建議後續研究可嘗試使用或設計其它演算法（譬如鄰域搜尋法、禁忌搜尋法等）。
2. 本研究僅針對「單台」無人船規劃路線，因無人船大都體積小、速度快，可能較適宜多台協同作業。針對多台無人船的協同路線規劃問題，本研究有試圖修改 M1 整數規劃模型於附錄 A 中，可以預期該模型可能會因為變數與限制式數量而更無

法處理中大規模的問題。在此延伸問題的求解演算法發展部分，我們預期其難度會更大幅度地提高。

3. 若再進一步將無人船的續航力列入考量，可考慮探討以母船帶數台子船的方式處理本研究的巡邏路線問題，其難度又將再提高（但實用性更高）。
4. 本研究將水域切割成格點式的離散網路求解，應該也算是簡化問題。若是考慮原來連續的水域區間，應該更符合現實，但又該如何求解出平面連續空間的船軌跡，應該也會是個值得探究的困難研究題目。



## 參考文獻

- Arfaoui, A. (2017). Unmanned aerial vehicle: Review of onboard sensors, application fields, open problems and research issues. *International Journal of Image Processing (IJIP)*, 11(1), 12.
- Bertram, V. (2008). Unmanned surface vehicles –A survey. In *Proceedings of skibsteknisk selskab*, Copenhagen, Denmark.
- Bland, G., Miles, T., Logan, M., Motter, M., & Bretthauer, J. (2007, May). Mini UAVs for atmospheric measurements. In *2007 AIAA InfoTech at Aerospace Conference*, AIAA-2007-2759 (pp. 461-470).
- Breivik, M. , Hovstein, V. E. , & Fossen, T. I. (2008). Straight-line target tracking for unmanned surface vehicles. *Modeling, Identification and Control*, 29 (4), 131–149.
- Breivik, M. (2010). Topics in guided motion control of marine vehicles . Norwegian University of Science and Technology (Ph.D. thesis).
- Caccia, M. , Bono, R. , Bruzzone, G. , Spirandelli, E. , Veruggio, G. , Stortini, A. M. , et al. (2005). Sampling sea surfaces with SESAMO: An autonomous craft for the study of sea-air interactions. *IEEE Robotics & Automation Magazine*, 12 (3), 95–105.
- Caccia, M. , Bibuli, M. , Bono, R. , Bruzzone, G. , Bruzzone, G. , & Spirandelli, E. (2007). Unmanned surface vehicle for coastal and protected waters applications: The charlie project. *Marine Technology Society Journal*, 41 (2), 62–71.
- Casbeer, D. W., Kingston, D. B., Beard, R. W., & McLain, T. W. (2006). Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6), 351-360.
- Current, J., ReVelle, C., & Cohon, J. (1984). The Shortest Coverage path problem-an application of locational constraints to network design. *Journal of Regional Science*, 24(2), 161-183.
- Current, J., Pirkul, H., & Rolland, E. (1994). Efficient algorithms for solving the Shortest Coverage path problem. *Transportation Science*, 28(4), 317-327.

