國立成功大學

工業與資訊管理學系碩士班

碩士論文

考量靜態標靶節線偵測機率之搜救隊與無人機群最佳協同搜救路線規畫研究

Optimal Human-UAV Collaborative Search Path Planning for an Immobile Target with Edge Detection Probabilities

研 究 生： 陳 彥 瑋

指導教授： 王 逸 琳 教授

中華民國一百零九年五月

doi:10.6844/NCKU202002825

國立成功大學

碩士論文

考量靜態標靶節線偵測機率之搜救隊與無人機群最佳協同搜救路線規劃研究

Optimal Human-UAV Collaborative Search Path Planning for an Immobile Target with Edge Detection Probabilities

研究生：陳彥瑋

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
丁慶榮
[illegible]

指導教授：王逸琳

系(所)主管：[illegible]

中　華　民　國　109　年　5　月　29　日

doi:10.6844/NCKU202002825

摘要

近年來，不時發生登山客因天災人禍等諸多因素受困途中，導致政府耗費不少人力物力搜救。由於搜救行動有其立即與危險性，搜救單位往往需要立即調派大量人力或直升機，在不易行走的地區沿路搜救，以期能盡快找到停留於某路段上的靜態標靶(譬如受傷無法行走的登山客)。即使目標可能在的路段與其存在機率為已知，為了在這些可能的路段上搜索，傳統只能以徒步方式沿路逐步前進，而直升機雖能快速抵達欲搜索之路段上方，但成本過高，且可能因障礙物遮蔽而難以貼近地面搜索。此時若使用無人機(Unmanned Aerial Vehicle，UAV)搜索，其較快的三維空間移動能力可跳脫傳統人力在二維網路上的路段移動限制，也較直升機有更低的成本與貼地搜索之優勢；只要續航力足夠，同時指派多架無人機分工協同搜索，應可更快完成搜索任務。為讓無人機群之搜索續航力無虞，本研究提議讓徒步的搜救隊攜帶無人機之充換電設備，權充類似地面載具 (Ground Vehicle，GV)之「動態充換電站」角色，讓無人機能在其電力耗盡前能與同步搜救的人員會合，以充換電池繼續搜救任務，形成一個「無人機-地面載具的二階層協同式路徑規劃問題」(Two-Echelon GV and UAV cooperated Routing Problem，2E-GU-RP)。本研究假設在欲搜索之網路上，某單一靜態標靶在其可能停留之各段節線上之存在機率、無人機之電量及其不同移動模式之電力消耗與速度、與搜救隊的移動速度皆為已知，以時空(time-space)網路為架構，建構「可共乘」和「非共乘」兩種整數規劃模式，來規劃多架無人機、多位搜救隊的最佳協同移動路線，以最短之總期望搜索時間來遍訪該標靶可能存在之所有節線，而在實際進行數值測試後，我們發現兩種數學模式皆需要花費大量的求解時間，且無法處理中大型網路，因此本研究再開發啟發式演算法以在短時間內獲得高品質的近似最佳解(nearly optimal solution)，參考過去針對多鄉村郵差問題(K-rural postman problem，KRPP)常使用的編碼方式，將需要尋訪的標靶節線分配給不同的無人機和搜救隊，

doi:10.6844/NCKU202002825

形成一組編碼，設計概念類似於貪婪演算法的解碼演算法，先規劃無人機的移動路徑，並記錄其需要充換電的節點與時間，再規劃動態充換電站的移動路徑，以提供無人機電力和尋訪標靶節線，將編碼轉化為實際的移動路徑。最後再利用多種區域搜尋演算法交換迭代，以求解最小化期望搜尋時間。透過數值分析，可證明即使是不同的區域搜尋演算法組合，啟發式演算法在面對中大型網路時，皆可在短時間內求得近似最佳解。因此我們推薦使用本論文發展之數學模式處理規模較小的網路，而中大型的網路則可使用本研究開發之啟發式演算法處理之。

關鍵字：無人機；搜救；動態充換電站；整數規劃 ；啟發式演算法

doi:10.6844/NCKU202002825

Optimal Human-UAV Collaborative Search Path Planning for an Immobile Target with Edge Detection Probabilities

Yen-Wei Chen

I-Lin Wang

Department of Industrial and Information Management

SUMMARY

Unmanned Aerial Vehicles (UAVs) are mobile, faster, remotely controllable for search and rescue (SAR) missions. This thesis investigates a SAR mission by SAR teams and a fleet of UAVs to search over arcs of positive probability to identify an immobile target (e.g., an injured hiker) with minimum expected time. To deal with the range anxiety, we assume each SAR team serves as a mobile battery-swapping station, similar to the Ground Vehicle (GV) role in the 2-Echelon UAV & GV Routing Problem (2E-GU-RP). In particular, while a SAR team searches along hiking routes on foot, a UAV can be launched and retrieved at some nodes. A retrieved UAV will become fully charged right after its rendezvous with teams. Both teams and UAVs have two moving modes: a slower searching mode, and a faster passing mode. The searching mode searches the target along arcs, but a UAV on the passing mode might fly between nonadjacent nodes. Two integer programming formulations, carried model and non-carried model are proposed on a time-space network, which can only deal with smaller networks. A local search based heuristic algorithm (LS) is designed to calculate solutions of good qualities in a short time. Our computational experiments' numerical results indicate that our heuristic with the 2-opt and cross-exchange LS mechanisms can get a nearly optimal solution in a short time.

Key words: Unmanned Aerial Vehicle, Search and Rescue, Ground Vehicle, Integer Programming, Heuristic Algorithm

doi:10.6844/NCKU202002825

INTRODUCTION

We focus on the Search and Rescue (SAR) mission for a single immobile target over some remote area, where the probability of finding the target on each arc is estimated beforehand. It is difficult, dangerous, and time-consuming for a SAR team to conduct such a mission on foot. Nevertheless, UAVs are perfect for such a task, since they are more mobile, faster, and can cover a broader range in a shorter time if their range anxiety can be resolved. In this thesis, we equip the SAR teams with UAVs for the collaborative search. The UAVs can be launched and retrieved on some nodes. Each UAV has two moving modes: a slower searching mode to search target along ground arcs, and a faster passing mode to fly between any adjacent nodes not necessarily connected by ground arcs, within its flight range. Before a UAV runs out of battery, it rendezvous with some team at a node for battery swapping. In some sense, a team serves as a ground vehicle (GV), and the entire mission becomes a 2-Echelon UAV & GV Routing Problem (2E-GU-RP) in some recent logistics distribution literature. Such a setting has appeared in practice, but we may arguably the first, to the best of our knowledge, to tackle it by mathematical programming models and solution methods.

Our problem is more challenging than the K-Rural Postman Problem (KRPP) because we not only need to pass through all the arcs of positive target probabilities, we also need to record the arrival time of passing a node to calculate the expected searching time. The probability values on different arcs affect path planning. As a result, it is NP-hard. Besides, considering the range of anxiety and two moving modes for both UAVs and teams further complicates the problem.

MATERIALS AND METHODS

To calculate an exact optimal solution, we first formulate this problem as an integer program (IP) over a time-space network. Our IP formulations can avoid subtours and can incorporate with both searching and passing modes using parallel searching arcs and passing arcs, respectively, in the time-space graph. To rendezvous with some team for battery swapping, we set a binary variable to record if UAVs and GVs recharge when they meet at the same time and node. We also developed two IP formulations, carried mode, and non-carried mode. The former one considers a GV to carry a UAV and move together, but the latter one assumes both to move independently. The carried model can consider more situations but consume more computational time. Unfortunately, Both IP formulations can only deal with small scale networks. Therefore, we further design

doi:10.6844/NCKU202002825

heuristic algorithms based on the Local Search (LS) mechanism, which encodes each routing solution as a sequence of arcs to be traversed. We exploit the 2-opt*, Or-opt, and cross exchange heuristics to search the neighborhood for better solutions, as proposed in Hashimoto and Yagiura (2008). To make our LS more effective, we prefer selecting the UAV routings and GV routings of the longest and shortest completion time to conduct the exchange operation. Our decoding algorithm strategy is similar to the Greedy Algorithm, which contains two stages, the UAV scheduling stage and the GV scheduling stage. The former stage schedules UAV routes based on its remaining power to search for the target arc of the shortest expected time. If a UAV is out of power, recording the location and time to recharge. In the GV scheduling stage, we would like to schedule a GV route to search for a target arc and swap UAV battery.

RESULTS AND DISCUSSION

We have some steps to generate problem instances: First, we generate a random grid network. Then, we remove some arcs yet ensure the connectivity of the network. Finally, we remove some nodes of two edges to increase the variety of networks. We test two different settings in mathematical models and three different settings in heuristic algorithms. Based on our testing, we have found that all our proposed IP models, carry or non-carry modes, can only deal with small-scale problems. The non-carry mode is a better choice in most situations. On the other hand, our heuristic algorithms can calculate nearly optimal solutions in a shorter time, and the setting without using or-opt but with IP models for rendezvous points has much better performance than the other two settings. We also provide some managerial insights in the UAV purchase strategy. In short, we observe that the quantity to purchase is more crucial than other factors such as the range.

CONCLUSIONS

To enhance the SAR efficiency and effectiveness, we propose a collaborative search path planning problem for both the teams and the UAVs. By making the team serve as a mobile battery-swapping station for UAVs, we can extend their range. Our problem settings are progressive and expected to be applicable in the near future, as the progress of the UAV and sensing technologies evolves quickly recently.

We have proposed two integer program formulations and several LS heuristic algorithms to solve this NP-hard problem. The results of our computational experiments show that IP models can only deal with problems of small sizes. For larger cases, our

doi:10.6844/NCKU202002825

heuristic algorithms can calculate a nearly-optimal solution in a shorter time. If the budget is limited, purchasing more cheap UAVs would be more effective than fewer long-range UAVs. For future related research topics, we suggest considering to rendezvous in the middle of arcs rather than nodes only. More investigation on theories and techniques of developing mathematical models or solution methods that produce exact optimal solutions are also very needed.

doi:10.6844/NCKU202002825

誌謝

將這一頁獻給所有曾幫助過我的師長、家長、朋友們。時光匆匆，2 年前踏入成大如今也將要畢業離開，在研究所這 2 年的時間，我認為是自己成長最多的階段，謝謝逸琳老師的指導，讓我有機會參與許多比賽、專案，順利完成論文，並在過程中學習老師對於事情的處理態度與思考方式，也特別謝謝老師帶我們去美國參加比賽，那是一次難忘的回憶，希望學弟妹們未來也能有機會再去。我相信這些寶貴的經驗，對於未來都非常有幫助。

謝謝爸媽長久以來無怨無悔的付出，讓我能無後顧之憂地完成大學與研究所的學位，順利完成論文，我真的非常的幸運、幸福。此外，也謝謝你們當初讓我來就讀成大工資管，我在這接觸到很多領域也學習到很多有趣的知識。

後謝謝錢博、吳博、政達、大哥等等陪我打麻將、玩遊戲、吃飯，度過研究所兩年的朋友們。

最後謝謝自己，這兩年的進步，也源自於自己的努力與堅持，希望未來可以堅持下去，研究所的畢業不是終點，未來還有很長的道路要走，希望一切順利。

doi:10.6844/NCKU202002825

目錄

摘要 I
目錄 VIII
圖目錄 X
表目錄 XII
第一章 緒論 1
1.1 研究背景與動機 1
1.2 本研究問題之設定與目的 2
1.3 本研究與其它相關研究之比較說明 3
1.4 研究範圍 5
1.5 論文架構 5
第二章 文獻回顧 6
2.1 無人機-載具的協同作業之相關研究 6
2.1.1 無人機-載具同步規劃 7
2.1.2 無人機-載具非同步規劃 11
2.2 搜救路線規劃相關研究 12
2.2.1 搜索動態標靶 13
2.2.2 搜索靜態標靶 14
2.3 節線途程問題相關文獻 16
2.4 小結 18
第三章 結合動態充換電站之無人機群搜索路徑 之數學規劃模式 20
3.1 問題描述 20
3.2 問題假設 24
3.3 網路結構 24

doi:10.6844/NCKU202002825

3.4 參數與變數定義 26
3.5 最小化目標被尋獲之期望搜索時間 29
3.6 可共乘之時空數學模式 31
3.7 非共乘之時空數學模式 38
3.8 小結 41
第四章 結合動態充換電站之無人機群搜索路徑 之求解演算法設計 42
4.1 編碼方式(Encode) 42
4.2 解碼演算法(Decoding Algorithm，DA) 43
4.2.1 演算法步驟 44
4.2.2 範例說明 50
4.3 建置初始解 52
4.4 區域搜尋法(Local Search，LS) 58
4.5 小結 66
第五章 數值分析 68
5.1 測試資料參數設定 68
5.2 測試網路圖設定 71
5.3 結合動態充換電站之無人機群搜索路徑規劃問題之測試 72
5.3.1 數學規劃模式比較 72
5.3.2 啟發式演算法測試 75
5.3.3 無人機選購策略 79
5.4 小結 80
第六章 結論與未來研究建議 82
6.1 結論 82
6.2 未來研究 83
參考文獻 85

doi:10.6844/NCKU202002825

圖目錄

圖 1.1 搜救路線之網路圖 3
圖 2.1 無人機-地面載具二階層式路徑規劃(Luo et al., 2017)..... 8
圖 2.2 考慮動態充換電站之無人機路徑規劃(Yu et al., 2018) 10
圖 2.3 TSP-D (Ha et al., 2018).....10
圖 2.4 待搜索網路圖 Campbell et al.(2018).....18
圖 3.1 本研究使用網路之示意圖 21
圖 3.2 無人機可行走之網路圖 22
圖 3.3 動態充換電站可行走之網路圖.....22
圖 3.4 無人機與動態充換電站需共乘之特殊情況.....23
圖 3.5 原問題之網路(呂昀軒,2019).....25
圖 3.6 以搜索節線與移動節線建構之網路(呂昀軒,2019) 25
圖 3.7 本研究非共乘之網路結構 26
圖 3.8 可共乘之無人機時空網路圖.....32
圖 3.9 非共乘之時空網路圖 38
圖 4.1 演算法流程圖.....44
圖 4.2 DA 說明範例圖 51
圖 4.3 K-means 範例(分群前).....54
圖 4.4 K-means 範例(分群後).....54
圖 4.5 動態充換電站覆蓋路徑規劃.....56
圖 4.6 兩組編碼 59
圖 4.7 cross change 示意圖(Taillard et al., 1997).....60

doi:10.6844/NCKU202002825

圖 4.8 2-opt* vs 2-opt(Potvin et al., 1996).....61
圖 4.9 Or-opt 交換示意圖 64
圖 4.10 純 2-opt*執行流程.....65
圖 4.11 純 cross exchange 執行流程 65
圖 4.12 先 cross exchange 再 2-opt*執行流程.....66
圖 4.13 先 2-opt*再 cross exchange 執行流程 66
圖 4.14 加入 or-opt 的先 2-opt*再 cross exchange 執行流程.....66
圖 5.1 網格圖(未刪除部分節線).....68
圖 5.2 網格圖(刪除部分節線).....69
圖 5.3 網格圖(刪除部分節點).....70
圖 5.4 網格圖(含無人機移動節線).....70
圖 5.5 非共乘數學模式測試(求解品質與效率).....73
圖 5.6 可共乘數學模式測試(求解品質與效率).....75
圖 5.7 數學演算法測試-第一種設定(求解品質與效率).....76
圖 5.8 數學演算法測試-第二種設定(求解品質與效率).....77
圖 5.9 Network4 第二種設定收斂速度.....78
圖 5.10 數學演算法測試-第三種設定(求解品質與效率).....78
圖 5.11 台灣日月潭網路圖 80

doi:10.6844/NCKU202002825

表目錄

表 2.1 無人機與載具協同作業相關文獻比較 12
表 2.2 搜救路線規劃相關文獻比較.....15
表 4.1 編碼範例 43
表 4.2 無人機路徑規劃流程 46
表 4.3 無人機充換電點順序 48
表 4.4 動態充換電站負責節點 48
表 4.5 動態充換電站移動組合 48
表 4.6 動態充換電站路徑規劃流程.....48
表 4.7 cross exchange 流程.....62
表 4.8 2-opt 流程.....63
表 4.9 Or-opt 演算法流程 64
表 5.1 測試資料參數設定 71
表 5.2 網路圖資訊.....72
表 5.3 數學演算法參數設定 76
表 5.4 無人機性價比分析 80

doi:10.6844/NCKU202002825

第一章 緒論

第一章首先介紹研究背景與動機，說明為何我們將無人機(Unmanned Aerial Vehicle，UAV)使用於搜救任務上，以及以搜救隊權充動態充換電站的優點；在研究問題設定部分解釋了本研究的場景設定考量，比較傳統搜救與無人機搜救的差異；接著我們說明了本研究與「無人機-地面載具的二階層協同式路徑規劃問題」(Two-Echelon GV and UAV cooperated Routing Problem，2E-GU-RP)、「搜救路線規劃問題」(Search Path Planning Problem，SPPP)，和「多鄉村郵差問題」(K-Rural Postmen Problem，KRPP) 等相關的數個議題之差異處；最後再說明研究範圍及論文架構。

1.1 研究背景與動機

搜救任務往往需要救難人員跟時間賽跑，黃金 72 小時搜救時間，即是代表越早搜救到目標，越能提高失蹤者的存活機率。然而為了減少搜救時間，救難人員同時也承受著極大的危險，台灣近年來也不時傳出救難人員在搜救過程中發生意外的新聞。例如 2019 年 10 月，消防員在無受困者的火場中搜救時，被坍陷的鐵皮困在火場，導致兩名消防員殉職；2017 年 3 月，搜救隊在山區的搜救過程中墜崖殉職。上述搜救任務，若能先派無人載具搜索，應可更快確認出受困者位置、狀況與其周遭環境，協助搜救隊以更安全迅速方式完成搜救任務。

近年來，無人機的研究越來越多，因其有高移動速度、低成本、易操控、較少空間限制等特性，可廣泛地使用在農業、測量、運輸和軍事用途等各種領域。雖然無人機的相關研究蓬勃發展，但其在搜救任務的使用上卻常僅流於紙上談兵，針對諸如多機協同搜索、考量充換電續航力之路線規劃等問題，文獻中大多仍僅以啟發式演算法處理，使用數學理論基礎來分析與設計數學模型的研究則相對少見。

doi:10.6844/NCKU202002825

本研究乃探討在山徑網路中，如何調度指揮多架無人機，以有效地越早搜索出靜態標靶（譬如受傷的登山客）。相較於指揮搜救隊於地面沿著山徑移動的傳統搜尋方式，無人機有較快的速度以及搜救能力，譬如可透過紅外線探測或是相機拍攝即時照片、掛載救生圈或是應急照明、擴音廣播、繩索等設備。此外，無人機具備三維空間移動能力，針對懸崖峭壁、山谷等較難移動的區域進行搜尋，可以避免搜救隊為了搜救目標而發生危險，讓搜救隊專注於搜索平面路段以及救出受困者的任務。我們預期若能適度指揮多台無人機同步搜索，應可提高搜救效率，且較徒步緩慢前進的搜救隊更能先行發現靜態標靶，之後再交由直昇機或人力背行等方式救出受困者。然而，無人機最大的缺點在於其續航力有限，且在偏遠地區無法隨意設立充換電站供其充換電。只要能解決其續航力問題，無人機應能成為搜救隊極佳的輔助搜索利器。因此本論文建議搜救隊攜帶無人機充換電設備（譬如電池），如此即可讓行進中的搜救隊同時擔任類似地面載具(Ground Vehicle，GV)般的「動態充換電站」角色，其與無人機的路線規劃問題將形成一個 2E-GU-RP。亦即，在搜救隊徒步於山徑中執行搜救任務的同時，也能施放多台無人機同步搜索其它路段；而當無人機續航力將耗盡前，只要回到搜救隊的所在處會合，即可立即抽換電池，再以滿電狀態返航繼續其搜索任務，期以最小化能搜索到靜態標靶的期望時間。

1.2 本研究問題之設定與目的

傳統執行搜救任務時，首先要指派指揮官，再由指揮官將所有人力分組，決定搜救路線，但其路線往往都是根據指揮官的主觀經驗，並無使用最佳化路徑的方法進行規劃。本研究乃同時規劃搜救隊與無人機之路徑，且在無人機的路徑規劃中亦加上其電力損耗因素。欲增加無人機之續航力，勢必讓無人機能夠在途中於充換電站設施進行充換電。過去之相關研究大多假設充換電站設施之位址為固定，但在此搜救問題的偏遠地區情境中，現實上因其地域限制、技術困難等諸多

doi:10.6844/NCKU202002825

因素，幾乎不可能在短時間內設置該類定址類之充換電站設施。因此，本研究假設救難人員在搜救時會攜帶無人機之電池，以救難人員權充「動態充換電設施」，只要無人機與搜救隊同時在節點上會合，便可以立即更換電池，增加無人機之續航力。本研究可視為考量無人機之續航力下，進行多組救難人員與多台無人機之二階層式路徑規劃(2E-GU-RP)，目標為最小化搜索到靜態標靶之期望搜索時間。

此外，傳統人力搜救，因僅能使用徒步方式沿路搜索，會因行進速度太慢、網路路段連結方式等許多限制，而導致搜救進度過慢。以圖 1.1 為例，若山區網路為一環狀圖，而靜態標靶在節線(3,4)時；若搜救隊自點 1 出發，欲以最短路徑前往節線(3,4)時，受限於原網路的平面連結，僅能經過節線(1,2)、(2,3)，才能抵達(3,4)。但若使用無人機，只要無人機剩餘電量足夠，即可直接行走立體空間的飛行路段(1,3)而直達點 3，再進入節線(3,4)，因而省下許多時間。由此可知，使用無人機搜救可以大幅縮短搜救時間，但也同時造成網路之可能路徑分段數大幅增加，選擇也會變多。此外由於無人機的電量限制，還要考慮每一台無人機需在何時、何地與何組搜救隊會合、更換電池等，導致此 2E-GU-RP 變得更加困難。

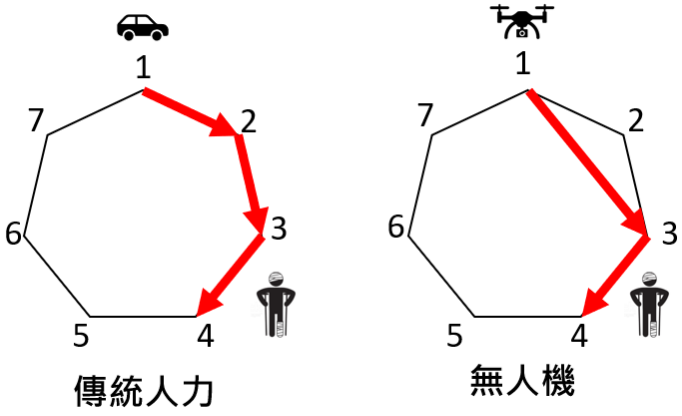


圖 1.1 搜救路線之網路圖

1.3 本研究與其它相關研究之比較說明

本研究將探討救難人員及無人機群之二階層式路徑規劃，我們將靜態標靶可

doi:10.6844/NCKU202002825

能出現的位置設於「節線」上，救難人員以及無人機在行經每條節線時皆有「搜索模式」、「移動模式」此兩種行進模式。若以搜索模式尋訪完此節線後，該節線便不用再被救難人員或無人機以搜索模式經過，因為我們假設搜索模式都不會遺漏標靶。由於無法確定靜態標靶會出現在哪一條節線上，因此救難人員與無人機需尋訪所有可能存在標靶（亦即其存在機率為正數）之節線，類似於「鄉村郵差問題」，並記錄行經標靶節線之時間點，用以計算期望搜索時間，將目標式定為最小化期望搜索時間。

相關文獻中，與本研究有關係的幾個議題為「無人機-地面載具的二階層協同式路徑規劃問題」(2E-GU-RP)、「搜救路線規劃問題」(SPPP)，與「鄉村多郵差問題」(K Rural Postmen Problem，KRPP)。

2E-GU-RP 類似傳統二階層式路徑規劃(Two-Echelon Routing Problem，2E-RP)，2E-RP 通常探討在物流配送系統中，規劃各物流中心車輛的配送路線，因各車輛皆有自己一層的網路圖，多車輛則構成多階層網路問題。而 2E-GU-RP 則是考慮無人機及載具之二階層式協同路徑規劃問題，與 2E-RP 不同之處在於無人機有電力及感應器的限制(不能離操作人員太遠)，其與載具可能有共乘及充換電的配合。在本研究因為無人機與搜救隊移動之網路圖有所差異，因此需區分為兩階層，如此各階層中的節線可設定不同的移動時間(譬如無人機常比搜救隊速度更快)。搜救隊及無人機群除了皆有共同的搜救目標外，因無人機必須和搜救隊 會合才可以充換電，所以兩者的路徑間存在著相依性，本文探討在兩者有共同目標與電力限制的情況下，如何規劃兩階層之移動路徑。

搜救路線規劃有別於一般的路徑規劃，目標通常為最小化移動時間或距離，SPPP 的目標可能為尋訪單一或多個標靶，標靶的類型又可分為靜態或是動態且隨機或均勻的分布於節線或節點上。本研究則針對搜救隨機分布於節線上之單一靜態標靶問題，探討如何指揮無人機與動態充換電站，以最小化期望時間尋訪所有可能存在目標之節線。

doi:10.6844/NCKU202002825

KRPP 則是探討如何以最短的移動路徑尋訪所有必須經過的節線，而本問題與 KRPP 一樣必須尋訪中所有靜態標靶可能存在之節線，但目標式與 KRPP 所廣為使用的最短移動路徑不同，而是再乘以節線上的標靶存在機率而成的期望搜索時間最小化為目標，而期望搜索時間計算公式的定義，將會在第三章詳細說明。

