

國立成功大學

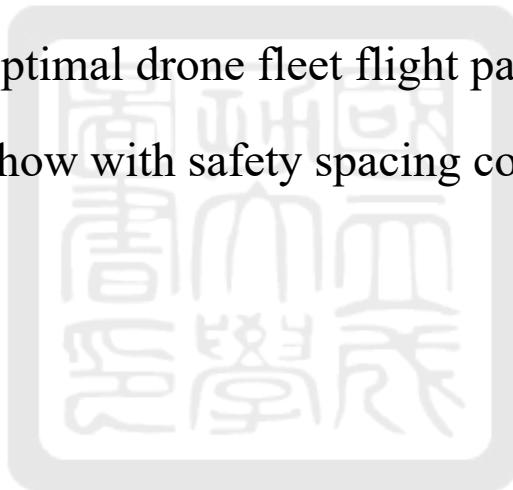
資訊管理學研究所

碩士論文

考慮安全距離之無人機群最佳飛行路線規劃研究：

以群飛展演為例

A study on the optimal drone fleet flight path planning for  
drone light show with safety spacing consideration



研究 生： 張 書 桓

指 導 教 授： 王 逸 琳 教 授

中 華 民 國 一 百 一 十 年 六 月

國立成功大學

碩士論文

考慮安全距離之無人機群最佳飛行路線規劃研究：以  
群飛展演為例

A study on the optimal drone fleet flight path planning for  
drone light show with safety spacing consideration



研究生：張書桓

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

周詩梵  
吳浩軒

指導教授：

王逸琳

單位主管：

翁美京

(單位主管是否簽章授權由各院、系（所、學位學程）自訂)

中華民國 110 年 6 月 11 日

## 摘要

無人機展演的過程由轉場及展演交錯而成，展演時，根據腳本中的圖形設計可知需要無人機亮燈的座標位置，而除了在各個展演畫面變換的過程中避免無人機互相撞擊之外，由於無人機的電池續航力的限制，需要在電池耗盡前返回地面充換電。因此，如何經由適妥的路徑規劃，讓表演順暢進行、降低整個表演的總消耗電量，並有效地減少成本支出，將會是無人機展演所面臨的重要議題。然而傳統上，無人機展演的路徑規劃主要是以程式輔助逐一檢驗各架無人機之飛行路徑，檢驗其與它架無人機飛行路徑是否全程保持安全距離，若有需要只能以人工手動方式調整，也因此容易顧此失彼，難以兼顧全局，其規劃與調整過程十分費時耗力，又無法保證求解品質。

本研究將無人機的電池續航力限制、展演座標上可進行移動的路徑、以及腳本中預計亮燈之時刻與座標位置等設定視為已知，以層空網路為架構，將燈光秀過程分為「展演」與「轉場」等兩類互相交錯的時間區段，建立整數規劃模式以規劃無人機在展演時段的位置及轉場時段的飛行路線，期望在考量無人機行進間安全距離的前提下，用最少的能量耗費以及無人機架數來完成演出。

相較於現行無人機群飛展演所採用的人工編排路徑規劃方式，本研究提出的整數規劃模式可根據展演腳本計算出理論上耗能最少且具安全距離保證的無人機群路線規劃方式，然而此精確解法求解十分耗時，因此我們針對現實較大規模的展演再設計滾動式求解機制與貪婪演算法等兩種啟發式演算法機制以加速求解。由多組不同情境的數值模擬測試結果中，我們觀察到貪婪演算法求解極快但可能啟用較多架無人機，而滾動式求解機制則以整數規劃模式為基礎，兼顧求解效率與品質，是具學理根據的系統性創新作法，特別推薦給相關展演業者及學術研究，用於規劃考量安全距離的無人機展演群飛路線。

**關鍵詞：**無人機、路徑規劃、整數規劃、電池充換電、貪婪啟發式演算法

# **A study on the optimal drone fleet flight path planning for drone light show with safety spacing consideration**

Shu-Huan Chang

I-Lin Wang

Institute of Information Management

## **Summary**

This research investigates a path planning problem for a fleet of drones making a light show. In particular, assuming that the figures in the light show are known beforehand, we need to deploy at least a drone in specific locations in the sky at a particular time. To the best of our knowledge, this problem is solved manually in practice. We propose an integer program (IP) based on a time-space network to calculate the movements for each drone with the minimum costs, including the fixed equipment costs and the total flying distance. We divide the entire show period into two types: the show period and the transition period, where the former displays the figures, and the latter is used for drones to transit to new positions in the sky. However, our IP model is too time-consuming. We, consequently, introduce a greedy heuristics algorithm that can calculate a good solution in a much shorter time. In addition, we propose a rolling horizon mechanism, which divides the entire planning horizon into smaller ones and then consecutively solves each IP model of a shorter planning horizon. The proposed mechanism is efficient and effective in our computational experiments.

**Keywords:** Drone, Path Planning, Integer Programming, Battery Swapping, Greedy Heuristics

## INTRODUCTION

Traditionally, the light shows in the sky at night are conducted by fireworks. Recently, drones, also known as Unmanned Aerial Vehicles (UAVs), have been used to display figures in the night sky, to replace the fireworks for light shows. UAVs have more advantages than fireworks because they are reusable, environmentally friendly, and a symbol of technological advancement. The major challenge for planning a good drone light show lies in careful plot planning. To display the designed figures in the sky by deploying UAVs as light dots, one requires meticulous path planning for all the UAVs. To the best of our knowledge, the path planning problem for a fleet of UAVs to make a light show is solved manually in practice. Thus, it takes lots of time and labor, but the results are still not guaranteed to be optimum.

This problem is similar to and, in fact, more challenging than the Location Routing Problem (LRP). We need to confirm that all the lighting locations would be passed through by a drone at a particular time and consider multiple UAVs and if their mutual distances are sufficiently safe simultaneously. Furthermore, to minimize the total costs, we also consider dispatching fewer drones to plan their flight paths as short as possible.

## MATERIALS AND METHODS

Given the script, which includes the figures to display in the light show, we seek an optimal path planning for each drone to complete the show with minimum total costs. To calculate an optimal solution in these two scenarios, we formulate the problem as an integer program (IP) that considers the necessity of battery swapping over a time-space network. If a UAV needs to swap its battery, it would return to land on the base and then launch to resume a planned flight path. Our IP formulations also consider safety spacing between any two UAVs during the entire planning horizon, which might arguably be the first analytical path planning model in the world that calculates optimal paths with such a setting, to the best of our knowledge. Our proposed complete IP models thus consume a tremendous amount of computational time, as expected. Therefore, we propose a rolling horizon mechanism, which divides the original planning horizon into shorter ones so that its corresponding IP can be solved quickly. Our rolling-horizon mechanism can solve much larger cases with good solution qualities by iteratively solving smaller IPs corresponding to a smaller version of the original problem with overlapped partial periods. For instance, suppose

the original script covers 10 frames, and more than 200 nodes are considered in each frame, then we will have 11 transition periods. Using our 3-fix-1 rolling-horizon mechanism, we may solve an IP for the 3 consecutive transition periods (e.g., period 1, 2, and 3), fix the optimal decisions for the first period, relax the calculated decisions for the second and third periods, and then solve the next IP for the next 3 consecutive periods (i.e., period 2, 3, and 4). By doing so, we give the new IP some freedom to calculate better decisions for the new periods based on the result of the previous IP. In our experiments, this rolling-horizon mechanism does provide the best results in our computational experiments.

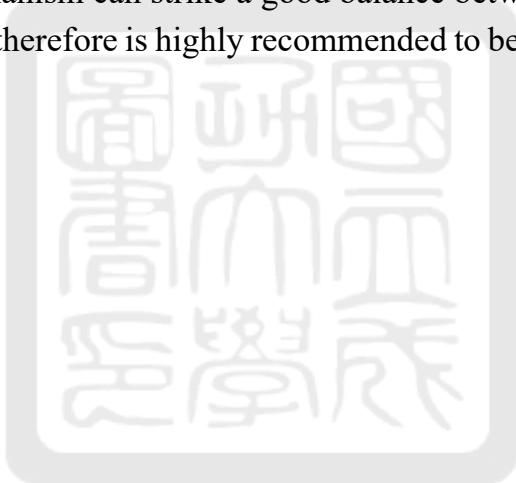
Moreover, to calculate a good solution more efficiently, we also introduce a greedy heuristic algorithm based on Dijkstra's algorithm. The mechanism would route for one UAV at a time to pass through light positions with the shortest total distances, and then repeat this until all the light positions are passed by some UAVs. During this greedy path searching procedure, we also consider the safety spacing. In the worst case, one may launch a new UAV to fulfill a light position requirement, assuming the new route would always satisfy the safety spacing constraint.

## RESULTS AND DISCUSSION

We have randomly generated 12 different network families with varying numbers of nodes and frames to increase testing data diversity. Then we used them to test our proposed IP formulation (M), the rolling horizon mechanism (R), and greedy heuristic algorithm (A). The results indicate that M can produce the best result but with a longer computational time, and M and R can produce good feasible ones with much short computational time. As for the more practical cases of large scales, we also generated 8 network families for testing. We compared the performance of our proposed greedy heuristic algorithm and two rolling horizon mechanisms. The latter is composed of two different settings: (1) the 2-fix-1 setting (R21) that solves a 2-period IP and fix its first-period result in each rolling horizon, and (2) the 1-fix-1 setting (R11) that solves a single period every time without considering overlapped horizon. Based on our experiments, the R21 mechanism produces the best results. In summary, M can calculate the best solution for users with powerful hardware and sufficient quota in computational time, A can be useful in an emergency due to its excellent efficiency, and R can strike a good balance between the efficiency and the quality of solutions.

## CONCLUSIONS

This research seeks optimal planned paths for UAVs to conduct a drone light show smoothly. We hope to design the UAV paths so that the safety spacing between each UAV at any time is enforced during the flight for displaying the designed figures in the sky. At the same time, the total costs in equipment and energy consumption are minimized. To the best of our knowledge, this research might arguably be the first that proposes analytical models and algorithms to resolve this problem systematically. In particular, we propose IP models on a multi-level time-space network. The IP models can calculate optimal paths, yet they are quite time-consuming. Thus to deal with real-world cases, we have also designed a greedy heuristic algorithm and an IP-based heuristic that iteratively solves smaller IPs associated with consecutive partial planning horizons in a rolling fashion. The greedy heuristic algorithm is suitable for urgent needs since it can quickly calculate a good feasible solution. On the other hand, the rolling-horizon mechanism can strike a good balance between efficiency and the quality of solutions, and therefore is highly recommended to be used for practitioners.



## 誌謝

兩年的時光很快的就過去，手邊抖邊拿著資料到系館找指導教授好像還是昨天的事情，而現在就即將畢業離開了。在成大資管所的這些日子是讓我改變、成長最多的階段，我要特別特別感謝王逸琳老師，老師一開始就知道我的基礎不穩固，在一次次的比賽、專案、研究中，卻還是非常有耐心且細心的指導、與我討論每一個問題，並補充許多的教學資源讓我加強自己；除了教學以外，老師處理事情的態度與思考方式，更是讓我學到許多，並期許自己未來能繼續努力與老師看齊。

感謝實驗室的學長姐，冠緯、嘉豪、雪湄、昀軒、彥瑋、宗瀚及晏慈，在我剛加入 lab 到現在，一直給了我很多幫助，不論是日常的吃飯、打球、玩桌遊，或是 RAS 時在旁邊打雜生測資、練習老師給的計程作業、選課，到之後的研究、研討會及找工作，學長姐們都不吝嗇給予我們建議，讓我們順利通過各種難關。感謝學弟妹，威銓、郁恩及叢仔，除了與我吃飯聊天外，你們在程度好的同時還都很認真，讓我著實佩服也警惕自己，也抱歉沒有幫助你們什麼，祝你們未來一年研究順利。還要特別感謝同屆的承中，你一直是這兩年幫助我最多的夥伴之一，雖然你一直說自己程度不好等等謙虛的話，但是你總是在關鍵時刻給予很重要的建議及陪伴，也很抱歉這兩年一直接受很多你默默的幫助，卻沒有給你什麼特別的回饋，期望你未來一切順遂。

感謝從大學到研究所，一路上都一起努力的夥伴們，亭鈞、佳晨、羿霈、雅婷、昀儒、書廷、子權……，兩年的研究生生活中，課業及生活上的苦悶、歡樂、緊張都有你們一起互相討論及鼓勵，大家都辛苦了。感謝成大流舞社的嘻哈學弟妹與陶家幫的每一位一起跳舞、聊天、分享迷因的老師及朋友們，讓我在研究之餘，還多了許多歡樂的回憶。感謝碩班打球趣團一起舒壓及同屆的各位幫過我忙的每位同學們。

感謝我的家人，給予我關懷及打氣甚至是課業及找工作的幫助，讓我在良好的環境中完成學業，度過這兩年研究生時光。最後也要感謝自己，即使在碩一的時候天天跟不上進度一度想休學，卻還是堅持過來了，雖然不是真的變成多厲害的人，但仍是慢慢在進步、成長，希望未來繼續加油，畢業後一切順利、繼續努力。



# 目錄

摘要 .....	I
目錄 .....	VIII
表目錄 .....	X
圖目錄 .....	XI
第一章 緒論 .....	1
1.1 研究背景與動機 .....	1
1.2 研究問題設定與目的 .....	4
1.3 研究範圍與限制 .....	6
1.4 論文架構 .....	7
第二章 文獻探討 .....	8
2.1 無人機及群飛展演相關文獻 .....	8
2.2 電力消耗及充換電相關文獻 .....	10
2.3 無人機飛行安全距離相關文獻 .....	12
2.4 無人機路線規劃相關文獻 .....	13
2.5 小結 .....	16
第三章 無人機群飛展演之最佳飛行路線整數規劃模式 .....	17
3.1 問題描述 .....	17
3.2 問題假設 .....	19
3.3 固定式位置換電之整數規劃模式 .....	20
3.3.1. 網路結構 .....	20
3.3.2. 飛行安全距離 .....	23
3.3.3. 參數與變數定義 .....	24
3.3.4. 最小化展演總成本 .....	27

3.3.5. 不考慮電量消耗之整數規劃模式.....	28
3.3.6. 考慮電量消耗之整數規劃模式.....	32
3.4 小結.....	34
<b>第四章 無人機群飛展演之最佳飛行路線規劃演算法.....</b>	<b>35</b>
4.1 滾動式求解機制 .....	35
4.2 貪婪演算法 .....	41
4.2.1 演算法步驟及邏輯 .....	41
4.2.2 路線選擇懲罰值設定 .....	44
4.3 簡例說明與求解方法比較.....	46
4.4 小結 .....	50
<b>第五章 數值測試與分析 .....</b>	<b>51</b>
5.1 測試資料參數設定 .....	51
5.2 測試網路圖設定 .....	52
5.3 無人機群飛展演之最佳飛行路線規劃之測試.....	53
5.4 滾動式求解機制與貪婪演算法比較 .....	55
5.5 小結 .....	57
<b>第六章 結論與未來研究建議.....</b>	<b>59</b>
6.1 結論 .....	59
6.2 未來研究 .....	61
<b>參考文獻 .....</b>	<b>65</b>
中文 .....	65
英文 .....	67

## 表目錄

表 1 UAV 應用範疇 .....	9
表 2 無人機展演相關文獻比較表 .....	10
表 3 充換電相關文獻比較表 .....	12
表 4 無人機路線規劃相關文獻比較表 .....	15
表 5 三個畫面期數說明 .....	25
表 6 集合與索引 .....	26
表 7 參數說明 .....	26
表 8 變數說明 .....	27
表 9 滾動式求解機制之預處理過程 .....	37
表 10 滾動式求解機制流程 .....	38
表 11 估算變數與限制式個數之參數設定值 .....	38
表 12 變數總個數比較 .....	39
表 13 限制式總個數比較 .....	39
表 14 貪婪演算法設計邏輯 .....	44
表 15 路線選擇懲罰值設定 .....	46
表 16 三種求解方法比較表 .....	49
表 17 範例求解結果 .....	50
表 18 測試資料參數設定 .....	52
表 19 網路圖資訊 .....	53
表 20 求解方法數值測試比較表 .....	54
表 21 大型測試資料參數設定 .....	56
表 22 滾動式與貪婪演算法求解結果比較表 .....	56

## 圖目錄

圖 1 飛行演練示意圖 .....	4
圖 2 UAV 移動方式 .....	5
圖 3 移動方式示意圖 .....	6
圖 4 無人機移動方式限制_起訖位置組合相反之對飛路線 .....	18
圖 5 無人機移動方向限制_兩路徑間距小於安全距離 .....	18
圖 6 網格型態的網路圖形 .....	20
圖 7 聯集後重新定義之網路圖形 .....	21
圖 8 單一（某第 $k$ ）層之層空網路示意圖 .....	22
圖 9 碰撞情況示意圖 .....	22
圖 10 飛行距離計算範例 .....	23
圖 11 座標位置—節點編號對照說明 .....	25
圖 12 滾動式求解機制設定 .....	36
圖 13 贊婪演算法求解例圖 .....	41
圖 14 贊婪演算法執行流程 .....	42
圖 15 避免選擇路線情況 .....	45
圖 16 簡例腳本畫面 .....	47
圖 17 預處理後之簡例腳本畫面 .....	47
圖 18 以整數規劃模式求解簡例結果 .....	48
圖 19 以滾動式求解機制求解簡例結果 .....	48
圖 20 以贊婪演算法求解簡例結果 .....	48
圖 21 範例圖像設計 .....	49
圖 22 網路生成過程 .....	51
圖 23 實務表演大型案例 .....	55

# 第一章 緒論

英特爾公司 (Intel) 在西元 2018 年的韓國平昌冬季奧運會的開幕儀式上，在夜空中，以上千台無人飛行載具 (Unmanned Aerial Vehicle, UAV) 進行燈光秀表演，在點亮天際的同時，也使無人機應用於群飛燈光秀的展演方式被全世界所注目。然而，多台無人機進行聯合展演的過程中，如何經由適妥的路徑規劃，讓整個表演順暢進行，不致中途撞機，以完整呈現各種腳本的圖形設計，同時降低整個表演的總消耗電量，並有效地減少成本支出，兼顧美感及環保，將會是無人機展演所面臨的重要議題。