- Current, J. R., Velle, C. R., & Cohon, J. L. (1985). The maximum covering/shortest path problem: A multiobjective network design and routing formulation. *European Journal of Operational Research*, 21(2), 189-199.
- Current, J. R., & Schilling, D. A. (1989). The covering salesman problem. *Transportation Science*, 23(3), 208-213.
- DeBusk, W. M. (2010, April). Unmanned aerial vehicle systems for disaster relief: Tornado alley. In *Infotech@ Aerospace Conferences*.
- Garey, M., & Johnson, D. (1979). *Computers and intractability: A guide to the theory of np-completeness*. Freeman, San Francisco.
- Gendreau, M., Laporte, G., & Semet, F. (1997). The covering tour problem. *Operations Research*, 45 (4), 568-576.
- Golden, B., Naji-Azimi, Z., Raghavan, S., Salari, M., & Toth, P. (2012). The generalized covering salesman problem. *INFORMS Journal on Computing*, 24(4), 534-553.
- Goudey, C. A. , Consi, T. , Manley, J. , Graham, M. , Donovan, B. , & Kiley, L. (1998). A robotic boat for autonomous fish tracking. *Marine Technology Society Journal*, 32 (1), 47-53.
- Hachicha, M., Hodgson, M. J., Laporte, G., & Semet, F. (2000). Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, 27(1), 29-42.
- He, Q., Chu, C. H. H., & Camargo, A. (2012). Architectural building detection and tracking in video sequences taken by Unmanned Aircraft System (UAS). *Computer Technology and Application*, 3(9).
- Healey, A. J., Horner, D. P., Kragelund, S. P., Wring, B., & Monarrez, A. (2007). *Collaborative Unmanned Systems For Maritime and Port Security Operations*.
- Herwitz, S. R., Johnson, L. F., Dunagan, S. E., Higgins, R. G., Sullivan, D. V., Zheng, J., & Slye, R. E. (2004). Demonstration of UAV-based imaging for agricultural surveillance and decision support. *Computers and Electronics in Agriculture*, 44, 49-



- Hodgson, M. J., Laporte, G., & Semet, F. (1998). A Covering Tour Model for Planning Mobile Health Care Facilities in Suhum District, Ghana. *Journal of Regional Science*, 38(4), 621-638.
- Horcher, A., & Visser, R. J. (2004). Unmanned aerial vehicles: applications for natural resource management and monitoring. *Proceedings of the Council on Forest Engineering Proceedings*.
- Huang, Y., Thomson, S. J., Ortiz, B. V., Reddy, K. N., Ding, W., Zablotowicz, R. M., & Bright, J. R. (2010). Airborne remote sensing assessment of the damage to cotton caused by spray drift from aerially applied glyphosate through spray deposition measurements. *Biosystems Engineering*, 107(3), 212-220.
- Kucik, D. (2004). U.S. Patent no. 6,712,312 . Washington, DC: U.S. Patent and Trade-mark Office.
- Liu, Z., Zhang, Y., Yu, X., & Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41, 71-93.
- Miller, J. W. (2005). Report on the development and operation of an UAV for an experiment on unmanned application of pesticides. AFRL, USAF.
- Murphy, R. R. , Steimle, E. , Griffin, C. , Cullins, C. , Hall, M. , & Pratt, K. (2008). Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma. *Journal of Field Robotics*, 25 (3), 164–180.
- Naeem, W. , Xu, T. , Sutton, R. , & Tiano, A. (2008). The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 222 (2), 67–79.
- Naji-Azimi, Z., Renaud, J., Ruiz, A., & Salari, M. (2012). A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, 222(3), 596-605.

- Niblett, T. J., & Church, R. L. (2016). The Shortest Covering Path Problem: A New Perspective and Model. *International Regional Science Review*, 39(1), 131-151.
- Oliveira, W. A. D., Moretti, A. C., & Reis, E. F. (2015). Multi-vehicle covering tour problem: Building routes for urban patrolling. *Pesquisa Operacional*, 35(3), 617-644.
- Pastore, T. , & Djapic, V. (2010). Improving autonomy and control of autonomous surface vehicles in port protection and mine countermeasure scenarios. *Journal of Field Robotics*, 27 (6), 903–914.
- Puri, A. (2005). A survey of unmanned aerial vehicles (UAV) for traffic surveillance. Department of computer science and engineering, University of South Florida, 1-29.
- Rahim, F., & Sepil, C. (2014). A location-routing problem in glass recycling. *Annals of Operations Research*, 223(1), 329-353.
- Rasal, K. (2013). Navigation & control of an automated SWATH surface vessel for bathymetric mapping . Santa Clara University (Ph.D. thesis).
- Roberts, G. N., & Sutton, R. (Eds.). (2006). *Advances in unmanned marine vehicles* . The Institution of Engineering and Technology.
- Salari, M., & Naji-Azimi, Z. (2012). An integer programming-based local search for the covering salesman problem. *Computers & Operations Research*, 39(11), 2594-2602.
- Salari, M., Reihaneh, M., & Sabbagh, M. S. (2015). Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem. *Computers & Industrial Engineering*, 83, 244-251.
- Sarıçiçek, İ., & Akkuş, Y. (2015). Unmanned Aerial Vehicle hub-location and routing for monitoring geographic borders. *Applied Mathematical Modelling*, 39(14), 3939-3953.
- Shaelaie, M. H., Salari, M., & Naji-Azimi, Z. (2014). The generalized covering traveling salesman problem. *Applied Soft Computing*, 24, 867-878.
- Svec, P. , & Gupta, S. K. (2012). Automated synthesis of action selection policies for