1.4 研究範圍

本研究的研究範圍如下:

1. 已知每台無人機之最大電量、無人機和搜救隊可移動之網路圖、各節線尋獲靜態標靶之機率、無人機與搜救隊使用不同模式在各節線之移動時間，在無人機電力限制之下尋訪所有可能存在靜態標靶之節線，建立 L 組搜救隊及 K 台無人機群之路徑規劃，目標為最小化期望搜索時間。
2. 開發「可共乘」與「不可共乘」兩種數學模式，適用於不同的情況。
3. 開發啟發式演算法在短時間內得到近似最佳解(nearly optimal solution) 。
4. 以多組不同大小與結構之網路圖，並且考慮不同數量的搜救隊和無人機的組合，測試兩種數學模式與啟發式演算法之求解效能與效率。
5. 考慮固定預算下的不同價格與電量無人機的選購策略。

1.5 論文架構

本論文之架構如下：第二章為文獻回顧，針對無人機-載具的協同作業、搜救路線規劃、節線途程問題(Arc Routing Problem，ARP)進行探討；第三章討論考慮電力限制下，無人機群與動態充換電站之二階層路徑規劃問題，定義最小化期望搜索時間之公式，並針對無人機與動態充換電站是否可共乘提出不同的數學模式；第四章提出啟發式演算法，改善數學模式對於中大型網路圖無法在短時間內取得可行解之問題；第五章對數學模式和啟發式演算法進行數值測試與分析，並在管理層面下，考慮無人機的選購策略；第六章總結本論文，並列出針對未來研究方向的建議。

doi:10.6844/NCKU202002825

第二章 文獻回顧

本章共分為三小節，首先針對無人機-載具的協同作業進行探討，包含 2E-GU-RP 在內，將探討在不同應用或限制下，先前研究如何解決此類問題；接著會討論有關 SPPP 的相關文獻，分類為靜態與動態標靶再進行介紹；最後針對 ARP 進行探討，主要介紹節線途程中的鄉村郵差問題 RPP。

2.1 無人機-載具的協同作業之相關研究

無人機雖有移動速度快、高機動性等優勢，卻同樣有著一些缺點，例如感應器的限制、電量限制等，需要被克服。Otto, Agatz, Campbell, Golden, and Pesch (2018) 除了整理許多無人機在不同應用的文獻，該篇和 Ferrandez, Harbison, Weber, Sturges, and Rich (2016) 皆提到無人機和載具的協同作業，會比單獨使用無人機或載具還要節省能源及成本。在此類的研究中，無人機的速度通常較快，但由於電池因素，其移動距離較短；而其它交通工具，如船、車子、人等，皆可通稱為載具。載具的移動速度較慢，卻可以移動較久、較長的距離，而無人機可以透過停留在載具上進行「共乘」，達到長距離的移動並減少電量的消耗。此時載具可被視為「動態充換電站」，當無人機與載具交會便可將其充換電，解決無人機的電量限制，且操作人員透過載具進行移動，也較不會違反感應器的距離限制。

其中無人機與載具的協同作業，廣泛地被討論在各類應用，如 Garone, Naldi, Casavola, and Frazzoli (2010) 將其用於搜救、Mathew, Smith, and Waslander (2015) 應用在運輸問題、Tavana, Khalili-Damghani, Santos-Arteaga, and Zandi (2017) 應用在最後一哩運輸問題上、Tokekar, Vander Hook, Mulla, and Isler (2016)則是使用在建築測量，協同作業的研究範疇包含兩者的路徑規劃、設備的改善、訊號的傳遞方式等，而本研究則是將無人機與載具應用在搜救上，討論兩者的移動路徑規

doi:10.6844/NCKU202002825

劃，因此後續的文獻探討也是以路線規劃為主。

Luo, Liu, and Shi (2017)將無人機及載具的路徑規劃定義為 2E-GU-RP，和 2E-RP 相同之處為有兩階層的路徑規劃，不同之處在於 2E-RP 的第一階層為由大型車輛將產品從工廠載到倉庫，第二階層則是使用小型車輛將產品從倉庫載到商家，兩階層在空間上的路徑規劃彼此獨立、互不影響；在時間上同樣也是，第一階層大多有較長周期時間（例如一個月），第二階層則為較短周期時間（例如一周），兩階層的時間並不同步，然而 2E-GU-RP 為考慮無人機及載具的二階層式路徑規劃，因為無人機通常有電力及感應範圍的限制，需考慮與載具的共乘或充換電的配合，甚至其起飛降落所需的準備時間亦需和載具進行配合，故兩階層皆須在時間、空間上相依，後續亦會有探討許多 2E-GU-RP 的相關文獻。而根據規劃兩者移動路徑的方式，可再分為「同步規劃」或是「非同步規劃」兩類。其中，「同步規劃」指的是要同時考慮無人機和載具兩者路徑的依存關係，反之「非同步規劃」則指個別路徑最佳化即可，不用一起考量。舉例來說，若是無人機和載具有共同的任務要去分配，或無人機和載具有共乘、充換電等互動或因通訊設備而不能距離太遠等限制，都可能導致無人機需要停留在某個點進行等待，或是規劃不同的路徑使兩者交會。而若要使無人機和載具的移動方式同時最佳化，則需要更全面地同步規劃兩者的路徑。本研究因為無人機必須與載具共同搜索靜態標靶可能所在之節線，目標式與無人機和載具以搜索模式行經節線之時間相關，且兩者之間亦有共乘及充換電的搭配，因此屬於同步規劃的範疇，希望全面性地最佳化無人機與載具兩者互相配合的路徑。

2.1.1 無人機-載具同步規劃

Sujit, Sousa, and Pereira (2009) 探討無人機與水下載具的配合，以載具為主、無人機為輔，無人機從基地出發後會收集資訊，提供給載具去規劃新的探索任務。

doi:10.6844/NCKU202002825

Luo et al. (2017) 探討 2E-GU-RP，如圖 2.1 所示。其中，載具和無人機有不同的移動網路，標靶存在於無人機所在之移動網路的節點上，故會讓無人機和地面載具共乘到一定點後，放出無人機去完成任務，再回來進行充換電。該篇論文設計 0-1 整數規劃模式在考慮無人機的電力限制下，規劃無人機和地面載具的移動路徑，以最小化時間用無人機尋訪所有標靶節點，故亦可視為一個旅行銷售員問題(Traveling Salesman Problem，TSP)。該研究發展兩個啟發式演算法，皆使用基因演算法進行疊代，只是切入的角度有所不同。其第一個演算法先建立無人機所有尋訪標靶的路徑，但因為無人機需考慮電力因素，必須靠中途補充換電力才能沿此路徑尋訪完所有標靶，因此必須在電力耗盡前將某些節線切斷，選擇無人機與載具要在哪個點進行交會以補充換電力，如此才能建構出無人機的路徑；而其地面載具的路徑，則是根據剛剛選擇的交會點再去規劃，此演算法的概念也被應用於本研究的解碼演算法當中。其第二個演算法則是先建構一個可以覆蓋所有交會點之完整的地面載具路徑，選擇無人機與載具的交會點，再藉由交會點去規劃無人機的移動路徑。倘若無人機的電力充足，則將無用的交會點剃除，以合併無人機的移動節線。

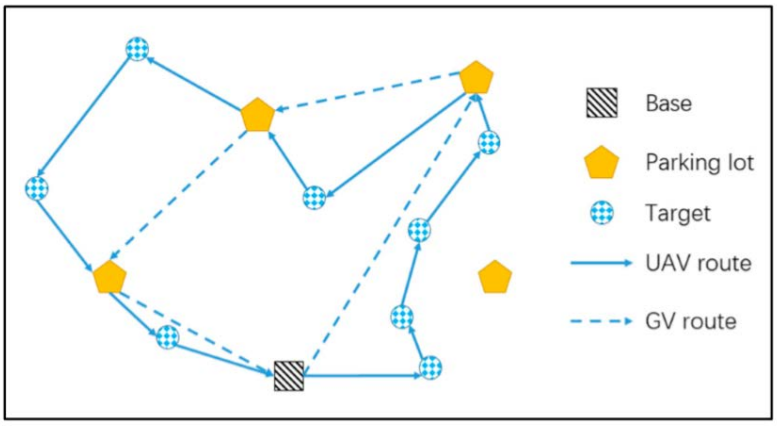


圖 2.1 無人機-地面載具二階層式路徑規劃(Luo et al., 2017)

Yu, Budhiraja, and Tokekar (2018) 提及無人機雖已開始廣泛使用在旅行銷售

doi:10.6844/NCKU202002825

員問題(TSP)，但多數未考慮到續航力限制，或是設定節點上有靜態充換電站，而該研究除了靜態充換電站以外，還加入了地面載具當作動態充換電站。如圖 2.2 所示，其中藍線表示無人機移動之路徑，紅色虛線則為無人機與地面載具共乘並充換電之路徑，紅色實線表示地面載具的移動路徑，形成 2E-GU-RP。該論文探討無人機如何以最短時間去尋訪所有節點，並針對充換電的型態不同，發展 Generalized TSP(GTSP)-based 演算法去解決多個靜態充換電站或單一動態充換電站問題。而針對多個動態充換電站，該研究發展整數規劃模式求解，除了無人機的移動路徑，同時探討地面載具如何移動最短的路徑去提供無人機充換電和共乘，使其保持電力。

Ha, Deville, Pham, and Ha (2018) 提到無人機的速度較地面載具快，同時成本也較低，故適用於運輸問題。如圖 2.3 (a)為傳統運輸問題的最佳解，圖 2.3(b)則為將無人機加入載具後的運輸路徑，其中實線為載具移動路徑，虛線則為無人機移動的路徑，明顯可看出(b)相較(a)大幅減少運輸成本。該論文將其問題定義為包含無人機之旅行銷售員問題(Traveling Salesman Problem with Drone，TSP-D)，將運輸成本和等待的時間成本設為目標式，考量無人機的電量限制下，如何使用無人機和載具服務所有顧客，並開發一個數學模式和兩種演算法：第一種演算法為 TSP by Local Search (TSP-LS)，為一常見的演算法，透過不斷計算節省值，交換尋訪的節點，尋找鄰近解看是否能降低目標值，與一般區域搜索演算法不同之處為此研究為 2E-GU-RP，交換後的兩節點需重新判別為無人機或載具之節點，用以規劃兩者的移動路徑；第二種演算法為貪婪隨機自適應搜索法(Greedy Randomized Adaptive Search Procedure，GRASP)，此演算法主要分成「建構」及「區域搜索」等兩個階段。其中，「建構階段」將每個待選元素排序成一個待處理的列表，並從頭逐一加入，以建構成一個初始的可行解。在選擇每個加入元素時，利用一個衡量函數為準則來決定選取方式。而「區域搜索階段」則將第一階段所產生的初始解進行擾動，去尋找其解空間中的其它鄰近解。如此反覆這兩個

doi:10.6844/NCKU202002825

階段，直到達成停止條件為止。

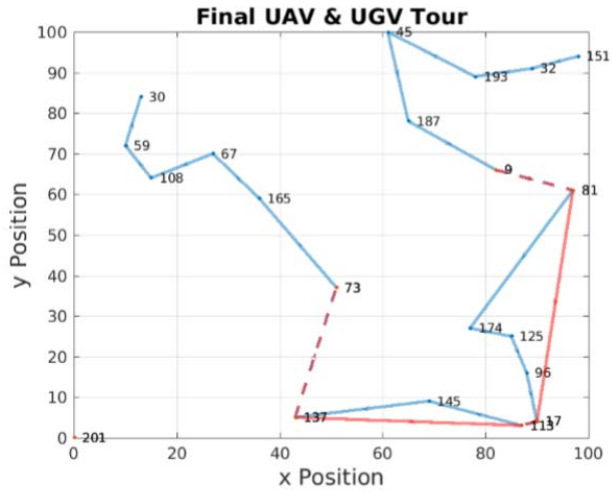


圖 2.2 考慮動態充換電站之無人機路徑規劃(Yu et al., 2018)

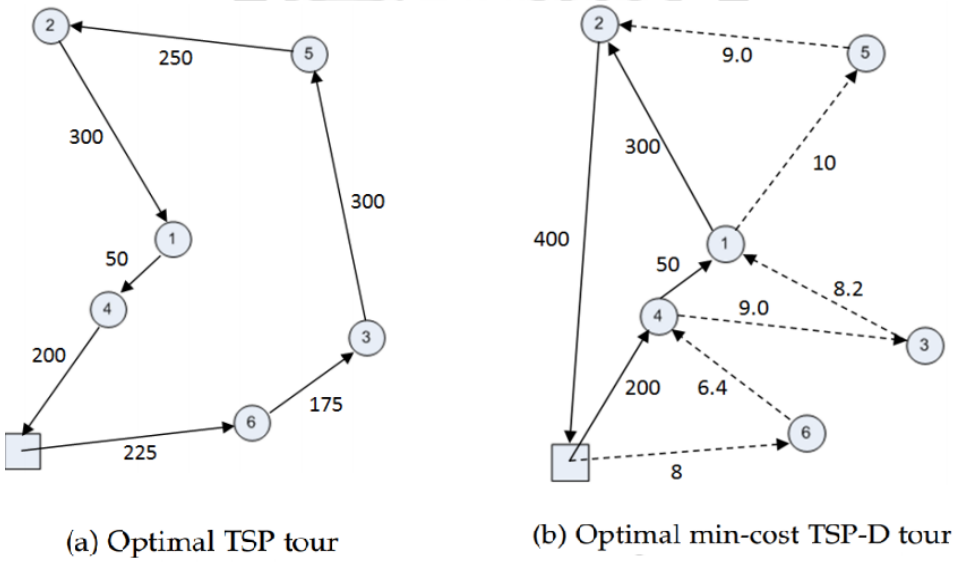


圖 2.3 TSP-D (Ha et al., 2018)

馬哈曼(2019)使用無人機和地面載具去求解覆蓋路徑規劃問題（Coverage Path Problem，CPP），由於兩者有不同的移動網路圖，故亦屬於 2E-GU-RP。在考慮電力限制的情況下，使用地面載具去提供電力及共乘，期望能使用無人機以最短的時間去搜索所有的網格。該研究開發單一地面載具和多台無人機之整數模

doi:10.6844/NCKU202002825

式，並設計貪婪演算法，使用深度優先搜索機制在時空網路中規劃不錯的找到可行無人機路線，以覆蓋所有目標區域。

本研究主要參考馬哈曼(2019)之設定，其相同處包括無人機和地面載具有各自的網路，並同樣使用地面載具對無人機充換電，且同步規劃兩者之路徑；而不同之處主要在於本研究的無人機和地面載具有不同的行進模式（即「搜索模式」與「移動模式」），若使用不同的行進模式，會有不同的移動速度與電量消耗，並將任務由探索所有節點的 CPP，改為搜索目標所有可能存在之節線的 RPP。

2.1.2 無人機-載具非同步規劃

無人機與載具的非同步規劃問題，大多為先固定無人機(載具)的移動路徑，再去規劃載具(無人機)的移動路徑，和本論文的設定亦不同，故只列下述兩篇作為參考。

Savuran and Karakaya (2015) 以基因演算法求解 TSP-D，假定地面載具的移動路徑固定，規劃無人機的路徑以最小距離去尋訪所有標靶，該篇論文並無考慮無人機之電力限制，但其有考慮到無人機和地面載具不同的移動速度，根據兩者速度差和不同時間點的相對距離，去計算無人機的降落點。

Ferrandez et al. (2016)研究無人機與載具的協同運輸，是否能減少單純使用無人機或載具的時間及能源消耗，使用混合牛頓法(hybrid Newton's method)，求解最短運輸時間以及最佳停靠站個數，其最後的結論為無人機與載具的協同運輸，相較單純使用無人機或載具，會有較佳的表現。

doi:10.6844/NCKU202002825

表 2.1 無人機與載具協同作業相關文獻比較

作者	方法	無人機限制	同步	多載具	多無人機	應用
Ferrandez et al. (2016)	H,Exp	只能載一個包裹			V	運輸
Garone et al. (2010)	M,H,Exp	飛行時間限制	V			搜救
Mathew et al. (2015)	Exp	只能載一個包裹、飛行時間限制				運輸
Tavana et al. (2017)	M,Exp	負載重量限制		V		運輸
Tokekar et al. (2016)	H,Exp	飛行時間限制、起飛與降落準備時間				建築
Sujit et al. (2009)	H,Exp	飛行時間限制	V	V		環境
Luo et al. (2017)	M,H,Exp	飛行時間限制	V			通用
Yu et al. (2018)	Exp	電力限制	V			運輸
Ha et al. (2018)	M,H,Exp	只能載一個包裹	V			運輸
馬哈曼 (2019)	M,H,Exp	電力限制	V	V	V	通用
Savuran and Karakaya (2015)	H,Exp	無				通用
本研究	M,H,Exp	電力限制	V	V	V	搜救

註：M 代表模式、H 代表啟發式演算法、Exp 代表實驗分析

2.2 搜救路線規劃相關研究

此節將回顧傳統人力搜救路線規劃以及加入無人機的搜救路線規劃相關文獻。Yao, Xie, and Ren (2017) 提到無人機十分適合用在搜索問題(search problem)，因為其相對傳統的地面或水上載具，有更好的機動性，但也由於路徑的選擇變多，使得問題變得更困難。而搜索問題早期大多源自 Stone (1976)的搜索理論，主要探討資源分配問題，研究如何分配投入的搜索資源量，而最大化搜索到標靶的機

doi:10.6844/NCKU202002825

率。此問題可進一步分為「單側搜索問題」(one-sided search problem)和「雙側搜索問題」(two-sided search problem)，前者假設標靶所在的位置或移動方式並不會受搜索者影響，譬如搜救行動或是尋寶任務等；後者則假設標靶的位置和移動與搜索者的行動相互影響，譬如警察追捕犯人和捉迷藏等應用。Dobbie (1968)針對單側與雙側搜索問題，整理了許多相關的文獻。本研究因將標靶假設為可能存在某些節線上，且其存在機率因搜救隊而變動，因此為單側搜索問題。而 Berger, Lo, and Noel (2014)將問題特徵進一步細分為幾個類型:

1. 離散與連續時間:依照是否將時間離散化與否，區分為離散與連續時間。
2. 靜態與動態標靶:依照標靶是否隨時間變動而移動，可區分為靜態或動態標靶。
3. 投入離散與連續之搜索資源(search effort)：指搜索時間投入之離散或連續資源，前者例如:投入搜救之單位數、離散之時間、備用電池數量等，後者例如:耗電量、連續之時間等。

而近年來搜救方面的探討，則從資源分配問題轉變為路徑規劃問題，規劃多組救難人員的移動路線，期望能盡快搜索到標靶，則常見到以下兩種目標式類別：最大化搜救機率，或最小化期望搜索時間。以下我們則依照動態與靜態標靶分類，介紹相關的文獻。

2.2.1 搜索動態標靶

動態標靶代表標靶會隨時間的變化而改變其位置，Piacentini, Bernardini, and Beck (2019)使用多台無人機以特定的移動模式搜索動態標靶，使用蒙地卡羅模擬(Monte Carlo Simulation，MCS)去預測標靶的移動，產生標靶可能移動之路徑以及其機率，並使用約束規劃數學模式(Constraint Programming Approach)、貪婪演算法(Greedy Algorithm)找出路徑以最大化發現目標之機率。

doi:10.6844/NCKU202002825

2.2.2 搜索靜態標靶

靜態標靶代表標靶不隨時間的變化而改變其位置，假定有一網路$G=(N,A)$，其中N為節點集合，A所有無向節線之集合，$A^s \subseteq A$為所有節線中可能存在標靶之節線，標靶出現在節線$(i,j) \in A^s$之機率密度函數(p.d.f)為α_{ij}，目標是找出一條路徑，而使整體尋獲標靶之機率最大或是期望搜索時間最短。若是尋獲標靶機率之最大，則目標式可以很直觀地以$\sum_{(i,j)\in A^s} \alpha_{ij}$去計算；但若是期望搜索時間最短，則會根據每個研究的不同，而有不同的計算方式，本研究則引用呂昀軒(2019)對期望搜索時間的之計算方式再進行修改。

Jotshi and Batta (2008)將問題設定為單一靜態標靶、均勻分布在所有節線上之中國郵差問題(Chinese Postman Problem，CPP)，要求搜救隊需尋訪所有的標靶存在機率為正之節線。該研究發展兩種啟發式演算法求解，而與傳統 CPP 問題不同的是其目標式由「最小化移動距離」變更為「最大化期望搜索時間」，假設標靶均勻分布於所有單位長度的節線上，則尋獲標靶的機率將與節線長度成正比。若搜救隊並非第一次經過行經該線段，則機率為 0；反之，尋獲標靶的機率則為該線段長度除以所有有標靶所在的線段總長度，並設計期望搜索時間之計算公式。假設任一線段之中間點為搜救隊期望尋獲標靶之位置，由起點出發至尋獲標靶之距離再乘上尋獲標靶機率之加總，即為期望搜索時間。本論文對於期望搜索時間之設定主要參考呂昀軒 (2019)的公式，而其公式亦由 Jotshi and Batta (2008)所提出的公式進行延伸，詳細會在第三章再進行比較。

Li and Huang (2018)探討多搜救隊於單位節線長度網路(unit graph，亦即所有節線的長度皆為單位長度)搜索靜態標靶，而標靶隨機分配在各節線上。該研究證明當(a)節線同時被兩組搜救隊搜救，與(b)節線僅被一組搜救隊搜救，兩者之期望搜索時間之差額為 0.25，由於該差距極小，因此可忽略不計。他們並以層空(Level-Space)網路來建構一數學模式求解此問題，目標式為最小化期望搜索時間。

doi:10.6844/NCKU202002825

Li, Patankar, Moridian, and Mahmoudian (2018)使用基因演算法(Genetic Algorithm)在一連續平面上求解多台無人機搜救問題，該論文假設每台無人機服務的節點個數相同，染色體則由節點組合，可由此染色體解碼為每台無人機移動之順序，而每台無人機有電力消耗且充換電站個數及位置皆為已知，靜態標靶位於節點上，且各標靶的重要性不同，故需先尋訪重要節點，目標則為最小化完成時間、完成重要節點尋訪時間、違反續航力限制與前往充換電距離之加權總合。

呂昀軒(2019)將問題歸類為 KRPP，使用多台無人機進行搜救，靜態標靶則隨機分布在特定節線上，和傳統鄉村郵差問題相同的是皆需行經所有可能存在標靶之節線，不同處在於其目標式由最小化行走距離變為最小化期望搜索時間。該研究亦使用層空網路當作架構數學模式和開發多種啟發式演算法，該論文同時假設無人機有移動模式和搜索模式，兩種模式的移動速度和電力消耗不同，而無人機需在特定節點上的充換電站進行充換電。

表 2.2 搜救路線規劃相關文獻比較

作者	**搜救隊**	**無人機**	**期望搜索時間最小**	**標靶分布**			**充電方式**
				靜態		動態	
				均勻	隨機		
Berger et al. (2014)	V				V		無電力因素
Piacentini et al. (2019)		V				V	不考慮電量消耗
Jotshi and Batta (2008)	V		V	V			無電力因素
Li and Huang (2018)	V		V		V		無電力因素
Li et al. (2018)		V		V			定點充電站
呂昀軒 (2019)		V	V		V		定點充電站
本研究	V	V	V		V		動態充換電站

本論文主要參考呂昀軒(2019)之設定，靜態標靶皆隨機分布在特定節線上，並需行經所有可能存在靜態標靶之節線。其無人機亦有搜索與移動兩種行進模式。該研究與本論文相異處在於本論文使用動態充換電站取代靜態充換電站，因此搜

doi:10.6844/NCKU202002825

索靜態標靶之任務可由無人機群和動態充換電站共同完成。而本論文因為無人機和動態充換電站需同時交會才能進行充換電，故使用時空(time-space)網路而非該研究的層空網路，有關本論文與 Jotshi and Batta (2008)、呂昀軒 (2019)等文獻相異處的更多細節，將會於第三章闡述與本論文進行比較。

2.3 節線途程問題相關文獻

節線途程問題(Arc Routing Problem，ARP)探討在網路圖 $G=(N,A)$ 中，存在某些路徑 $A^s \subseteq A$ 必須經過，而每條節線 A 皆存在其經過所需之距離成本，當所有線段皆需服務時，此為著名的 CPP，目標為從起點出發找出一條包含所有線段之封閉路徑並回到起點，而使總成本最低。而若僅有特定部分節線需被服務，則為 RPP。本研究欲指派 K 組無人機和 L 組動態充換電站進行搜救，給定網路圖 $G=(N,A)$，無人機因可在任兩點所構成之節線 A 移動，故無人機移動之網路為完全圖(Complete Graph)；而動態充換電站則因山區或災害地區的特性，其移動之網路可視為各點可連通之平面網路圖(Planar Graph)，兩個網路圖皆包含需要經過之節線 $A^s \subseteq A$，此設定與求解多組郵差必須經過某些節線的 KRPP 類似；而差異處則為本研究有無人機和動態充換電站(即搜救隊)等兩種移動載具，且兩者皆有不同的移動模式，此外無人機還需考慮電力消耗的因素，需和動態充換電站會合進行充換電，而目標式也從常見的最小化移動距離變為最小化期望搜索時間。

RPP 最早由 Orloff (1974)提出，由於 CPP 需尋訪所有節線，並不符合現實多數案例，因此發展只需服務特定節線的 RPP。Lenstra and Kan (1976)證明 RPP 若在無向圖且所有必須被尋訪的節線皆相連時，可能可以在多項式時間內求解，否則至少為 NP-HARD 問題，若延伸為 KRPP 則同樣也是 NP-HARD 問題。