本章節將分為四節，1.1 節說明研究背景與動機，介紹 UAV 群飛燈光秀的應用案例，以及其目前在巡演時所面臨的各項挑戰；1.2 節再從其中引導出本研究針對所欲求解之問題提出的環境設定與考量，與一般常見的多種路線規劃方式比較說明；接著於 1.3 節釐清本研究主題的範圍與限制；最終於 1.4 節解釋本論文的架構。

## 1.1 研究背景與動機

在許多重要節日慶典中，最受眾人矚目的往往是聲光斑斕、璀璨動人的燈光表演，無論是傳統的煙火表演、結合建築結構與光影美感的光雕秀，到近年來最熱門的無人機群飛燈光展演，每每讓觀眾歎為觀止、意猶未盡。而以無人機進行展演這樣的節目在國外已行之有年，在台灣的首度展演則由 Intel 團隊於 2019 年 2 月在屏東舉辦的台灣燈會中拔得頭籌，該展演動超過 300 架無人機「摘星號」(Shooting star)，在夜空中排出「屏東」、「TAIWAN」及黑鮪魚等複雜圖像，迴響熱烈，讓燈會的參觀人數突破千萬人次，更以感官的震撼顛

覆觀展民眾對於 UAV 應用的既定印象。而台灣的本土團隊亦於同一年內，由南台灣無人機技術團隊 Taiwan Drone 100 主導，在台南漁光島藝術節的閉幕儀式、臺南市國際龍舟邀請賽以及 U12 世界少棒錦標賽等活動中以數十至上百架 UAV 進行群飛展演。在一次次的演出中，民眾可看到無人機在空中秀出各式各樣與主題相關的圖像符號，導致此類科技應用逐漸受到重視，相關演出也因此如雨後春筍般蓬勃發展。無人機展演能夠以重複使用的無人機取代部分成本花費高的煙火表演，因此能降低煙火表演所造成空氣及環境污染，在提昇表演品質的同時也兼顧節約及環保的美意。

無人機展演的過程由「轉場」及「展演」等兩類互相交錯的時間區段組合而成。在「展演」時段中，根據腳本中的圖形設計可得知需要無人機亮燈的座標位置，然而，由於無人機有電池續航力以及路線避撞的限制，在各個展演畫面間變換的「轉場」時段中，需要妥善的路線規劃，避免無人機中途因電力不足掉落或互相撞擊導致損壞。蔡成鴻（2020）指出於無人機展演畫面中光點所構成的集合、各光點的移動軌跡、以及定點漸層光影變換等都是路徑規劃上的應用與展現。因此，如何經由適當的 UAV 路徑規劃方法，讓表演中各個畫面完整呈現、避免無人機碰撞、降低整個表演的總消耗電量，並有效地減少成本支出，是無人機展演所面臨的一大挑戰。

飛行路線規劃是飛行應用時首先會碰到的問題，陳冠岐（2016）指出在 UAV 自動化研發中最具挑戰性問題之一是路徑規劃，因 UAV 之感應範圍、耐力有限，需進行優化。據我們訪談燈光秀主辦單位的經驗得知，實務上目前規劃燈光秀的無人機飛行路線規動作法多採用程式嘗試連結前後展演畫面光點位置，連結完後再逐一用人工方式計算每兩條飛行路徑之間距是否足夠遠（以避撞）；若太近，則必須重新調整飛行路徑。然而這種手動的檢驗調整方式經常會導致牽一髮動全身的漣漪效應，導致先前好不容易排好的飛行路徑又要重新調整，且又不能保證如此調整方式是否會導致後續更差的結果；若實在太難調整的話，

可能必須多飛新的 UAV 去補位。這類人工調整路線的規劃方法在無人機數量較少的案例中或許還能接受，但顯而易見，此種手動的排路線方式勢必無法有效地處理更複雜或多光點的展演圖案需求。正因為目前實務上土法煉鋼的路線規劃方式，尚未有系統性的科學作法或學理分析相關文獻，本研究希望透過作業研究中的整數規劃法，為本應用所有畫面及轉場切換過程的 UAV 最佳飛行路線提供學理基礎與系統性的科學作法，以達到最小耗能的目的，進而使群飛展演的規劃過程能更有效率。

當前相關的路徑規劃研究方法眾多，但皆有其適用條件與限制，Huang et al. (2018) 指出「人工勢場法」(Artificial Potential Field Method , APF) 無法在密集障礙物的情況下產生路徑；「蟻群演算法」(Ant Colony Optimization , ACO) 則取決於參數的配置且需要較高的計算資源運算、「A\* Search Algorithms」則無法適用在 UAV 有移動限制的情況，而 Oz, Topcuoglu & Ermis (2014) 提出運用「基因演算法」(Genetic Algorithms , GA) 來解決地理高低差異情境下的路徑規問題。然而，上述文獻皆僅考量單次或單層的路徑移動規劃，不適合直接用在需多次展演以及轉場階段的 UAV 燈光秀上。舉例來說，在燈光秀期間皆需考量每架 UAV 該如何從不同起點移動到另一個節點上（每架 UAV 在空中各畫面的起點與終點都不同），且前一階段的移動方式將直接影響下一階段的起始位置(可參考圖 1)，如圖 1 中 3 號無人機在轉場時刻  $t'=1$  時，可以選擇移動至任一點，若其選擇由原點 0 出發移動至點 5，在轉場時刻  $t'=3$  時，便必須由點 5 出發。此外，不同的節點在轉場期間所可到達的節點亦不盡相同。因此，若需呈現的畫面越多、畫面中待飛抵之節點將越多，導致整體考量規劃的移動路線將越複雜。故可預期的是，若需排出圖案的光點數越多、畫面越多，則需規劃的 UAV 架數與架次將越多，因而使得整個從頭到尾的可能飛行路線將形成一個十分錯綜複雜的網路結構。

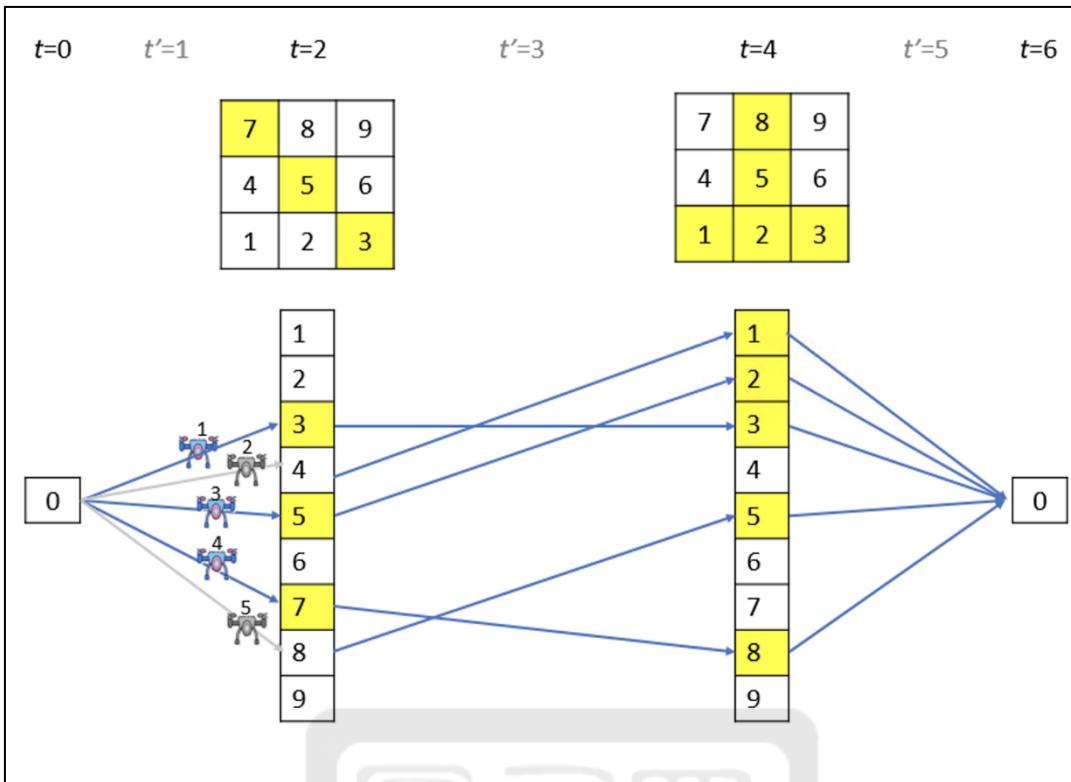


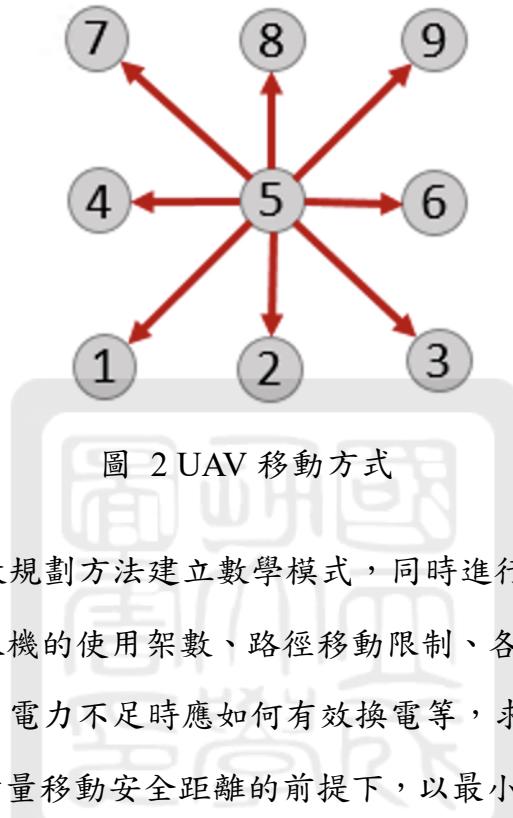
圖 1 飛行演練示意圖

## 1.2 研究問題設定與目的

傳統 UAV 展演路線規劃方法，往往是考慮單一畫面中路線的移動安全性與電力消耗，因為缺少整體性的規劃，不僅費時耗力還無法達成最佳的規劃，容易造成不必要的電力及成本浪費，若因此造成使用無人機展演失去了兼顧環保的意義，將非常可惜。然而，現實中因為電力的限制、諸多空中飛行時的不可抗力因素以及人力規劃疏失，使 UAV 在展演過程中容易發生碰撞意外，若因而損失 UAV，不但虧損更多成本且增加展演整體的安全疑慮。

為了考量移動間安全距離，本研究將問題中的 UAV 在一轉場時段內可移動的距離上限設定為  $\sqrt{2}$  單位長度，亦即 UAV 移動的方式為上下直行、左右橫行以及斜行至鄰近的點。以圖 2 為例說明，若一架 UAV 位於點 5，可以經由路線 (5,2)、(5,8) 進行上下移動，(5,4)、(5,6) 進行左右移動，以及 (5,1)、

(5,3)、(5,7)、(5,9) 進行斜行移動。而本研究為了詳細計算無人機移動時的路線是否與其它無人機低於行徑間的安全距離，將在第三章中詳細介紹計算方式。



本研究將使用整數規劃方法建立數學模式，同時進行多架 UAV 飛行路徑的編排，其中考慮無人機的使用架數、路徑移動限制、各時刻的層數、網路空間位置節點數間關係、電力不足時應如何有效換電等，求解出多架 UAV 飛行最佳的移動路徑，在考量移動安全距離的前提下，以最小化成本的方式來求得消耗的電力與規劃 UAV 使用架數，而討論在眾多圖像畫面須呈現時項目包含：

- 如何配置每架 UAV 的路線移動配置使其能在指定位置變換燈號。
- 如何使每架 UAV 移動路線互相不低於安全距離確保移動安全。
- 考量電池消耗的情況下，待換電的 UAV 如何安排其移動路徑。
- 規劃所需的最少 UAV 架數，以降低租機成本。

### 1.3 研究範圍與限制

實驗依據設定圖像位置進行計算，評估 UAV 在路徑移動方式與不同時間區段上選擇的差異，在排除不可控制的外在變數後，專注於路線規劃並進行假設如下：

- 假設不考慮外在機構設計、承載重量造成影響。
- 假設不考慮天氣風力影響。
- 假設 UAV 換電所需時間不列入考量 (需要時能立即補上)。
- 假設有足夠的 UAV 可供使用，且每架的移動能力速率相同。
- 假設 UAV 移動簡化成一轉場時段可移動至鄰近(橫行、直行、斜行皆可)的位置，可參考圖 3。

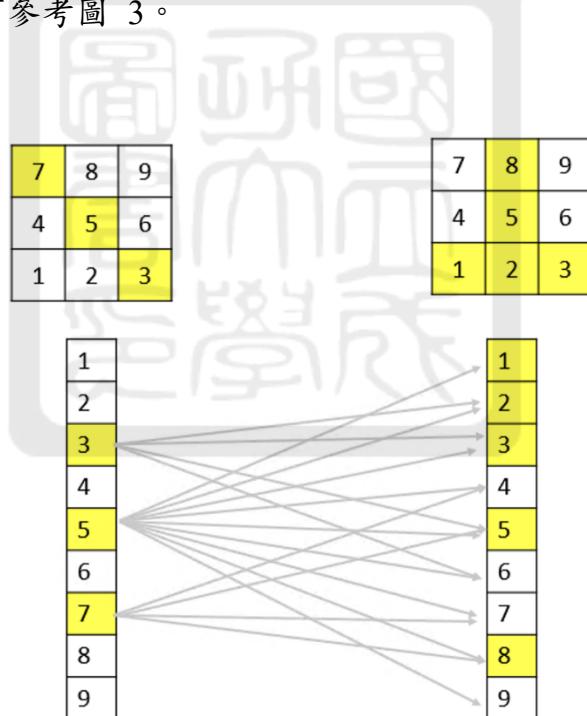


圖 3 移動方式示意圖

## 1.4 論文架構

本論文流程共分為六章，下述介紹各章內容。

### 第一章 緒論

以 UAV 在燈光表演的案例為開頭，說明其現行方式求解問題與限制，逐步從研究背景與動機、研究問題設定與目的、範圍及限制與研究流程說明來介紹。

### 第二章 文獻探討

解釋 UAV 群飛展演中面臨到的各種問題，並回顧近年來考慮 UAV 電力消耗需充換電模式以及考量行徑間安全距離的研究，再將 UAV 路線規劃的研究方法進行整理，最後說明本研究中欲採用的求解方法。

### 第三章 UAV 移動路徑之整數規劃模式

針對問題進行建構整數規劃模式，並說明提出的模式參數、限制式與最佳化目標，最後以範例進行演示。

### 第四章 UAV 移動路徑之求解演算法設計

針對中大型問題求解時，提出啟發式演算法說明，解決數學模式無法在短時間內求解的缺點，並以範例進行演示，也指出在優化的過程中面臨的挑戰與改善方式。

### 第五章 整數規劃模式與貪婪演算法之數值測試與分析

運用小例子來說明使用 Python 結合 Gurobi 進行規劃求解與貪婪演算法作法比較，其次嘗試用演算法與其應用上的變形來進行模擬測試並從中分析差異。

### 第六章 結論與未來研究議題建議

總結本研究論文成果，如何作為飛行展演時的規劃應用，並提出的未來研究方建議。

## 第二章 文獻探討

無人載具路線規劃的相關研究繁多，每一種問題所考慮的議題以及希望達到的目標也都有所不同，其中包含 UAV 行動時的電力消耗、規劃最短的路線、保障行進間的飛行安全、以最短的時間完成或是使用最少架無人機完成任務等等諸多議題。本章分五小節，首先會討論有關無人載具及其應用在群飛展演中的相關研究議題；並回顧過去 UAV 路徑規劃有考量電池消耗及充換電模式的相關研究，說明其問題設定；接著整理目前文獻中解決 UAV 飛行安全距離的方法，針對無人機的路線規劃方式進行探討，包含各研究中在不同情況限制之下，用什麼方法如何解決此類問題等；最後以小結統整本研究與過去研究的異同與目的。

### 2.1 無人機及群飛展演相關文獻

無人載具泛指不需人為駕駛在機器上，即可透過遙控技術或甚至自主判斷進行移動的載具，其中又分做 UAV、無人地面載具 (Unmanned Ground Vehicle, UGV) 以及無人海上載具 (Unmanned Surface Vehicle, USV) 等等不同應用領域。而此項技術起源於第一次世界大戰後，軍用靶機訓練與投擲炸藥，伴隨著科技進步可安裝許多微型晶片模組，例如全球定位系統 (Global Positioning System, GPS)、慣性測量單元 (Inertial Measurement Unit, IMU) 裝置、相機數位、微型計算晶圓、藍芽與紅外線通訊、物聯網 (Internet of Things, IoT)、人工智慧等，機構上也不斷研發改良突破承載限制 (蔡成鴻, 2020)。如表 1 整理，可見隨著無人載具技術的進步，除了傳統軍事用途外也廣泛用於農業、工業、商業領域。

表 1 UAV 應用範疇

農業植保	藥物施灑、輔助授粉
物流空投	投擲信件、包裹配送、血液配送
氣象科研	大氣取樣、人工降雨、行星探索
災難應變	偏遠地區物資補給、災情監測評估、人員搜索
警備巡防	海上巡邏、邊境偵測、跨境走私監控
設施檢查	高架橋檢查、石油管道與水力廠巡查、建築物勘查
通訊服務	平流層氣球網路、偏遠地區的網路覆蓋率
繪測辨識	地理繪圖、航太攝影、生態環境監測
運動新聞	賽事實況追蹤、新聞拍攝、災難戰區報導
娛樂直播	無人機群演燈光秀、微型投影、電影取景拍攝