unmanned vehicles operating in adverse environments. *Autonomous Robots*, 32 (2), 149–164.

Svec, P. , Thakur, A. , Raboin, E. , Shah, B. C. , & Gupta, S. K. (2014b). Target follow- ing with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots*, 36 (4), 383–405.

Tan, H., & Hill, R. (2013). The in-transit vigilant covering tour problem for routing unmanned ground vehicles. *Military Operations Research*, 243-250.

Turner, D., Lucieer, A., & Watson, C. (2011, April). Development of an Unmanned Aerial Vehicle (UAV) for hyper resolution vineyard mapping based on visible, multispectral, and thermal imagery. In *Proceedings of 34th International symposium on remote sensing of environment*.



## 附錄 A

### 多期風力 K 台無人船設計其整數規劃模型和目標函式與限制式

如圖 A-1 所示，除了考慮在受不同時段風力影響下，當有 2 台無人船同時執行巡邏任務時，以藍色線段和紅色線段區分為 2 台無人船所被分配的執行路徑，能在花費最少的時間內完成，其目標函式與限制式，如公式 A.1.1 至公式 A.1.18 所示。

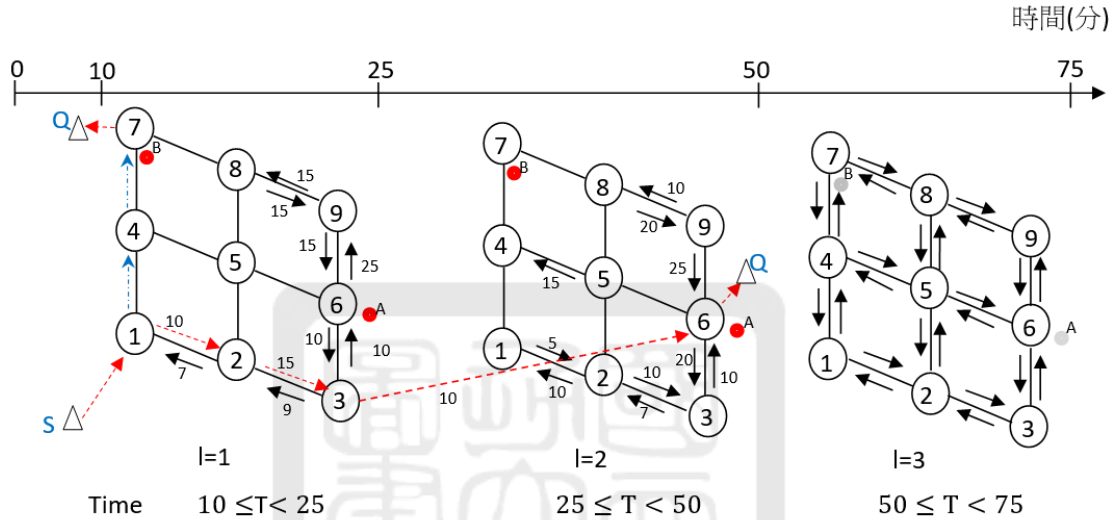


圖 A-1 多期風力 K 台無人船路徑規劃

#### 符號定義

$N$  : 網路的節點之集合  $i, j = 1, \dots, |N|$

$A$  : 網路經過的路徑的集合， $(i, j) \in A$

$L$  : 虛擬層的集合  $l = 1, \dots, |L|$

$K$  : 無人船的集合  $k = 1, \dots, |K|$

$S$  : 虛擬的起點

$U$  : 起點的節點集合

$Q$  : 虛擬的目的地

$V$  : 目的地的節點集合

$T_{\max}^{lk}$  : 在第  $l$  層的最大時限

$X_{ij}^{lk}$  : 在同一層第 $l$ 層的節線 $(i, j)$ 是否被經過，1代表是，反之為0

$Y_{ij}^{lk}$  : 在跨層第 $l$ 層的節線 $i$ 到第 $l-1$ 層的 $j$ 是否被經過，1代表是，反之為0

$C_{ij}^{lk}$  : 在第 $l$ 層的節線 $(i, j)$ 的行經路徑時間成本

$Z_i^{lk}$  : 在第 $l$ 層是否會經過 $i$ ，1代表是，反之為0

$T_i^{lk}$  : 在第 $l$ 層的抵達節點 $i$ 的時刻

其餘同CSP

### 數學模式

$$\text{Minimize } \sum_{k=1}^K \sum_{l=1}^L \sum_{(i,j) \in A} C_{ij}^{lk} X_{ij}^{lk} + \sum_{k=1}^{K-1} \sum_{l=1}^{L-1} \sum_{(i,j) \in A} C_{ij}^{lk} Y_{ij}^{lk} + \varepsilon \sum_{k=1}^K \sum_{l=1}^L \sum_{i \in N} T_i^{lk}$$

(A.1.1)

Subject to

- For layer 1 ( $l = 1$ ):