Hà, Bostel, Langevin, and Rousseau (2014) 針對「足夠接近之節線途程問題」(Closed-Enough Arc Routing Problem，CEARP)發展三種混整數數學模式(Mixed Integer Programming，MIP)並使用分支切面法(Branch-and-Cut Algorithm，B&C)求

doi:10.6844/NCKU202002825

解。其中，CEARP 與有向圖之鄉村郵差問題(Directed Rural Postman Problem，DRPP)類似，差別在於後者需尋訪某些特定節線，而前者則為須尋訪在特定節線附近之顧客集即可，現實中可用「抄錶問題」為例。亦即，若帶有感應器的車輛行經一定距離時可接收到附近住戶的電錶、水錶的資訊，則車輛不必穿越每條道路或是到達每個節點，只要移動到可以感應器接收到資訊的位置即可。而如何根據感應器範圍規劃最短移動距離去獲得所有客戶的資料，即為 CEARP 之應用。該論文發展的三種數學模式皆有其優缺點；第一種數學模式有最少的變數及限制式，但卻需要設定極大值參數(BIG-M)，參數的設定若無法有效設定，可能會導致模式效率不佳；而其第三種數學模式則反之，有較多的參數及限制式，但不需設定 BIG-M，因此也有不錯的效率；而其第二種模式的表現則介於前兩者之間。此外，該研究也發展演算法去計算 BIG-M 的設定值，希望其 M 值越小越好，以改善效能。同時，該研究針對第三種數學模式所加入的新限制式，證明其為有效切割的限制式(Valid Equalities and Inequalities)。該模式針對每條節線引進決策變數 y_a 和 x_a，其中 y_a 為 0-1 變數，代表該節線 a 是否有被服務；而 x_a 為整數變數，代表該節線 a 在未被服務時被經過之次數。其假設若 $x_a>0$ 則 $y_a=1$，並證明以此想法去增加限制式，可有效地將可行解集合簡化，以更快求得高品質的解，此 CEARP 的想法也被應用在本研究的演算法當中。

Campbell, Corberán, Plana, and Sanchis (2018)使用無人機求解節線途程問題(Drone Arc Route Problem，DRAP)，該論文提到使用無人機可大幅降低成本及時間，因為無人機可無視道路限制，而在任兩點間直接移動。如下圖 2.4 所示，A 點到 B 點、C 點皆需 1 單位時間，而 BC 兩點之距離則為一極小值 ε，但 BC 之間並無道路連通，故若為 CPP，以傳統車輛行經需 4 單位時間，但若使用無人機，只需 $2+\varepsilon$ 即可完成任務，大大減低所需要的時間和成本，該論文提出三個假設: (1)當無人機服務完某線段後可直接前往任一節線進行服務、(2)無人機可在節

doi:10.6844/NCKU202002825

線上之任一處開始或結束服務、(3)節線並不一定為直線可為任一形狀，並使用分支切割演算法(Branch-and-Cut Algorithm)進行求解。

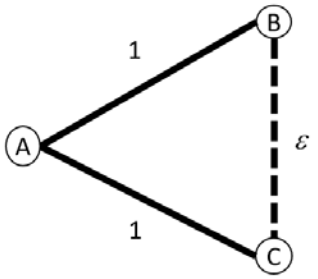


圖 2.4 待搜索網路圖 Campbell *et al.*(2018)

Groves and Van Vuuren (2005) 針對 KRPP 提出啟發式演算法，其針對標靶節線的交換提出區域搜尋法 (Local Search，LS)的方法，將 LS 中常見的兩元素優化(2-opt)、三元素優化(3-opt)等方法應用在求解 KRPP 問題中，使用多次交換疊代，重新分配不同人之搜索任務，再搭配最短路徑去規劃搜索路徑，並記錄較好的解，此方法也被參考使用於本研究的演算法中。

2.4 小結

本節針對無人機與載具的協同作業、搜救路線規劃以及節線途程問題的文獻回顧，可發現相關文獻大多數設定為節點排程問題，如協同作業中大多數文獻的目標皆為求解 TSP，搜救路線規劃的標靶亦多設立於節點上，關於無人機之節線途程的應用明顯較少。本研究則不同於節點排程問題，將問題定義為 KRPP，隨機設定靜態標靶可能存在於某節線上的機率，再使用無人機及動態充換電站尋訪所有靜態標靶可能存在之節線。與傳統節線途程問題不同的是，我們以「最小化期望搜索時間」取代「最小化移動距離」作為目標式。另外相關文獻中也較少針對無人機不同行進模式的移動速率以及充換電方式提出討論，若有考慮電力因素的文獻，大多亦使用靜態充換電站來充換電，本研究考慮山區地形不易設立靜態充換電站，因而使用動態充換電站，其可行走於傳統平面網路，同時搭配可行走

doi:10.6844/NCKU202002825

於完全圖之無人機進行搜救，形成 2E-GU-RP；除了可解決充換電問題外亦可加速搜救任務，並更符合現實情況。在數學模式的相關文獻探討中，過去大多以層空網路來建構模式，而本研究則使用時空網路，使無人機及動態充換電站可以互相配合，在下一章我們將針對本研究所探討之問題提出整數規劃模式。

doi:10.6844/NCKU202002825

第三章 結合動態充換電站之無人機群搜索路徑之數學規劃模式

本章探討結合動態充換電站之無人機群二階層路徑規劃問題，先描述研究問題，說明無人機與動態充換電站「可共乘」與「非共乘」之差別，再詳細敘述本研究之假設與網路架構，接著根據參考文獻說明本問題中期望搜索時間的計算方式，將其應用在數學模式中。為使建立數學模式更為容易，我們將原問題之網路結構更改為以時空網路為架構之整數規劃模式。3.6 節將說明主要的無人機與動態充換電站「可共乘」之時空網路模式，該模式較一般之模式更能被應用在大部分的網路圖中，但其求解時間卻相對較長，該節將說明相關之參數、變數、目標式與限制式。而 3.7 節則以 3.6 節所提出的數學模式為基礎，減少變數及限制式，修改為「非共乘」之時空網路模式。由於「非共乘」模式具有求解時間較短的優點，因此我們建議先使用它來處理大部分相關問題，而若遇到特定需使用「可共乘」時空網路的問題時，再將之變更為「可共乘」模式來處理。

3.1 問題描述

如圖 3.1 所示，假設有一無向完全網路 $G=(N,A)$，N 為所有節點的集合，A 為任兩點間所有無向節線之集合。$A_s \subseteq A$ 為可能存在靜態標靶之節線，如圖 3.1 上的黑色節線，已知 α_{ij} 為節線 $(i,j) \in A_s$ 上尋獲標靶之機率。假設有 K 架相同的無人機、L 台相同的動態充換電站共同負責搜索單一靜態標靶，如圖 3.2 所示。無人機可行經任一節線 $(i,j) \in A$，而動態充換電站只可行走於平面網路圖之節線 $A_G \subseteq A$，如圖 3.3 所示。平面網路圖上任意兩節線 A_G 互不相交，任意兩節點不相疊，但保持任意兩點之連通性，如此即可確保動態充換電站不論從哪個點出發，皆可到任意節點。而由圖 3.1~圖 3.3 可知 $A_s \subseteq A_G \subseteq A$。假設動態充換電站之電

doi:10.6844/NCKU202002825

力永遠充足，而無人機之續航力上限為 $P_{\max}$，需在電力耗盡前與動態充換電站交會處充換電。我們假設充換電可瞬間完成，只要兩者交會即可立即完成充換電。而無人機和動態充換電站皆有兩種行進模式：以 s 代表「搜索模式」以及 m 代表「移動模式」。對於無人機而言，搜索模式的速度較移動模式慢，電力消耗也較多；動態充換電站和無人機相同之處為搜索模式的速度較移動模式慢，不同之處為動態充換電站之一動假設沒有電力消耗。由於無人機不論何種行進模式，其移動速度皆較動態充換電站還快，故四種行進模式之速度由快而慢排序如下：無人機的移動模式>無人機的搜索模式>動態充換電站的搜索模式>動態充換電站的移動模式。

依上述設定，無人機與動態充換電站在行經節線 $(i, j) \in A$ 時，將共有四種不同的行進時間 $d_{ij}^{U_s}$、$d_{ij}^{U_m}$、$d_{ij}^{G_s}$、$d_{ij}^{G_m}$，分別為「無人機搜索模式」、「無人機移動模式」、「動態充換電站搜索模式」、「動態充換電站移動模式」，而其花費的時間由小而大之排序則為 $d_{ij}^{U_m} < d_{ij}^{U_s} < d_{ij}^{G_m} < d_{ij}^{G_s}$。本研究之目標為派遣 K 架無人機及 L 組動態充換電站，去尋訪所有可能存在標靶之節線，以使尋獲標靶之期望搜索時間最小化。無人機有許多可能之行進路徑，且同時須考慮當下電力狀況、尚未尋訪之節線、動態充換電站所在處、尋獲標靶之機率與行走節線所需時間等，加上動態充換電站亦有許多可能性，而這些不同之設定皆使本問題變成十分複雜難解。

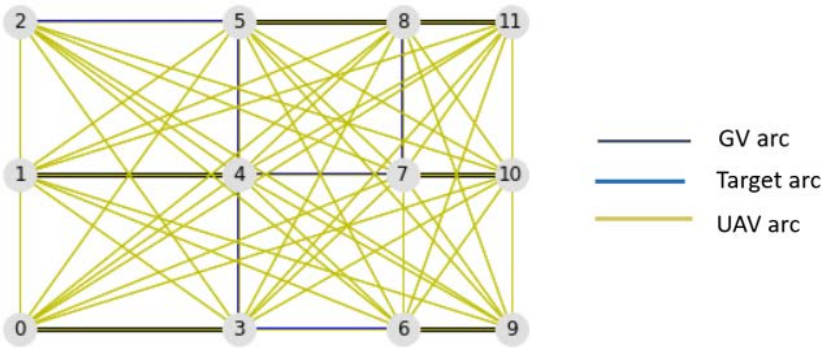


圖 3.1 本研究使用網路之示意圖

doi:10.6844/NCKU202002825

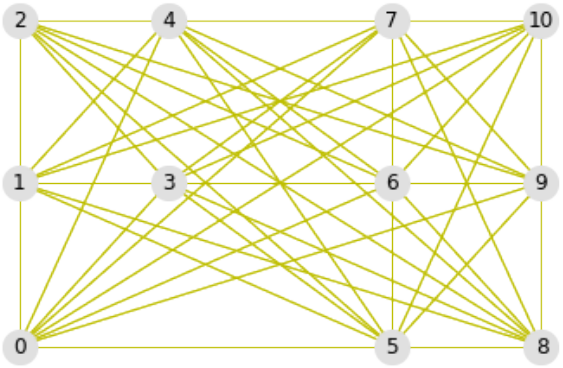


圖 3.2 無人機可行走之網路圖

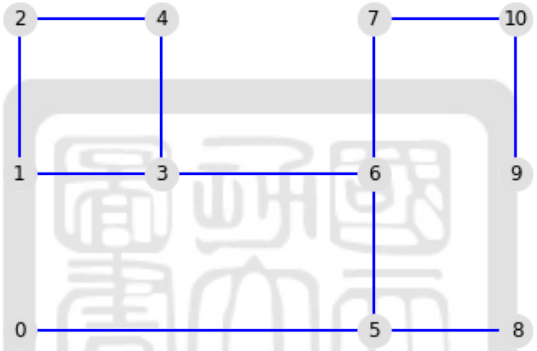


圖 3.3 動態充換電站可行走之網路圖

在過去許多無人機與載具之二階層路徑規劃的相關文獻中，無人機大多可與動態充換電站共乘，其優點為若無人機與載具需同時經過某一條節線時，使用共乘可以減少電量的消耗，且無人機可邊充換電邊進行移動，故發展出 3.6 節中的可共乘之時空數學模式。然而，由於本研究設定充換電可在極短時間內完成(因此可假設不需耗時)，只要無人機與動態充換電站交會於同一節點即可充換電，且針對單一節線的移動速率之排序為：無人機的移動模式>無人機的搜索模式>動態充換電站的移動模式>動態充換電站的搜索模式；由於我們假設無人機的最大電量可行經任意平面道路，故大多情形無人機並無與動態充換電站共乘之需求，亦即我們僅需使用 3.7 節中的非共乘之時空數學模式即可求解。然而，現實中還是有一些特殊情況仍需要無人機與動態充換電站共乘。譬如圖 3.4 中，無人機從

doi:10.6844/NCKU202002825

節點 1 出發，動態充換電站於節點 2 出發，無人機的最大電量僅能移動五單位時間，實線為平面道路 A_G。而如圖所示，行經紅色實線與橘色實線將有不同的移動時間，假設所有實體節線皆可能存在靜態標靶，故節線(1,3)、(2,3)、(3,4)、(4,5)與(4,6)皆須被尋訪，然而無人機的最大電量無法供無人機於橘色實線(3,4)上移動。若兩者無法共乘，則無人機僅需移動節線(1,3)後即被迫完成任務，而動態充換電站需隨著節線(2,3)、(3,4)、(4,5)、(5,4)、(4,6)移動，此將花費 18 單位時間才可完成搜索任務；但若無人機可與動態充換電站共乘，則兩者可一起通過節線(3,4)，再由無人機尋訪節線(4,5)；而動態充換電站尋訪節線(4,6)，只需 12 單位時間即可完成任務；上述例子，將較非共乘模式為快。

圖 3.4 是為了方便說明而設立之極端案例，譬如若無人機的最大電量仍無法通過過長的某些節線，或是某些區域不適合(或不准許)無人機飛越過去的話，此時若無人機被動態充換電站攜帶著一起移動(亦即「共乘」)，則除了可減少無人機電量之消耗外，同時也可能大量縮短完成任務的時間。因此，可共乘之時空網路模式或有其優勢。然而可共乘之時空網路模式需再加入許多變數以及限制式，導致求解時間也會較長。所以，實務上或可先使用非共乘之模式為主來計算，直至有必要使用特定需共乘之網路圖時，再改用可共乘之模式求解即可，詳細的內容亦會在 3.6 節與 3.7 節中敘述。

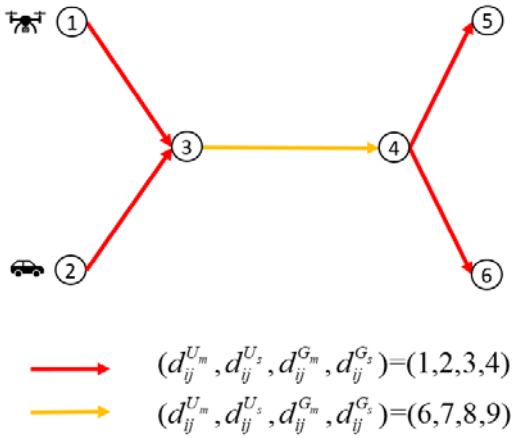


圖 3.4 無人機與動態充換電站需共乘之特殊情況

doi:10.6844/NCKU202002825

3.2 問題假設

為了適當簡化數學模式與其使用限制，本研究有以下基本假設：

1. 無人機和動態充換電站不存在誤警(false-alarm)之情形(譬如，感應器偵測到標靶時，標靶卻不存在於節線)。
2. 救難人員具備先備知識(prior knowledge)，已知各節線上尋獲靜態標靶之機率(稱之為「標靶存在機率」)，且所有節線的標靶存在機率總和為 1。
3. 靜態標靶隨機分布在節線而非節點上，且標靶出現在任一節線之位置服從均勻分佈。
4. 假設每架無人機之硬體皆相同，因此其最大電力（續航力）相同，且使用同一種模式在同一條節線上移動所需之移動時間和電力耗損等皆相同。
5. 假設每組動態充換電站在同一條節線上移動所需之時間相同。
6. 假設無人機充換電之時間可忽略（亦即，瞬間完成充換電）。
7. 假設每架無人機和動態充換電站在節線上搜索時，一定會將目前的工作完成後才會進行下一個工作；不會工作到一半中止，放棄當下節線而去尋訪其它節線 (此即排程問題之 non-preemptive 假設)。
8. 無人機所行走之網路為完全無向圖，動態充換電站行走之網路為無向平面圖，兩者皆包含連通性，即任意節點 A 皆可透過節線到達任意節點 B。
9. 當無人機停留在同一節點時，不消耗電力(可想成先定點降落熄火休息)。
10. 假設搜救隊可由多個登山口進出，即動態充換電站可有數個起點。

3.3 網路結構

呂昀軒(2019)除了決定所有無人機的尋訪路徑以外，同時需決定兩者在各節線時的行進模式。由於原網路圖結構僅能了解兩者的尋訪順序，無法判斷通過節

doi:10.6844/NCKU202002825

線時的行進模式。如圖 3.5，若使用無人機移動，給定一路徑 ABCBD，在經過節線 CB 時，因已搜索過節線 BC，故不需再搜索一次，此時會使用速度較快的移動模式。但因原網路結構無法判斷使用何種模式，所以可使用圖 3.6 的網路結構，將節線分為「移動節線」與「搜索節線」兩種。其中，移動節線可被經過多次，而搜索節線則只能被經過一次，便可判斷經過節線時之行進模式。以圖 3.6 為例，可知經過節線 BC 時使用的是移動模式，經過節線 CB 則是使用搜索模式。

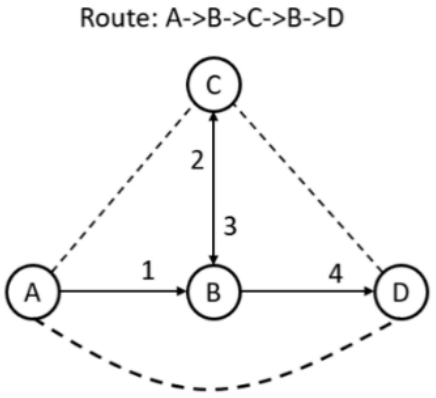


圖 3.5 原問題之網路(呂昀軒,2019)

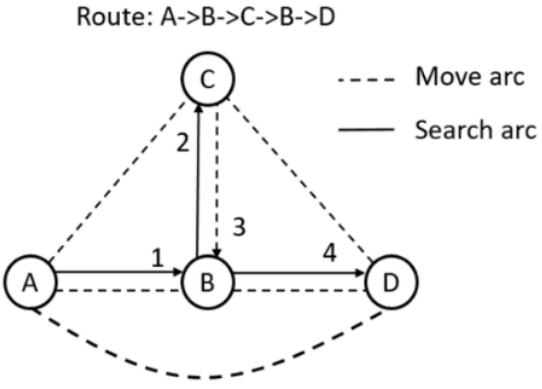


圖 3.6 以搜索節線與移動節線建構之網路(呂昀軒,2019)

本研究除了無人機以外，同時使用動態充換電站，且兩者速率不同。若為「非共乘」模式，如圖 3.7 所示，兩節點之間共有四條節線，分別為無人機與動態充

doi:10.6844/NCKU202002825

換電站，使用移動或搜索模式行經節線。而若為「可共乘」模式，則兩節點中間還要再加上一條節線，代表無人機以動態充換電站的速率來經過該節線。

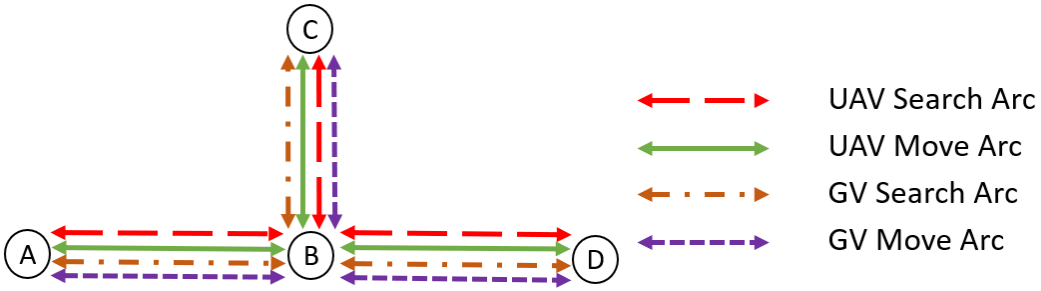


圖 3.7 本研究非共乘之網路結構

3.4 參數與變數定義

集合

V	節點之集合(包含虛擬起訖點O、D)
O	虛擬起點
D	虛擬訖點
A	所有節線之集合
A_{start}	起始節線集合，$(i,j)=a\in A_{start}\subseteq A\forall \quad i\in O$
A_{end}	終止節線集合，$(i,j)=a\in A_{end}\subseteq A \quad \forall j\in D$
A^s	搜索節線集合，$A^s\subseteq A$
A_h^s	搜索節線節線(i,j)中滿足$i<j$之集合，
A_U	無人機所有節線集合$A_U\subseteq A$
A_U^s	無人機搜索節線集合$A_U^s\subseteq A_U, A_U^s\subseteq A^s$
A_U^m	無人機移動節線集合$A_U^m\subseteq A_U$

doi:10.6844/NCKU202002825

A_G 動態充換電站所有節線集合 $A_G \subseteq A$

A_G^s 動態充換電站搜索節線集合 $A_G^s \subseteq A_G$, $A_G^s \subseteq A^s$

A_G^m 動態充換電站移動節線集合 $A_G^m \subseteq A_G$

U_s 無人機使用搜索模式進行移動

U_m 無人機使用移動模式進行移動

G_s 動態充換電站使用搜索模式進行移動

G_m 動態充換電站使用移動模式進行移動

參數

T 最大單位時間數

K 無人機的數量

L 動態充換電站的數量

P_{max} 無人機的最大電量

$d_{ij}^{U_s}$ 節線 $(i,j)=a \in A_U^s$ ，使用無人機搜索模式所需的時間

$d_{ij}^{U_m}$ 節線 $(i,j)=a \in A_U^m$ ，使用無人機移動模式所需的時間

$d_{ij}^{G_s}$ 節線 $(i,j)=a \in A_G^m$ ，使用動態充換電站搜索模式所需的時間

$d_{ij}^{G_m}$ 節線 $(i,j)=a \in A_G^m$ ，使用動態充換電站移動模式所需的時間

α_{ij} 在搜索節線 $(i,j)=a \in A^s$ 尋獲標靶之機率

M 極大的數

ε 極小的數

$mode$ 用於電量的移動模式，1 代表搜索模式、2 代表移動模式、3

doi:10.6844/NCKU202002825

表示無人機使用動態充換電站的移動模式進行移動、4 表示無人機使用動態充換電站的搜索模式進行移動。

註：$i, j \in V$ ，$i \neq j$ ，$(i, j) = a \in A$ ，$tail(a) = i$ ，$head(a) = j$ ；定義符號 $\bar{a} = (j, i)$

變數

x_{ij}^{tk} 無人機 k 在第 t 單位時刻行走搜索節線 $(i, j) = a \in A_U^s$ 為 1;反之為 0

y_{ij}^{tk} 無人機 k 在第 t 單位時刻行走移動節線 $(i, j) = a \in A_U^m$ 為 1;反之為 0

u_{ij}^{tk} 無人機 k 在第 t 單位時刻以動態充換電站之搜索模式移動速率行走節線 $(i, j) = a \in A_{G_s}$ 為 1;反之為 0

v_{ij}^{tk} 無人機 k 在第 t 單位時刻以動態充換電站之移動模式移動速率行走節線 $(i, j) = a \in A_{G_m}$ 為 1;反之為 0

w_{ij}^{tl} 動態充換電站 l 在第 t 單位時刻行走搜索節線 $(i, j) = a \in A_G^s$ 為 1;反之為 0

z_{ij}^{tl} 動態充換電站 l 在第 t 單位時刻行走移動節線 $(i, j) = a \in A_G^m$ 為 1;反之為 0

$outb_{ij}^{tkmode}$ 無人機 k 在第 t 單位時刻使用模式 $mode$ 進入 $(i, j) = a \in A_U$ 的電量

inb_{ij}^{tkmode} 無人機 k 在第 t 單位時刻使用模式 $mode$

doi:10.6844/NCKU202002825

離開 $(i, j) = a \in A_U$ 的電量

a_i^{tk} 無人機 k 和動態充換電站 l 在第 t 單位時刻

在節點 i 交會進行充換電為 1;反之為 0

ω_{ij}^{tk} 無人機 k 和動態充換電站在第 t 單位時刻

使用移動模式共乘行走節線 $(i, j) = a \in A_G^m$;反之為 0

μ_{ij}^{tk} 無人機 k 和動態充換電站在第 t 單位時刻

使用搜索模式共乘行走節線 $(i, j) = a \in A_G^s$;反之為 0

3.5 最小化標靶被尋獲之期望搜索時間

本研究主要參考呂昀軒(2019)和 Jotshi and Batta (2008)對於期望搜索時間的計算方式，並加以進行修改以符合本研究之網路結構。本節將說明本研究主要參考呂昀軒(2019)和 Jotshi and Batta (2008)對於期望搜索時間的計算方式，加以修改以符合本研究之網路結構，並會說明本研究與另外兩篇論文對於期望搜索時間的計算方式。

Jotshi and Batta (2008)

此論文假設在任意線段上尋獲標靶之機率為均勻分佈，線段之中間點為期望尋獲搜索標靶之位置。式(3.5.1)代表線段 $i(P)$ 尋獲標靶之機率，$i(P)$ 代表搜索路徑 P 中第 i 條線段，$l_{i(P)}$ 則為線段 $i(P)$ 之長度，而 $L_{i(P)}$ 為搜索路徑 P 中第 1 條到第 i-1 條線段之集合，故 $L_{1(P)}=\Phi$；當 $i(P) \in L_{i(P)}$ 時代表此線段過去曾被服務過，因為此研究為靜態標靶且誤警率為 0，故在此線段尋獲標靶之機率為 0，否則該機

doi:10.6844/NCKU202002825

率應為 $\frac{l_{i(P)}}{L}$ (該線段的長度除以搜索路徑 P 之總長度)，而期望搜索時間之計算則為式(3.5.2)，從起點出發至尋獲標靶之距離乘上尋獲標靶機率之加總。

$$\Pr(i(P)) = \begin{cases} 0 & \text{if edge i(P)} \in L_i\text{(P)} \\ \frac{l_{i(\mathrm{P})}}{L} & \text{otherwise} \end{cases} \tag{3.5.1}$$

$$E\left[T_s \mid P\right] = \sum_{i=1}^{|\mathrm{E'(P)}|}\left(\sum_{j=1}^{i-1} l_{j(\mathrm{P})} + \frac{l_{i(\mathrm{P})}}{2}\right) * \Pr(\mathrm{i(P)}) \tag{3.5.2}$$