其中，將 UAV 應用於展演當中，研究需考慮的面向也相當多元。在設計一場 UAV 燈光展演的過程中，從圖像設計、圖像轉換為網路圖形中的點座標、無人機群飛路線規劃，到考量行進間安全距離以及電力限制等等。接著就整理有關無人機應用於展演相關領域的研究。

Chen, Wang, Chen & Han (2018) 為了顯示每架無人機需亮燈的位置，提出一套將圖像轉換為點座標形式的方法，首先是將預計顯示在天空上的圖像柵格化。每一個圖像可以通過 MATLAB Image Viewer Toolbox 離散化，並由像素矩陣表示。再將像素矩陣中的灰階色彩，經由數學式轉換為需要進行亮燈的離散化點座標。

Sun, Qi, Wu & Huang (2020) 基於無人機的燈光展演，提出一種用於密集無人機形成的自主路徑規劃方法。基於改良的人工勢場演算法，提出了無人機編隊的路徑規劃方法。可以有效地限制了飛行路線，提高了 UAV 路徑互相影響的缺點，有效解決了無人機安全距離的疑慮。也將其研究中提出的路徑規劃方法應用於實際飛行中。

蔡成鴻 (2020) 以考量電力消耗為重點，提出整數規劃數學模式，其中將展演過程以實務為依據分做轉場及展演時段，並以總消耗成本為目標，求解最

佳的路線規劃應用。並且，同時提出隨機貪婪演算法，以解決數學模式無法在短時間內求解中大型問題的缺點。

將相關文獻其所研究的主題及對象整理成表 2，蔡成鴻(2020)的研究與本研究最為相似，同時對多架無人載具進行考慮充換電力的路線規劃，而本研究與之不同之處在於，該研究假設 UAV 在飛行時存在自我避撞的機制，故不考慮路線衝突的危險，而本研究則將 UAV 飛行間的安全距離計算納入路線規劃的考量因素，以確定 UAV 行進安全，解決因意外事故而展演失誤的可能。

表 2 無人機展演相關文獻比較表

作者	研究主題	規劃種類		研究考量範圍		
		載體	多架	安全距離	路線規劃	充換電考量
Chen et al.(2018)	圖像設計 路線規劃	UAV		V	V	
Sun et al.(2020)	路線規劃	UAV		V	V	
蔡成鴻(2020)	路線規劃	UAV	V		V	V
本研究	路線規劃	UAV	V	V	V	V

## 2.2 電力消耗及充換電相關文獻

在無人機路徑規劃研究上，其中一項需要深入探討的是無人機作業時的電力限制，過去為了達到無人機輕巧的特性，也造成其電量續航力較差，無法負荷規模、範圍較大的任務，如搜救、展演以及農業檢測方面；然而，即使近年來，UAV 技術越來越進步，大部分展演實務上所用之無人機已經可以負擔展演的時間，本研究仍將電力消耗納入考慮，以更全面地解決某些續航力不足的情況。因此，在路徑規劃當中加入考慮可充換電的情況，是擴大無人機可應用範圍中重要的一環，以下將近期研究進行回顧，並在本小節最後整理成表 3。

馬哈曼（2019）在考慮電力限制的情況下，使用無人機及地面載具協同合作求解覆蓋路經規劃問題（Coverage Path Problem，CPP），其中，使用店面載具輔助 UAV 作業，提供共乘、電力充換的功能，期望能使用 UAV 在最短的時間內搜尋範圍內所有網格。

蔡成鴻（2020）提出考量定點充換電站的無人機群飛展演路徑規劃，研究設定充換電站固定位於原點，無人機於展演當中若有充換電需求，皆須回到原點，並且假設忽略充換電時間，以方便後續建立整數規劃模式求解方法。而實務上，群飛展演將多個無人機充換電、異常檢修站設置於原點，因此，本研究將主要蔡成鴻學者的設定。

Yu, Budhiraja, Buebel & Tokekar (2019) 分別研究此三種不同的充電模式下，透過 UAV 結合 UGV 依自訂的數學模式來計算總運行路徑最小的問題，結果得知在特定大小的條件下，該演算法運行時間是可求解出不錯品質的解法。

本文引述其對於充換電模式的 3 種分類說明如下：

- (1) 多個固定式充電站，如現行的 UAV 展演及採用此方式進行群演。
- (2) 單一移動式充電站，如 Luo et al. (2017) 使用 UGV 進行主路線規劃與充電，UAV 則作為延伸使用。同樣在 Ropero et al. (2019) 的研究上亦採用單一移動式方式進行研究，探討在崎嶇不平的道路上，使用 UAV 進行延伸。
- (3) 多個移動式充電站，Yilmaz et al. (2019) 則使用 UAV 配合船艇來達成機體充電與不同目標地點的巡訪的目的。

表 3 充換電相關文獻比較表

作者	規劃種類		充電方式		
	載體	多架	單個 移動式	多個 移動式	多個 固定式
Luo et al. (2017)	UGV		V		
Ropero et al. (2019)	UAV+UGV		V		
Yilmaz et al. (2019)	船+UAV	V		V	
馬哈曼(2019)	UAV+USV	V	V		
蔡成鴻(2020)	UAV	V			V
本研究	UAV	V			V

## 2.3 無人機飛行安全距離相關文獻

在應用無人載具在實務上最為關鍵的問題之一，便是如何能讓無人機群在飛行時能夠互相保持安全距離，在沒有不可抗力的天氣因素之下，可以免去規劃不善導致的衝撞，因而降低無人機的損壞或遺失。

Prats et al.(2012)的研究中，統整了無人飛行載具相關的現有法規、推薦避撞方法和標準。其將防撞的預期要求區分為不同的重點：

- (1) 根據通行權規則檢測並避免與其它飛行交通的空中碰撞。
- (2) 檢測並避開其它飛行物體（如鳥類）。
- (3) 檢測並避開地面車輛（在地面上機動時）。
- (4) 檢測並避開地形和其它障礙物（如建築物或電力線）。
- (5) 避免危險的天氣狀況。

然而，由於不可抗力的因素，如天氣、非預期的障礙物等，需要機體結構設計上加入傳感器等機制，本研究將專注於該研究中所歸納的第一個重點，也就是避免與其它 UAV 發生衝突。

Wang, Yadav & Balakrishnan(2007)將無人機的協作飛行任務，依照安全性區分為實驗環境不存在障礙物的「安全模式」與存在障礙物並可能發生碰撞「危險模式」。並針對危險模式提出了一種使用改進的 Grossberg 神經網絡（GNN）的新演算法來避免 UAV 與障礙物/碰撞。

He et al.(2017)的研究中，使用 Leader & follower 演算法，將無人機群中區分成領導無人機群及跟隨無人機，其中由領導群接收飛行時的訊息進行整隊，藉由領導群發出接下來行動時的變數，如速度、加速度，進行有障礙環境中的避撞。

Majd et al.(2020)的研究中所提出的避撞方法由五個部分組成，包括離線部分、動態進化、關鍵指令、運行時安全監控和決策中心。而其中的離線部分將先明確定義應該驗證以確保群體運動安全的條件，即群體確實與靜態物體、彼此和/或動態出現在群體飛行區域中的物體發生碰撞。並將路線規劃視為一個最佳化問題，目的是最大限度地提高安全性，同時最大限度地減少每架無人機的路徑長度，以實現更高的效率。

## 2.4 無人機路線規劃相關文獻

在無人載具路徑規劃上，必須依照其功能及使用環境，規劃出一套包含各架機器，在何時需要前往何地的移動路線，廣義來說屬於區位途程問題（Location Routing Problem，LRP）的範疇。Schneider & Drexel(2017) 整理 LRP 的應用分類上，定義了這類問題的特徵主要包含：相關數據設定為已知、有個規劃好的周期、潛在的決定設施位置為有限、有一個可量化的目標函數、每一

個需求都必須由某條路徑來滿足等。可以看出其中設定，能夠套用至無人機群飛展演的問題設定上，假設將分做轉場展演兩時段；問題設定需要在展演時段，在特定節點發光；可使用之無人機群數量有限；期望求得以最小化總成本的目標完成演出。而 LRP 中主要應用在移動載體的問題，分類至 Vehicle Routing Problems (VRPs) 的領域，其廣泛運用於運籌學的物流與運輸問題上。

Mor & Speranza (2020) 將典型的 VRP 問題進行列舉說明，思考哪些車輛該從哪一個位置依何順序最後返回相同的起始倉庫，在過程中每台車最多只會有一條路線可遍行，並依據此移動路線來為客戶提供服務，在最後給出整體所需使用的車輛數。這樣的描述也跟 UAV 展演規劃求解問題中的移動安全限制相似，多架 UAV 在空中移動、滯空停留，每一架只會有一條唯一的移動軌跡，並且整體不會重複移動到相同位置。

近年來由於 UAV 技術上的提升，機體輕巧、機動性高的優勢，使其應用上的限制越來越少，也更被各界所接受及使用，也因此 UAV 的路徑規劃問題上，就有多位學者提出相關研究結果，相關的學術研究分別依照其不同的研究方法、數學模式、演算法等介紹如下。

Yilmaz, Yakici & Karatas (2019) 結合 UAV 與 USV 兩種無人載具，在限制船提供 UAV 停靠充電使用與可停靠的數量有限制的情況下，使用混整數線性規劃公式與改良過的蟻群優化演算法，來解決在限定時間內 UAV 可訪問特定目標最多的問題。陳彥瑋 (2020) 結合 UAV 以及地上載具作為 UAV 行動式充換電站，減少搜救任務受限地形限制的缺點，並以整數規劃模式及啟發式演算法求解兩者的協同合作的搜救路線規劃，使其可以在最短的期望時間內搜索範圍內的靜態標靶（即不會移動的搜救目標）。

Chen, Wang, Chen & Han (2018) 則指出，可以透過整數規劃模式找出 UAV 展演中最佳的無人機部署方式，以便在燈光秀期間以最少的時間改變無人機的位置以達成演出圖像正確的優點；另外，也從中提出路徑規劃過程中，由於皆

以最短移動路線（直線），作為 UAV 的移動路線，這種方式容易產生路線互相交叉進而導致路線間低於安全距離而碰撞的情形，並依此提出檢查碰撞與否的判斷方式。然而他們並沒有實際建立完整數學模式及演算法以實踐兩種想法。

Sun, Qi, Wu & Huang (2020) 於研究中指出，Wang et al. (2019) 基於 Leader-Follower 方法，實現了 21 架固定機翼無人機的規劃和飛行控制，該方法適用於分層結構中的編隊飛行，並且取決於機體設計中前導桿的穩定性。

將以上路徑規劃研究所使用的研究對象、考量因素、求解目標及方法整理成表 4，從中可以看出本研究與其它研究不同之處在於，將提出考量安全距離及電力消耗的數學規劃模式及求解演算法，期望可以以最小成本完成展演。

表 4 無人機路線規劃相關文獻比較表

作者	規劃種類		求解方式		考量因素		求解目標		
	載具	多架	規劃模式	演算法	電力消耗	安全距離	最小移動距離	最大覆蓋率	最小耗電量
Chen et al. (2018)	UAV		V				V		
Wang et al. (2019)	UAV	V							
陳彥瑋 (2020)	UAV+GV	V	V	V	V			V	
Yilmaz, Yakici & Karatas (2019)	USV+UAV	V	V	V	V		V		
蔡成鴻(2020)	UAV	V	V	V	V		V		V
本研究	UAV	V	V	V	V	V	V	V	V

## 2.5 小結

本研究與上述 UAV 路徑規劃文獻上，相似處都是在解決 LRP 問題，但由於 UAV 展演須全面考量每階段的移動配置來達成全面的最佳化評估，故適合採用整數規劃模式來求解各個不同畫面間的移動路徑選擇與最小所需的 UAV 數。且目前文獻當中，雖有以電力消耗為研究方向提出數學模式求解方法，也有以飛行安全距離考量為題，以提出啟發式演算法的研究。但是缺乏一套同時考量成本、安全距離以及電力耗損三項 UAV 展演路線規劃中的重點的研究，故本研究希望提出一套整數規劃模式，同時考量上述三個面向進行最小化總成本的最佳化求解方法。



### 第三章 無人機群飛展演之最佳飛行路線整數規劃模式

本章節探討結合安全距離以及考量 UAV 充換電需求的路徑規劃問題，先描述問題，說明多台無人機如何在可充換電的情況下，規劃每架 UAV 的安全路徑，再詳細敘述本研究的假設以及網路架構。接著，依據參考文獻當中說明本問題期望之最低成本消耗的計算方式，將其應用在本文的數學模式中。另一方面，為了使建立數學模式更為容易，我們將欲解決問題的網路結構設計為以時空網路為架構之整數規劃模式。並使用小問題範例來做模式求解的演示，最後針對本章節論述以小結說明與進行章節彙整。

#### 3.1 問題描述

本研究以近年來在各大活動中大放異彩的無人機群飛展演為研究主題，在展演過程中，多架 UAV 從一固定的起點起飛，在不同的時間區段上應用多架 UAV，以變換燈號的方式呈現各種不同的圖像設計，其中考量多架 UAV 是否移動、移動距離、移動路線、期間是否亮燈等等狀態以完成整場演出。但是，實務上往往受限於表演的時間長度、飛行移動的距離、排列圖案的複雜程度、機體移動時的安全性、機體本身的電容量與移動耗電率等，都會影響無人機群的配置量與整個活動的成本。

以 Taiwan Drone 100 於 2020 年的臺南安平國慶煙火晚會中展出畫面為例，UAV 從「民主台灣，自由前行」的字樣轉換成以愛心包圍的「Taiwan Can Help」圖像，兩畫面可以清楚地發現其需配置的 UAV 數量是不同的，且每架 UAV 為了完成圖像，移動的位置、方向也完全不同。同時，還需要考慮行進過程中的燈號切換(亮與不亮)，若移動距離太長、頻繁切換燈號都會讓 UAV 耗電率過高，導致中途需要換電補給的情況；另一方面，多台 UAV 不同的移動位置以

及方向，更容易造成碰撞意外發生，導致機體虧損，甚至使表演圖像有缺失。以圖 4、圖 5 表示 UAV 移動將會發生碰撞的路徑範例，其中包含兩無人機路線有相同終點、起訖位置組合相反之對飛路線（如圖 4 所示），以及兩行徑路線間距小於安全距離（如圖 5 所示）。

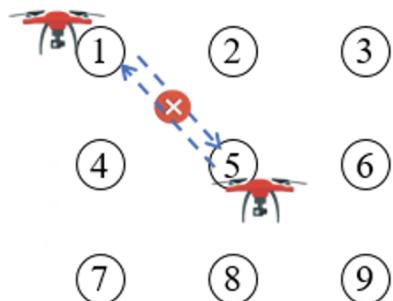


圖 4 無人機移動方式限制\_起訖位置組合相反之對飛路線

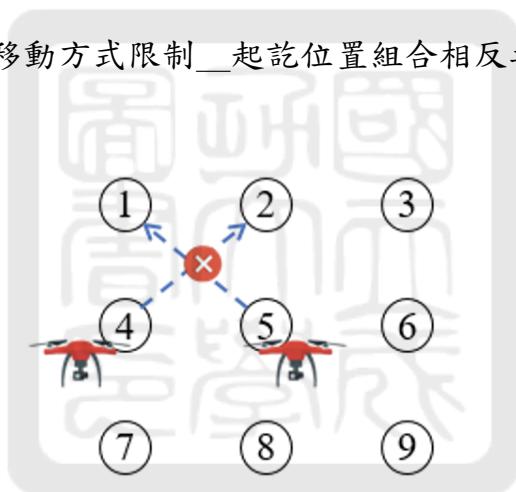


圖 5 無人機移動方向限制\_兩路徑間距小於安全距離

故本研究考量若能有效規劃 UAV 移動路徑，讓 UAV 移動時儘量只做鄰近位移，希望能讓整體的移動距離縮小，減少各階段每架 UAV 的耗電量來達成整體最小耗電成本的目標。

## 3.2 問題假設

本研究依據實務上展演的方式，假設 UAV 都是由相同位置出發，且每台 UAV 的性能、外觀、電池容量與耗電率相同，飛行後會回到相同的出發位置，唯每台以不同代號編制，且每台移動路線軌跡不會相同。移動過程中若需進行換電也是返回其原起訖位置進行補給或更換。

為方便將問題使用整數規劃模式來配置，本研究進行下述基本假設：

1. 每架 UAV 都是相同起點出發，表演後回到相同原點。
2. 每架 UAV 繼航力有限制且相同，若飛行中辨別為電力不足需返回原點進行換電。
3. 展演過程中需換電，需回到原點換電站。
4. 每一個亮燈點由 UAV 移動或不移動亮燈來完成，亮燈位置為已知且定義，但無須指定由特定哪一架來完成。
5. 每台 UAV 在避免大範圍移動與避免碰撞的情況下，設定在每單位移動距離為鄰近一單位，可橫向、直向或斜行，且依移動距離與是否亮燈有不同的耗電支出。
6. 每台 UAV 移動路線不可出現低於安全距離的情況。
7. 假設 UAV 飛行時，不考量飛行時的外在天氣、干擾、災難等不可控制因素，每架的飛行速率相當。
8. 假設每期展演的畫面持續時間及轉場所需時間可透過設定來控制。

### 3.3 固定式位置換電之整數規劃模式

#### 3.3.1. 網路結構

蔡成鴻（2020）決定以層空網路模型（Level space network）為基礎建立整數規劃模式。以  $k$  作為 UAV 數量的索引，則每一層代表每一架 UAV 可能飛行的路徑網路，總共  $K$  層 ( $K$  架)，圖 8 中以單一層為例，不同架 UAV 的飛行路徑都不相同，以點 0 做為出發起始點代號；不同的畫面設定為不同時間區段  $t$ ，以  $t$  逐次累加 1 設為不同畫面的時間區段，若畫面增加則展演時刻  $t$  亦增加；展演過程中，若需要換電，需要回到原始點 0；而最後，展演結束時也皆需要回到起始點 0，表示完成此次表演。