$$\sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) = \sum_{(j,i) \in A} X_{ji}^{lk} \quad \forall i \in N - U - V \quad l = 1 \quad k = 1, 2, 3, \dots, K \quad (\text{A.1.2})$$

$$Y_{iQ}^1 + \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) = \sum_{(j,i) \in A} X_{ji}^{lk} \quad \forall i \in V \quad l = 1 \quad k = 1, 2, 3, \dots, K \quad (\text{A.1.3})$$

$$\sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) = \sum_{(j,i) \in A} X_{ji}^{lk} + Y_{si}^0 \quad \forall i \in U \quad l = 1 \quad k = 1, 2, 3, \dots, K \quad (\text{A.1.4})$$

- For other layer ( $l = 2 \sim L$ , and  $Y_{ij}^{Lk} = 0$ ):

$$\sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) = \sum_{(j,i) \in A} (X_{ji}^{lk} + Y_{ji}^{(l-1)k}) \quad \forall i \in N - V \quad l = 2, 3, 4, \dots, L \quad k = 1, 2, 3, \dots, K \quad (\text{A.1.5})$$

$$Y_{iQ}^{lk} + \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) = \sum_{(j,i) \in A} (X_{ji}^{lk} + Y_{ji}^{(l-1)k}) \quad \forall i \in V \quad l = 2, 3, 4, \dots, L \quad k = 1, 2, 3, \dots, K \quad (\text{A.1.6})$$

- For source node :

$$\sum_{(S,i) \in A} Y_{Si}^{0k} = 1 \quad k=1,2,3,...K \quad (A.1.7)$$

- For destination :

$$\sum_{k=1}^K \sum_{l=1}^L \sum_{(i,Q) \in A} Y_{iQ}^l = 1 \quad (A.1.8)$$

- For Z variable(  $l=1 \sim L$ , and  $Y_{ij}^{Lk} = 0$ ):

$$Z_i^{lk} = \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) \quad \forall i \in N-V \quad l=1,2,3...L \quad k=1,2,3....K \quad (A.1.9)$$

$$Z_i^{lk} = Y_{iQ}^{lk} + \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) \quad \forall i \in V \quad l=1,2,3...L \quad k=1,2,3....K \quad (A.1.10)$$

$$MZ_i^{lk} \geq \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) \geq \varepsilon Z_i^{lk} \quad \forall i \in N-V \quad l=1,2,3...L \quad k=1,2,3....K$$