呂昀軒(2019)

呂昀軒(2019)假設靜態標靶隨機分布在特定節線上，且發現標靶之機率可為相異，故修改式(3.5.1)為式(3.5.3)，$\alpha_{i(P)}$ 為節線 $i(P)$ 上尋獲標靶之機率，其介於 0 到 1 之間且所有存在可能存在靜態標靶之機率總和為 1，由於此研究中無人機有不同的移動模式，故修改式(3.5.2)為式(3.5.4)，將 $l_{i(P)}$ 替換為 $t_{i(P)}$，其中 $t_{i(P)}$ 為線段 $i(P)$ 之移動時間，$\frac{t_{i(P)}}{2}$ 則為期望尋獲標靶的時間，但由於該論文使用層空式網路架構，將每層無人機到達的累積距離設為變數，故式(3.5.4)中累積距離的變數又乘上該節線是否為首次尋訪的變數使之成為非線性，因此需將式(3.5.4)改寫為式(3.5.5)，記錄經過搜索節線的時刻，t_a^s 為經過搜索節線 a 之時刻，t_a 則為無人機以搜索模式經過節線 a 之移動時間，這樣便可不必判斷是否首次尋訪節線 a。

$$\Pr(i(P)) = \begin{cases} 0 & \text{if edge i(P)} \in L_i\text{(P)} \\ \alpha_{i(P)} & \text{otherwise} \end{cases} \tag{3.5.3}$$

$$E\left[T_s \mid P\right] = \sum_{i=1}^{\mathrm{E'(P)}}\left(\sum_{j=1}^{i-1} t_{j(\mathrm{P})} + \frac{t_{i(\mathrm{P})}}{2}\right) * \Pr(\mathrm{i(P)}) \tag{3.5.4}$$

$$E(T \mid P) = \sum_{a \in \mathrm{A}_h^s} (t_a^s - \frac{t_a}{2})\alpha \tag{3.5.5}$$

doi:10.6844/NCKU202002825

本研究

而本研究同樣假設靜態標靶隨機分布在特定節線上，因此對於發現標靶之機率與式(3.5.3)相同，但式(3.5.5)對於期望搜索時間之計算需改為式(3.5.6)，因為本研究除了無人機可尋訪標靶外，加入了動態充換電站，以完成搜索任務。而在尋訪單一節線時，因無人機與動態充換電站兩者速率不同，導致式(3.5.5)中的t_a需改成t_{ua}或t_{ga}，以記錄經過節線a所需的時間。但此方法需新增變數與限制式，以區別每一條搜索節線是何者(無人機或動態充換電站)所經過，藉以判斷經過該節線所需的時間，然而，本研究採用時空網路架構建構數學模式，此與呂昀軒(2019)之層空網路架構不同，不需另外設變數與限制式去記錄可能存在靜態標靶之節線被尋訪的時刻，故將式(3.5.6)改為式(3.5.7)。當無人機和動態充換電站使用搜索模式尋訪該節線時，由$x_{ij}^{tk}=1$或$w_{ij}^{tk}=1$即可區分節線(i,j)被何者經過，可根據t得知尋訪該節線的時刻，再使用$d_{ij}^{U_s}$與$d_{ij}^{G_s}$去計算到達節線(i,j)所需的時間，便可計算期望搜索時間。

$$E(T\mid P)=\sum_{a\in A_h^s}(t_{ga}^s-\frac{t_{ga}}{2})\alpha_a+\sum_{a\in A_h^s}(t_{ua}^s-\frac{t_{ua}}{2})\alpha_a \qquad (3.5.6)$$

$$E(T\mid P)=\sum_{a\in A^s}x_{ij}^{tk}(t+\frac{d_{ij}^{U_s}}{2})\alpha_{ij}+\sum_{a\in A^s}w_{ij}^{tk}(t+\frac{d_{ij}^{G_s}}{2})\alpha_{ij} \qquad (3.5.7)$$

3.6 可共乘之時空數學模式

本研究之充換電站為動態可移動，此與呂昀軒(2019)使之用靜態式充換電站不同。呂昀軒(2019)的模式只需決定無人機的節線移動順序，因此使用層空式網路架構，決定每一層無人機所到達的節點順序即可；反之，本研究因為使用動態式充換電站，無人機與動態充換電站需在同一時刻到達相同節點才可進行充換電，

doi:10.6844/NCKU202002825

故需要使用時空網路架構，以辨別是否停留在節點上等候充換電。圖 3.8 為單一無人機與單一動態充換電站可共乘之時空網路圖，O 為虛擬起點、D 為虛擬訖點，無人機或動態充換電站從 O 出發，若沒有要繼續移動則會立即回到 D。假設節線(1,2)與(2,1)存在靜態標靶之可能性，故無人機和動態充換電站可使用搜索模式經過此兩條節線，而節線(2,3)與(3,2)不存在靜態標靶，則沒有搜索節線。由於行經單一節線的移動時間為 $d_{ij}^{U_m} < d_{ij}^{U_s} < d_{ij}^{G_m} < d_{ij}^{G_s}$，故在時空網路中以節線(1,2)為例，動態充換電站使用搜索模式需要 4 單位時間，動態充換電站使用移動模式需要 3 單位時間，無人機使用搜索模式需 2 單位時間，使用移動模式只需 1 單位時間。然而無人機若需共乘，則需使無人機可以動態充換電站之速率移動，故以橘色虛線(UAV move with GV arc)或粉紅色虛線(UAV search with GV arc)表示兩者共乘，以此建立可共乘之時空網路模式。

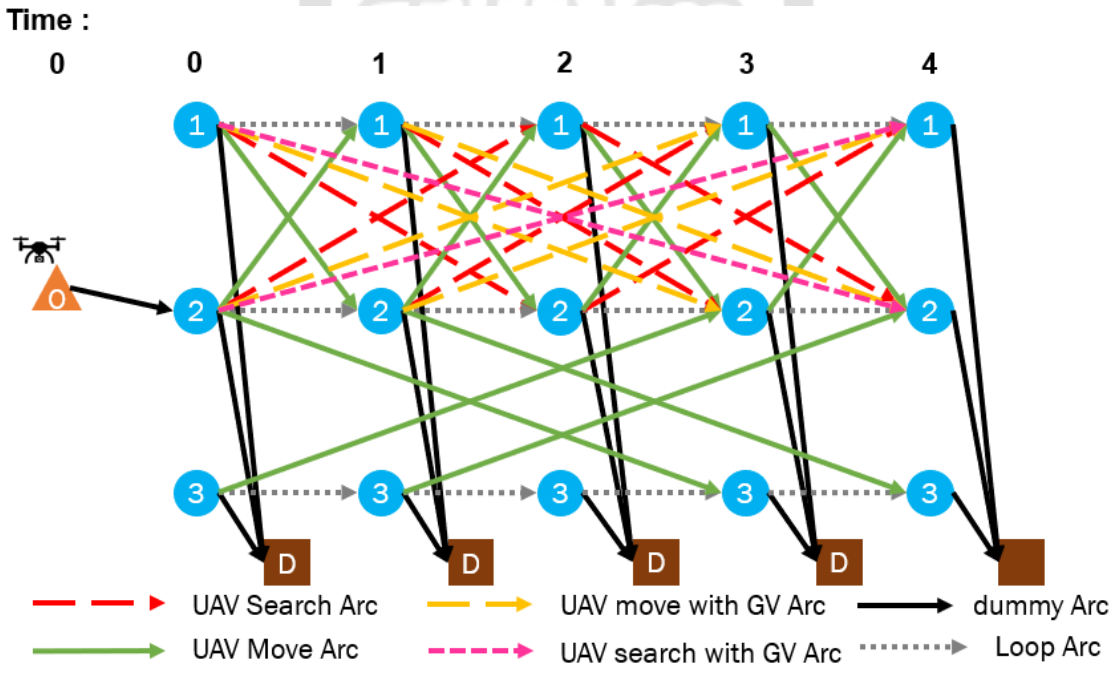


圖 3.8 可共乘之無人機時空網路圖

目標式為式(3.6.1)，代表最小化期望搜索時間，其中 ε 則為一個極小的數，將其

doi:10.6844/NCKU202002825

乘上無人機和動態充換電站的飛行距離，旨在希望兩者在完成任務後，可立即回到虛擬訖點，不會多繞遠路。

$$\begin{aligned}
&\textit{Minimize} \quad \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A^s} x_{ij}^{tk}\left(t+\frac{d_{ij}^{U_s}}{2}\right)\alpha_{ij} \\
&+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A^s} w_{ij}^{tl}\left(t+\frac{d_{ij}^{G}}{2}\right)\alpha_{ij}+\varepsilon\Big(\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_U^s} x_{ij}^{tk}d_{ij}^{U_s} \\
&+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_U^m} y_{ij}^{tk}d_{ij}^{U_m}+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^s} u_{ij}^{tk}d_{ij}^{G_s} \qquad (3.6.1)\\
&+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^m} v_{ij}^{tk}d_{ij}^{G_m}+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^s} w_{ij}^{tk}d_{ij}^{G} \\
&+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^m} z_{ij}^{tk}d_{ij}^{G}\Big)
\end{aligned}$$

限制式部分，限制式(3.6.2)到(3.6.10)為無人機和動態充換電站的移動限制，限制式(3.6.2)為限制每架無人機從虛擬起點出發。

$$\sum_{(i,j)\in A_{start}} y_{ij}^{0k}=1 \qquad \forall k=1,.....,K \qquad (3.6.2)$$

限制式(3.6.3)則為每組動態充換電站從虛擬起點出發。

$$\sum_{(i,j)\in A_{start}} z_{ij}^{0l}=1 \qquad \forall l=1,.....,L \qquad (3.6.3)$$

限制式(3.6.4)為每架無人機的流量守恆限制。

$$\begin{aligned}
&\sum_{(i,j)\in A_U^s} x_{ij}^{tk}+\sum_{(i,j')\in A_U^m} y_{ij'}^{tk}+\sum_{(i,j'')\in A_G^m} v_{ij''}^{tk}+\sum_{(i,j''')\in A_G^s} u_{ij'''}^{tk} \\
&=\sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k}+\sum_{(h',i)\in A_U^m} y_{h'i}^{(t-d_{h'i}^{U_m})k}+\sum_{(h'',i)\in A_G^m} v_{ij}^{(t-d_{h''i}^{G_m})k}+\sum_{(h''',i)\in A_G^m} u_{ij}^{(t-d_{h'''i}^{G_s})k} \qquad (3.6.4)\\
&\forall i\notin O; i\notin D; k=1,.....,K; t=0,.....,T
\end{aligned}$$

限制式(3.6.5)則為每組動態充換電站的流量守恆限制。

$$\begin{aligned}
&\sum_{(i,j)\in A_G^s} w_{ij}^{tl}+\sum_{(i,j)\in A_G^m} z_{ij}^{tl}=\sum_{(h,i)\in A_G^s} w_{hi}^{(t-d_{hi}^{G})l}+\sum_{(h',i)\in A_G^m} z_{h'i}^{(t-d_{h'i}^{G})l} \qquad (3.6.5)\\
&\forall i\notin O; i\notin D; l=1,.....,L; t=0,.....,T
\end{aligned}$$

限制式(3.6.6)為每架無人機結束任務後回到虛擬訖點。

$$\sum_{t=1}^{T}\sum_{(i,j)\in A_{end}} y_{ij}^{tk}=1 \quad \forall k=1,.....,K \qquad (3.6.6)$$

限制式(3.6.7)則為每組動態充換電站結束任務後回到虛擬訖點。

$$\sum_{t=1}^{T}\sum_{(i,j)\in A_{end}} z_{ij}^{tl}=1 \quad \forall l=1,.....,L \qquad (3.6.7)$$

doi:10.6844/NCKU202002825

限制式(3.6.8)表示所有搜索節線皆需被尋訪一次。

$$\sum_{k=1}^{K}\sum_{t=0}^{T}x_{ij}^{tk}+\sum_{k=1}^{K}\sum_{t=0}^{T}x_{ji}^{tk}+\sum_{k=1}^{K}\sum_{t=0}^{T}w_{ij}^{tl}+\sum_{k=1}^{K}\sum_{t=0}^{T}w_{ji}^{tl}$$
$$+\sum_{k=1}^{K}\sum_{t=0}^{T}u_{ij}^{tk}+\sum_{k=1}^{K}\sum_{t=0}^{T}u_{ji}^{tk}=1 \qquad \forall(i,j)=a\in A^{s};(j,i)=\overline{a}\in A^{s} \tag{3.6.8}$$

限制式(3.6.9)為每架無人機在單一時刻只能選擇一條節線尋訪。

$$\sum_{(i,j)\in A_U^s}x_{ij}^{tk}+\sum_{(i,j)\in A_U^m}y_{ij}^{tk}+\sum_{(i,j)\in A_G^m}v_{ij}^{tk}+\sum_{(i,j)\in A_G^s}u_{ij}^{tk}\le 1$$
$$\forall t=0,.....,T;k=1,.....,K \tag{3.6.9}$$

限制式(3.6.10) 為每組動態充換電站在單一時刻只能選擇一條節線尋訪。

$$\sum_{(i,j)\in A_G^s}w_{ij}^{tk}+\sum_{(i,j)\in A_G^m}z_{ij}^{tk}\le 1 \qquad \forall t=0,.....,T;l=1,.....,L \tag{3.6.10}$$

接下來從限制式(3.6.11)到限制式(3.6.21)為電力限制相關的限制式，包含初始電力的設置、電力消耗限制與充換電限制。

限制式(3.6.11)到(3.6.18)為無人機不論經過哪一條節線，進入節線時的電量與離開節線時的電量皆不大於最大電量。

$$inb_{ij}^{tk1}\le P_{max}x_{ij}^{tk} \quad \forall k=1,.....,K;t<T-d_{ij}^{U_s};(i,j)\in A_U^s \tag{3.6.11}$$

$$outb_{ij}^{tk1}\le P_{max}x_{ij}^{tk} \quad \forall k=1,.....,K;t<T-d_{ij}^{U_s};(i,j)\in A_U^s \tag{3.6.12}$$

$$inb_{ij}^{tk2}\le P_{max}y_{ij}^{tk} \quad \forall k=1,.....,K;t<T-d_{ij}^{U_m};(i,j)\in A_U^m \tag{3.6.13}$$

$$outb_{ij}^{tk2}\le P_{max}y_{ij}^{tk} \quad \forall k=1,.....,K;t<T-d_{ij}^{U_m};(i,j)\in A_U^m \tag{3.6.14}$$

$$inb_{ij}^{tk3}\le P_{\max}v_{ij}^{tk} \quad \forall k=1,...,K;t<T-d_{ij}^{G_m};(i,j)\in A_G^m \tag{3.6.15}$$

$$outb_{ij}^{tk3}\le P_{\max}v_{ij}^{tk} \quad \forall k=1,...,K;t<T-d_{ij}^{G_m};(i,j)\in A_G^m \tag{3.6.16}$$

$$inb_{ij}^{tk4}\le P_{\max}u_{ij}^{tk} \quad \forall k=1,...,K;t<T-d_{ij}^{G_s};(i,j)\in A_G^s \tag{3.6.17}$$

$$outb_{ij}^{tk4}\le P_{\max}u_{ij}^{tk} \quad \forall k=1,...,K;t<T-d_{ij}^{G_s};(i,j)\in A_G^s \tag{3.6.18}$$

限制式(3.6.19)表示無人機從虛擬起點出發時的電量為最大電量 P_{max} 。

doi:10.6844/NCKU202002825

$$out_{0j}^{0k2} = P_{\max} y_{0j}^{0k} \qquad \forall k = 1,.....,K \tag{3.6.19}$$

限制式(3.6.20)到限制式(3.6.24)為行經不同節線時的電力消耗。

限制式(3.6.20)為$i = j$時，代表無人機待在同個節點上並未移動，故沒有消耗電力，此情況大多為無人機停在某個節點上等待動態充換電站。

$$out_{ij}^{tk2} = inb_{ij}^{tk2} \qquad \forall k = 1,.....,K; t < T; i = j \tag{3.6.20}$$

限制式(3.6.21)為無人機行經搜索節線時，消耗移動所需時間的電力。

$$inb_{ij}^{tk1} = outb_{ij}^{tk1} - x_{ij}^{tk} d_{ij}^{U_s} \qquad \forall k = 1,.....,K; t + d_{ij}^{U_s} < T; (i, j) \in A_U^s \tag{3.6.21}$$

限制式(3.6.22)為無人機行經移動節線時，消耗移動所需時間的電力。

$$inb_{ij}^{tk2} = outb_{ij}^{tk2} - y_{ij}^{tk} d_{ij}^{U_m} \qquad \forall k = 1,.....,K; t + d_{ij}^{U_m} < T; (i, j) \in A_U^m \tag{3.6.22}$$

限制式(3.6.23)表示若無人機共乘時，則經過節線$(i, j) \in A_G^m$不消耗電量。

$$inb_{ij}^{tk3} = outb_{ij}^{tk3} - (v_{ij}^{tk} - \omega_{ij}^{tk}) d_{ij}^{G_m} \qquad \forall k = 1,...,K; t + d_{ij}^{G_m} < T; (i, j) \in A_G^m \tag{3.6.23}$$

限制式(3.6.24)表示若無人機共乘時，則經過節線$(i, j) \in A_G^s$不消耗電量。

$$inb_{ij}^{tk4} = outb_{ij}^{tk4} - (u_{ij}^{tk} - \mu_{ij}^{tk}) d_{ij}^{G_s} \qquad \forall k = 1,...,K; t + d_{ij}^{G_s} < T; (i, j) \in A_G^s \tag{3.6.24}$$

限制式(3.6.25)到限制式(3.6.27)為控制充換電變數a_{ik}^t的限制式，需三個限制式皆滿足才能充換電，無人機與動態充換電站不論使用搜索或移動模式需在同樣時刻t到達節點i，才能使a_{ik}^t為 1 以充換電。限制式(3.6.25)夾擠a_{ik}^t的下界，如果無人機和動態充換電站只有其一到節點i，則$a_{ik}^t \geq 0$；而若有一架無人機和m組動態充換電站皆到達節點i則$a_{ik}^t \geq m$。而 M 則為一極大的數，用以保證在多組動態充換電站到達節點i時，此限制式仍可正常運作。限制式(3.6.26)和限制式(3.6.27)則夾擠a_{ik}^t的上界，在式(3.6.26)中，若無人機使用移動或搜索節線在時刻t到達節

doi:10.6844/NCKU202002825

點i，則$a_{ik}^{t} \le 1$；反之若沒到達，則$a_{ik}^{t} \le 0$，式(3.6.27)則如同式(3.6.26)根據動態充換電站的移動決定a_{ik}^{t}的範圍，以此種上下夾擠的方式，確保a_{ik}^{t}只有在無人機和動態充換電站同時到達節點i時為 1，其餘皆為 0。

$$Ma_{ik}^{t} \ge (\sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k} + \sum_{(h,i)\in A_U^s} y_{h'i}^{(t-d_{h'i}^{U_m})k} + \sum_{l=1}^{L}\sum_{(h",i)\in A_G^m} z_{h"i}^{(t-d_{h"i}^{G})l} + \sum_{l=1}^{L}\sum_{(h'",i)\in A_G^s} w_{h'i}^{(t-d_{h'"i}^{G})l})\text{-}1 \quad \forall k=1,.....,K; i\in V \tag{3.6.25}$$

$$a_{ik}^{t} \le \sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k} + \sum_{(h',i)\in A_U^m} y_{h'i}^{(t-d_{h'i}^{U_m})k} \quad \forall k=1,.....,K; i\in V \tag{3.6.26}$$

$$a_{ik}^{t} \le \sum_{l=1}^{L}\sum_{(h",i)\in A_G^m} z_{h"i}^{(t-d_{h"i}^{G})l} + \sum_{l=1}^{L}\sum_{(h'",i)\in A_G^s} w_{h'i}^{(t-d_{h'"i}^{G})l} \quad \forall k=1,.....,K; i\in V \tag{3.6.27}$$

限制式(3.6.28)到(3.6.33)為控制共乘變數ω_{ij}^{tk}和μ_{ij}^{tk}的限制式，其原理類似上述限制式(3.6.25)到(3.6.27)中對於充換電變數a_{ik}^{t}的限制。其中，式(3.6.28)需要無人機與動態充換電站同時通過節線$(i,j)\in A_G^m$，決定共乘變數的下界$\omega_{ij}^{tk} \ge 1$。式(3.6.29)與(3.6.30)則分別根據無人機與動態充換電站的移動，決定共乘變數的下界$\omega_{ij}^{tk} \le 1$，以這三個限制式夾擠ω_{ij}^{tk}，使其唯有在無人機與動態充換電站同時在時刻t，以動態充換電站移動，其速率經過節線$(i,j)\in A_G^m$時，將使ω_{ij}^{tk}=1，其餘皆為 0。

$$\omega_{ij}^{tk} \ge (\sum_{(i,j)\in A_U^m} v_{ij}^{tk} + \sum_{(i,j)\in A_G^m} z_{ij}^{tl}) - 1 \quad \forall k=1,...,K; l=1,...,L; (i,j)\in A_G^m \tag{3.6.28}$$

$$\omega_{ij}^{tk} \le \sum_{(i,j)\in A_G^m} v_{ij}^{tk} \quad \forall k=1,...,K; l=1,...,L; (i,j)\in A_G^m \tag{3.6.29}$$

$$\omega_{ij}^{tk} \le \sum_{(i,j)\in A_G^m} z_{ij}^{tl} \quad \forall k=1,...,K; l=1,...,L; (i,j)\in A_G^m \tag{3.6.30}$$

而μ_{ij}^{tk}的設置則類似於ω_{ij}^{tk}，只是將移動節線變更為搜索節線。

$$\mu_{ij}^{tk} \ge (\sum_{(i,j)\in A_U^s} u_{ij}^{tk} + \sum_{(i,j)\in A_G^s} w_{ij}^{tl}) - 1 \quad \forall k=1,...,k; l=1,...,L; (i,j)\in A_G^s \tag{3.6.31}$$

$$\mu_{ij}^{tk} \le \sum_{(i,j)\in A_G^s} u_{ij}^{tk} \quad \forall k=1,...,K; l=1,...,L; (i,j)\in A_G^s \tag{3.6.32}$$

doi:10.6844/NCKU202002825

$$\mu_{ij}^{tk} \le \sum_{(i,j)\in A_G^s} w_{ij}^{tl} \quad \forall k=1,...,K; l=1,...,L; (i,j)\in A_G^s \tag{3.6.33}$$

限制式(3.6.34)到限制式(3.6.37)為電力的流量平衡限制式。

限制式(3.6.34)表在時刻 $t=0$ 從虛擬起點到預設起點時不消耗電量。

$$\sum_{(i,j)\in A_{source}} inb_{ij}^{(t-d_{ij}^{U_m})k2} = \sum_{(j,h)\in A_U^m} outb_{jh}^{tk2} + \sum_{(j,h')\in A_U^s} outb_{jh}^{tk1} \tag{3.6.34}$$
$$\forall k=1,.....,K; j\notin O; j\notin D; t=0$$

限制式(3.6.35)到限制式(3.6.37)為除了 $t=0$ 以外的電量平衡，一般的電量平衡應為流入電量等於流出電量，但因為需考慮到充換電時，電量會變為最大電量，導致流出電量大於流入電量，故使用上下界夾擠的方式。若無人機在時刻 t 時於節點 i 充換電則 a_{ik}^t=1；限制式(3.6.35)和限制式(3.6.36)則能使流出節點 i 的電量 $outb_{ij}^{tkmode}$ 大於流入電量 inb_{ij}^{tkmode}，再搭配限制式(3.6.37)使充換電後流出節點 i 的電量 $outb_{ij}^{tkmode}$ 需大於等於最大電量 P_{max}，如此 $outb_{ij}^{tkmode}$ 則一定為最大電量 P_{max}。反之，若兩者沒有交會進行充換電則 a_{ik}^t=0；限制式(3.6.35)和限制式(3.6.36)則保持流入等於流出的電量平衡。

$$\sum_{(i''',j)\in A_G^s} inb_{i'''j}^{(t-d_{ij}^{G_s})k4} + \sum_{(i'',j)\in A_G^m} inb_{i''j}^{(t-d_{ij}^{G_m})k3} + \sum_{(i',j)\in A_U^m} inb_{i'j}^{(t-d_{i'j}^{U_m})k2}$$
$$+ \sum_{(i,j)\in A_U^s} inb_{ij}^{(t-d_{ij}^{U_s})k1} + P_{max}a_j^{tk} \ge \sum_{(j,h''')\in A_G^s} outb_{jh'''}^{tk4} + \sum_{(j,h'')\in A_G^m} outb_{jh''}^{tk3} \tag{3.6.35}$$
$$+ \sum_{(j,h')\in A_U^m} outb_{jh'}^{tk2} + \sum_{(j,h)\in A_U^s} outb_{jh}^{tk1} \quad \forall k=1,.....,K; j\notin O; j\notin D; t\ge 0$$

$$\sum_{(i''',j)\in A_G^s} inb_{i'''j}^{(t-d_{ij}^{G_s})k4} + \sum_{(i'',j)\in A_G^m} inb_{i''j}^{(t-d_{ij}^{G_m})k3} + \sum_{(i',j)\in A_U^m} inb_{i'j}^{(t-d_{i'j}^{U_m})k2}$$
$$+ \sum_{(i,j)\in A_U^s} inb_{ij}^{(t-d_{ij}^{U_s})k1} + P_{max}a_j^{tk} \le \sum_{(j,h''')\in A_G^s} outb_{jh'''}^{tk4} + \sum_{(j,h'')\in A_G^m} outb_{jh''}^{tk3}$$
$$+ \sum_{(j,h')\in A_U^m} outb_{jh'}^{tk2} + \sum_{(j,h)\in A_U^s} outb_{jh}^{tk1} \quad \forall k=1,.....,K; j\notin O; j\notin D; t\ge 0 \tag{3.6.36}$$

$$\sum_{(i,j)\in A_U^s} outb_{ij}^{tk1} + \sum_{(i,j')\in A_U^m} outb_{ij'}^{tk2} + \sum_{(i,j'')\in A_G^m} outb_{ij''}^{tk3} + \sum_{(i,j''')\in A_G^s} outb_{ij'''}^{tk4} \tag{3.6.37}$$
$$\ge P_{max}a_i^{tk} \quad \forall k=1,.....,K; i\in V; t=1,.....,T$$

doi:10.6844/NCKU202002825

3.7 非共乘之時空數學模式

在 3.6 節當中，我們規劃了一個可共乘之時空數學模式，其滿足了過往無人機和載具的協同作業的文獻中，無人機可共乘之特性，使之可用於一般的例子。由於本研究在設計網路圖時，即避免了無人機無法通過平面網路的情況，且假設無人機的充換電並不需要時間，因此不會出現一邊共乘移動一邊充換電的情況，所以在本研究所設計的例子，皆可先以「非共乘」模式求解。故我們可刪減變數 v_{ij}^{tk} 、w_{ij}^{tl} 、inb_{ij}^{tk3} 與 $outb_{ij}^{tk3}$ ，修改或刪除某些限制式，將圖 3.8 的可共乘時空網路圖修改為圖 3.9 的非共乘時空網路圖，以大幅改善求解速度。接下來會列出非共乘之時空網路數學模式，並說明與可共乘時空網路數學模式不同之處。