本模式設定已知的腳本圖形由一網格隨機生成，在求解時每層網路都考慮網格中所有節點，如圖 6，其中都不亮燈的區塊在每一層網路中仍會進行運算，因此容易消耗過多的運算時間。

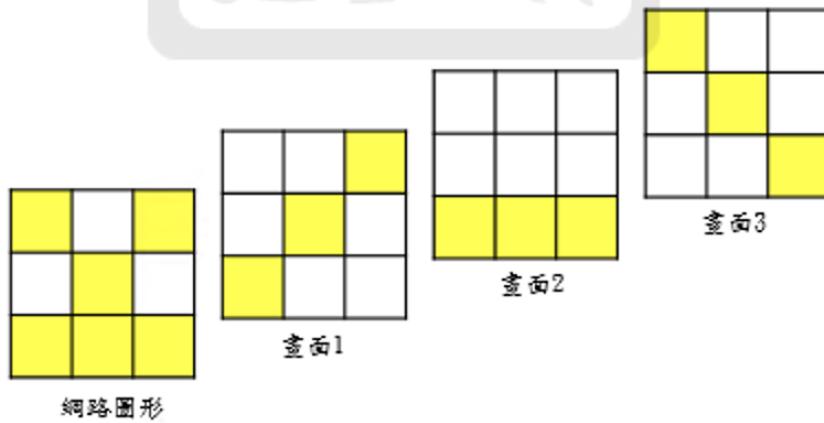


圖 6 網格型態的網路圖形

因此，本研究將網路圖形進行預處理，將腳本中所有展演時段需要亮燈的節點聯集起來，並將此聯集做成新的網路圖形，如圖 7 所示，圖中聯集後的網路僅包含都需要亮燈的區塊，以減少求解時需要考慮的節點數量，並建立圖 8 所示之層空網路結構。

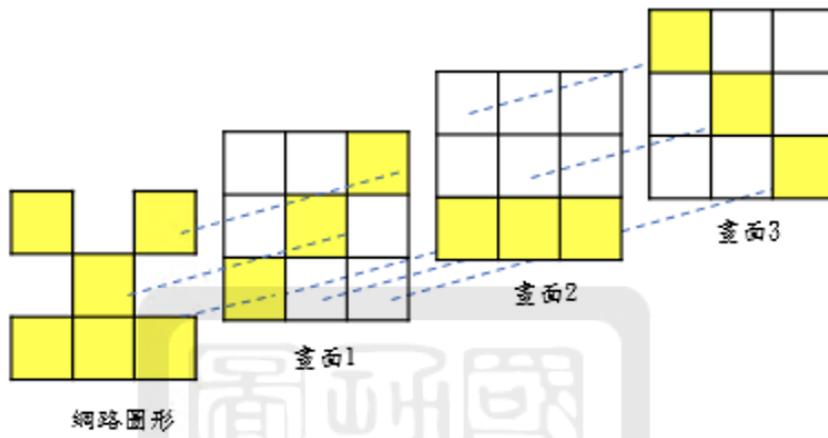


圖 7 聯集後重新定義之網路圖形

另一方面，蔡成鴻（2020）提出的整數規劃中並未考量 UAV 的行進間安全距離，而是假設 UAV 在飛行時有自我防撞的機制；而本研究並不考慮自我防撞機制，如圖 9，給定一架 UAV 將沿路徑  $(5,1)$  移動，有另一架 UAV 沿路徑  $(2,4)$  移動，此時會有機會發生碰撞意外；給定一架 UAV 將沿路徑  $(5,9)$  移動時，若有另一架 UAV 沿路徑  $(9,5)$  移動，則此亦將可能發生碰撞意外；反之，若兩 UAV 移動路線在不同的時刻交叉，則也可能是安全可行的。然而，上述情況目前皆尚未被考慮在模式求解過程。本研究針對考慮上述未被考量的設定，提出考量不同時刻的行徑間安全距離計算方式，並將其計算結果應用於整數規劃模式中，避免 UAV 於網路中移動時，出現安全疑慮。

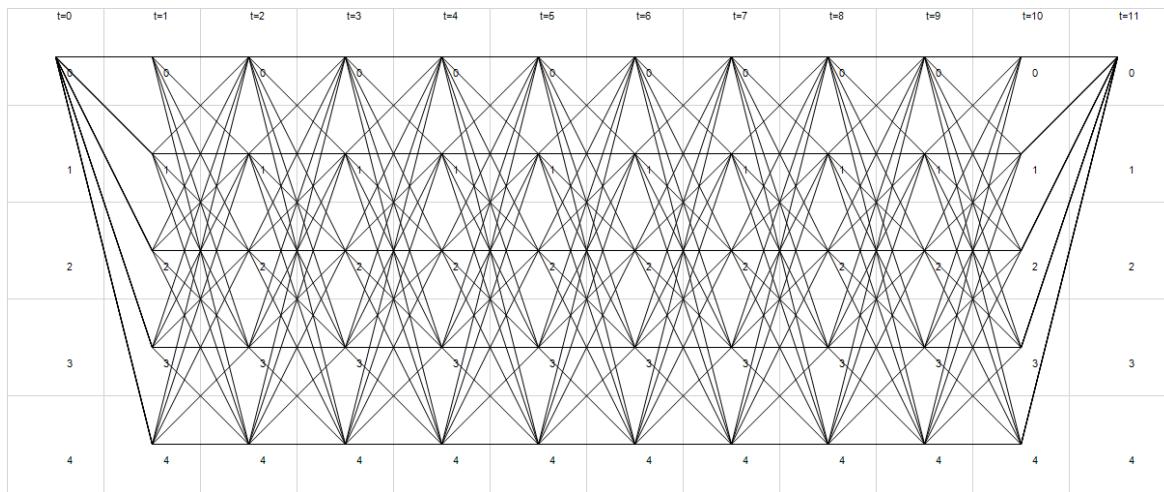


圖 8 單一（某第  $k$ ）層之層空網路示意圖

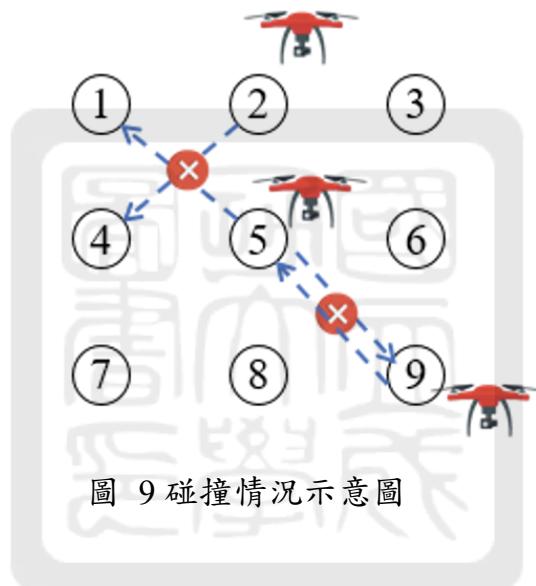


圖 9 碰撞情況示意圖

### 3.3.2. 飛行安全距離

在本節當中，我們將介紹一種兩架無人機行徑時的安全距離計算方式，其假設每台無人機在轉場時段開始時，開始移動，轉場時段結束時抵達展演時段需停留的位置，故每架無人機速度視其移動距離而改變，不一定相同，以下說明計算方式。

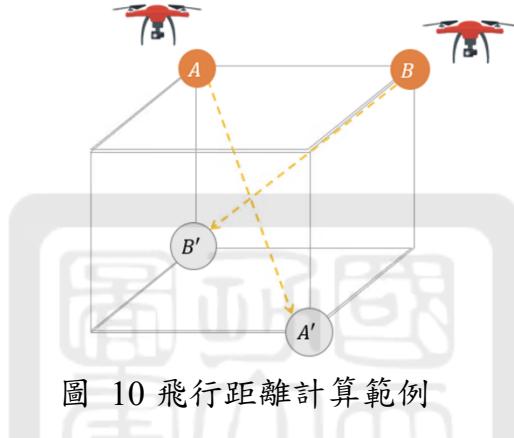


圖 10 飛行距離計算範例

假設一架無人機  $a$  在轉場時段  $t'$  時自  $A$  點，座標以  $A(x_a, y_a, z_a)$  表示，沿  $\overline{AA'}$  路徑移動至  $A'$  點，座標為  $A'(x_{a'}, y_{a'}, z_{a'})$ ，另一架無人機  $b$  在同一時段自  $B$  點，座標以  $B(x_b, y_b, z_b)$  表示，並沿  $\overline{BB'}$  路徑移動至座標  $B'(x_{b'}, y_{b'}, z_{b'})$ ，如圖 10 所示，可以推得兩架無人機移動之向量分別為：

$$\overrightarrow{AA'} = (x_{a'} - x_a, y_{a'} - y_a, z_{a'} - z_a) \quad , \quad \overrightarrow{BB'} = (x_{b'} - x_b, y_{b'} - y_b, z_{b'} - z_b) \quad ,$$

兩路徑長分別為：

$$D_A = |\overrightarrow{AA'}| = \sqrt{(x_{a'} - x_a)^2 + (y_{a'} - y_a)^2 + (z_{a'} - z_a)^2} \quad ,$$

$$D_B = |\overrightarrow{BB'}| = \sqrt{(x_{b'} - x_b)^2 + (y_{b'} - y_b)^2 + (z_{b'} - z_b)^2} \quad ,$$

由此可以計算兩架無人機  $a$ 、 $b$ ，在時刻  $t_i$  時，兩者的座標將分別為：

$$A_{t_i} = (x_a, y_a, z_a) + D_A t_i \frac{\overrightarrow{AA'}}{|AA'|} = (x_a + t_i(x_{a'} - x_a), y_a + t_i(y_{a'} - y_a), z_a + t_i(z_{a'} - z_a))$$

$$B_{t_i} = (x_b, y_b, z_b) + D_B t_i \frac{\overrightarrow{BB'}}{|BB'|} = (x_b + t_i(x_{b'} - x_b), y_b + t_i(y_{b'} - y_b), z_b + t_i(z_{b'} - z_b))$$

並藉由歐式距離公式計算兩架無人機在時刻  $t_i$  的距離：

$$d_{t_i} = \sqrt{\left\{[x_b + t_i(x_{b'} - x_b)] - [x_a + t_i(x_{a'} - x_a)]\right\}^2 + \left\{[y_b + t_i(y_{b'} - y_b)] - [y_a + t_i(y_{a'} - y_a)]\right\}^2 + \left\{[z_b + t_i(z_{b'} - z_b)] - [z_a + t_i(z_{a'} - z_a)]\right\}^2}$$

令  $S = (x_b - x_a)$ 、 $S' = (x_{b'} - x_b - x_{a'} - x_a)$ 、 $T = (y_b - y_a)$ 、 $T' = (y_{b'} - y_b - y_{a'} - y_a)$ 、

$U = (z_b - z_a)$ 、 $U' = (z_{b'} - z_b - z_{a'} - z_a)$ ，則可以將算式簡化為

$d_{t_i} = \sqrt{(S + S't_i)^2 + (T + T't_i)^2 + (U + U't_i)^2}$ ，並經由配方法推導後，整理為

$$d_{t_i} = \sqrt{t_i + \left( \frac{SS' + TT' + UU'}{S'^2 + T'^2 + U'^2} \right)^2 + \frac{(S^2 + T^2 + U^2)}{S'^2 + T'^2 + U'^2} - \frac{(SS' + TT' + UU')^2}{S'^2 + T'^2 + U'^2}} \text{，可以得知當}$$

$t_i = -\left( \frac{SS' + TT' + UU'}{S'^2 + T'^2 + U'^2} \right)$  時，兩路徑會出現最短距離

$\sqrt{\frac{(S^2 + T^2 + U^2)}{S'^2 + T'^2 + U'^2} - \frac{(SS' + TT' + UU')^2}{S'^2 + T'^2 + U'^2}}$ ，因此當此距離小於腳本所設定的安全距離時，

代表  $A(x_a, y_a, z_a) \rightarrow A'(x_{a'}, y_{a'}, z_{a'})$ 、 $B(x_b, y_b, z_b) \rightarrow B'(x_{b'}, y_{b'}, z_{b'})$  兩路徑並不安全、有碰撞危險，便將此兩節點對加入會發生衝突的路線集合中。

### 3.3.3. 參數與變數定義

此節將說明整數規劃模式中，所需的參數及變數的定義。

首先，本研究將 UAV 展演的時間區段分為兩類：展演時段、轉場時段，兩者互相間隔構成整場演出，其排列方式如表 5 所示，將展演開始前，定義為第一個展演時段，而此時所有 UAV 皆須在原點（節點 0）；另在演出結束時新增一個展演畫面，規定所有無人機，此時皆須在原點並且不亮燈。

另外，為了使建構數學模式更加方便，將網路中各個節點座標轉換為節點代號，其轉換方式如圖 11 所示，圖中將畫面轉換為一個 3x3 的網格狀網路圖形，並將每個點的座標位置轉換為節點代號，而後續整數規劃模式中則以各節點代號將其進行運算及限制。

表 5 三個畫面期數說明

$t=0$	$t'=1$	$t=2$	$t=3$	$t=4$	$t'=5$	$t=6$	$t=7$	$t=8$
停滯	轉場	停滯	轉場	停滯	轉場	停滯	轉場	停滯

[註] 包含起訖原點共計 2 個起訖點、3 個展演停滯時段、與 4 個轉場移動時段，表示 T 的區間其值為 8。

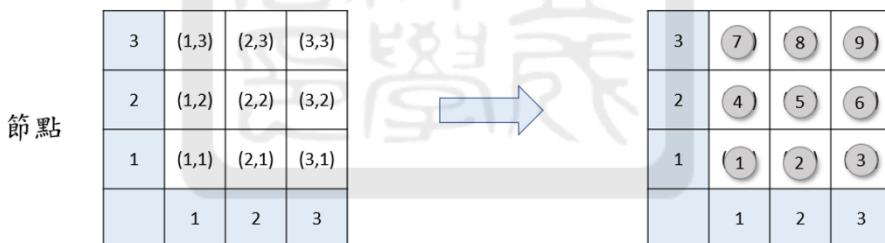


圖 11 座標位置—節點編號對照說明

以下說明集合、參數及變數。

## 集合與索引

表 6 集合與索引

集合	索引	說明
節點代號 (Nodes)	$i, j$	$P = \{0, 1, 2, \dots,  P  - 1\}$
無人機 (Drones)	$k$	$K = \{0, 1, 2, \dots,  K  - 1\}$
展演的時間集合 (Stay Time)	$t$	$T^s = \{0, 2, 4, \dots,  T  - 2\}$
轉場時刻的集合 (Move Time)	$t'$	$T^m = \{1, 3, \dots,  T  - 1\}$
		其中 $T = T^s \cup T^m$ $ T $ 為偶數 = 2 (畫面數+1)
時間集合 (Time)	$\hat{t}$	$T = \{0, 1, 2, \dots,  T  - 1\}$

## 參數

表 7 參數說明

名稱	描述
$\alpha_1$	停留一個單位所耗電量(單位 W)，文中以 1 來做實驗。
$\alpha_2$	移動一個單位所耗電量(單位 W)，文中以 1.5 來做實驗。
$\beta$	亮燈一個單位所耗電量(單位 W)，文中以 2 來做實驗。
$C_c$	UAV 每次的換電費用，文中採用數值 100 單位來做實驗。
$C_k$	每架 UAV 的租機費用，文中採用數值 1000 單位來做實驗。
$C_w$	每單位消耗電量的折合費用，文中採用數值 1 來做實驗。
$P_t$	表示腳本設定要亮燈的位置
$M$	極大值
$\bar{W}$	初始電量(單位 W)，文中採用數值 100 來做實驗。
$W$	最小電量(單位 W)，文中採用數值 30 來做實驗。
$A = \{(i, j) : R_{ij} \leq \bar{R}\}$	可在相距 $\bar{R}$ 範圍內移動的節線代號配對 $(i, j)$ 組合
$\bar{A}_j = \{(i, j) \in A\} \cup \{(0, j)\}$	進入 $j$ 點的節線集合
$\hat{A}_i = \{(i, j) \in A\} \cup \{(i, 0)\}$	離開 $i$ 點的節線集合
$XA$	節線對 $(i, j, i', j')$ 其中 $(i, j) \in A$ $(i', j') \in A$ 兩距離小於安全距離
$RL_{ij}$	$(i, j)$ 兩點間的長度
$TC$	Total Cost，預計求算的總成本

## 變數

表 8 變數說明

變數	描述
$x_{t'ij}^k$	無人機 $k$ 在時刻 $t'$ 時是否從 $i$ 移動到 $j$ ，有則為 1；無則為 0
$L_{ti}^k$	無人機 $k$ 在時刻 $t$ 時 $i$ 點是否亮燈，有則為 1；無則為 0
$\bar{L}_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時是否亮燈，有則為 1；無則為 0
$s_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時是否停留，有則為 1；無則為 0
$m_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時是否移動，有則為 1；無則為 0
$\bar{w}_t^k$	無人機 $k$ 在時刻 $t$ 時的起始電量(單位 W)
$\hat{w}_t^k$	無人機 $k$ 在時刻 $t$ 時的結束電量(單位 W)
$\bar{w}_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時的起始電量(單位 W)
$\hat{w}_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時的結束電量(單位 W)
$R_{t'}^k$	利用歐式路徑計算 $k$ 在 $t'$ 時的移動距離
$g_{t'}^k$	無人機 $k$ 在時刻 $t'$ 時的是否停留在地面(基地)，有則為 1；無則為 0
$g_t^k$	無人機 $k$ 在時刻 $t$ 時的是否停留在地面(基地)，有則為 1；無則為 0
$u^k$	無人機 $k$ 是否有使用到 (用來計算實際需要的 UAV 數)，有則為 1；無則為 0