(A.1.9-A)

$$MZ_i^{lk} \geq Y_{iQ}^{lk} + \sum_{(i,j) \in A} (X_{ij}^{lk} + Y_{ij}^{lk}) \geq \varepsilon Z_i^{lk} \quad \forall i \in V \quad l=1,2,3...L \quad k=1,2,3....K$$

(A.1.10-A)

$$Y_{ij}^{Lk} = 0 \quad \forall (i,j) \in A \quad \forall k \in K(k=1,2,3...,K) \quad (A.1.11)$$

(時間限制式)

- IF  $X_{ij}^{lk} = 1$  then  $T_j^{lk} = (T_i^{lk} + C_{ij}^{lk})$

$$(T_i^{lk} + C_{ij}^{lk}) - M(1 - X_{ij}^{lk}) \leq T_j^{lk} \leq (T_i^{lk} + C_{ij}^{lk}) + M(1 - X_{ij}^{lk}) \quad \forall l \in L \quad \forall i \in N \quad \forall k \in K$$

(A.1.12)

- IF  $Y_{ij}^{lk} = 1$  then  $T_j^{(l+1)k} = (T_i^{lk} + C_{ij}^{lk})$

$$(T_i^{lk} + C_{ij}^{lk}) - M(1 - Y_{ij}^{lk}) \leq T_j^{(l+1)k} \leq (T_i^{lk} + C_{ij}^{lk}) + M(1 - Y_{ij}^{lk}) \quad \forall l \in L - \{1\} \quad \forall i \in N \quad \forall k \in K$$

(A.1.13)

- IF  $Z_i^{lk} = 1$  then  $T_{\max}^{(l+1)k} \leq T_i^{lk} \leq T_{\max}^{lk}$

$$T_{\max}^{(l-1)k} \times Z_i^{lk} \leq T_i^{lk} \leq T_{\max}^{lk} \times Z_i^{lk} \quad \forall l \in L \quad \forall i \in N \quad \forall k \in K \quad (A.1.14)$$

- IF  $Y_{Sj}^{0k} = 1$  then  $T_j^{1k} = 0$

$$-M(1 - Y_{Sj}^{0k}) \leq T_j^{1k} \leq M(1 - Y_{Sj}^{0k}) \quad \forall j \in U \quad \forall k \in K(k=1,2,3...,K) \quad (A.1.15)$$

(覆蓋限制式)

$$\sum_{k=1}^K \sum_{l=1}^L \sum_{i \in N} b_{ir} Z_i^{lk} \geq 1 \quad \forall r \in N \quad (\text{A.1.16})$$

$$b_{ir} = \begin{cases} 1 : \text{if node } i \text{ covers area } r \\ 0 : \text{o.w.} \end{cases}$$

(其它限制式)

$$T_{\max}^0 = 0, T_{\max}^1 = 60, T_{\max}^2 = 120, T_{\max}^3 = 180 \quad (\text{A.1.17})$$

$$X_{ij}^{lk} = \{1, 0\}, Y_{ij}^{lk} = \{1, 0\}, Z_i^{lk} = \{1, 0\} \quad (\text{A.1.18})$$

其中公式 A.1.1 為整數規劃模型的目標函數，代表使用  $K$  台無人船完成必須巡邏的地點，其總路徑所花的費時間成本最小化，與多期風力單台無人船之最短覆蓋路徑相較，主要的差異在於變數多加了一個無人船編號的索引  $k$ 。