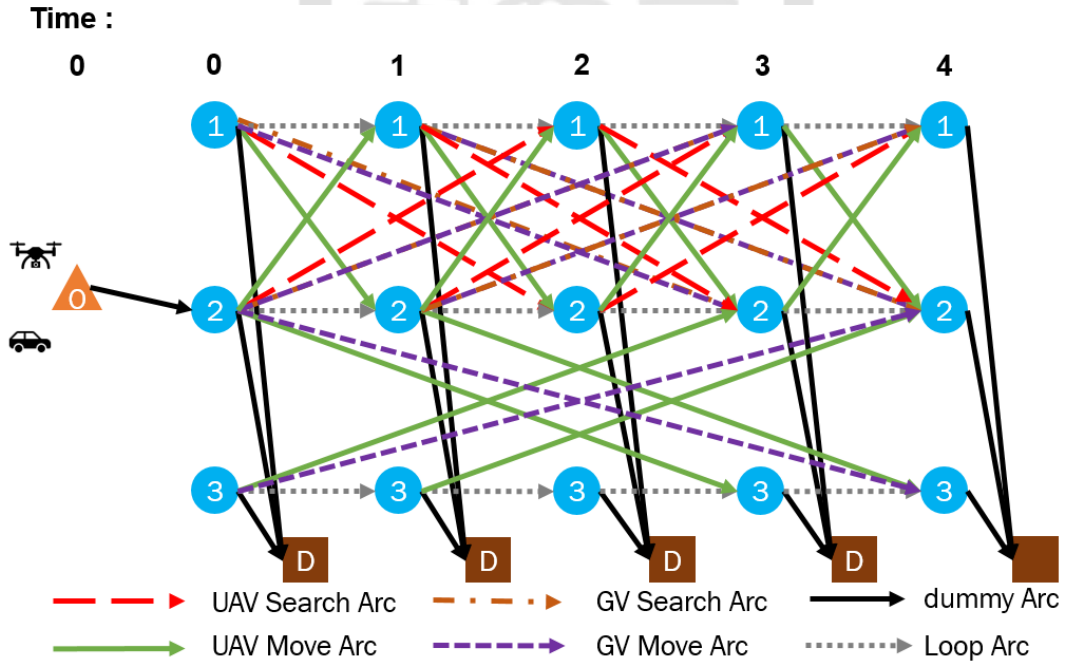


圖 3.9 非共乘之時空網路圖

doi:10.6844/NCKU202002825

$$\text{Minimize} \quad \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A^s} x_{ij}^{tk}\left(t+\frac{d_{ij}^{U_s}}{2}\right)\alpha_{ij}$$
$$+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A^s} w_{ij}^{tl}\left(t+\frac{d_{ij}^{G}}{2}\right)\alpha_{ij}+\varepsilon\Big(\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_U^s} x_{ij}^{tk}d_{ij}^{U_s} \tag{3.7.1}$$
$$+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_U^m} y_{ij}^{tk}d_{ij}^{U_m}+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^s} w_{ij}^{tk}d_{ij}^{G}$$
$$+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{(i,j)\in A_G^m} z_{ij}^{tk}d_{ij}^{G}\Big)$$

$$\sum_{(i,j)\in A_{start}} y_{ij}^{0k}=1 \qquad \forall k=1,\ldots\ldots,K \tag{3.7.2}$$

$$\sum_{(i,j)\in A_{start}} z_{ij}^{0l}=1 \qquad \forall l=1,\ldots\ldots,L \tag{3.7.3}$$

$$\sum_{(i,j)\in A_U^s} x_{ij}^{tk}+\sum_{(i,j')\in A_U^m} y_{ij'}^{tk}$$
$$=\sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k}+\sum_{(h',i)\in A_U^m} y_{h'i}^{(t-d_{h'i}^{U_m})k} \tag{3.7.4}$$
$$\forall i\notin O; i\notin D; k=1,\ldots\ldots,K; t=0,\ldots\ldots,T$$

$$\sum_{(i,j)\in A_G^s} w_{ij}^{tl}+\sum_{(i,j)\in A_G^m} z_{ij}^{tl}=\sum_{(h,i)\in A_G^s} w_{hi}^{(t-d_{hi}^{G})l}+\sum_{(h',i)\in A_G^m} z_{h'i}^{(t-d_{h'i}^{G})l} \tag{3.7.5}$$
$$\forall i\notin O; i\notin D; l=1,\ldots\ldots,L; t=0,\ldots\ldots,T$$

$$\sum_{t=1}^{T}\sum_{(i,j)\in A_{end}} y_{ij}^{tk}=1 \quad \forall k=1,\ldots\ldots,K \tag{3.7.6}$$

$$\sum_{t=1}^{T}\sum_{(i,j)\in A_{end}} z_{ij}^{tl}=1 \quad \forall l=1,\ldots\ldots,L \tag{3.7.7}$$

$$\sum_{k=1}^{K}\sum_{t=0}^{T} x_{ij}^{tk}+\sum_{k=1}^{K}\sum_{t=0}^{T} x_{ji}^{tk}+\sum_{k=1}^{K}\sum_{t=0}^{T} w_{ij}^{tl}+\sum_{k=1}^{K}\sum_{t=0}^{T} w_{ji}^{tl}=1 \tag{3.7.8}$$
$$\forall (i,j)=a\in A^s; (j,i)=\bar{a}\in A^s$$

$$\sum_{(i,j)\in A_U^s} x_{ij}^{tk}+\sum_{(i,j)\in A_U^m} y_{ij}^{tk}\le 1 \qquad \forall t=0,\ldots\ldots,T; k=1,\ldots\ldots,K \tag{3.7.9}$$

$$\sum_{(i,j)\in A_G^s} w_{ij}^{tk}+\sum_{(i,j)\in A_G^m} z_{ij}^{tk}\le 1 \qquad \forall t=0,\ldots\ldots,T; l=1,\ldots\ldots,L \tag{3.7.10}$$

$$inb_{ij}^{tk1}\le P_{max}x_{ij}^{tk} \quad \forall k=1,\ldots\ldots,K; t<T-d_{ij}^{U_s}; (i,j)\in A_U^s \tag{3.7.11}$$

$$outb_{ij}^{tk1}\le P_{max}x_{ij}^{tk} \quad \forall k=1,\ldots\ldots,K; t<T-d_{ij}^{U_s}; (i,j)\in A_U^s \tag{3.7.12}$$

$$inb_{ij}^{tk2}\le P_{max}y_{ij}^{tk} \quad \forall k=1,\ldots\ldots,K; t<T-d_{ij}^{U_m}; (i,j)\in A_U^m \tag{3.7.13}$$

$$outb_{ij}^{tk2}\le P_{max}y_{ij}^{tk} \quad \forall k=1,\ldots\ldots,K; t<T-d_{ij}^{U_m}; (i,j)\in A_U^m \tag{3.7.14}$$

doi:10.6844/NCKU202002825

$$out_{0j}^{0k2} = P_{\max} y_{0j}^{0k} \qquad \forall k = 1,.....,K \tag{3.7.15}$$

$$out_{ij}^{tk2} = inb_{ij}^{tk2} \qquad \forall k = 1,.....,K; t < T; i = j \tag{3.7.16}$$

$$inb_{ij}^{tk1} = outb_{ij}^{tk1} - x_{ij}^{tk} d_{ij}^{U_s} \qquad \forall k = 1,.....,K; t + d_{ij}^{U_s} < T; (i,j) \in A_U^s \tag{3.7.17}$$

$$inb_{ij}^{tk2} = outb_{ij}^{tk2} - y_{ij}^{tk} d_{ij}^{U_m} \qquad \forall k = 1,.....,K; t + d_{ij}^{U_m} < T; (i,j) \in A_U^m \tag{3.7.18}$$

$$\begin{aligned} Ma_{ik}^{t} \geq (&\sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k} + \sum_{(h,i)\in A_U^s} y_{h'i}^{(t-d_{h'i}^{U_m})k} \\ &+ \sum_{l=1}^{L} \sum_{(h'',i)\in A_G^m} z_{h''i}^{(t-d_{h''i}^{G_m})l} + \sum_{l=1}^{L} \sum_{(h''',i)\in A_G^s} w_{h'i}^{(t-d_{h''';i}^{G_s})l}) - 1 \\ &\forall k = 1,.....,K; i \in V \end{aligned} \tag{3.7.19}$$

$$a_{ik}^{t} \leq \sum_{(h,i)\in A_U^s} x_{hi}^{(t-d_{hi}^{U_s})k} + \sum_{(h',i)\in A_U^m} y_{h'i}^{(t-d_{h'i}^{U_m})k} \qquad \forall k = 1,.....,K; i \in V \tag{3.7.20}$$

$$a_{ik}^{t} \leq \sum_{l=1}^{L} \sum_{(h'',i)\in A_G^m} z_{h''i}^{(t-d_{h''i}^{G_m})l} + \sum_{l=1}^{L} \sum_{(h''',i)\in A_G^s} w_{h'i}^{(t-d_{h''';i}^{G_s})l} \qquad \forall k = 1,.....,K; i \in V \tag{3.7.21}$$

$$\begin{aligned} &\sum_{(i,j)\in A_{source}} inb_{ij}^{(t-d_{ij}^{U_m})k2} = \sum_{(j,h)\in A_U^m} outb_{jh}^{tk2} + \sum_{(j,h')\in A_U^s} outb_{jh}^{tk1} \\ &\forall k = 1,.....,K; j \notin O; j \notin D; t = 0 \end{aligned} \tag{3.7.22}$$

$$\begin{aligned} &\sum_{(i',j)\in A_U^m} inb_{i'j}^{(t-d_{i'j}^{U_m})k2} + \sum_{(i,j)\in A_U^s} inb_{ij}^{(t-d_{ij}^{U_s})k1} + P_{max} a_j^{tk} \\ &\geq \sum_{(j,h')\in A_U^m} outb_{jh'}^{tk2} + \sum_{(j,h)\in A_U^s} outb_{jh}^{tk1} \quad \forall k = 1,.....,K; j \notin O; j \notin D; t \geq 0 \end{aligned} \tag{3.7.23}$$

$$\begin{aligned} &\sum_{(i',j)\in A_U^m} inb_{i'j}^{(t-d_{i'j}^{U_m})k2} + \sum_{(i,j)\in A_U^s} inb_{ij}^{(t-d_{ij}^{U_s})k1} + P_{max} a_j^{tk} \\ &\leq \sum_{(j,h')\in A_U^m} outb_{jh'}^{tk2} + \sum_{(j,h)\in A_U^s} outb_{jh}^{tk1} \quad \forall k = 1,.....,K; j \notin O; j \notin D; t \geq 0 \end{aligned} \tag{3.7.24}$$

$$\begin{aligned} &\sum_{(i,j)\in A_U^s} outb_{ij}^{tk1} + \sum_{(i,j')\in A_U^m} outb_{ij'}^{tk2} \geq P_{max} a_i^{tk} \\ &\forall k = 1,.....,K; i \in V; t = 1,.....,T \end{aligned} \tag{3.7.25}$$

3.7 節的非共乘時空數學模式，相較 3.6 節中的可共乘數學模式，前者減少了與共乘相關的變數v_{ij}^{tk}、w_{ij}^{tl}、inb_{ij}^{tk3}、$outb_{ij}^{tk3}$、inb_{ij}^{tk4}與$outb_{ij}^{tk4}$有關的限制式(3.6.15)到式(3.6.18)、式(3.6.23) 、式(3.6.24)、式(3.6.28)到式(3.6.23)。並修改了目標式(3.6.1)成式(3.7.1)，減少共乘模式下無人機的移動距離的計算$\sum_{a\in A_G} v_{ij}^{tk} d_{ij}^{G_m}$。將式(3.6.4)變成式(3.7.4)，修改流量平衡的限制式，減少的共乘模式下的無人機移動

doi:10.6844/NCKU202002825

節線 v_{ij}^{tk} 。同理將式(3.6.9)修改為式(3.7.9)，限制無人機的節線選擇。最後則是將式(3.6.35)到式(3.6.37)修改為式(3.7.23)到式(3.7.25)，將記錄共乘模式的電量變數 inb_{ij}^{tk3} 、$outb_{ij}^{tk3}$ 、inb_{ij}^{tk4} 和 $outb_{ij}^{tk4}$ 刪除。

3.8 小結

本節首先針對問題描述，說明無人機與動態充換電站之移動網路、靜態標靶可能存在之節線、經過同樣節線時不同移動模式的差別與可共乘與非共乘模式使用時機及差異，接著說明此問題的假設與參數定義，再比較 Jotshi and Batta (2008)與呂昀軒(2019)與本研究對於期望搜索時間公式的差異，最後再開發可適用於通用案例的可共乘之時空數學模式，與適用於本研究設定的非共乘之數學模式。但兩種模式皆由於整數規劃變數與限制式過多，對於中大型網路，無法在短時間內求得最佳解，因此下一節將會針對此無人機與動態充換電站之二階層式區為路徑規劃問題提出較整數規劃模式更有效率之演算法。

doi:10.6844/NCKU202002825

第四章 結合動態充換電站之無人機群搜索路徑之求解演算法設計

在第三章我們以數學模式為基礎並利用求解器進行求解，然而當面對較大規模之問題時數學模式需耗時甚久，但在現實中的搜索任務，往往需要與時間賽跑，因此本節提出啟發式演算法，針對非共乘網路進行求解，使其能在短時間內獲得高品質的解，以符合現實的需求。4.1 節將針對路徑進行編碼；並在 4.2 節提出解碼演算法，以獲得無人機和動態充換電站的路徑，而在 4.3 節中提出建置初始解的方式，期望能夠獲得高品質的初始解，以減少收斂的時間，最後將上述的編碼方式應用在 4.4 節中的區域搜尋法，使用迭代的方式進行收斂。

4.1 編碼方式(Encode)

本研究參考 Groves and Van Vuuren (2005) 針對 KRPP 問題所提出的一組編碼方式，記錄每台車負責的標靶節線。假設有一無向完全網路 $G=(N,A)$，N 為所有節點的集合，A 為任兩點間所有無向節線之集合，$A_s \subseteq A$ 為可能存在靜態標靶之節線，其中 $A=\{1,2,3,4,5,6,7\}$，$A_s=\{1,3,4,5,7\}$，共有 K 台無人機以及 L 組動態充換電站，而載具編號包含無人機和動態充換電站，先從無人機開始編號 $(1\sim K)$，再編號動態充換電站 $(K+1\sim K+L)$，故如表 4.1 所示，可知第一架無人機分配到節線 1，第二架無人機分配到節線 3、4，第一組動態充換電站則分配到節線 5、7，其解空間為 $\frac{(|A_s|+K+L-1)!}{|A_s|!(K+L-1)!}$，然而此編碼方式不包含順序以及方向，需要在解碼(Decode) 4.2 節決定。

doi:10.6844/NCKU202002825

表 4.1 編碼範例

載具編號	目標節線
1	1
2	3、4
3	5、7

4.2 解碼演算法(Decoding Algorithm，DA)

本研究參考第二章文獻回顧中提到的 Luo et al. (2017)提出的演算法，該研究提出兩種演算法：1.先規劃無人機移動路徑，再規劃動態充換電站路徑 2.先規劃動態充換電站路徑，再規劃無人機移動路徑。起先，我們嘗試第二種演算法，先固定動態充換電站路徑，但發現即使我們將由數學模式得出的最佳動態充換電站移動路徑當作演算法的已知資訊，進而規劃無人機的移動路徑，仍可能因路徑太多，無法得到高品質的解。因此我們轉而嘗試第一種演算法，我們先使用貪婪演算法規劃無人機的路徑，由於動態充換電站已知各無人機需充換電的時間和地點，因此動態充換電站只需決定負責充換電的無人機順序即可，大幅縮減了其可能的移動路徑。

從本研究第三章所發展的數學模式觀察其解，可發現因為無人機的移動速度較快，因此會負責大部分的搜索任務；而動態充換電站則主要負責提供電力，僅可能在與無人機會合的途中負責部分的搜索任務，因此我們在已知每架無人機和動態充換電站需負責的標靶節線後，可先規劃出無人機的移動路徑，記錄無人機需要充換電的地點與時間，再去規劃動態充換電站的移動路徑以滿足任務及協助無人機充換電，此方法可有效地進行解碼，後續在 4.2.2 小節中我們也會以一個範例進行說明。

doi:10.6844/NCKU202002825

4.2.1 演算法步驟

我們將演算法分為「無人機路徑規劃階段」以及「動態充換電站路徑規劃階段」兩個階段，如圖 4.1 所示。我們輸入無人機和動態充換電站的初始位置 start_node、負責的標靶節線 TASK，以及無人機的最大電量 max_P 和可以停靠的充換電點 C，C 為各動態充換電站負責的充換電點之集合，提供無人機的充換電選擇，其如何設定則會在 4.3 節的初始解建置進行更詳細的說明。在無人機路徑規劃階段，我們將使用最短路徑讓無人機行至標靶節線，若電量不足，則會看是否曾經過充換電點或尋找距離最近的充換電點。完成此階段後，我們會得到每架無人機需要充換電的節點資訊，包含位置以及時間，將該資訊輸入至下個階段。在動態充換電站路徑規劃階段，我們要規劃其路徑以提供無人機電量以及尋訪標靶節線，並同時修正無人機的路徑，使兩者可以在同樣的時刻交會以充換電，最終輸出兩者的路徑，並記錄目標值。

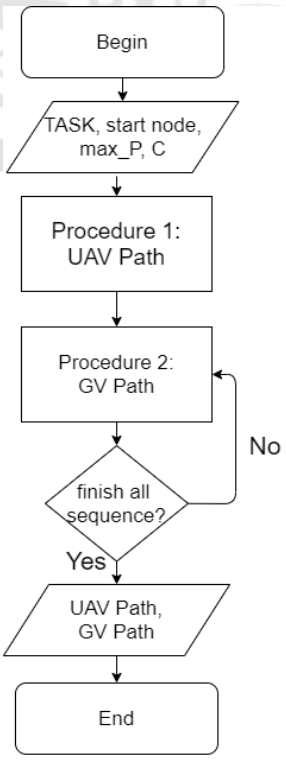


圖 4.1 演算法流程圖

doi:10.6844/NCKU202002825

無人機路徑規劃階段(UAV Path Programming Phase)

表 4.2 為無人機路徑規劃的步驟，假設已知網路$G=(N,A)$，N 為所有節點的集合，A 為任兩點間所有無向節線之集合，而目前第k架無人機已走的路徑為$S_k=(s_1,...,s_{i_k})$，目前的位置為s_{i_k}，未完成標靶節線$\mathrm{TASK}_k=(task_1,...,task_{i_k})$，無人機當前電量$p_k$、無人機可停靠進行充換電的點$C=(c_1,....,c_i)$，$Park_k=(park_1,...,park_{i_k})$為之前進行充換電的點，其值代表 S 中的索引值，如$park_1=2$，代表在s_2進行第一次充換電，記錄用以計算電量以及後續動態充換電站的路徑規劃，利用上述的資訊使無人機k尋訪所有標靶節線，並記錄每個充換電點的位置以及時刻。

doi:10.6844/NCKU202002825

表 4.2 無人機路徑規劃流程

UAV Path Programming

Data: $G, TASK_k, C, S_k, p_k, t_k, Park_k$

1: ***function*** *UAV_Path(*$G, TASK_k, C, S_k, p_k, t_k, Park_k$*)*

2: ***while*** $TASK_k$!=Null

3: *Dijkstra(*$G, s_{i_k}, Task_k$*)*

4: $nextTask \leftarrow$ *closest task* can *arrive with* p_{i_k} ***(1)***

5: ***if*** *have nextTask* ***then***

6: ***Update*** S_k *from* s_{i_k} *to nextTask* ***(2)***

7: ***else:***

8: ***if*** *C in the path from* $s_{park_{i_k}}$ *to* s_{i_k} ***then***

9: ***Update*** $Park_k, p_k, S_k$ *from* s_{i_k} *to nextTask* ***(3)***

10: ***else***

11: *Dijkstra(*G, s_{i_k}, C*)*

12: ***if*** *there is no C can arrive with* p_k

13: ***Update*** $Park_k, C$ *with* s_{i_k}, p_k

14: $nextTask \leftarrow$ *closest task* can *arrive with* p_{i_k}

15: ***Update*** S_k *from* s_{i_k} *to nextTask* ***(4)***

16: ***else***

17: ***for*** *potentialCharge* ***in*** *the C which can arrive with* p_{i_k} ***(5)***

18: *Dijkstra(G, potentialCharge,Task)*

19: ***Record*** *minimal combination(chargeNode,nextTask)*

20: ***Update*** S_k *from* s_{i_k} *to chargeNode*

21: ***Update*** S_k *from chargeNode to nextTask*

22: ***end While***

23: ***end function***

Result: $S_k, Park_k$

(1) 尋找當前電量 p_k 足夠完成的最小期望搜尋時間的標靶節線。

doi:10.6844/NCKU202002825

(2) 更新路徑 S_k，加入無人機從現在的位置到標靶節線的最短路徑。

(3) 如果現有電量 p_k 不足以走完下一個標靶節線，尋找從上個充換電點 $park_{i_k}$ 到現在的節點 s_{i_k}，是否有經過停靠進行充換電的點 C，若有加入 $park_{i_k}$，則更新電量 p_k，更新 S_k 加入無人機從現在的位置到標靶節線的最短路徑。

(4) 若過往路徑沒有經過充換電點 C，且當前電量不足以到其它的停靠點，則將現在的點 s_{i_k} 當作停靠點 C，再尋找最小期望搜尋時間的標靶節線，更新 S_k 加入無人機從現在的位置到標靶節線的最短路徑。

(5) 若過往路徑沒有經過充換電點 C，尋找當前電量可到的所有充換電點 *potentialCharge*，再找從這些充換電點到所有剩餘任務 $Task_k$ 中的最短路徑組合，並更新路徑 S_k。

動態充換電站路徑規劃階段(UAV Path Programming Phase)

從無人機路徑規劃階段，我們可以得到每一台無人機的移動路徑 $S=(S_1,....,S_k)$ 和充換電點順序 $Park=(Park_1,...,Park_k)$，譬如表 4.3 中，第一架無人機須依序尋訪節點 3、5，第二架須依序尋訪節點 1、3，而表 4.4 為每組動態充換電站各自負責的充換電點 c_i，第一組動態充換電站負責節點 1、3，第二台負責節點 5。以表 4.3、表 4.4 的範例可得出表 4.5 的兩種組合，其中[1,2]代表尋訪第一台無人機的第二個充換電點，故若[1,1]尚未被完成，便無法去[1,2]進行充換電；而表 4.5 中的兩種組合，差別為第一組動態充換電站先完成任務[1,1]或是[1,2]，雖然其順序對該組動態充換電站並沒有影響，但卻會對另一台負責的[1,2]有影響，因為即使動態充換電站先到[1,2]，依然要等[1,1]先完成，故尋訪組合 1 理應比尋訪組合 2 快。表 4.6 為動態充換電站路徑規劃的流程，而在這個階段我們會排列所有可能的組合 $Seq=(seq_1,...,seq_n)$ 為無人機路徑規劃的步驟，將每種

doi:10.6844/NCKU202002825

組合搭配動態充換電站 l 本身需要去尋訪的標靶節線 $Task_l$ 去規劃每一組動態充換電站的路徑，而目前第 k 組動態充換電站已走的路徑為 $Q_k=(q_1,...,q_{i_k})$，目前的位置為 q_{i_k}，其中還可能會修改無人機的移動時間，在充換電點上等待動態充換電站到達，以進行充換電。