### 3.3.4. 最小化展演總成本

本研究主要參考蔡承鴻（2020）對於無人機展演過程中所花費的總成本計算方式。然而，在現行展演方式當中，UAV 的電池續航力日益增強，對於換電的需求也日益降低，因此，本研究在參考蔡承鴻（2020）的總成本計算方式之外，也提出一種不考慮電量消耗的情況下，最小化展演總成本的方法。並將兩種模式加以修正及新增設定以滿足本研究之網路結構，本節將針對總成本的計算方式進行說明。而本研究中的數值測試皆以不考慮電量消耗的目標函式進行測試。

### 不考慮電量消耗最小化展演總成本

在不考慮電力消耗的情況下，期望能最小化所有 UAV 展演過程中的成本，本研究將(1)實際使用的 UAV 租機成本以及(2)無人機在展演過程中的總移動距離納入考量視為總成本的計算方式，其求解目標式定義與限制式規範如下 3.3.4 小節所示。

*Minimize TD*

$$= \sum_{k \in K} \sum_{t' \in T^m} R_{t'}^k + M \times \sum_{k \in K} u^k$$

### 考慮電量消耗最小化展演總成本

在考慮電力消耗的情況下，期望能最小化所有 UAV 展演過程中的成本，包含耗電支出與相關機體費用，成本將 4 種成本支出納入考量，依序採計為：(1)分別計算停滯狀態時段所有 UAV 的電量消耗、(2)移動狀態時段所有 UAV 的電量消耗、(3)所有 UAV 的換電費用加總與(4)實際使用的 UAV 租機成本等，其求解目標式定義與限制式規範如下所示。

*Minimize TC*

$$\begin{aligned} &= C_w \times \sum_{k \in K} \sum_{t \in T^s} \left( \bar{W}_t^k - \hat{W}_t^k \right) + C_w \times \sum_{k \in K} \sum_{t' \in T^m} \left( \bar{W}_{t'}^k - \hat{W}_{t'}^k \right) \\ &+ C_c \times \sum_{k \in K} \sum_{t' \in T^m} \sum_{j \in P} x_{t'0j}^k + C_k \times \sum_{k \in K} u^k \end{aligned}$$

### 3.3.5. 不考慮電量消耗之整數規劃模式

#### 目標式

*Minimize TD*

$$= \sum_{k \in K} \sum_{t' \in T^m} R_{t'}^k + M \times \sum_{k \in K} u^k \quad (3.2.1)$$

## A. 移動限制

限制式(3.2.2) 限制每架 UAV 在第一個轉場時段  $t' = 1$  時，從原點出發。

$$\sum_{j \in P} x_{10j}^k = 1 \quad \forall k \in K \quad (3.2.2)$$

限制式(3.2.3) 限制每架 UAV 在最後一個轉場時段  $t' = |T| - 2$  時，回到原點。

$$\sum_{i \in P} x_{(|T|-2)i0}^k = 1 \quad \forall k \in K \quad (3.2.3)$$

限制式(3.2.4) 限制每架 UAV 在一轉場時段中的移動路線加總只會有一條。

$$\sum_{i \in P} \sum_{(i,j) \in \bar{A}_j} x_{t'ij}^k = 1 \quad \forall t' \in T^m, k \in K \quad (3.2.4)$$

限制式(3.2.5) 限制每架 UAV 的移動路線終點不可相同。

$$\sum_{(i,j) \in \bar{A}_j} \sum_{k \in K} x_{t'ij}^k \leq 1 \quad \forall t' \in T^m, j \in P - \{0\} \quad (3.2.5)$$

限制式(3.2.6) 流量守恆，UAV 進入數節點的數量與飛出會相同，若在轉場時段  $t'$  進入節點  $j$ ，下一個轉場時段  $t'+2$  就必須由節點  $j$  出發。

$$\sum_{(i,j) \in \bar{A}_j} x_{t'ij}^k - \sum_{(j,i) \in \bar{A}_j} x_{(t'+2)ji}^k = 0 \quad \forall j \in P \cup \{0\}, t' \in T^m, k \in K \quad (3.2.6)$$

限制式(3.2.7) 紀錄 UAV 的使用需求數，若 UAV 自原點出發至其它不為原點的節點，則視為有使用。

$$\sum_{j \in P - \{0\}} x_{t'0j}^k \leq u^k \quad \forall k \in K, t' \in T^m \quad (3.2.7)$$

限制式(3.2.8) 移動路線不可出現兩路徑低於安全距離的情況，因此，若兩節線 $(i, j)$   
 $(i', j')$ 為會發生衝突的節線配對組合，則不可以同時存在。

$$x_{t'ij}^k + \sum_{k' \in K \setminus \{k\}} x_{t'i'j'}^{k'} \leq 1 \quad \forall k \in K, t' \in T^m, (i, j, i', j') \in XA \quad (3.2.8)$$

## B. 亮燈限制

限制式(3.2.9) 須有一架 UAV 於展演時段在腳本設定之亮燈位置停留。

$$\sum_{k \in K} x_{tii}^k = 1 \quad \forall i \in P_t, t \in T^s \quad (3.2.9)$$

限制式(3.2.10) 限制腳本設定位置要亮燈。

$$\sum_{k \in K} L_{ti}^k = 1 \quad \forall t \in T^s, i \in P_t \quad (3.2.10)$$

限制式(3.2.11) 限制若一架 UAV $k$ 需於展演時段 $t$ 在腳本設定的節點 $i$ 發亮

( $L_{ti}^k = 1$ )，則該 UAV 必須於前一期轉場時段 $(t-1)$ 移動至此

$$(\sum_{(j,i) \in \bar{A}_t} x_{(t-1)ji}^k = 1) \circ$$

$$L_{ti}^k \leq \sum_{(j,i) \in \bar{A}_t} x_{(t-1)ji}^k \quad \forall i \in P, t \in T^s, k \in K \quad (3.2.11)$$

限制式(3.2.12) 限制若一架 UAV $k$ 於兩展演時段 $t$ 、 $(t+2)$ 需要發亮( $\sum_{i \in P} L_{ti}^k = 1$ 、

$\sum_{i \in P} L_{(t+2)i}^k = 1$ )，則轉場時段 $t+1$ 也需要發光( $\bar{L}_{(t+1)}^k = 1$ )。

$$\sum_{i \in P} L_{(t)i}^k + \sum_{j \in P} L_{(t+2)j}^k \leq \bar{L}_{(t+1)}^k + 1 \quad \forall k \in K, t \in T^m \quad (3.2.12)$$

限制式(3.2.13) 設定原點不發光。

$$L_{t0}^k = 0 \quad \forall k \in K, t \in T^s \quad (3.2.13)$$

### C. 狀態判斷

限制式(3.2.14) 區分 UAV 的狀態，只會是空中移動  $m_{t'}^k$ 、滯空停留  $s_{t'}^k$  或留在原點  $g_{t'}^k$  的其中一種。

$$m_{t'}^k + s_{t'}^k + g_{t'}^k = 1 \quad \forall k \in K, t' \in T^m \quad (3.2.14)$$

限制式(3.2.15) 紀錄移動狀態，若一架 UAV  $k$  在轉場時段  $t'$  時沿路線  $(i, j)$  移動  $(\sum_{(i,j) \in A} x_{t'ij}^k = 1)$ ，則紀錄為空中移動狀態 ( $m_{t'}^k = 1$ )。

$$\sum_{(i,j) \in A} x_{t'ij}^k \leq m_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.15)$$

限制式(3.2.16) 紀錄滯空狀態，若一架 UAV  $k$  在轉場時段  $t'$  時停留在非原點的節點  $i$   $(\sum_{i \in P \setminus \{0\}} x_{t'i}^k = 1)$ ，則紀錄為滯空狀態 ( $s_{t'}^k = 1$ )。

$$\sum_{i \in P \setminus \{0\}} x_{t'i}^k \leq s_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.16)$$

限制式(3.2.17) 紀錄停留原點狀態，若一架 UAV  $k$  在轉場時段  $t'$  時停留 在原點 ( $x_{t'00}^k = 1$ )，則紀錄為留在原點狀態 ( $g_{t'}^k = 1$ )。

$$x_{t'00}^k \leq g_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.17)$$

限制式(3.2.18) 由於展演時段不可移動，因此若轉場時段留在原點， 則展演時段也會在原點。

$$g_{t'}^k \leq g_{t'+1}^k \quad \forall k \in K, t' \in T^m \quad (3.2.18)$$

限制式(3.2.19) 紀錄不在地面的狀態(可能為移動或滯空)。

$$\sum_{(i,j) \in \bar{A}} x_{t'ij}^k \leq (1 - g_{t'}^k) \quad \forall k \in K, t' \in T^m \quad (3.2.19)$$

#### D. 移動距離

限制式(3.2.20) 計算 UAV k 在移動時所飛行的總距離。

$$\sum_{(i,j) \in A} (RL_{ij} \times x_{t'ij}^k) = R_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.20)$$

### 3.3.6. 考慮電量消耗之整數規劃模式

在 3.3.4 小節中，我們提出了一種不考慮電量消耗的數學模式，本節則是在考慮電量消耗的情況下所需要增加的限制式以及最小化展演總成本之目標函數。

#### 目標式

*Minimize TC*

$$\begin{aligned} &= C_w \times \sum_{k \in K} \sum_{t \in T^s} \left( \tilde{w}_t^k - \hat{w}_t^k \right) + C_w \times \sum_{k \in K} \sum_{t' \in T^m} \left( \tilde{w}_{t'}^k - \hat{w}_{t'}^k \right) \\ &+ C_c \times \sum_{k \in K} \sum_{t' \in T^m} \sum_{j \in P} x_{t'0j}^k + C_k \times \sum_{k \in K} u^k \end{aligned}$$

限制式(3.2.22) 初始電量設定為電量上限。

$$\tilde{w}_0^k = \bar{W} \quad \forall k \in K \quad (3.2.22)$$

限制式(3.2.23)、(3.2.24) 以夾擠的方式，限制換電後的電量為電量上限。其機制為，若 UAV  $k$  有回到原點( $\sum_{j \in P} x_{t'oj}^k = 1$ )，此時轉場期初的電量為電力上限(需同時滿足 $\bar{w}_{t'}^k \leq \bar{W}$  、 $\bar{w}_{t'}^k \geq \bar{W}$ )；若沒有回到原點( $\sum_{j \in P} x_{t'oj}^k = 0$ )，則不予限制。

$$\bar{w}_{t'}^k \leq \bar{W} + M \times \left(1 - \sum_{j \in P} x_{t'oj}^k\right) \quad \forall k \in K, t' \in T^m \quad (3.2.23)$$

$$-M \times \left(1 - \sum_{j \in P} x_{t'oj}^k\right) + \bar{W} \leq \bar{w}_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.24)$$

限制式(3.2.25) 展演時段期初電量為上期轉場時段的期末電量。

$$\bar{w}_{t'-1}^k = \hat{w}_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.25)$$

限制式(3.2.26)、(3.2.27) 以夾擠的方式，限制 UAV 在不回到原點的情況下，轉場時段的期初電量為前一期展演時段的期末電量。其機制如下：若 UAV  $k$  沒有回到原點( $\sum_{j \in P} x_{t'oj}^k = 0$ )，此時轉場時段的期初電量為前一期展演時段的期末電量(需同時滿足 $\hat{w}_{t'-1}^k \leq \bar{w}_{t'}^k$  、 $\hat{w}_{t'-1}^k \geq \bar{w}_{t'}^k$ )；若有回到原點( $\sum_{j \in P} x_{t'oj}^k = 1$ )，則不予限制。

$$(-M \times \sum_{j \in P} x_{t'0j}^k) + \hat{w}_{t'-1}^k \leq \bar{w}_{t'}^k \quad \forall k \in K, t' \in T^m \quad (3.2.26)$$

$$\bar{w}_{t'}^k \leq \hat{w}_{t'-1}^k + (M \times \sum_{j \in P} x_{t'0j}^k) \quad \forall k \in K, t' \in T^m \quad (3.2.27)$$

限制式(3.2.28) 計算展演時段為滯空狀態的消耗電量。

$$\bar{w}_t^k - \beta \times \sum_{i \in P} (L_{ti}^k \times T_{t'}^s) - \alpha_1 \times T_{t'}^s (1 - g_t^k) = \hat{w}_t^k \quad \forall k \in K, t \in T^s \quad (3.2.28)$$

限制式(3.2.29) 計算轉場時段為移動狀態的消耗電量。

$$\bar{w}_{t'}^k - \left[ \alpha_1 \times T_{t'}^m \times (s_{t'}^k) + \alpha_2 \times m_{t'}^k \times (R_{t'}^k) + \beta \times T_{t'}^m \times (\bar{L}_{t'}^k) \right] = \hat{w}_t^k \quad (3.2.29)$$

$$\forall k \in K, t' \in T^m$$

限制式(3.2.30) 期末電量恆大於最小值。

$$\hat{w}_t^k \geq \underline{W} \quad \forall t \in T^s, k \in K \quad (3.2.30)$$

### 3.4 小結

本章節以 UAV 群飛展演問題作討論與假設作為開場，並針對問題提出本研究之層空網路結構，接著針對問題盡可能以貼近現實展演情況為前提，其中包含分割展演及轉場時段、亮燈方式及移動狀態限制等等情形，提出考慮電量消耗及不考慮電量消耗兩種整數規劃模式，以解決多架 UAV 群飛移動路線的 LRP 問題。然而，因為限制式繁多，對於中大型的網路（期數多、節點數增加、亮燈數量差異大等等），無法在短時間內求得最佳解。因此下一章節將針對考慮安全距離之無人機群飛展演最佳飛行路線規劃問題提出較整數規劃模式更有效率的演算法，以作為求解此問題的替代方案。

## 第四章 無人機群飛展演之最佳飛行路線規劃演算法

在第三章中，我們建立整數規劃數學模式並使用求解器求解，然而在問題規模增加時，例如腳本設定畫面需要發光的節點變多、畫面數量增加，則需要消耗大量時間。近來在現實中的無人機展演常有更多畫面而更複雜，如此則容易出現無法求解的問題，因此本章節在觀察 UAV 飛行方式後，提出啟發式演算法，使其在短時間內便能獲得高品質的解。4.1 節中，將介紹滾動式求解方式；4.2 節中提出求解演算法，4.2.1 解釋演算法步驟及邏輯，並在 4.2.2 說明將不同的路線給定懲罰值以增加該路線的成本的方式；在 4.3 以簡例比較說明兩種演算法及模式的差異，期望使用不同方法獲得高品質的解並減少求解時間。

### 4.1 滾動式求解機制

由於一次求解所費時間過長，本研究嘗試將畫面進行分組批次求解，期望藉由減少每一次求解的問題規模，進而達到加速的效果。然而，畫面的分組方式將會影響求解的效果。本研究將三個畫面分為一組，且為了避免局部最佳解，本研究一輪將會求解 UAV 在二個轉場的移動路線，並將該輪第一個轉場所決定的路線固定，以第二個展演時段 UAV 停留的位置作為下一輪求解的起始位置，如圖 12 所示。

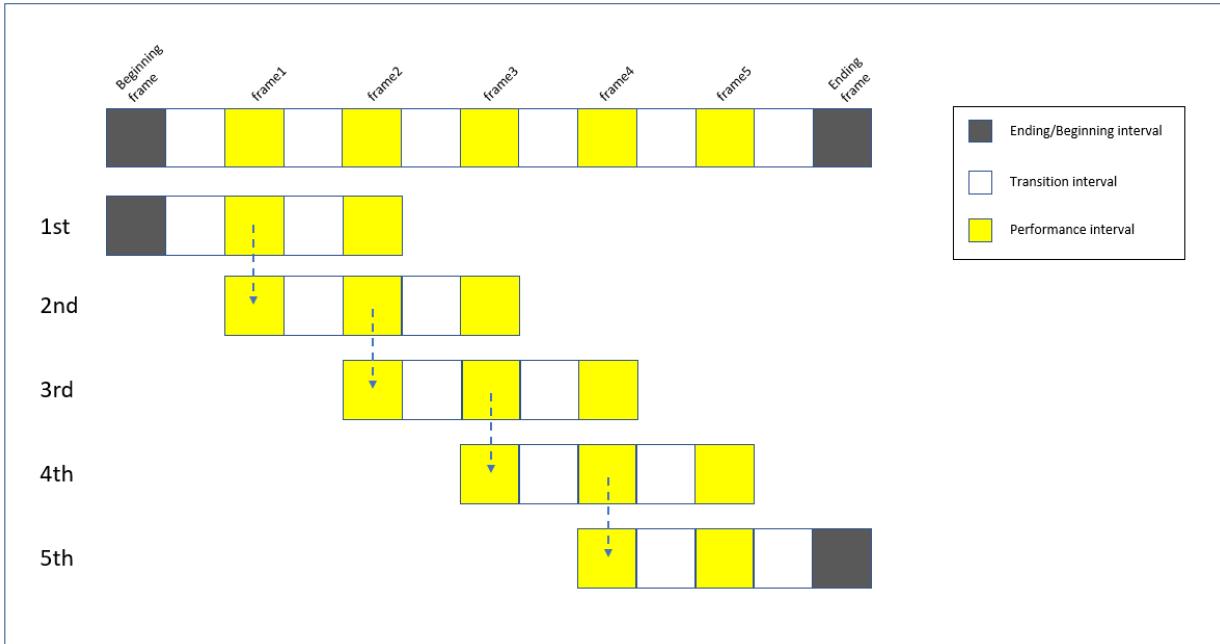


圖 12 滾動式求解機制設定

為了達到縮小問題規模的目的，我們將原始的腳本進行預處理如表 9，已知全部展演畫面之腳本  $Script_{t,j} = (Script_{0,0}, \dots, Script_{t,j})$ ， $Arc_{i,j} = (Arc_{0,0}, \dots, Arc_{i,j})$  為所有可到達的任兩點間的無向節線集合， $N$  為所有節點之集合， $Xarc_i = (Xarc_0, \dots, Xarc_i)$  則為所有會發生衝突之兩節線的集合， $f$  是包含起始及結束的所有展演時段數量， $fp$  代表每一次欲求解的展演時段數量。先將畫面進行分組，將每一輪將用到的三個畫面中需要發亮的點以及當前有 UAV 所在的點進行聯集，如表 9 中的(a)段迴圈，形成新的網路圖形中的節點集合  $R\_node_n$ ；並將該圖形所需要用到的路線集合  $R\_arc_n$ ，如表 9 中的(b)段迴圈，以及低於安全距離的路線集合  $R\_xarc_n$ ，如表 9 中的(c)段迴圈，做為該輪數學模式的輸入資料。

表 9 滾動式求解機制之預處理過程

---

**Rolling Solution Preprocessing**

---

**Data:**  $Script_{t,j}$ ,  $Arc_{i,j}$ ,  $Xarc_i$ ,  $N$ ,  $f$ ,  $fp$ ,  $R\_node_i$ ,  $R\_xarc_i$ ,  $R\_arc_i$

```

1: function Rolling_Prep ( $Script_{t,j}$ ,  $Arc_{i,j}$ ,  $Xarc_i$ ,  $N$ ,  $f$ ,  $fp$ )
2:   for  $i = 0$  to  $f - fp$  do                                (a)
3:     for  $j$  in  $N$  do
4:       for  $t$  in  $(0, fp)$  do
5:         if  $Script_{i+t,j} == 1$  then
6:           Update  $j$  to  $R\_node_i$ 
7:         end for
8:       end for
9:       for  $j$  in  $Xarc_j$  do                                (b)
10:      if  $j < R\_node_i$  then
11:        Update  $j$  to  $R\_xarc_i$ 
12:      end for
13:      for  $j$  in  $Arc_j$  do                                (c)
14:        if  $j < R\_node_i$  then
15:          Update  $j$  to  $R\_arc_i$ 
16:        end for
17:      end for
18:    end function

```

---

**Result:**  $R\_node_i$ ,  $R\_xarc_i$ ,  $R\_arc_i$

---

並依此設計了滾動式求解的流程，如表 10。經表 9 所示之預處理方式處理過後的資料當作輸入，由數學模式求解，如表 10 中的(d)，並將第二個展演時段 UAV 停留的位置紀錄於  $location(t)$ ，如表 10 中的(e)，做為下一輪求解的起始位置，直到全部畫面皆求解完成為止。

表 10 滾動式求解機制流程

---

**Rolling Solution**

---

**Data:**  $f, fp, R\_node_n, R\_xarc_n, loc_n$

- 1: **function** Rolling ( $f, fp, R\_node_n, R\_xarc_n, loc_n$ )
- 2:     **for**  $i = 0$  to  $f - fp$  **do**
- 3:         Model( $R\_node_n, R\_xarc_n, loc_n$ ) (d)
- 4:         **if**  $i == f - fp$  **then** (e)
- 5:             **for**  $t = 0$  to  $f$  **do**
- 6:                 Update result<sub>k,t,i,j</sub> to loc<sub>n</sub>
- 7:                 **end for**
- 8:         **else then**
- 9:             Update result<sub>k,0,i,j</sub> to loc<sub>n</sub>
- 10:        **end for**

---

**Result:** loc<sub>n</sub>

---

而滾動式求解機制之所以能夠達到縮小問題規模，加速求解的主要原因在於，每次求解所包含的畫面數、節點數、可使用無人機數都比一次性求解來的少，因此求解的變數數量也較少，如表 11 參數建立的網路架構為範例，將滾動式求解機制之變數與限制式個數估算，並與一次性求解數學模式做比較，如表 12 為變數總個數的比較，表 13 為限制式個數比較。

表 11 估算變數與限制式個數之參數設定值

節點 數量	無人機 數量	展演 時長	轉場 時長	節線 數量	衝突節線對 數量
P	K	T <sup>s</sup>	T <sup>m</sup>	A	XA
36	40	12	11	330	330

表 12 變數總個數比較

一次性求解			滾動式求解機制		
Variable	Calculation	Number	Variable	Calculation	Number
$x_{t'ij}^k$	$ K  T^m  A $	145,200	$x_{t'ij}^k$	$2 K  A $	26,400
$L_{ti}^k$	$ K  T^s  P $	17,280	$L_{ti}^k$	$3 K  P $	4,320
$L_{t'}^k$	$ K  T^m $	440	$L_{t'}^k$	$2 K $	80
$s_{t'}^k$	$ K  T^m $	440	$s_{t'}^k$	$2 K $	80
$m_{t'}^k$	$ K  T^m $	440	$m_{t'}^k$	$2 K $	80
$R_{t'}^k$	$ K  T^m $	440	$R_{t'}^k$	$2 K $	80
$g_{t'}^k$	$ K  T^m $	440	$g_{t'}^k$	$2 K $	80
$g_t^k$	$ K  T^s $	480	$g_t^k$	$3 K $	120
$u^k$	$ K $	40	$u^k$	$ K $	40
變數總個數		165,200	變數總個數		31,280

表 13 限制式總個數比較

一次性求解			滾動式求解機制		
Constraint	Calculation	Number	Constraint	Calculation	Number
1	$ K $	40	1	$ K $	40
2	$ K $	40	2	$ K $	40
3	$ K  T^m $	440	3	$2 K $	80
4	$ P  T^m $	396	4	$2 P $	72
5	$ K  T^m  P $	15,840	5	$2 K  P $	2,880

6	$ K  T^m $	440	6	$2 K $	80
7	$ K  T^m  XA $	145,200	7	$2 K  XA $	26,400
8	$ \mathbf{P}  T^s $	432	8	$3 \mathbf{P} $	108
9	$ \mathbf{P}  T^s $	432	9	$3 \mathbf{P} $	108
10	$ P  T^s  K $	17,280	10	$3 P  K $	4,320
11	$ K  T^m $	440	11	$2 K $	80
12	$ K  T^s $	480	12	$3 K $	120
13	$ K  T^m $	440	13	$2 K $	80
14	$ K  T^m $	440	14	$2 K $	80
15	$ K  T^m $	440	15	$2 K $	80
16	$ K  T^m $	440	16	$2 K $	80
17	$ K  T^m $	440	17	$2 K $	80
18	$ K  T^m $	440	18	$2 K $	80
19	$ K  T^m $	440	19	$2 K $	80
限制式總個數		184,540	限制式總個數		34,888

## 4.2 貪婪演算法

貪婪演算法的構想來自於觀察現實情況及數學模式所求得之最佳解。為了避免一次中的每一段轉場路線選擇皆選擇最短路線，進而導致局部最佳解，本研究之貪婪演算法將問題分為每一架 UAV 分開求解，而每次決定一架 UAV 在全部轉場時段的最短移動路線，如圖 13 所示；此外每決定一條路線，與其發生衝突的路線就必須被排除，因此在演算法的迭代中，每次結束前都會進行可使用路徑的修正，以確保不發生衝突。

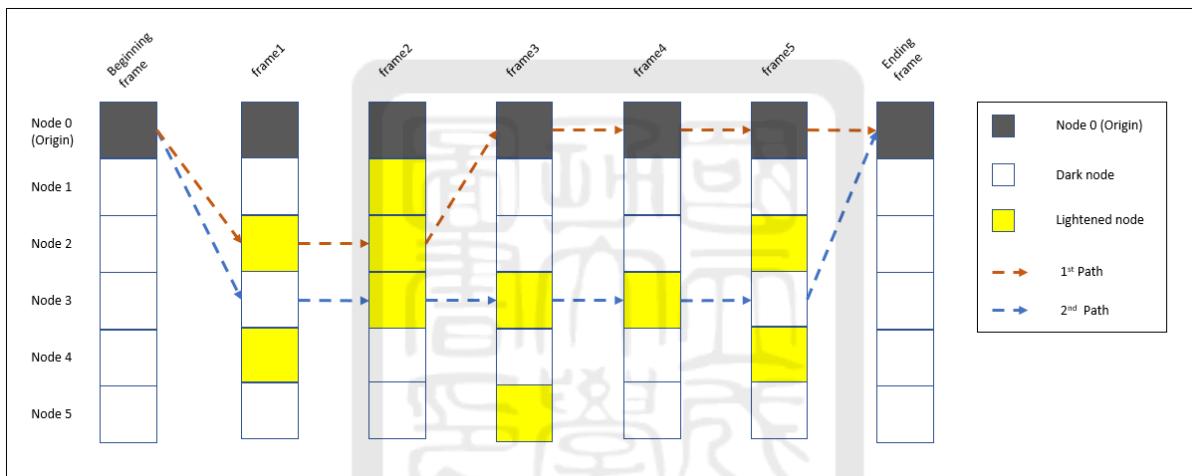


圖 13 貪婪演算法求解例圖

### 4.2.1 演算法步驟及邏輯

本研究提出的演算法主要分做三個部分：節線預處理、最短路徑搜尋、更新待選節線，如圖 14 中流程所示。

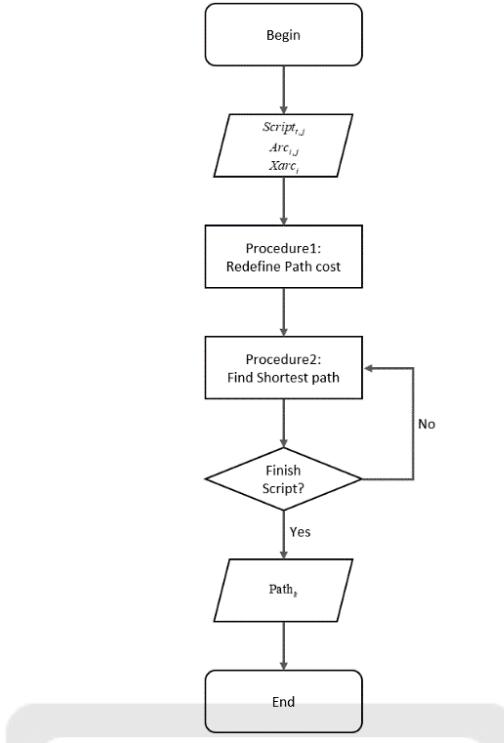


圖 14 貪婪演算法執行流程

首先，為了避免搜尋最短路徑時出現停滯原點、無故長期停滯在空中一點、移動至不需亮燈的位置等狀況，我們將不同情況下的移動，增加節線選擇的懲罰值，詳細的計算方式將在 4.2.2 節中說明；第二部分，為了避免選擇最短路徑時，每次皆選擇該轉場時待選路徑的最短者，造成局部最佳解，本研究採用 Dijkstra's 最短路徑搜尋演算法，計算一無人機從開始制結束的最短路徑；最後依據所選擇的路線，對照可能發生衝突的節線並刪除，將可選擇節線更新。

其邏輯設計如表 14， $Path_i$  代表每一次迭代所決定的路徑， $c\_graph_{i,j}$  為經過路線懲罰值修正過後的網路圖形， $Xarc_i$  為與節線 $i$ 會發生重圖的節線集合， $O$ 、 $D$  分別代表欲找尋最短路徑的起點及終點，在此兩者皆為 0 代表原點。我們先將已知網路的相鄰矩陣  $Arc_{i,j}$ 、節線長度  $length_{i,j}$  透過 Redefine 函式處理，產生新的網路圖  $c\_graph_{i,j}$ ；接著將腳本完成與否作為判斷，若未完成，則使用 Dijk 函式搜尋自 O 至 D 的最短路徑，如表 14 中的(f)，並對選擇的路線檢查是

否完成腳本中的亮點，如表 14 中的(g)段迴圈；接著，使用產生的路徑更新下一次迭代所使用的網路圖形，首先將該輪所選擇路線所經過的節線移除，如表 14 中的(h)段迴圈不予使用，再將所有會與已經過之節線發生衝突的節線移除不予使用，如表 14 中的(i)段迴圈。

接著將以上的步驟反覆執行，直到所派出的無人機能夠正確完成腳本中所設計的圖像後才停止。



表 14 貪婪演算法設計邏輯

---

**Algorithm**

---

**Data:**  $Path_t, Script_{t,j}, Arc_{i,j}, length_{i,j}, c\_graph_{i,j}, Xarc_i, N, O, D$

1:  $c\_graph_{i,j} = Redefine(Arc_{i,j}, length_{i,j})$

2: **function**  $Algo(N, Script_{t,j}, c\_graph_{i,j}, O, D)$

3:     **while**  $Script_{t,j} \neq Null$  **do**

4:          $Path_t = Dijk(c\_graph_{i,j}, O, D)$  (f)

5:         **for**  $i$  in  $Path_t$  **do** (g)

6:             **Update**  $Script_{t,i} = 0$

7:             **end for**

8:         **for**  $i$  in  $Path_t$  **do** (h)

9:             **Update**  $c\_graph_{i,j}$  by  $del(i)$

10:          **for**  $j$  in  $Xarc_i$  **do** (i)

11:             **if**  $j < i$  **then**

12:                 **Update**  $c\_graph_{i,j}$  by  $del(j)$

13:             **end for**

14:          **end for**

15:         **end while**

16:     **end function**

**Result:**  $Path_t$

---

## 4.2.2 路線選擇懲罰值設定

本節在觀察數學模式所求之最佳解方式中，發現移動路線會避免以下三種狀況(如圖 15 所示)：

- (1) 停滯在原點不移動
- (2) 無故且長期停留在空中的一點
- (3) 無故移動至不需亮燈的位置

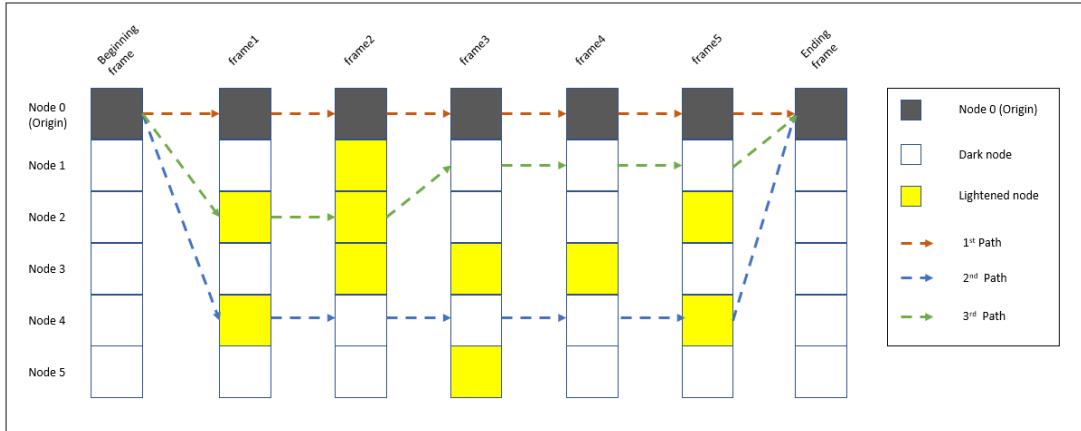


圖 15 避免選擇路線情況

為了解決三種情況，我們將其選擇時加入不同的懲罰值，邏輯如表 15，首先停留在原點將會導致腳本無法順利完成最為嚴重，將懲罰值定義為畫面數與節點數之積，如表 15 中的(j)；若停滯在空中，可能的原因為該點後續其它畫面有發亮的需求，因此將該情形分做兩個處理方式：若該點下個畫面仍必須發亮，則不予懲罰；若下個畫面不需要發亮，則將該節線懲罰值定義為該轉場中所有移動路線的最長距離，如表 15 中的(k)；若無故移動至不需亮燈的位置，不必要的移動將使總成本及總移動距離增加，故將該節線懲罰值定義為該轉場中所有移動路線的最長距離。

表 15 路線選擇懲罰值設定

---

### Redefine

---

**Data:**  $Script_{t,j}, Arc_{i,j}, length_{i,j}, c\_graph_{i,j}, O, f, N$

- 1: **function** Redefine ( $Script_{t,j}, Arc_{i,j}, length_{i,j}, c\_graph_{i,j}, O, f, N$ )
- 2:     **for**  $t = 0$  to  $f$  **do**
- 3:         **for**  $(i,j)$  in  $Arc$  **do**
- 4:             **if**  $t \neq 0$  and  $(i,j) == (O,O)$  **then** (j)
- 5:                 Update  $c\_graph_{i,j} = (i,j, f \times (x-1))$
- 6:             **else if**  $i == j$  and  $Script_{t,j} == 0$  **then** (k)
- 7:                 Update  $c\_graph_{i,j} = (i,j, \max(length_{i,k \in N}))$
- 8:             **else then**
- 9:                 Update  $c\_graph_{i,j} = (i,j, length_{i,j})$
- 10:         **end for**
- 11:     **end for**
- 12: **end function**

---

**Result:**  $c\_graph_{i,j}$

---

## 4.3 簡例說明與求解方法比較

本節以一個小規模的範例如圖 16，將其畫面進行聯集的預處理過後顯示如圖 17，接著進行數學模式求解、滾動式求解、貪婪演算法求解，並進行比較及說明。已知此次展演共有五個畫面、六個節點，而將使用三種不同方法求解六個轉場時段無人機該如何移動、需要使用幾架無人機。