表 4.3 無人機充換電點順序

無人機編號 k	充換電順序 $Park_k$
1	3、5
2	1、3

表 4.4 動態充換電站負責節點

動態充換電站編號 l	負責的充電站 c_l
1	1、3
2	5

表 4.5 動態充換電站移動組合

動態充換電站編號 l	尋訪組合 1	尋訪組合 2
1	[1,1]、[2,1]、[2,2]	[2,1]、[1,1]、[2,2]
2	[1,2]	[1,2]

表 4.6 動態充換電站路徑規劃流程

GV Path Programming

Data: $G, TASK, Seq, S, Q$

1: ***function*** *GV_Path(G,TASK,Seq,S,Q)*

2: ***for*** *nowSeq* ***in*** *Seq* (1)

3: ***while*** *TASK*!=*Null* && *nowSeq*!= *Null* (2)

4: ***for*** *nowGV* ***in*** *L*

5: ***if*** *seq[nowSeq][nowGV]* =Null

6: ***Update*** *Q to finish all* $TASK_{nowGV}$ *by shortest path* (3)

doi:10.6844/NCKU202002825

7: ***else***

8: ***if*** *seq[nowSeq][nowGV].top() isn't next UAV park node* ***then***(4)

9: ***Continue***

10: ***else***

11: ***if*** any GV stay on *seq[nowSeq][nowGV].top()at the same time* ***then***

12: **Pop** seq[*nowSeq*][*nowGV*] (5)

13: **else**

14: *Dijkstra(* G, q_{i_k} , *seq*[*nowSeq*][*nowGV*].*top*() *)*

15: *Dijkstra(* $G, seq[nowSeq][nowGV].top(), Task_l$ *)*

16: *nextTask* ← *closest task*

17: $task\_time \leftarrow$ *the time from* q_{i_k} *to nextTask*

18: $chg\_time \leftarrow$ *the time from nextTask to seq*[*nowSeq*][*now*

19: ***if*** $time < uav\ arrive\ time + \lambda$ ***then*** (6)

20: ***Update*** Q_j *from* q_{i_k} *to nextTask to seq*[*nowSeq*][*nowGV*].*top*()

21: ***Update*** $Task_{nowGV}$

22: **Pop** seq[*nowSeq*][*nowGV*]

23: ***else***

24: $chargeTime \leftarrow$ *the time* GV_j *arrive seq*[*nowSeq*][*nowGV*].*top*()

25: ***if*** $chargeTime \le UAV\ arrive\ time$ ***then*** (7)

26: ***Update*** Q_{nowGV} *from* $q_{i_{nowGV}}$ *to seq*[*nowSeq*][*nowGV*].*top*()

27: **Pop** seq[*nowSeq*][*nowGV*]

28: ***else*** (8)

29: ***Update*** Q_{nowGV} *from* $q_{i_{nowGV}}$ *to seq*[*nowSeq*][*nowGV*].*top*()

30: ***Update*** *S to wait GV arrive*

31: **Pop** seq[*nowSeq*][*nowGV*]

32: **end While**

33: **end function**

Result: S, Q

doi:10.6844/NCKU202002825

(1) 掃過所有尋訪組合，每個組合會重置 GV、UAV 的路徑和 GV 的標靶節線。

(2) 若此組合 *nowSeq* 尚有停靠點或標靶節線 *TASK* 未被尋訪，繼續迴圈。

(3) 如果第 nowGV 組動態充換電站皆已尋訪所有該去的充換電點，但仍有標靶節線未被尋訪，則依最短路徑去找最近且仍未被尋訪之標靶節線，並更新路徑。

(4) 如表 4.5 的動態充換電站 2 的任務[1,2]，因動態充換電站 1 尚未完成任務[1,1]，故動態充換電站 2 最前面的任務尚無法完成，故先跳過更新其路徑。

(5) 若該動態充換電站要負責的充換電點已有其它動態充換電站在同樣時刻抵達，則代表此任務已完成。

(6) 若動態充換電站尋訪標靶節線後再到達充換電點的時間，小於無人機到達該充換電點加上等待常數λ，則可先去尋訪標靶節線，再去進行充換電。

(7) 如果動態充換電站較無人機早到，則更新動態充換電站路徑，並等待無人機。

(8) 如果動態充換電站較無人機晚到，則更新動態充換電站路徑，並更新無人機路徑，以等待動態充換電站前來進行充換電。

4.2.2 範例說明

假設有一網格網路圖如圖 4.2，為方便說明，假設無人機和動態充換電站只可於實線節線上移動，無人機的移動模式需 1 單位時間、搜索模式需 2 單位時間，動態充換電站移動和搜索模式皆需 3 單位時間；藍色實線節線則為標靶節線，其存在各實線之機率均為 25%。假設現有兩台無人機及一組動態充換電站，無人機當下電量 $p_1 = p_2 = 3$、無人機最大電量 $P_{\max} = 3$、無人機所在節點 $n_1 = 1$、$n_2 = 2$，動態充換電站所在節點 $n_3 = 2$，等待常數 $\lambda = 2$，充換電點 $C = 2,5,6$。無人機和動態充換電站各自負責的標靶節線則為 $Task_1 = (1,4),(4,5)$、$Task_2$

doi:10.6844/NCKU202002825

$=(3,6)$、$Task_3=(2,5)$。

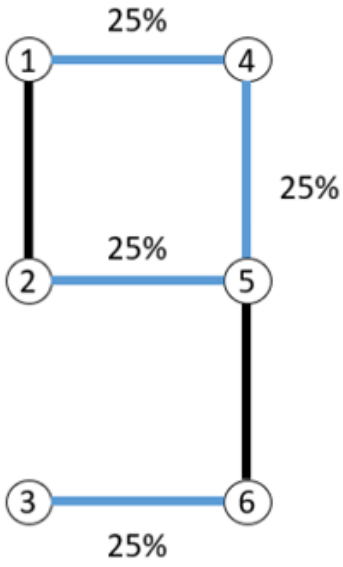


圖 4.2 DA 說明範例圖

無人機路徑規劃階段

在此以第一架無人機為例，其出發點為節點 1，標靶節線為節線(1,4)、(4,5)，以最短路徑去搜尋標靶節線，可知節線(1,4)較近，且使用搜索模式只需 2 單位時間，現有電量 p_1 仍足以應付，故可更新移動路徑 $S_1=(1,4)$ 、 $p_1=1$。

但當要從節點 4，尋訪標靶節線(4,5)時，發現現有電量不足以使用搜索模式通過節線(4,5)，且現有路徑 S 中不包含 C，故須尋找現有電量可到達的充換電點，因此使用移動模式前往節點 5，更新移動路徑 $S_1=(1,4,5)$，記錄充換電點 $Park_1=(5)$，並更新電量 $p_1=3$。

充換電後便可使用搜索模式通過節線(4,5)，並完成此架無人機的搜尋任務，最終路徑 $S_1=(1,4,5,4)$ 、 $Park_1=(5)$。

doi:10.6844/NCKU202002825

第二架無人機也可以得到最終路徑 $S_2=(2,5,6,3)$、$Park_2=(6)$，其中 $Park_1$ 的節點 5、$Park_2$ 的節點 6 皆會在第 3 單位時間到達。

動態充換電站路徑規劃階段

從無人機路徑規劃階段，我們可以得到動態充換電站需尋訪 $Park=(Park_1,Park_2)$，可得到兩種組合 $seq_1=([1,1],[2,1])$、$seq_2=([2,1],[1,1])$，其中[1,1]為第一架無人機的第 1 個充換電點、[2,1]則為第二架無人機的第 1 個充換電點，且已知動態充換電站亦有標靶節線 $Task_3=(3,6)$、所在位置 $n_3=2$。

若以第一種組合 $seq_1=([1,1],[2,1])$ 為例，先尋訪第一架無人機的第一個充換電點，需在第 3 單位時間到達節點 5，此時先判斷是否能先完成標靶節線(3,6)，而完成標靶節線的時間為 3，小於無人機到達的時間 $3+\lambda$，故先完成標靶節線(3,6)，路徑為 $S_3=(2,5)$，此時 $n_3=5$，且剛好和無人機同時到達節點 5，故[1,1]已完成。

接著動態充換電站需到達節點 6 完成[2,1]，故須花 3 單位時間通過節線(5,6)，完成[2,1]的時間為 6，動態充換電站路徑為 $S_3=(2,5,6)$，但因為第二架無人機在第 3 單位時間就到了，故需要等待動態充換電站到達才能充換電，須更新第二架無人機的路徑 $S_2=(2,5,6,6,6,6,3)$。

故可知若動態充換電站選擇 $seq_1=([1,1],[2,1])$，$S_1=(1,4,5,4)$ $S_1=(1,4,5,4)$、$S_2=(2,5,6,6,6,6,3)$、$S_3=(2,5,6)$，便可以此去計算目標值，規劃並記錄較好的移動路徑。

4.3 建置初始解

在建立 4.2 節的 DA 後，只要我們分配任務到不同的無人機和動態充換電站，形成 4.1 節的 Encoding，並且確立了動態充換電站負責的充換電點 C，就能得到

doi:10.6844/NCKU202002825

一組解。而好的初始解，可以節省許多時間，例如在 DA 的過程中，若發現在更新路徑時，目標值已經超過現有的最佳解，便可以提前結束 DA。因此我們便設計了一種用於產生初始解的流程，該流程分成兩種部份：第一部分用於產生初始解 Encoding 的 K-平均演算法(K-Means Clustering Algorithm)，第二部分則是用於產生充換電點 C 的覆蓋路徑規劃(Coverage Path Planning)。

K-平均演算法(K-Means Clustering Algorithm)

K-Means 最早為 MacQueen (1967)學者所提出的，其使用數學模型進行求解，概念為對所有數據進行分組，將相似的數據歸類為一起，每一筆數據的能有一個分組，每一組稱作為群集 (Cluster)，而以本研究來說，便是將標靶節線進行分群，將距離相近的靜態標靶分配給相同的無人機，縮小每架無人機的移動距離。

Xu and Wunsch (2005)學者對於分群演算法進行整理，其引用 Forgy (1965)學者所提出的演算法，概念類似於數學中找重心的方式，方法如下。

1. 我們先決定要分 K 組，並隨機選 K 個點做群集中心。
2. 將每一個點分類到離自己最近的群集中心。
3. 重新計算各組的群集中心。
4. 反覆 2、3 動作，直到群集不變，群集中心不動為止。

本研究便參考上述的演算法，而從第三章設計的數學模式所得出的解，可以發現多數的靜態標靶為無人機所尋訪，動態充換電站主要是為無人機提供充換電服務，並在途中服務少數節線，故我們將 K 設定為無人機的個數，避免動態充換電站分到過多的標靶節線，卻又無法在充換電途中尋訪，只能完成充換電任務後，再去尋訪，使得動態充換電站的路徑過長。

而需要分群的點，則設定為每條標靶節線的中心點，此方法便可避免節線的兩端屬於不同分群的情形。如圖 4.3 的網路圖，所有節線皆為動態充換電站可移動的節線，而其中以黑色實線則為需要尋訪的標靶節線。圖 4.4 則為分群後的結

doi:10.6844/NCKU202002825

果，藍色節點為標靶節線的中心點，綠色節點則為每一個分群的中心點，所有連接到同一個綠色節點的藍色節點，便可視為同一個分群。

使用 K-means 得到每架無人機需要負責的標靶節線後，我們再將其中最靠近每組動態充換電站初始位置的的靜態標靶分配給該動態充換電站，如此每台無人機和動態充換電站，皆有需要尋訪的標靶節線，便可組成編碼，再使用 4.2 節中的 DA 進行求解。

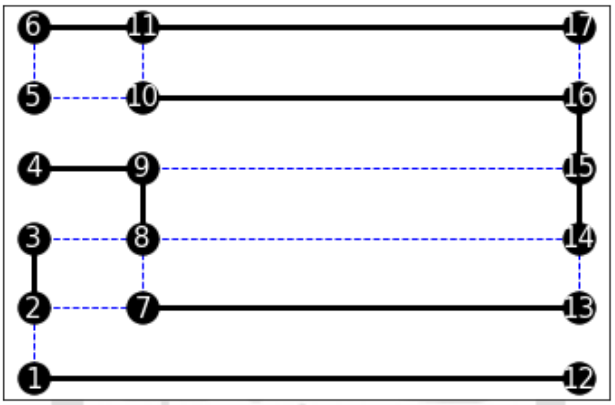


圖 4.3 K-means 範例(分群前)

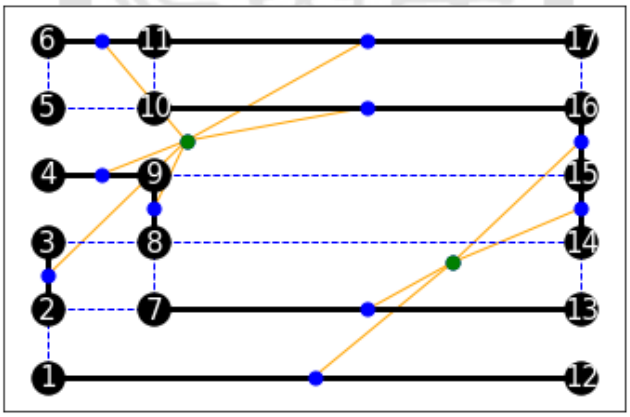


圖 4.4 K-means 範例(分群後)

覆蓋路徑規劃(Coverage Path Planning)

由於無人機的移動會受限於動態充換電站何時提供電力，若無人機因電力因素任意的停靠在節點上，可能使得原先便移動較慢且受限於二維平面移動的動態

doi:10.6844/NCKU202002825

充換電站，需花更多時間進行移動，才能提供充換電服務，如圖 4.3 左圖的網路圖中，若無人機與動態充換電站在節點 8 進行充換電，接著無人機通過節線(8,7)以尋訪標靶節線(7,13)，但在要尋訪下一個標靶節線(12,1)時，無人機剛好在通過節線(13,12)後消耗完所有電量，使其無法尋訪標靶節線(12,1)，只能停靠於節點 12 等待動態充換電站到達，而對於無人機來說，其可以突破傳統二維網路，直接從節點 13 飛往節點 12，但動態充換電站若要行經節點 12，需通過距離較遠的節線(1,12)，且很可能完成充換電後，還要通過節線(12,1)回來以提供其它充換電服務或尋訪標靶節線，但若我們決定了充換點為節點 13，則動態充換電站便不須移動較長的節線(12,1)，大幅縮短移動距離，故決定哪些節點可以提供充換電服務，除了可以有效地縮短動態充換電站的移動距離，同時也降低無人機等待充換電的時間，進而減少目標值。

而在 2.3 節的節線途程問題相關文獻中有回顧到 Hà et al. (2014)學者提出針對足夠接近節線途程問題，其將目標從需尋訪之特定節線修改為在節線附近之顧客集，便可應用於此。我們假設動態充換電站可覆蓋一定的範圍，意同於無人機若行駛至該覆蓋範圍，皆可與動態充換電站會合進行充換電，則如何規劃動態充換電站的移動路徑使其可以覆蓋全部的節點，便是我們要考慮的。

如圖 4.5 所示，綠色圓圈為動態充換電站所可以涵蓋的範圍，而若假設現有兩組動態充換電站進行搜索任務，且出發點分別於節點 7、10，則可規劃兩組動態充換電站路徑分別為 7→13 和 10→16，如此便可覆蓋所有的節點，使無人機不管行駛到哪，皆有鄰近的充換電點可以前往充換電，並分配充換電點給不同的動態充換電站去負責，以設定充換點 $C=(c_1,c_2)$, $c_1=(7,13), c_2=(10,16)$ 。

而針對上述的概念，我們設計兩種方法以求得充換電點，第一個方法為使用覆蓋路徑規劃的數學模式以求得動態充換電站的最短路徑；另一個方法則是較接近一般搜救隊的想法，使用貪婪演算法挑選節點，直到涵蓋範圍能覆蓋所有節點為止，而接下來我們則會針對兩種方法說明。

doi:10.6844/NCKU202002825

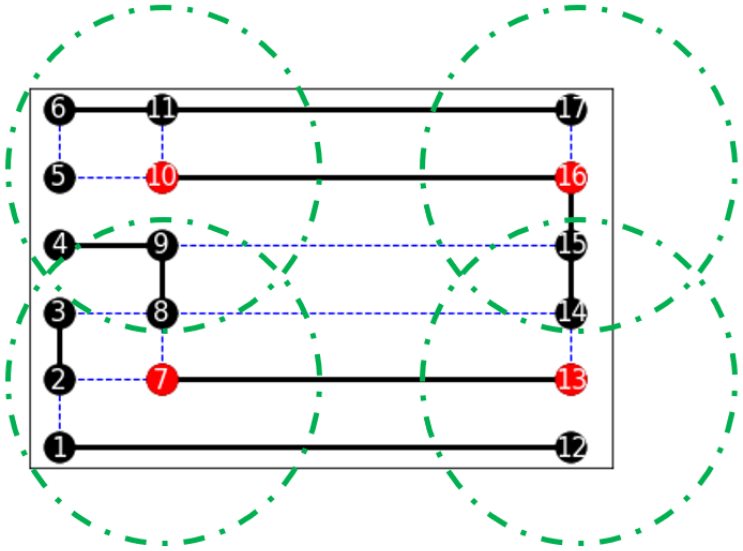


圖 4.5 動態充換電站覆蓋路徑規劃

充換電點設置之覆蓋路徑規劃數學模式

以下介紹以覆蓋路徑規劃數學模式設置充換電站，其中參數和集合可參考 3.4 節的定義，並新增集合 R_i 表示節點 i 可以涵蓋到的點，變數則修改為下列所述。

變數

x_{ij}^{tl}　　動態充換電站 l 在第 t 單位時刻行走移動節線 $(i,j)=a\in A_G^m$ 為 1;反之為 0

z_i^{tl}　　動態充換電站 l 在第 t 單位時刻到達節點 i $\forall i\notin O, i\notin D$ 為 1;反之為 0

w_i^{tl}　　節點 i 在第 t 單位被動態充換電站 l 涵蓋為 1;反之為 0

$visited_i$　　節點 i 被動態充換電站涵蓋的次數

doi:10.6844/NCKU202002825

目標式為式(4.3.1)，目標為最小化動態充換電站移動距離。

$$min \quad \sum_{t=1}^{T}\sum_{l=1}^{L}\sum_{(i,j)\in A_G} x_{ij}^{tl} d_{ij}^{G_m} \tag{4.3.1}$$

限制式(4.3.2)則為每組動態充換電站從虛擬起點出發。

$$\sum_{(i,j)\in A_{start}} x_{ij}^{0l} = 1 \qquad \forall l = 1,.....,L \tag{4.3.2}$$

限制式(4.3.3)則為每組動態充換電站的流量守恆限制。

$$\sum_{(i,j)\in A_G^m} x_{ij}^{tl} = \sum_{(h,i)\in A_G^m} x_{hi}^{(t-d_{hi}^G)l} \\ \forall i \notin O; i \notin D; l = 1,.....,L; t = 0,.....,T \tag{4.3.3}$$

限制式(4.3.4)則為每組動態充換電站結束任務後回到虛擬訖點。

$$\sum_{t=1}^{T}\sum_{(i,j)\in A_{end}} x_{ij}^{tl} = 1 \qquad \forall l = 1,.....,L \tag{4.3.4}$$

限制式(4.3.5)為記錄動態充換電站l在單位時間t到節點i。

$$\sum_{(j,i)\in A_G^m} x_{ji}^{(t-d_{ji}^G)l} = z_i^{tl} \qquad \forall i \notin O; i \notin D; l = 1,.....,L; t = 0,.....,T \tag{4.3.5}$$

限制式(4.3.6)為記錄節點j在單位時間 t 是否有被動態充換電站l涵蓋。

$$\sum_{i\in R_j} z_i^{tl} = w_j^{tl} \qquad \forall i \notin O; i \notin D; l = 1,.....,L; t = 0,.....,T \tag{4.3.6}$$

限制式(4.3.7)為記錄節點i被涵蓋的次數。

$$\sum_{t=1}^{T}\sum_{l=1}^{L} w_i^{tl} = visited_i \quad \forall i \notin O; i \notin D \tag{4.3.7}$$

限制式(4.3.8)為限制每個節點皆至少要被涵蓋過一次。

$$visited_i \geq 1 \quad \forall i \notin O; i \notin D \tag{4.3.8}$$

根據上述的數學模式，便可取得每組動態充換電站的移動路徑，而第l架的動態充換電站移動路徑，即可視為充換電點c_l，集合所有c_l則形成無人機的可充換點C。

doi:10.6844/NCKU202002825

充換電點設置之貪婪演算法

此方法先挑選鄰近節點(degree)數較多的節點，由於 degree 數越多的節點，表其連接的道路越多，可以更迅速的移動的其它節點，因此適合拿來當作充換電點，不論是無人機要回到此點進行充換電，或是動態充換電站要在此等待，都會相較 degree 數較少的節點方便；而若 degree 數相同，則我們可依據 R_i 所涵蓋的範圍，優先挑選涵蓋範圍較大的節點，方法如下。

1.依照 degree 數，由多至少，排出每個節點的順序。

2.再將 degree 數相同的節點依照涵蓋的範圍 R_i 由大至小進行排序。

3.挑選 degree 數最大的節點集中，涵蓋範圍同樣最大的節點，並修正 R，將已被包含的節點從 R 之中移除，再重複步驟 2，針對剩餘未挑選的節點，依其所包含尚未被覆蓋的節點個數，由多至少排序之。

4.反覆步驟 3，直到所有節點皆被包含為止。

5.使用 K-means 將所有被挑選出來的節點進行分群，分配給不同的動態充換電站負責。

4.4 區域搜尋法(Local Search，LS)

區域搜尋法為節線或節點排程中常用的啟發式演算法，其概念為從一組初始解開始，向鄰近解進行搜索。若鄰近解較佳，則再從該解搜索其鄰近解；反之則返回當前解繼續搜索。本研究參考 Hashimoto and Yagiura (2008)所提出的區域搜尋演算法，該論文利用 2-opt*、cross exchange 與 Or-opt 來搜尋鄰域解，由於該論文為節點排程問題，本論文則再加以修改以用於節線排程。

而其中 Or-opt 演算法是由 Reiter and Sherman (1965)所提出，其概念是移除路徑中的某一段路徑，再插入路徑中的其它位置，以產生新的路徑。本研究的解碼演算法對於編碼並沒有先後順序的差異，如圖 4.6 所示，雖然左右兩圖在無人

doi:10.6844/NCKU202002825

機編號 2、3 的標靶節線有先後順序的差異，但我們在 DA 的過程中，皆是從現有位置去搜尋期望搜救時間較小的標靶節線，故左右兩圖並無差異。但若是無人機和動態充換電站架數較少時，修改部分演算法，使用 or-opt，可使得疊代的次數與組合較多，增加其區域解的多樣性，故我們在第六章中會做加入 or-opt 的實驗，分析是否有較佳的表現。接下來我們則會針對編碼方式、鄰域列表、cross exchange、2-opt*、or-opt 加以介紹。

載具編號	目標節線
1	1
2	3、4
3	5、7

載具編號	目標節線
1	1
2	4、3
3	7、5

圖 4.6 兩組編碼

編碼方式

參考 4.1 節中的編碼方式，記錄每組無人機和動態充換電站負責的標靶節線，以圖 4.5 中左邊編碼為例，若有 2 架無人機和 1 組動態充換電站，則無人機 1 負責節線 1，無人機 2 負責節線 3、4，動態充換電站 1 則負責節線 5、7。

鄰域列表(Neighbor List)

鄰域列表為區域搜尋法中常使用的技巧，概念為記錄每一條節線附近的節線，在兩條路徑進行交換時，交換後才連結的兩條節線若在節線的鄰域列表中，才進行交換，反之則不進行交換，如此即使能有效率地進行節線交換。我們在程式的一開始便建立鄰域列表，並記錄距離每條節線最短的 n_{nlist} 條節線。

cross exchange 演算法

doi:10.6844/NCKU202002825

cross exchange 演算法為 Taillard, Badeau, Gendreau, Guertin, and Potvin (1997) 提出，其概念為移除兩段路徑中各一段至多長度為 L^{cross} 的路徑，若交換後所連接的節線有在鄰域列表中，則進行交換。如圖 4.7，針對兩台不同的載具中的各擷取一段長度至多為 L^{cross} 的路徑，從節點 $X1$ 後擷取節點 $X1'$ 到節點 $Y1$ 的路徑與節點 $X2$ 後節點 $X2'$ 到節點 $Y2$ 的路徑，判斷節點 $X1$ 與節點 $X2'$ 或是節點 $X2$ 與節點 $X1'$ 是否在鄰域列表中。若是則進行交換，再將交換後的編碼餵入解碼演算法中，並判斷新的編碼 S 的目標值 $E(S)$ 是否小於全域解的目標值 $E(Gbest)$。若是，則從新的編碼 S 繼續做 cross exchange 演算法；若否則不進行交換。而為了增加交換的效率，我們則針對任務完成時間最大和最小的兩條路徑進行交換，因為每架無人機和動態充換電站的任務時間越平均，整體的目標值也會較小，詳細可參考表 4.7 的流程，此方法的鄰域解數量為 $O((L^{cross})^2 mn_{nlist})$。

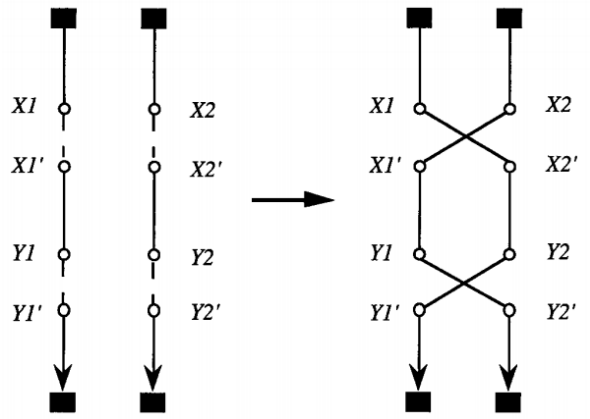
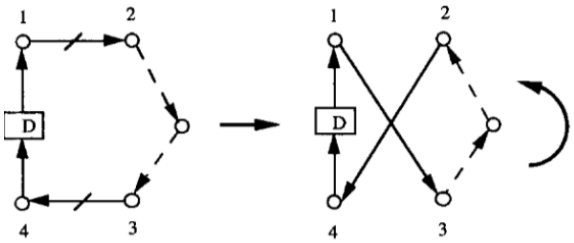
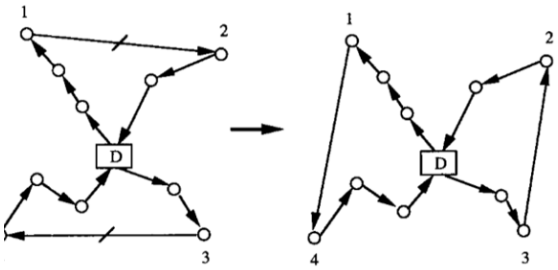


圖 4.7 cross change 示意圖(Taillard et al., 1997)

doi:10.6844/NCKU202002825



（a）2-opt 改變單一條路徑



（b）2-opt* 同時改變兩條路徑

圖 4.8 2-opt* vs 2-opt(Potvin et al., 1996)

doi:10.6844/NCKU202002825

表 4.7 cross exchange 流程

cross exchange

Data: Gbest, S

1: ***function cross_exchange***(Gbest,S)

2: ***for*** Vehicle1 sorted by task completion time in increasing order ***do***

3: ***for*** Vehicle2 sorted by task completion time in decreasing order ***do***

4: ***for*** node1 in Vehicle1 path ***do***

5: ***for*** node2 in Vehicle2 path ***do***

6 ***for*** pathlen1 in [1… L^{cross}] ***do***

7: ***for*** pathlen2 in [1… L^{cross}] ***do***

8: ***remove*** Vehicle1 path by node1 with the path length ***equal to*** pathlen1

9: ***remove*** Vehicle2 path by node2 with the path length ***equal to*** pathlen2

10: ***if*** node i and node $j+1$ or node j and node $i+1$ both are in the neighbor list ***then***

11: $S_{Vehicle1}, S_{Vehicle2} \leftarrow$ exchange two path

12: call DA function to get all vehicle path

13: calculate objective value E(S)

14: ***if*** E(Gbest) > E(S) ***then***

15: $Gbest \leftarrow S$

16: cross_exchange(Gbest,S)

17: ***end function***

doi:10.6844/NCKU202002825

2-opt* 演算法

2-opt*演算法由 Potvin, Kervahut, Garcia, and Rousseau (1996)其改善 Lin (1965)所提出的 2-opt 演算法而得，如 圖 4.8 所示。圖 4.8(a)為 2-opt 演算法，其將原先節線(1,2)、(3,4)交換成節線(1,3)、(2,4)，形成新的路徑；而圖 4.8 (b)的 2-opt*演算法同樣是交換節線，但用於多條路徑的交換，可同時改變多條路徑，而本研究需交換不同載具所負責的標靶節線，故使用 2-opt*演算法，詳細可參考表 4.8 的流程。此方法的鄰域解數量為 $O(mn_{nlist})$，m 為總節線數量，而同樣為了更有效率地交換，和 cross exchange 演算法相同，針對任務完成時間最大和最小的兩條路徑進行交換。

表 4.8 2-opt 流程

2-opt*
Data: *Gbest, S*
function 2-opt*(*Gbest, S*)
for *Vehicle1* sorted by completion time in increasing order **do**
for *Vehicle2* sorted by completion time in decreasing order **do**
for *node1* in *Vehicle1* path **do**
for *node2* in *Vehicle2* path **do**
separate *Vehicle1* path by *node1* into *path1A* and *path1B*
separate *Vehicle2* path by *node2* into *path2A* and *path2B*
if *path1A* last node and *path2B* first node in neighbor list or
path2A last node and *path1B* first node in neighbor list **then**
$S \leftarrow$ exchange *path1B* and *path2B*
call 2-opt* function(Gbest,S)
end function

doi:10.6844/NCKU202002825

Or-opt 演算法

Or-opt 演算法在本節初有介紹過，其概念是移除路徑中長度至多為 $L_{\text{path}}^{\text{intra}}$ 的路徑，並插入到距離原移除位置至多 $L_{\text{ins}}^{\text{intra}}$ 的位置。如圖 4.9，從節點 i 移除長度為 $L_{\text{path}}^{\text{intra}}$ 的路徑，將移除的路徑插入位置 $i+L_{\text{ins}}^{\text{intra}}$ ，並衡量交換完的新路徑的目標值是否較原本好。若是，則交換並從新的路徑開始尋找鄰域解；若否，則從原路徑繼續搜索。詳細的執行過程列於表 4.9。此方法的鄰域解數量為 $O(L_{\text{ins}}^{\text{intra}} L_{path}^{\text{intra}} m)$ 。

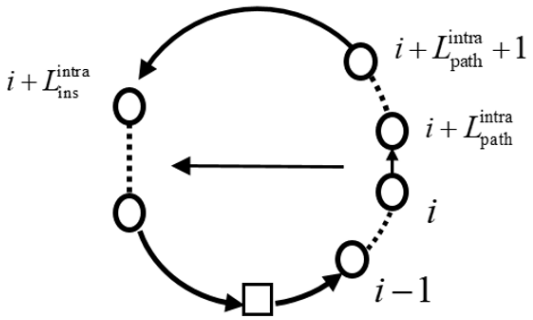


圖 4.9 Or-opt 交換示意圖

表 4.9 Or-opt 演算法流程

Or-opt

Data: S_k

1: **function** Or_opt(S_k)
2: **for** *node* in path **do**
3: **for** *len_intra_path* in [1,…, $L_{\text{path}}^{\text{intra}}$] **do**
4: **for** *loc_ins* in [*node*+1,…,*node*+ L_{ins}^{intra}] **do**
5: remove path from position *node* to *node*+ $L_{\text{path}}^{\text{intra}}$
6: $S_k' \leftarrow$ insert the path into position *node*+ L_{ins}^{intra}
7: **if** $E(S_k) > E(S_k')$ **then**
8: call Or-opt(S_k') function
9: **end function**

doi:10.6844/NCKU202002825

介紹上述的 2-opt*、cross exchange、or-opt 演算法後，我們嘗試了下述四種組合：(1)單純使用 2-opt*(如圖 4.10)、(2)單純使用 cross exchange(如圖 4.11)、(3)先使用 cross exchange 再使用 2-opt* (如圖 4.12)、(4)先使用 2-opt*再使用 cross exchange (如圖 4.13)。而若是加入 or-opt 的版本，則皆是在 initial solution 後使用 or-opt 再接續不同的區域搜尋方法，譬如圖 4.14 便是將圖 4.13 的執行流程加入 or-opt。

若單純使用 2-opt*或 cross exchange，便是一直交換直到停止條件滿足才停止程式。但若是先使用 cross exchange 再使用 2-opt*、先使用 2-opt*再使用 cross exchange，則是從下層的演算法開始，直到交換完後，再執行上層的演算法；若上層有更好的解出現，立即接受，並將新的解放回下層的演算法繼續交換，直到停止條件滿足才停止程式。

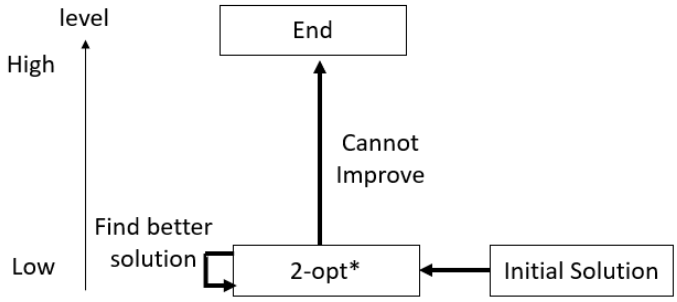


圖 4.10 純 2-opt*執行流程

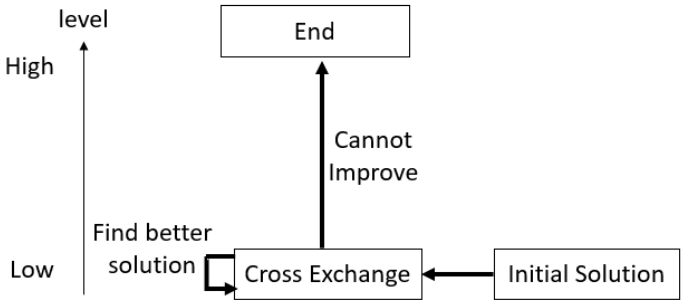


圖 4.11 純 cross exchange 執行流程

doi:10.6844/NCKU202002825

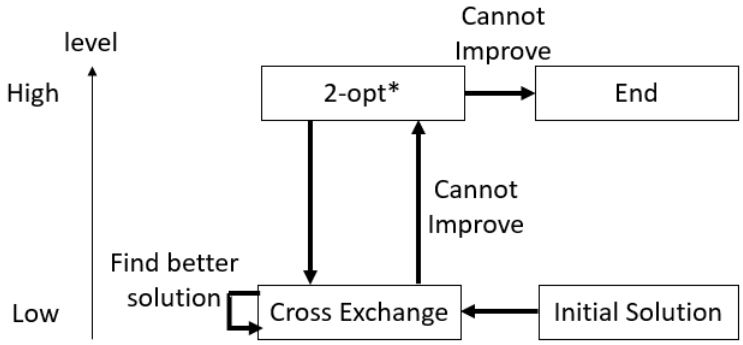


圖 4.12 先 cross exchange 再 2-opt*執行流程

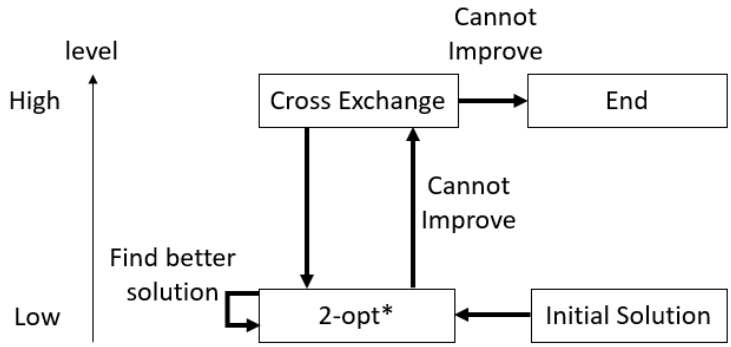


圖 4.13 先 2-opt*再 cross exchange 執行流程

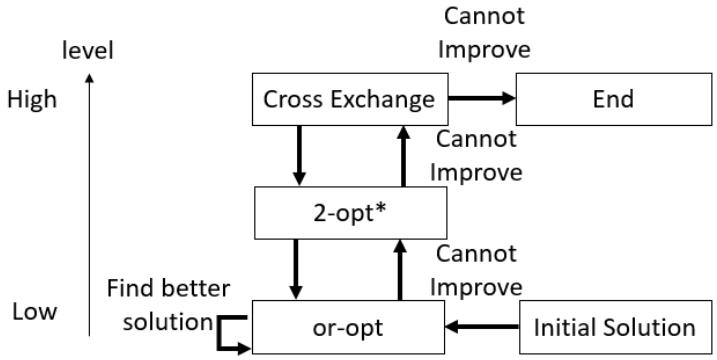


圖 4.14 加入 or-opt 的先 2-opt*再 cross exchange 執行流程

4.5 小結

在第三章時，我們提出了數學模式以計算最佳解，但數學模式在面對問題規模變大時，不僅求解時間會大幅增加，求解品質也會隨之下降，故我們在第四章開發演算法，使其在面對較大的問題規模時，也能在短時間內獲得高品質的解。

doi:10.6844/NCKU202002825

本章首先將解以編碼的方式呈現，記錄無人機和動態充換電站所負責的標靶節線，再利用解碼演算法去規劃兩者的路徑，使其能在考慮電力因素的情況下以最小化期望搜索時間去尋訪標靶節線，並使用部分方法建置初始解，縮小求解集。最後則是使用區域搜尋法，利用 2-opt*和 cross exchange、op-opt 演算法，交換兩架載具中的部分標靶節線，透過不斷迭代以找到最好的組合。

doi:10.6844/NCKU202002825

第五章 數值分析

本節針對第三章所提出的數學模式和第四章的啟發式演算法進行測試與分析，本研究的測試環境為 Windows 10 作業系統，Intel Core i7-6700，3.40GHZ 處理器，與 32G 記憶體，以 Python 為程式語言撰寫數學模型與數學演算法，並利用 Gurobi 9.0.2 版求解數學模型，其中求解時間 CPU TIME 設定為 1 小時(3600 秒)，以下說明測試網路參數設定，接著針對不同的參數設定與網路大小等不同情況來分析測試結果。

5.1 測試資料參數設定

本研究探討問題之網路圖為以網格圖當作動態充換電站的移動路徑，節點數為 n，網格的 row、column 數則為 n 最接近的一組公因數，且同一(直)行上的節點 x 軸座標相同、同一(橫)列上的節點 y 軸座標相同，如圖 5.1，$n=12$，row、column 則分別為 3、4，而同一列的節點(2,5,8,11)其 y 軸座標皆為 1，同一行的節點(4,5,6)其 x 軸座標皆為 4。

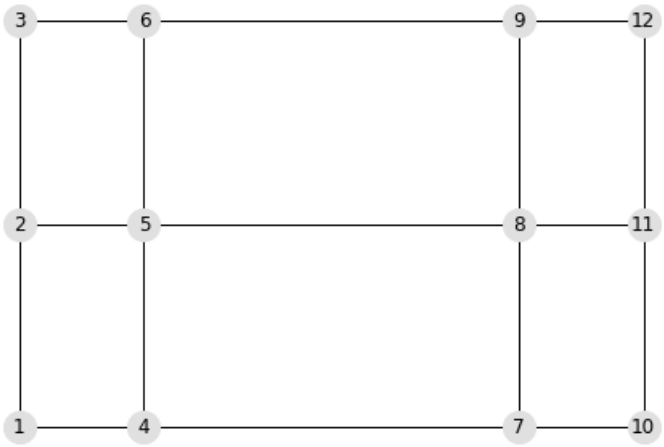


圖 5.1 網格圖(未刪除部分節線)

doi:10.6844/NCKU202002825

而為了增加網格圖的多變性，我們再隨機減少產生之節線，但又同時保持所有節點的連通性。以維持連通性，網路圖最少要有生成樹(spanning tree)為子圖，故節線數須至少為 $A_1 = n-1$，最多則為圖 5.2 中的原始產生的網格節線數 $A_2 = row(column+1)+column(row+1)$。而我們以設定新增節線比例 p 方式，來決定節線數 $m = A_1 + (A_2 - A_1)*p$，如圖 5.2 則為設定 $p = 25\%$ 刪除部分節線後的情況。

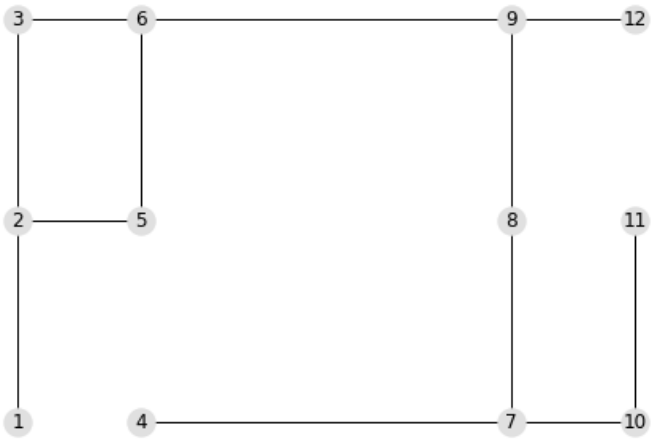


圖 5.2 網格圖(刪除部分節線)

而為了再增加節線長度的多變性，我們將一部分只有連接同一個行或列(亦即一進一出)的節點，如圖 5.2 中的節點 7 刪除，並連接其鄰近的兩節點，使節線長度增加，如圖 5.3。

doi:10.6844/NCKU202002825

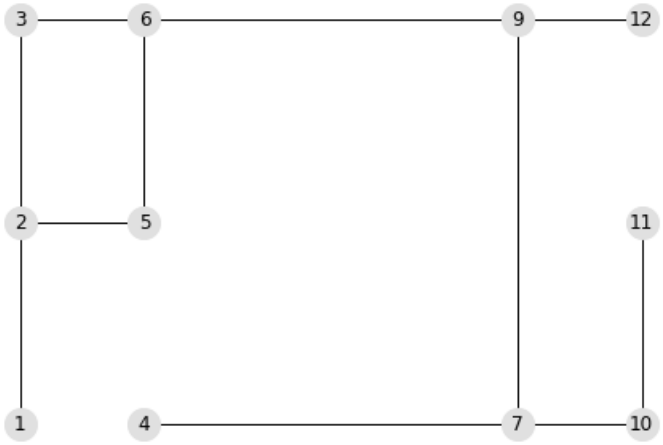


圖 5.3 網格圖(刪除部分節點)

圖 5.3 便為動態充換電站可移動之網路圖，可視為類似山區的傳統二維網路，而無人機可突破傳統二維網路，在任意兩節點中移動，故只要兩節點距離小於無人機的最大電量，無人機便可飛行，如圖 5.4 中的黃色節線。另外我們再將一半的節線設為標靶節線，如圖 5.4 中的藍色節線。而兩節點若以無人機的移動模式行進，所需時間則為兩節點的距離 $d=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$ ；而若使用無人機的搜索模式移動，所需時間則為 $2d$ ；動態充換電站的移動則不論搜索或移動模式皆為 $3d$ 。

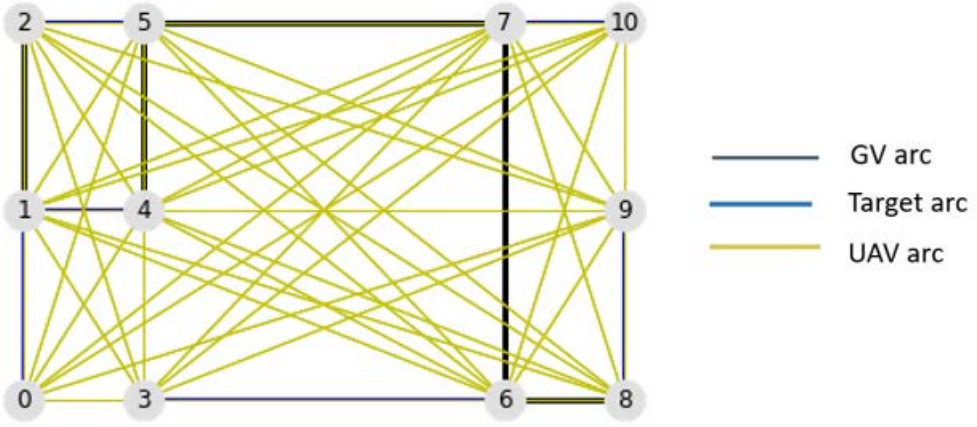


圖 5.4 網格圖(含無人機移動節線)

doi:10.6844/NCKU202002825

而雖然我們有可能刪除部分節點或節線，但由於數量不多，為方便記錄，仍以原先的節點數n和節線數m表示。除了上述網路圖的設定，其餘參數如無人機的數量為K、動態充換電站的數量為L、無人機的電量為 10。之後都以表 5.1 參數設定，針對不同網路規模與不同無人機數量和動態充換電站數量進行設定。

表 5.1 測試資料參數設定

			參數設定				
網路結構	**節線數**	**目標節線數**	**節點數**	**比例**	**無人機數**	**動態充換電站數**	**無人機電量**
網格圖	m	$\frac{m}{2}$	n	p	K	L	10

5.2 測試網路圖設定

本研究使用基於 python 開發的 NetworkX 套件(https://networkx.github.io)來產生隨機網格圖並檢查網路之連通性，表 5.2 定義 12 種n和 p 的組合以及節線數m，每種組合有 5 筆測資，每筆測資皆測試$K=1,2,3$和$L=1,2,3$的九種組合。

doi:10.6844/NCKU202002825

表 5.2 網路圖資訊

網路名稱	節點數 n	比例 p	節線數 m
Network1	12	50%	21
Network2	12	25%	16
Network3	12	12.5%	13
Network4	18	50%	31
Network5	18	25%	24
Network6	18	12.5%	20
Network7	20	50%	34
Network8	20	25%	26
Network9	20	12.5%	22
Network10	24	50%	40
Network11	24	25%	31
Network12	24	12.5%	27

5.3 結合動態充換電站之無人機群搜索路徑規劃問題之測試

針對結合動態充換電站之無人機群搜索路徑規劃問題的測試，首先我們針對第三章所設計的數學規劃模式進行測試，再比較數學啟發式演算法之求解效率與求解品質。

5.3.1 數學規劃模式比較

本小節針對第 3 節中所提出的兩種數學模式進行測試，圖 5.5 為非共乘數學模式在不同網路規模與型態下的求解表現，其中 *Num* 為該組網路測試資料數；*Num opt.*為該網路測試資料中求得最佳解之個數；*Num Arcs* 為所有節線的數量；*CPU Time* 為數學模式的求解時間；*Gap* 是利用最佳化軟體 Gruobi 之測試結果(求解時限為 3600 秒)，計算方式如下：

doi:10.6844/NCKU202002825

$Gap = \frac{UB - LB}{UB} \times 100\%$。$UB$ 及 LB 為 Gurobi 所求得之上下界。

Network	K	L	Num opt	Gap (%)	CPU Time(s)	K	L	Num opt	Gap (%)	CPU Time(s)	K	L	Num opt	Gap (%)	CPU Time(s)
Network1	1	1	4	20.0	728.74	2	1	5		123.08	3	1	3	40.0	1476.13
	1	2	5	0.0	1.31	2	2	5	0.0	0.70	3	2	5	0.0	3.70
	1	3	5	0.0	0.53	2	3	5	0.0	3.80	3	3	5	0.0	0.83
Network2	1	1	5	0.0	12.81	2	1	5	0.0	37.71	3	1	5	0.0	0.98
	1	2	5	0.0	12.46	2	2	5	0.0	0.75	3	2	5	0.0	3.63
	1	3	5	0.0	4.57	2	3	5	0.0	0.30	3	3	5	0.0	0.36
Network3	1	1	4	20.0	721.31	2	1	5	0.0	26.66	3	1	5	0.0	7.81
	1	2	5	0.0	1.97	2	2	5	0.0	1.28	3	2	5	0.0	0.43
	1	3	5	0.0	0.41	2	3	5	0.0	2.91	3	3	5	0.0	0.41
Network4	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	3	37.5	1667.38	2	2	3	40.0	1520.28	3	2	3	40.0	1450.32
	1	3	4	20.0	843.92	2	3	4	15.0	888.33	3	3	4	20.0	727.85
Network5	1	1	1	80.0	3073.79	2	1	1	80.0	3043.39	3	1	2	68.5	2314.67
	1	2	4	20.0	1083.86	2	2	2	62.3	2585.40	3	2	3	53.2	1462.30
	1	3	4	20.0	730.77	2	3	4	20.0	1098.30	3	3	4	20.0	736.58
Network6	1	1	2	60.0	2328.78	2	1	1	80.0	2907.55	3	1	3	60.0	1440.94
	1	2	3	20.0	1452.42	2	2	5	0.0	111.27	3	2	4	80.0	721.67
	1	3	5	0.0	10.27	2	3	5	0.0	4.03	3	3	5	0.0	10.51
Network7	1	1	1	80.0	3004.94	2	1	0	100	3600.00	3	1	0	100	3600.00
	1	2	2	60.0	2185.16	2	2	2	60.0	2449.75	3	2	2	60.0	2264.97
	1	3	3	40.0	1632.83	2	3	4	20.0	780.48	3	3	3	40.0	1699.09
Network8	1	1	3	40.0	2362.67	2	1	1	80.0	3446.86	3	1	1	60.22	2891.99
	1	2	3	40.0	1471.52	2	2	2	21.06	2371.24	3	2	3	40.0	1621.27
	1	3	5	0.0	29.01	2	3	5	0.0	253.70	3	3	4	20.0	749.59
Network9	1	1	2	60.0	2393.20	2	1	1	80.0	2906.23	3	1	0	100.0	3600.00
	1	2	2	60.0	2175.51	2	2	2	60.0	2262.28	3	2	3	22.6	1781.27
	1	3	5	0.0	41.95	2	3	3	40.0	1472.24	3	3	5	0.0	75.98
Network10	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	2	60.0	2987.23	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	1	80.0	2951.07	2	3	1	80.0	3154.95	3	3	1	63.0	2941.78
Network11	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	0	100.0	3600.00	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	1	80.0	2886.10	2	3	2	60.0	2700.97	3	3	2	60.0	2547.17
Network12	1	1	0	100	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	0	81.3	3600.00	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	2	60.0	2223.81	2	3	1	80.0	3508.54	3	3	2	60.0	2192.96

圖 5.5 非共乘數學模式測試(求解品質與效率)

由圖 5.5 可知，當無人機和動態充換電站數量少時，求解時間通常較長，例如同樣 Network 中兩架無人機和兩組動態充換電站的組合，比對三架無人機和兩組動態充換電站的組合，皆可以看到後者的 CPU Time 較低，判斷原因為前者每架載具要負責的標靶節線數較多，使得數學模式的時空層數、變數、限制式增加，造成求解時間增加。

動態充換電站的數量，同時也會影響到無人機的充換電的立即性。若動態充換電站較少，則需來回奔波幫無人機充換電，同樣會造成數學模式的時空層數、變數、限制式增加，造成求解時間增加。例如同樣 Network 中兩架無人機和一組

doi:10.6844/NCKU202002825

動態充換電站的組合，比對兩架無人機和兩組動態充換電站的組合，可看到後者的 CPU Time 較低。

當節點數n相同，但節線數m和標靶節線數皆較多時，如 Network4~Network6，可觀察在同樣的無人機數和動態充換電站數時，Network4 所需的 CPU Time 皆較多，判斷原因為標靶節線數較多，使得數學模式的時空層數、變數、限制式增加，造成求解時間增加。