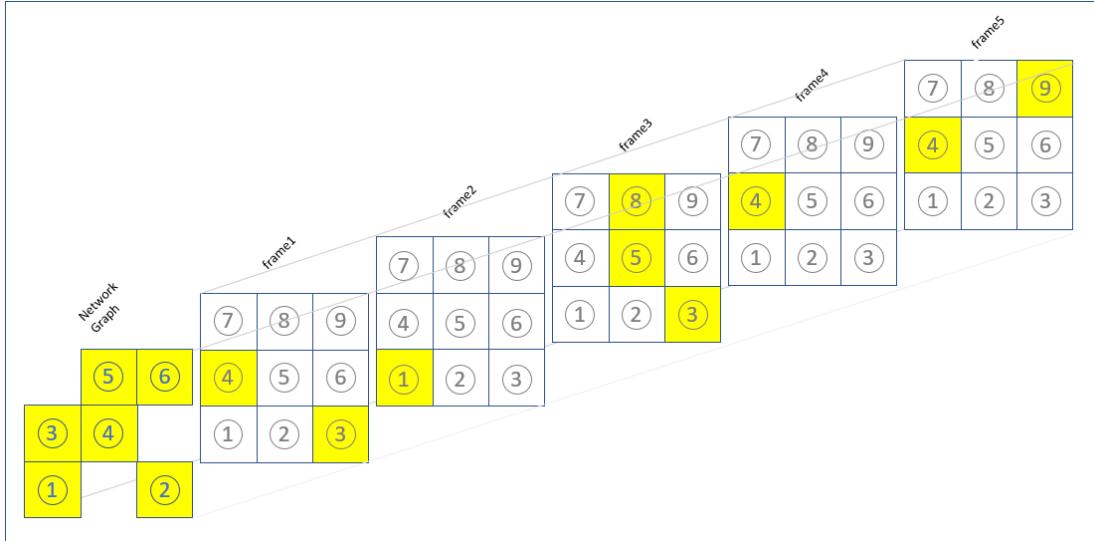


圖 16 簡例腳本畫面

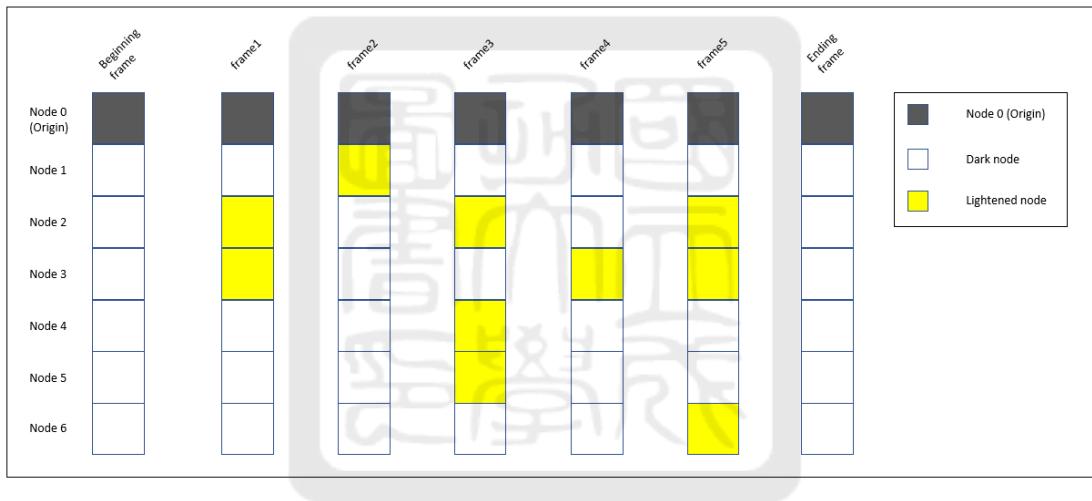


圖 17 預處理後之簡例腳本畫面

經過整數規劃模式求解後的結果，規劃路線如圖 18 所示；圖 19 為使用滾動式求解機制的結果；圖 20 為演算法求解的結果。在此簡例當中，使用數學模式一次性求解與滾動式求解的結果是相同的，總共使用三架無人機完成腳本；而演算法則使用了四架無人機。

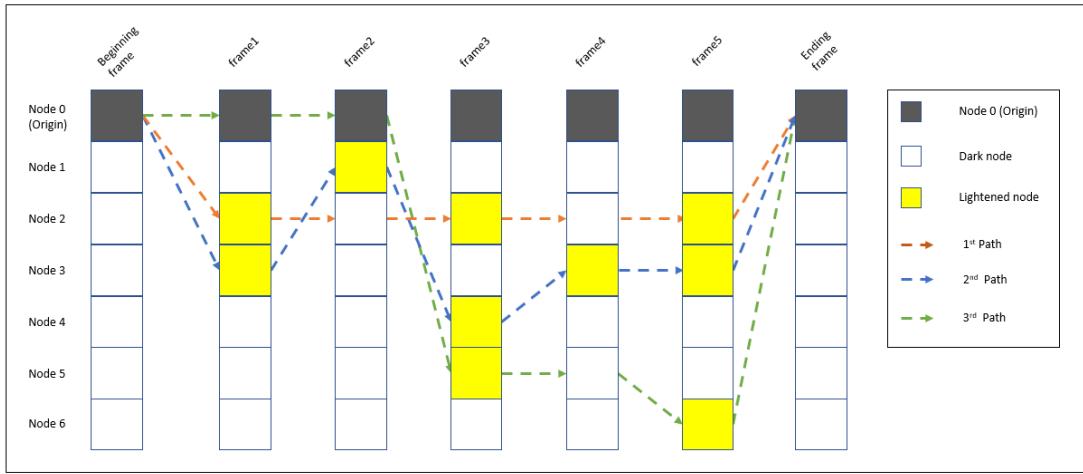


圖 18 以整數規劃模式求解簡例結果

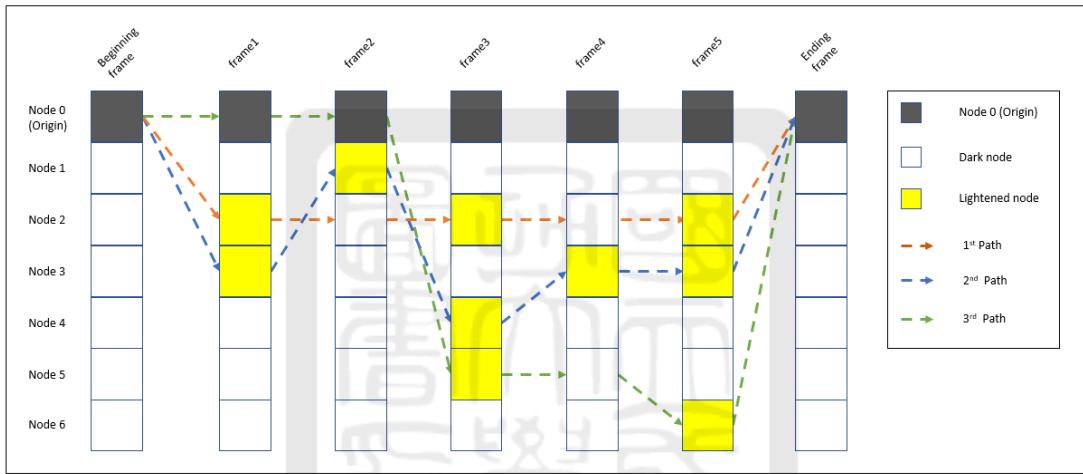


圖 19 以滾動式求解機制求解簡例結果

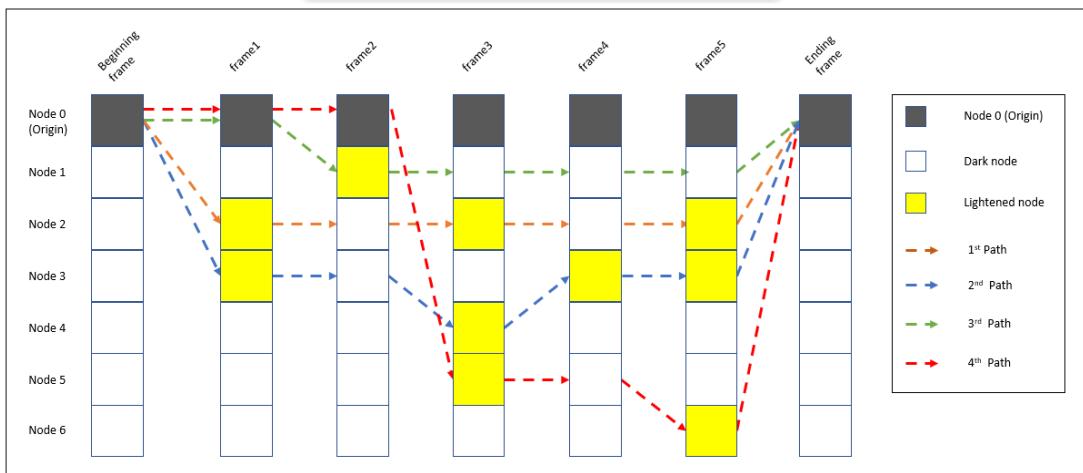


圖 20 以貪婪演算法求解簡例結果

將結果整理成表 16，其中  $Gap_g$  表示 Gurobi 求解器在一次性模式之求解過程中，所計算出的目標值與最佳目標（或其下限）值的相對差距百分比， $Gap$  則表示該求解方法所得之目標值與三種方法中最佳目標值的相對差距百分比。由於在小規模的例子當中，滾動式求解一次所求解的範圍與整體的大小差異不大、次數較多，造成滾動式求解機制與一次性求解所耗 CPU 時間差異並不明顯。而贪婪演算法則是因為在決定第二架無人機於第二個轉場時段的移動方式時，選擇停留不動，導致下一個轉場時段必須額外再多指派一架 UAV 至節點五以完成腳本，因此多用了一架無人機。

表 16 三種求解方法比較表

Method	Objective value	UAV used	CPU time (s)	Gap (%)	$Gap_g$ (%)
Model*	3064.41*	3	0.3493	0	0
Rolling*	3064.41*	3	0.2567	0	
Algorithm	4083.00	4	0.0179	33.2	

為了更加貼近實務應用，我們實際產生了一個簡單範例如圖 21，總共有三個畫面分別為 SHOW(展演)、UAV(無人機)、OPT(最佳化)，將此作為測試資料，產生畫面數為 3、總節點數為 84、最多節點的畫面有 53 個節點的網路圖形。

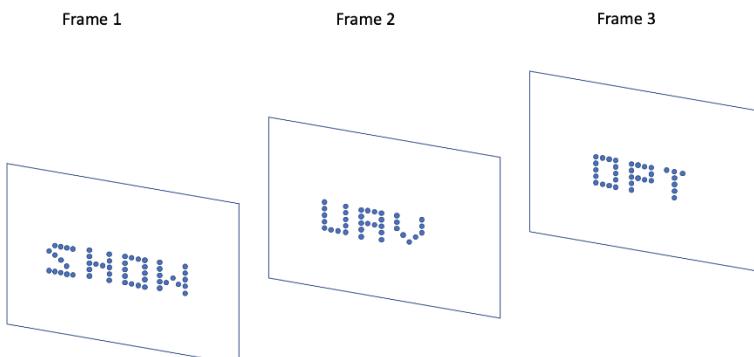


圖 21 範例圖像設計

分別測試三種不同的求解方法，將測試結果整理成表 17，其中  $Gap_g$  表示 Gurobi 求解器在一次性模式求解所回報的最佳可行解目標值與最佳目標(或下限)值的相對差距百分比， $Gap$  則表示該求解方法所得之目標值與三種方法中最佳目標值的相對差距百分比。

表 17 範例求解結果

Method	Objective value	UAV used	CPU time(S)	Gap %	$Gap_g$ %
<b>Model</b>	56164.21	55	2600.00	0.03	0.01
<b>Rolling*</b>	<b>56144.21*</b>	55	765.48	0.00	
<b>Algorithm</b>	61386.41	60	2.20	9.34	

在這個節點較多的範例中，可以很明顯地發現數學模式一次性求解不僅所耗費的 CPU 時間長，又無法保證在時限內可求得最佳解。相對來說，滾動式求解機制以及貪婪演算法所花的時間則少了非常多，且有機會在限定時間內，求解出比一次性模式更好的目標值。而這僅僅只是在 3 個畫面的展演情況下，若畫面數與節點數再增加，所耗費資源及時間將會越高是顯而易見的，這也更突顯出貪婪演算法以及滾動式求解機制的實用性。

## 4.4 小結

由於第三章所提出的數學模式，在面對規模較大的問題時不僅需要大量求解時間，求解的品質也會逐漸下降；因此，在本章節中，我們期望能夠在較短的時間內得到高品質的解，為此提出了兩種演算方法：滾動式求解機制、貪婪演算法，說明兩種方法的邏輯及其所需要的資料預處理，並在貪婪演算法中提出一種路線選擇懲罰值的計算方法；最後藉由一個小規模及一個較大規模的簡單範例展示三種方法的求解結果，記錄並比較三種作法的求解效率與效能並分析差異的可能原因。

## 第五章 數值測試與分析

本章節針對第三章提出的數學模式及第四章所提出的演算法進行數值測試與分析。本研究的測試環境為 Windows10 作業系統，Intel Core i9-9900，3.6GHz 處理器，與 32GB 記憶體，以 Python 程式語言撰寫數學模式與求解演算法，並使用 Gurobi9.0.0 版求解器，以下將分別介紹測試資料的參數設定、網路圖形設定方式、及三種求解方法數值測試的比較分析。

### 5.1 測試資料參數設定

由於在實務上，展演圖形中的每一個節點都以點座標表示。本研究預先生成一組總節點數  $N_0 = (x \times y)$  的  $R^2$  座標平面或  $N_0 = (x \times y \times z)$  的  $R^3$  座標空間如圖 22，並從中使用 python 中的 random 套件隨機生成一組低於上限數量  $ub$  之需要發光的節點集合如圖 22。

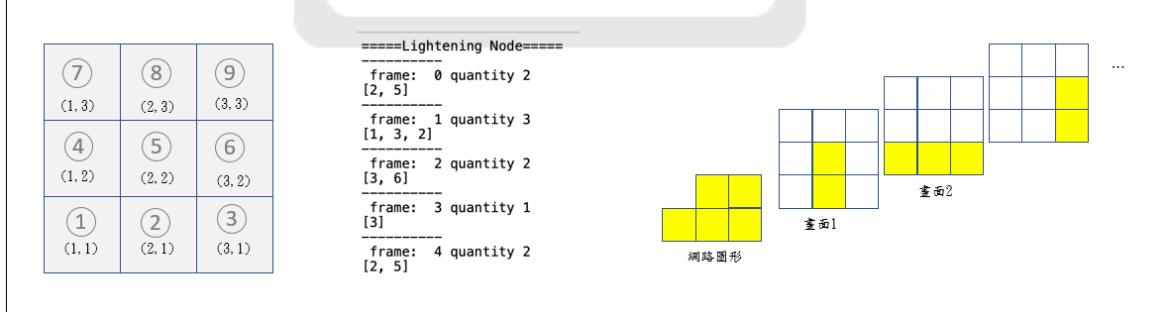


圖 22 網路生成過程

依照所設定的畫面數量  $f$  生成腳本完畢後，再將每一個畫面必須亮的節點聯集生成總節點數為  $N$  的網路圖形  $G$ ，接著應用各個點的點座標計算轉場時段

內能夠移動到的點，形成節線，本研究將最長移動距離 $l$ 設定為 $\sqrt{2}$ ，換句話說，一個轉場時段內可以移動的範圍為上下左右以及斜行至鄰近的一點；另外，針對每一條節線，計算與之短於安全距離 $d$ 的衝突節線。

除了上述的參數以外，我們還額外還設定可使用之無人機上限 $K$ ，所有參數整理於表 18。

表 18 測試資料參數設定

參數設定						
畫面 數量	平面/空間 $N_0$	網路 節點數量	亮點上限 $ub$	最長 移動距離	安全 距離	可用 無人機數
$f$	$x \times y \times z$	$N$	$\frac{1}{2}N_0$	$l$	$d$	$K$

## 5.2 測試網路圖設定

以 5.1 節當中所述的網路生成方式，生成 12 種不同規模的網路圖形，分別從  $N_0 = 3^2$ 、 $4^2$ 、 $5^2$ 、 $6^2$  等 4 類座標平面、上生成  $f = 5, 10, 15$  等 3 組  $ub = \frac{1}{2}N_0$  的展演腳本，並將其每一個畫面所應用到的點聯集以產生網路  $G$ 。將所有測資整理成表 19。

表 19 網路圖資訊

網路名稱	畫面數量 $f$	節點數量 $N$	生成空間/平面	單一畫面最多節點數
Network1	5	6	$3^2$	3
Network2	10	9	$3^2$	4
Network3	15	9	$3^2$	4
Network4	5	16	$4^2$	8
Network5	10	16	$4^2$	8
Network6	15	16	$4^2$	8
Network7	5	25	$5^2$	6
Network8	10	25	$5^2$	12
Network9	15	25	$5^2$	12
Network10	5	36	$6^2$	13
Network11	10	36	$6^2$	15
Network12	15	36	$6^2$	18

### 5.3 無人機群飛展演之最佳飛行路線規劃之測試

將所有測資之測試結果整理成表 20，其中 Solu. 欄位標示不同的求解方法類別：M 代表以整數規劃模式一次性求解、R 為滾動式求解機制、A 為貪婪演算法。而實驗結果的部分， $Gap_g$  表示 Gurobi 求解器在一次性模式求解所回報的最佳可行解目標值與最佳目標（或下限）值的相對差距百分比， $Gap$  則表示該求解方法所得之目標值與三種方法中最佳目標值的相對差距百分比，計算方式是以三種求解方法中可得之最佳目標值  $Obj_{Best}$  作為標準，以計算各求解方法  $m$  求得之目標值  $Obj_m$  與  $Obj_{Best}$  差距  $Gap$  值，其計算公式如下： $Gap = \frac{|Obj_{Best} - Obj_m|}{Obj_{Best}}$ 。