圖 5.6 則為可共乘數學模式在不同網路規模與型態下的求解表現，可明顯看出相較於圖 5.5，大多數設定下的求解時間皆增加了不少，甚至在某些規模下，在時限內可求得最佳解的次數也變少了。如 Network 8 中，一台無人機和一組搜救隊的組合，當使用「非共乘數學模式」時，五個測資中尚可已得到三個可行解，但若使用「可共乘數學模式」時，僅剩一個最佳解，且求解時間也從 2362 秒，大幅增加至 3427 秒。由此便可證實我們在第三章所說明的，「可共乘數學模式」雖可處理大多數的情況，但也因為限制式較多，會造成求解時間的增加。因此可先以「非共乘數學模式」進行求解，若遇到需使用可共乘的情況，再改用可共乘數學模式。

doi:10.6844/NCKU202002825

Network	K	L	Num opt	Gap (%)	CPU Time(s)	K	L	Num opt	Gap (%)	CPU Time(s)	K	L	Num opt	Gap (%)	CPU Time(s)
Network1	1	1	4	20.0	734.75	2	1	5	0.0	86.84	3	1	3	40.0	1465.74
	1	2	5	0.0	1.70	2	2	5	0.0	1.00	3	2	5	0.0	2.92
	1	3	5	0.0	0.79	2	3	5	0.0	3.62	3	3	5	0.0	0.92
Network2	1	1	5	0.0	30.10	2	1	5	0.0	30.28	3	1	5	0.0	1.17
	1	2	5	0.0	15.47	2	2	5	0.0	0.81	3	2	5	0.0	6.04
	1	3	5	0.0	6.04	2	3	5	0.0	0.37	3	3	5	0.0	0.43
Network3	1	1	4	20.0	721.34	2	1	5	0.0	31.81	3	1	5	0.0	4.57
	1	2	5	0.0	2.54	2	2	5	0.0	1.60	3	2	5	0.0	0.34
	1	3	5	0.0	0.48	2	3	5	0.0	3.68	3	3	5	0.0	0.52
Network4	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	3	40.0	1639.34	2	2	3	40.0	1558.54	3	2	3	40.0	1460.04
	1	3	4	20.0	871.49	2	3	4	20.0	952.35	3	3	4	20.0	728.58
Network5	1	1	1	80.0	3111.76	2	1	1	80.0	3026.64	3	1	2	60.0	2346.16
	1	2	4	20.0	1060.57	2	2	2	60.0	2281.55	3	2	3	40.0	1467.68
	1	3	4	20.0	738.74	2	3	3	40.0	1448.23	3	3	4	20.0	734.87
Network6	1	1	2	60.0	2301.01	2	1	1	80.0	2890.79	3	1	3	40.0	1441.81
	1	2	3	40.0	1780.85	2	2	5	0.0	54.63	3	2	4	20.0	721.87
	1	3	5	0.0	40.75	2	3	5	0.0	6.47	3	3	5	0.0	13.00
Network7	1	1	1	80.0	3221.85	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	2	60.0	2192.33	2	2	2	60.0	2471.73	3	2	2	60.0	2229.61
	1	3	2	60.0	2193.09	2	3	4	20.0	794.92	3	3	3	40.0	1876.84
Network8	1	1	1	80.0	3427.59	2	1	1	80.0	3113.03	3	1	1	60.18	3173.60
	1	2	3	40.0	1478.17	2	2	3	28.92	2524.17	3	2	2	60.0	2183.36
	1	3	5	0.0	36.88	2	3	5	0.0	544.04	3	3	4	20.0	756.75
Network9	1	1	2	60.0	2846.95	2	1	2	60.0	3159.72	3	1	0	100.0	3600.00
	1	2	2	60.0	2576.84	2	2	2	60.0	2633.83	3	2	3	22.85	1652.80
	1	3	5	0.0	67.06	2	3	3	40.0	1564.07	3	3	5	0.0	132.84
Network10	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	2	60.0	3062.83	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	1	80.0	2836.42	2	3	1	80.0	3258.61	3	3	1	80.0	2998.15
Network11	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	0	100.0	3600.00	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	1	80.0	3120.47	2	3	2	60.0	2992.66	3	3	2	60.0	3011.42
Network12	1	1	0	100.0	3600.00	2	1	0	100.0	3600.00	3	1	0	100.0	3600.00
	1	2	0	100.0	3600.00	2	2	0	100.0	3600.00	3	2	0	100.0	3600.00
	1	3	2	60.0	2255.31	2	3	0	100.0	3600.00	3	3	2	60.0	3154.37

圖 5.6 可共乘數學模式測試(求解品質與效率)

5.3.2 啟發式演算法測試

在演算法的部分，我們針對(1)2-opt*、(2)cross exchange、(3)先 2-opt*再 cross exchange、(4)先 cross exchange 再 2-opt*等四種區域搜尋演算法進行分析。表 5.3 為參數設定，鄰域解的個數 η_{nlist} 為 3，cross exchange 中交換最大長度 L^{cross} 為 3，LS 則是進行 10 分鐘或是 1000 次迭代沒進步就停止，上限 *combination* 為 500 次。

doi:10.6844/NCKU202002825

表 5.3 數學演算法參數設定

Local Search				
n_{nlist}	L^{cross}	Iter time(m)	Iter	*combination*
3	3	10	1000	500

而除了上述的參數，我們另外針對以下三種設定進行測試，以不同 Network 進行分類，分別針對 4 種不同的區域搜尋法進行測試，測試的標準則是依據數學模式求得的最佳解，演算法執行超過 30 分鐘即終止，每個 Network 最多有 45 筆結果，為 5 筆測資乘上 9 種不同的載具組合。而若數學模式無法產生可行解的 case，則不採入比對。

1. 不使用 or-opt，以覆蓋路徑規劃數學模式設置充換電點。(圖 5.7)
2. 使用 or-opt，以覆蓋路徑規劃數學模式設置充換電點。
3. 不使用 or-opt，以貪婪演算法設置充換電點。

	2-opt*		cross exchange		2-opt* cross exchange		cross exchange 2-opt*	
network	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)
Network1	6.22	18.86	6.26	16.86	4.22	17.86	2.58	41.58
Network2	1.57	45.48	1.57	38.31	1.57	40.59	1.62	47.83
Network3	1.74	40.85	1.78	37.29	1.77	35.09	1.73	61.53
Network4	448.07	2.69	363.61	7.52	432.12	2.65	161.79	5.49
Network5	396.91	1.48	366.05	1.18	376.39	1.66	196.31	2.83
Network6	442.76	2.87	411.18	2.19	402.33	1.66	214.24	4.22
Network7	252.39	6.93	232.38	6.33	226.77	5.87	123.07	9.97
Network8	173.38	12.3	172.73	12.10	167.42	10.34	65.36	16.76
Network9	15.02	10.69	12.77	13.29	13.53	12.62	7.27	14.68
Network10	320.56	4.57	316.63	6.24	266.57	3.03	89.56	4.82
Network11	143.56	16.35	139.51	18.46	118.01	17.39	51.49	23.75
Network12	170.57	9.93	165.53	1.14	181.39	13.27	71.07	20.67

圖 5.7 數學演算法測試-第一種設定(求解品質與效率)

圖 5.7 可看出在此設定中不同的區域搜尋演算法相對於數學模式皆可在 10

doi:10.6844/NCKU202002825

分鐘內收斂完成，且當節點數n相同時，節線數m越多的 Network，如 Network1~Network3 中的 Network1，可看出求解時間皆較長。其原因推測為LS的鄰域數量隨著節線數增加而大幅上升，且每架載具負責的標靶節線，和動態充換電站需負責的充換電點數量較多，因此要迭代的組合也較多，使得整體的求解時間上升。

而同時節點數n相同時，Gap 值的趨勢則多是由節線數較少的 Network 表現較差。推測原因為標靶節線較少時，單一條路徑交換有限，因此較易陷入區域解。

而不同的區域搜尋法組合，並沒有特別顯著的優劣，而是根據不同的網路設定，有不同的表現。

	2-opt*		cross exchange		2-opt* cross exchange		cross exchange 2-opt*	
network	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)
Network1	179.95	9.75	305.59	12.56	177.19	31.34	167.33	45.26
Network2	7.46	36.99	25.90	38.29	5.26	27.74	10.39	47.41
Network3	10.7	24.95	27.32	26.31	4.62	37.62	5.55	45.34
Network4	600.00	6.11	600.00	8.49	600.00	5.66	600.00	6.63
Network5	600.00	10.42	600.00	5.14	600.00	5.71	600.00	7.18
Network6	600.00	8.55	600.00	3.78	600.00	6.77	600.00	6.09
Network7	600.00	8.18	600.00	8.42	600.00	9.34	600.00	11.80
Network8	598.72	26.70	600.00	18.90	600.00	28.14	580.25	33.98
Network9	412.69	10.68	556.65	10.82	465.92	14.04	387.84	13.47
Network10	600.00	13.27	600.00	14.36	600.00	12.83	600.00	12.01
Network11	600.00	23.17	600.00	22.70	600.00	22.34	594.35	27.06
Network12	600.00	15.91	600.00	20.41	600.00	13.21	600.00	12.11

圖 5.8 數學演算法測試-第二種設定(求解品質與效率)

圖 5.8 可明顯看出使用 or-opt 後，由於 Encoding 機制的改變，使分配到的標靶節線順序不同，成為不同的組合，造成需考慮的組合數變多，因此大多無法在十分鐘內收斂完成，而 Gap 值大多也較第一種設定表現較差。但若是我們加長求解時間的限制，使較多的 Encoding 組合被解碼，亦可能使我們得到較好的解，

doi:10.6844/NCKU202002825

因此我們如何權衡求解品質和求解時間的重要程度，亦是可以深入探討的。

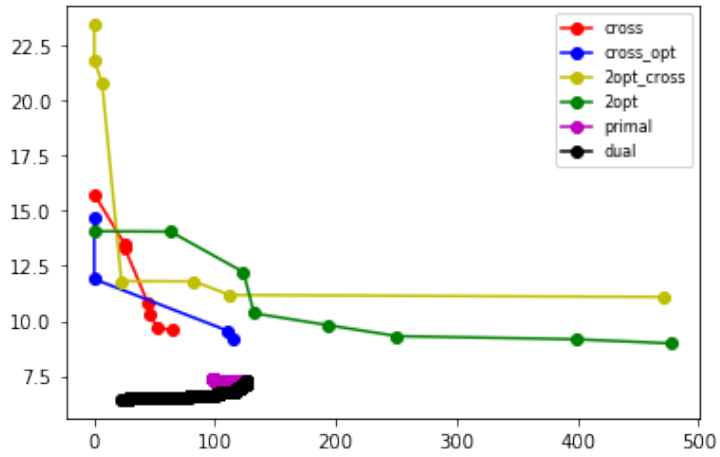


圖 5.9 Network4 第二種設定收斂速度

而雖然圖 5.8 中大多數的組合無法在十分鐘內收斂，但從圖 5.9 中，以 Network4 中的一個測資可知，此種設定亦可能在短時間內得到一個不錯的解，只是在後續的疊代中，大多的組合並無法再得到更好的解，因此才會使得收斂的時間較長。

	2-opt*		cross exchange		2-opt* cross exchange		cross exchange 2-opt*	
network	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)	CPU Time(s)	Gap (%)
Network1	3.30	17.22	6.40	21.19	4.21	18.23	1.03	54.04
Network2	0.22	40.16	0.21	47.73	0.22	40.82	0.17	84.74
Network3	0.16	46.27	0.16	12.08	0.18	52.04	0.14	70.04
Network4	392.29	4.55	341.10	6.13	427.69	2.46	144.86	5.85
Network5	367.01	1.96	330.87	12.51	348.73	1.48	169.11	6.69
Network6	383.59	6.38	329.41	4.18	356.23	2.73	152.45	5.69
Network7	286.07	6.88	278.18	6.82	264.12	6.07	104.42	12.22
Network8	168.16	9.82	153.01	12.06	138.42	11.21	46.88	22.32
Network9	11.53	18.84	11.90	19.63	11.48	16.63	2.38	20.87
Network10	320.75	6.56	285.24	8.53	289.58	2.86	108.58	8.04
Network11	159.23	16.23	144.80	16.82	141.00	19.71	41.79	31.47
Network12	168.41	17.49	167.63	22.33	161.59	17.79	68.7	33.34

圖 5.10 數學演算法測試-第三種設定(求解品質與效率)

doi:10.6844/NCKU202002825

圖 5.10 為不使用 or-opt 且用貪婪演算法設置充換電點的方法，此種方法較為直觀，類似一般搜救隊的作法，將充換電點設置於交通較方便的節點，希望用最少的充換電點數涵蓋所有網路中的節點。

而根據數值分析的結果可知，此設定相較於第二種設定而言，其 Gap 值和求解時間都較佳，表示不論是否使用數學模式或演算法求解充換電點，因 or-opt 會花太多時間嘗試許多不好的組合，導致在時限內其改善求解效益不若其它演算法。而相較使用數學模式的第一種設定，第三種設定的求解時間大多較短但 Gap 值較大，其原因可能是使用演算法所設置的充換電點仍較數學模式差，導致其求解組合較少，較容易收斂至品質較差的解，也應證了一般搜救隊的想法雖然直觀，但相較數學模式的求解，無法全面地考慮，使得其搜救表現較差。

而根據本小節的三種演算法設定測試結果，我們認為使用第一種設定較佳，雖然其求解時間相較第三種設定長，但皆不多於一分鐘。因此，在求解時間差異不大的情況下，第一種設定之求解品質較佳，最適合在短時間內獲得近似最佳解。

5.3.3 無人機選購策略

除了前兩小節的數學模式和演算法在求解效率與品質的比較外，我們在此亦針對無人機的電量做比較，旨在探討若搜救隊有同樣的資金，那該購買哪種類型的無人機叫好?譬如少量但價格較高、電量較充足的無人機，或是大量但較便宜、續航力較差的無人機該如何選購的決策問題。

因此我們以台灣的日月潭為例，如圖 5.11 所示，其節點數為 $n=15$、節線數為 $m=23$，並固定只有一組動態充換電站，且無人機和動態充換電站起點相同，其中當無人機數量越多時，電量越低，以數學模式進行測試。

doi:10.6844/NCKU202002825

圖 5.11 台灣日月潭網路圖

而我們透過數學模式的分析後，可以得到了表 5.4 的結果，可知當無人機數量較多時，即使其每架的電量較低，仍有較佳的期望搜尋時間，推測為每架無人機分配到的標靶節線較少，故可較快的完成搜尋任務，使得需充換電的次數較少，因此有較好的表現。當然這個初步測試可能不夠完整，其管理意涵為只要無人機的續航力尚可，則推薦以可採購較多數量（即使續航力差）的方案，對縮短搜尋時間較有效益。

表 5.4 無人機性價比分析

無人機架數	動態充換電站架數	期望搜尋時間
1	1	33.94
2	1	19.20
3	1	13.94

5.4 小結

本章首先介紹測試資料的參數設定，在網格圖的架構下，增加其多樣性，接著定義不同網路的節點數、節線數，以此測試兩種數學模式的求解效率與品質，

doi:10.6844/NCKU202002825

並分析不同網路圖設定下，數學模式的求解差異。接著再對第四章所開發的演算法進行數值測試，並測試了三種設定，可發現不論何種設定均可在短時間內求得近似最佳解，且對於不同的網路有其一致性的表現，而未加入 or-opt 的第一種和第三種設定，表現皆比僅加入 or-opt 的第二種設定較好，推測原因為加入 or-opt 後使得求解組合增加，造成求解時間增加，難以收斂；而使用數學模式求解充換電點的第一種設定，雖較使用貪婪演算法的第三種設定求解時間稍長，但其表現亦較佳，推測原因為使用貪婪演算法所求得的充換電點雖然直觀但品質較差，導致其求解組合較少，雖較容易收斂，卻同時無法求得部分表現較好的解。由於第一種和第三種設定求解時間差異並不大，因此我們最後仍推薦使用第一種設定，在短時間內求得近似最佳解，最後我們測試了購買無人機的性價比，結果為我們推薦以無人機數量為主要考量，亦即購買較多的無人機(即使其續航力較差)，其縮短搜索時間的效益較大。

doi:10.6844/NCKU202002825

第六章 結論與未來研究建議

6.1 結論

本研究以無人機和動態充換電站進行偏遠地區的搜救，考慮電力因素和不同的移動速度，決策無人機和動態充換電站的移動路徑，以最小化期望搜索時間，搜索所有可能存在靜態標靶之節線。為求貼近實務，本研究假設: (1)搜救隊於傳統二維平面道路上行走，並身背充換電設備，除了可以解決無人機的電力因素，也可以同步進行搜索任務，把握搜救黃金時間。(2)無人機在電力許可下可在任意兩節點間移動，而動態充換電站只能在固有道路上移動，且根據不同的目的採取不同的移動模式，造成不同的移動速率和消耗電量。以下臚列本論文的具體貢獻，並於 6.2 小節建議未來可能的研究方向：

具體貢獻：

1. 本研究以無人機和動態充換電站進行偏遠地區的搜救，為了解決搜救任務時，無人機的電力限制，以兩者的交會來進行電力的補給，避免偏遠地區難以設置充換電站所造成的困難。而此種無人機和動態充換電站的協同搜救任務，可視為 2E-GU-RP，且假定不同的移動速度，以更貼近實際情況。
2. 提出「以時空網路建構」之模式，利用 Time-Space 的方式展開網路，每一個時間空間網路代表一架無人機或動態充換電站，共 $K+L$ 個時間空間網路，以時間來判斷無人機和動態充換電站是否交會充換電或共乘。(第 3.3 節)
3. 提出「可共乘」與「非共乘」數學模式，前者適用性較為廣泛，但變數、限制式較多，以致於求解時間較長。後者則求解時間較短，但面對某些特殊的情況，可能得到較差的解，甚至無解。(第 3.6、3.7 節)
4. 提出「啟發式演算法」，先以「K-mean」分配標靶節線，並使用「覆蓋路徑規劃」求得充換電點和初始解，再以「解碼演算法」將一組必須搜索節線所組

doi:10.6844/NCKU202002825

成的解，考量電力限制與充換電站位置下，解碼為無人機之移動路徑。(第 4.1~4.3 節)

5. 使用多種不同的「區域搜尋法」，將 or-opt 演算法、2-opt*演算法和 cross exchange 演算法，進行先後順序的組合，搭配鄰域列表(Neighbor List)交換標靶節線。(第 4.4 章)
6. 針對兩種數學模式和啟發式演算法進行數值分析，推薦使用「非共乘數學模式」求解規模較小的網路，若遇到特定需共乘之網路，再改由「可共乘數學模式」求解；而中大型的網路，則以「不使用 or-opt，以覆蓋路徑規劃數學模式設置充換電點」的第一種啟發式演算法設定求解，以在短時間內求得近似最佳解。(第 5.3.1~5.3.2 章)
7. 在考慮有限成本的情況下，考慮無人機性價比的選購策略，發現購買多台低電量、較便宜的無人機，會有較短的期望搜尋時間。

6.2 未來研究

針對本研究之搜索路徑規劃問題，尚有許多問題需要進一步探討，如下所示:

1. 可持續精進及發展本研究之數學規劃模式與啟發式演算法：

數學模式

由於本研究之數學模式並無法求解中、大規模之網路(僅能處理約節點數$_n$小或等於 30 之網路)，而小規模之網路亦須耗費不少時間才能得到最佳解，除了考慮是否有其它建模方式外，亦可嘗試開發啟發式演算法，其結果先行縮小變數、限制式範圍或個數，以縮小解集合，增加求解效率與品質。

啟發式演算法

本研究提出解碼演算法將一組編碼，在考量無人機和動態充換電站的電力因素和潛在充換電點下，解碼為一組路徑。其中潛在充換電點的決定，很可能造成影響到後續路徑的優劣，目前決策方式為使用覆蓋式路徑規劃，或可再另尋其它

doi:10.6844/NCKU202002825

方式，譬如先決策充換電點數量，再使用選址問題等方式，決定潛在充換電站的位置及數量。

另外由於解碼演算法時，我們因考慮求解時間，而令無人機或動態充換電站優先尋找最近的標靶節線，而非以標靶節線的順序進行求解。此部分或也可以考慮更改成以標靶節線的順序進行求解，如此會大幅擴大求解集合，可能找到更好的近似最佳解(nearly optimal)，但同時會影響到區域搜索法中需要將單一路徑也進行Or-opt 的交換，將造成標靶節線的組合大幅增加，故求解時間和求解品質的權衡，是可以再考慮的。

2. 考慮無人機和動態充換電站初始位置之搜索路徑規劃研究：

目前對於無人機和動態充換電站的初始位置為隨機設定，然而其初始位置對於期望搜索時間、求解時間與載具的移動路徑都有極大的影響，故設置最佳的初始位置將有助於減少搜救的時間，然而目前尚未有演算法針對上述問題進行求解。

3. 考慮無人機和動態充換電站可隨時隨地充換電之搜索路徑規劃研究：

本研究目前假設無人機和動態充換電站在節點上交會即可立即更換電池，但實務上充換電可能需花費一些時間，此時若無人機可於動態充換電站邊共乘邊充換電，且不限於節點上交會，可於節線上交會充換電或離開，會更符合實際情況。

doi:10.6844/NCKU202002825

參考文獻

呂昀軒. (2019). *以無人機群搜尋單一靜態標靶之最佳協同搜尋路徑規劃問題研究*. 國立成功大學工業與資訊管理學系碩士論文, 台南市. Retrieved from https://hdl.handle.net/11296/58w5mx

馬哈曼. (2019). *執行區域覆蓋任務之可充換電無人機及其載具之最佳聯合路線規劃問題研究*. 國立成功大學工業與資訊管理學系碩士論文, 台南市. Retrieved from https://hdl.handle.net/11296/4ukecq

Berger, J., Lo, N., & Noel, M. (2014). A new multi-target, multi-agent search-and-rescue path planning approach. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *8*(6), 935-944.

Campbell, J. F., Corberán, Á., Plana, I., & Sanchis, J. M. (2018). Drone arc routing problems. *Networks*, *72*(4), 543-559.

Dobbie, J. M. (1968). A survey of search theory. *Operations Research*, *16*(3), 525-537.

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., & Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using K-means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM)*, *9*(2), 374-388.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, *21*, 768-769.

Garone, E., Naldi, R., Casavola, A., & Frazzoli, E. (2010). *Cooperative mission planning for a class of carrier-vehicle systems.* Paper presented at the 49th IEEE Conference on Decision and Control (CDC).

Groves, G., & Van Vuuren, J. (2005). Efficient heuristics for the rural postman problem. *ORiON*, *21*(1), 33-51.

doi:10.6844/NCKU202002825

Ha, Q. M., Deville, Y., Pham, Q. D., & Ha, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 86*, 597-621.

Hà, M. H., Bostel, N., Langevin, A., & Rousseau, L. M. (2014). Solving the close-enough arc routing problem. *Networks, 63*(1), 107-118.

Jotshi, A., & Batta, R. (2008). Search for an immobile entity on a network. *European Journal of Operational Research, 191*(2), 347-359.

Lenstra, J. K., & Kan, A. R. (1976). On general routing problems. *Networks, 6*(3), 273-280.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*(10), 2245-2269.

Luo, Z., Liu, Z., & Shi, J. (2017). A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. *Sensors, 17*(5), 1144.

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* Paper presented at the Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.

Mathew, N., Smith, S. L., & Waslander, S. L. (2015). Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1298-1308.

Orloff, C. (1974). A fundamental problem in vehicle routing. *Networks, 4*(1), 35-64.

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks, 72*(4), 411-458.

doi:10.6844/NCKU202002825

Piacentini, C., Bernardini, S., & Beck, J. C. (2019). Autonomous target search with multiple coordinated UAVs. *Journal of Artificial Intelligence Research, 65*, 519-568.

Potvin, J.-Y., Kervahut, T., Garcia, B.-L., & Rousseau, J.-M. (1996). The vehicle routing problem with time windows part I: tabu search. *INFORMS Journal on Computing, 8*(2), 158-164.

Reiter, S., & Sherman, G. (1965). Discrete optimizing. *Journal of the Society for Industrial and Applied Mathematics, 13*(3), 864-889.

Savuran, H., & Karakaya, M. (2015). Route optimization method for unmanned air vehicle launched from a carrier. *Lecture Notes on Software Engineering, 3*(4), 279.

Stone, L. D. (1976). *Theory of optimal search* (Vol. 118): Elsevier.

Sujit, P., Sousa, J., & Pereira, F. L. (2009). *UAV and AUVs coordination for ocean exploration.* Paper presented at the Oceans 2009-Europe.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science, 31*(2), 170-186.

Tavana, M., Khalili-Damghani, K., Santos-Arteaga, F. J., & Zandi, M.-H. (2017). Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products. *Expert systems with applications, 72*, 93-107.

Tokekar, P., Vander Hook, J., Mulla, D., & Isler, V. (2016). Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *IEEE Transactions on Robotics, 32*(6), 1498-1511.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks, 16*(3), 645-678.

doi:10.6844/NCKU202002825

Yao, P., Xie, Z., & Ren, P. (2017). Optimal UAV route planning for coverage search of stationary target in river. *IEEE Transactions on Control Systems Technology, 27*(2), 822-829.

Yu, K., Budhiraja, A. K., & Tokekar, P. (2018). *Algorithms for routing of unmanned aerial vehicles with mobile recharging stations.* Paper presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA).

doi:10.6844/NCKU202002825