表 20 求解方法數值測試比較表

<b>Network</b>	<b>f</b>	<b>N</b>	<b>ub</b>	<b>Solu.</b>	<b>UAV used</b>	<b>CPU Time(s)</b>	<b>Best</b>	<b>Gap_g (%)</b>	<b>Gap (%)</b>	<b>objective</b>
Network 1	5	6	3	<b>M*</b>	3	0.313	✓	0.000	0	3043.41
				<b>R*</b>	3	0.539			0	3043.41
				A	4	0.019			34.16	4083.00
Network 2	10	9	4	<b>M*</b>	5	109.000	✓	0.010	0	5111.00
				R	5	14.898			3.03	5266.00
				A	7	0.038			41.44	7229.00
Network 3	15	9	4	<b>M*</b>	5	126.097	✓	0.000	0	5119.41
				R	5	24.821			16.63	5971.00
				A	6	0.061			24.62	6380.00
Network 4	5	16	8	<b>M*</b>	9	32.691	✓	0.000	0	9188.00
				R	9	23.581			4.52	9603.00
				A	9	0.091			0.26	9212.41
Network 5	10	16	8	<b>M*</b>	9	42.792	✓	0.000	0	9192.00
				R	10	25.335			19.38	10973.00
				A	10	0.155			11.84	10280.41
Network 6	15	16	8	<b>M*</b>	9	290.521	✓	0.000	0	9212.24
				R	9	147.612			22.13	11251.24
				A	11	0.394			23.35	11363.41
Network 7	5	25	6	<b>M*</b>	7	10.650	✓	0.006	0	7175.66
				R	9	20.469			36.93	9825.41
				A	11	0.078			56.50	11230.00
Network 8	10	25	12	<b>M*</b>	13	661.051	✓	0.002	0	13284.00
				R	13	288.198			5.54	14020.56
				A	13	0.535			0.73	13380.66
Network 9	15	25	12	<b>M*</b>	12	5076.341	✓	0.004	0	12300.56
				R	14	649.030			24.53	15317.80
				A	15	1.167			27.17	15642.90
Network 10	5	36	13	<b>M*</b>	14	21.385	✓	0.000	0	14299.90
				R	14	55.689			2.96	14722.66
				A	16	0.172			14.52	16376.00
Network 11	10	36	15	<b>M*</b>	16	16790.530	✓	0.010	0	16386.97
				R	17	1450.942			5.25	17246.97
				A	18	1.762			13.77	18642.90
Network 12	15	36	18	<b>M*</b>	19	15248.000	✓	51.400	0	19438.38
				R	20	1755.401			4.85	20381.38
				A	23	1.852			22.063	23727.14

從表中可以發現，在節點數量較少的測資當中，貪婪演算法 A 所求得的解，有時可以優於滾動式求解機制 R，如 Network4 以及 Network5。然而，A

的表現時好時壞，求解品質不穩定。而當網路中節點數增加時，以數學模式求解的一次性求解 M 以及滾動式求解機制 R 的求解時間明顯上升，且在 15 個畫面，節點數超過 20 個時，數學模式一次性求解 M 時間更會超過一個小時；使用滾動式求解機制 R，會比一次性機制 M 求解更快但其解品質稍差（平均 gap 約為 12.15%），也會用到更多架無人機；若使用貪婪演算法 A，大多能在 2 秒以內解完，遠比上面兩種模式求解方式快速，但其求解品質也是最差的（平均 gap 約為 22.54%）。

## 5.4 滾動式求解機制與貪婪演算法比較

近年以來，實務上的表演多為圖形複雜但畫面數較少的展出。譬如可能是單一個畫面的節點數多、甚至為三維的立體圖像，但其總畫面個數卻可能僅 5 至 10 個不等。舉例來說，2020 年 12 月 31 日台南跨年晚會中的無人機表演中，該場表演僅設計了六個畫面，然而每一個畫面中卻包含數十甚至上百架無人機；另外，如圖 23 所示為 2020 年 9 月 20 日，中國無人機編組團隊大漠大智控在珠海的表演，其中雖然僅包含四個畫面，卻同時操作 3051 架無人機飛行。

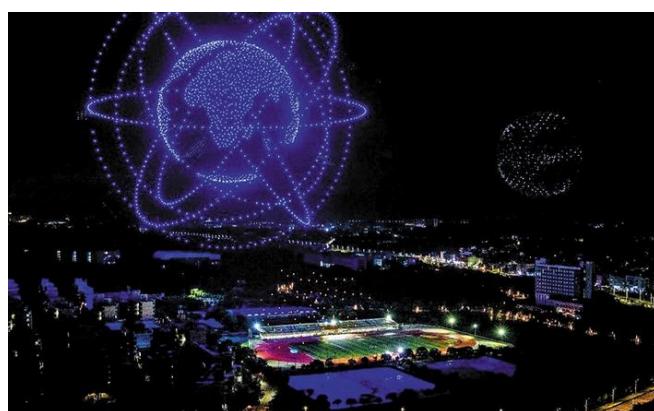


圖 23 實務表演大型案例

因此，本小節將生成較接近此類實務上多節點、少畫面數的大型測資進行實驗，如表 21 所示。

表 21 大型測試資料參數設定

網路名稱	畫面數量 $f$	節點數量 $N$	生成空間/平面	單一畫面最多節點數
Network13	5	57	$4^3$	29
Network14	10	63	$4^3$	32
Network15	5	103	$5^3$	56
Network16	5	77	$10^2$	40
Network17	5	86	$11^2$	46
Network18	5	90	$11^2$	46
Network19	5	109	$12^2$	49
Network20	5	133	$12^2$	69

表 22 滾動式與貪婪演算法求解結果比較表

Network	$f$	$N$	$ub$	Solu.	UAV used	CPU Time(s)	Best	Gap (%)	objective
Network 13	5	57	29	R11	30	351.611		0.03	30646.31
				<b>R21*</b>	30	974.48	∨	0	30637.90
				A	35	2.07		16.85	35800.24
Network 14	10	63	32	R11	34	565.43		9.14	36824.36
				<b>R21*</b>	33	1506.12	∨	0	33739.46
				A	37	7.78		13.08	38153.14
Network 15	5	103	56	R11	59	1125.84		5.271	62303.46
				<b>R21*</b>	58	3900.35	∨	0	59183.97
				A	62	9.83		7.255	63477.49
Network 16	5	77	40	R11	45	274.84		9.50	46974.31
				<b>R21*</b>	42	330.344	∨	0	42899.38
				A	48	3.52		14.58	49154.24
Network 17	5	86	46	R11	50	174.28		0.01	51069.53
				<b>R21*</b>	50	693.27	∨	0	51066.28
				A	58	3.38		16.11	59292.24
Network 18	5	90	46	R11	53	314.78		1.9	54129.28
				<b>R21*</b>	53	1017.53	∨	0	53119.94
				A	59	4.06		13.59	60338.66
Network 19	5	109	49	R11	59	290.76		13.49	60267.69
				<b>R21*</b>	52	1055.63	∨	0	53102.60
				A	65	7.16		25.28	66526.90
Network 20	5	133	69	R11	71	1018.77		2.95	72556.36
				<b>R21*</b>	69	2651.11	∨	0	70479.21
				A	80	24.14		16.39	82028.24

這類大型的問題在一次性的數學模式求解中，需要大量算力與記憶體，時常出現無法求解、或是所得的解品質太差的情形。本節中，將大型測資以兩種滾動式機制求解，分別為：(1)每次求解兩個畫面(一個轉場時段)的移動方式，且每次滾動沒有重疊的時段，以 R11 表示。(2)一次求解三個畫面(兩個轉場)的移動方式，固定第一回合轉場，並與下次滾動重疊一個轉場時段求解，以 R21 表示。最後，再與貪婪演算法一起比較，實驗結果如表 22 所示。

實驗結果觀察可知，一次求解三個畫面兩個轉場的滾動式求解機制 R21，由於其一次考慮的畫面多，即使所費時間較長，卻能得到三種方法中最好的結果；而貪婪演算法 A，依舊是三種方法中最快的，其主要缺點在於它會使用較多架無人機，但其飛行總距離反而較其它方法短，因此當租機成本並不佔總成本中較高的比例時，A 將會有不錯的成效。而實驗結果也提供給擁有資源不同的使用者，使用者可以依照不同的軟硬體資源，選擇合適的方法。

## 5.5 小結

在這個章節中，我們使用隨機套件生成 12 種節點數量、畫面數量、最多亮點數量不同的網路圖形，增加測試資料的多樣性，用以測試第三章所提出的數學模式(M)、第四章提出的滾動式求解機制(R)與貪婪演算法(A)，並從中發現貪婪演算法 A 及滾動式求解機制 R 可以有效降低求解時間並得到一個品質不錯的解增加效率；然而，若是時間充裕且有足夠算力，建議可以使用數學模式 M 求得最佳的解法。另外，若是在資源有限的情況下，為了貼近實務，我們生成 8 種大型的測資，並測試演算法以及兩種不同設定的滾動式求解機制（R11 與 R21）進行比較，建議若在時間算力不足的時候，可以依照軟硬體的能力去選擇求解方法，若時間、算力足夠可使用向後參照的滾動式求解機制；反之，

若求解時間為主要考量的話，貪婪演算法及考慮較少畫面的滾動式求解機制亦可提供質量不錯的解。



## 第六章 結論與未來研究建議

### 6.1 結論

近年來，慶典節目中的聲光表演越來越有創意，其中使用無人機群飛展演的項目，相較於傳統煙火，不僅能更直接地表現出更細緻的圖像，同時也能減少背後所帶來的高成本及環境污染議題。然而，如何能同時將多架 UAV 的移動方式妥善規劃，讓整個表演順暢進行，不只能完整呈現各種腳本的圖形設計，同時降低整個表演的總消耗電量並有效地減少成本支出，兼顧美感及環保，上述都是無人機展演規劃過程中必須處理的困難議題。

在無人機路線規劃的相關文獻研究中，雖已有部分研究有考量飛行安全距離以及電力消耗，但皆與本研究所探討的無人機群飛展演路線規劃問題無直接相關。無人機展演主要要求無人機需要在特定時刻，到達特定位置亮燈展出圖像，是一個 NP-Hard 的 LRP 問題。為了能考量無人機行進間安全距離，並以最小化成本為目標，做出多架無人機完成腳本設定畫面的最佳路線規劃決策，本研究以蔡成鴻(2020)提出的多層式時空網路數學規劃模式為基礎，並進一步將實務上安排飛行路徑時必須考量的安全距離加以限制，為相關領域中考量最完備的研究。

一次性求解之數學規劃模式因為要將所有變數全部列入規劃的考量，通常需要耗費極大的算力及記憶體等軟硬體條件，特別在求解規模較大的問題時，常會出現時限內無法求解或求解品質降低的情況。因此本研究再提出「滾動式求解機制」以及「貪婪演算法」等兩種啟發式解法。其中，滾動式求解機制將展演過程切割成數個時間區段，依序逐一處理規模較小（譬如僅規劃連續的一至三個展演畫面之對應的一至三個轉場時段飛行路線）的子問題，此時因其規模較小，可直接使用原先發展的數學規劃模式求解，再將其求解結果僅固定前

一(頂多至三)轉場時段飛行路線，再重複上述流程往後重新規劃新的子問題，依此滾動式機制依序逐一求解部分飛行路線，直至結束。而貪婪演算法則是每一次迭代僅決定一架無人機從開始到結束的每一段轉場時段的移動路線以及展演時段的位置，據以更新可跟已選擇路線保持安全距離之可行的飛行節線集合，再重複此步驟重新為新的一架無人機規劃其飛行路線，直至每一個展演畫面的亮燈位置都有無人機可以正確飛抵展演為止。

將上述的研究結果歸納出具體貢獻說明如下：

### 具體貢獻

1. 首度考量轉場時段長度與無人機速度，推導出兩條路徑間最短距離計算方式，以計算出不符合安全距離的路線組合，將此做為後續求解最佳路線的限制。
2. 以多層時空網路為架構，在考慮飛行安全距離的情況下，提出包含（與不包含）電量限制等兩種數學規劃模式，以規劃最低移動與建置總成本的無人機群飛行路線。
3. 由於一次性求解數學規劃模式(M)將消耗過多運算時間，經常無法求解較大規模的問題，因此本研究提出滾動式求解機制(R: R11 & R21)，將問題切割成數個規模較小的子問題，再用數學規劃模式求解，固定其部分求解結果，再重複此步驟依序逐一求解後續的子問題，以縮短整體求解時間，並據此求解更大規模問題。
4. 為能更快求解大規模問題，本研究亦設計貪婪演算法(A)，以最短路徑方式一次規劃一架無人機飛行路線，據此調整設定其鄰近之飛行路段懲罰值，以方便後續無人機可選取較佳之避撞路線，用於快速規劃出大規模問題的初始飛行路線。
5. 改良文獻中以飛行空域中的所有網格建構飛行網路的設定方式，僅考慮各展演畫面中各光點位置的聯集，以避免產生過多的無用節點，縮小數學規劃模式以加速求解。

6. 將本研究提出的數學規劃模式(M)、滾動式求解機制(R)、貪婪演算法(A)等三種路線規劃方法進行數值測試實驗，依不同的節點數量、展演畫面多寡，建構出 12 組展演腳本的路線規劃問題測試比較。結果顯示滾動式機制(R)的求解方式，最能兼顧效能與效率。
7. 我們再針對實務上畫面較少、每畫面較多光點的較大規模展演問題產生 8 組測資，比對兩類滾動式求解機制 R11 & R21 與貪婪演算法 A 的求解表現，發現一次求解三個畫面兩個轉場並固定第一個轉場決策的滾動式求解機制 R21 有最好的求解效能與效率表現，是本研究最後建議實務展演業者與未來研究最合適的路線規劃方法；反之，貪婪演算法 A 雖有較短的整體飛行總距離，但這是靠啟動更多架無人機才達成的，適用無人機單價便宜的情境。

## 6.2 未來研究

至目前為止，本研究將電力消耗、飛行安全距離納入考慮，為多架無人機展演最佳路線規劃問題建立數學模式，並為了加速求解，因應解決問題所需的軟硬體條件，提出滾動式求解機制以及貪婪演算法等兩種啟發式解法，並進行數值測試。然而，囿於時程較趕，尚有許多待改善之處說明於下：

### 數學規劃模式

本研究之數學規劃模式，由於限制式繁多，求解時需要花費較多時間，在求解規模大、畫面多的問題結構時，其目標函數值的收斂可能耗時太長甚至偶有無法求解的情況。因此，未來研究或可思考嘗試設計諸如縱列產生(column generation)等其它的建模方式。

## 滾動式求解機制

本研究中，使用兩種設定的滾動式求解機制，分別是(1)R21：一次求解三個畫面(兩個轉場時段)的移動方式，固定其中一個轉場時段的解，下一輪再往後移一個時間區段；以及(2)R11：每次求解僅兩個畫面(一個轉場時段)的移動方式，下一輪再往後移一個時間區段，彼此沒有重疊。然而，上述機制設定其實還有許多值得調整的可能性值得探討：

1. **每次滾動的區間範圍**：每一次滾動需要求解的畫面數，會直接影響該次中的節點數量多寡，因而造成求解時間的長短差異。而該如何將原問題分解，該分成多大的子問題，會最有效率地減少求解時間又得到品質最好的解，可以在未來多加以實驗探討。
2. **每次滾動的固定及重疊區間範圍**：在目前的研究的設定中，僅讓每次求解與前一次重疊一個時段、固定一個時段。然而，每一種不同大小、長度的問題中，可能都存在各自適合的設定方式，該如何設定才可以得到品質更接近最佳解的解，有待更多次的測試觀察。
3. **每次子問題求解時間限制**：在每次滾動的求解過程中的收斂情形都不同，可能會花了很長一段時間，但是求解的目標值仍未收斂好，這時若限制每次的求解時間，可以有效地減少不必要的時間浪費。然而，在什麼樣情形下適合怎麼樣的設定、該如何設定等，也是未來研究中值得實驗討論的。

## 啟發式路徑規劃演算法

本研究中所設計的貪婪演算法(A)，雖然可以有效降低求解時間，同時求得一個品質尚可的解。然而，這個解與數學模式所計算出的最佳解仍舊有一段差距，且通常會使用較多架無人機。期待未來也能在此基礎上做出改善：

1. **在原有貪婪演算法中建立路線檢查的機制**：在產生路徑後檢查每一個轉場時段的移動路線，是否存在選擇路線能夠降低移動距離同時又不影響路線

衝突的改變方式；或者，每次選擇的路線方法不採用最短路線，而是選擇第二短、第三短、甚至更長之路線來相互比較，或許亦有機會讓它們各自與不同的節線發生衝突，而使求解品質有所精進，不至被侷限於局部最佳解。

2. 嘗試實作其它機制的啟發式演算法：文獻中有許多使用諸如模擬退火法、蟻群最佳演算法、粒子群演算法等不同的機制來規劃路徑，目前本研究尚無想出各演算法對應的合宜路線編碼機制來求解，這應該也會是個十分值得探討的研究議題。

## 其它問題設定

無人機群飛展演路線規劃問題中所包含的議題很多，隨著實務上追求愈來愈繁複的展演圖像設計，未來研究也能夠從題目設定著手，包含：

1. 空中虛擬點的位置規劃：在現有設定中，我們將原點以及所有畫面會亮的點聯集成節點集合，在路線規劃時，若無人機所在的位置，無法在有限的轉場時刻內移動至其它需要發亮的點支援，或是因為其它無人機的路線互相牽制，導致必須停留在原處；此時，若有一些能夠讓該機先移動到的節點，使其下一回合能支援到其它更遠的亮點，進而減少需要派遣的無人機數量以降低租機成本，將對最小總成本有益，而這些點的數量應該是多少？又該在甚麼位置？才能在不浪費太多計算時間的前提下降低成本，這都是未來可以鑽研的議題。
2. 指定某一架無人機在某一時段的位置：目前的問題設定假設每一架無人機都是相同的，並不硬性規定某一個亮點要由哪一架完成，而未來可能會因為無人機的類型、功能不同，其所能營造的效果也不同，因此說不定有可能需要針對不同類型無人機的移動方式做限制。

3. 展演及轉場時段長度不同：現實展演中，由於每一個圖形的重要程度及意義不同，可能會造成每次的展演時段與轉場時段時間不相同，間接造成每次可移動時間與可移動長度不同，使無人機的衝突範圍因而改變。
4. 每一個時段皆可移動：目前我們假設無人機在展演時段時不可移動，然而，若放鬆這項設定是否可以使得無人機互相支援的能力更好，以減少無人機的使用量，亦是可加以探討的。



## 參考文獻

中文

工研院無人機講座 揭示台灣產業發展路向（民 107 年 9 月 27 日）。民 108 年 8 月 5 日，取自 <https://dronesplayer.com/drone-use/> 工研院無人機講座-揭示台灣產業發展路向

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碩士論文。取自 <http://etds.lib.ncku.edu.tw/html/>

陳冠岐（2016）。基於多目標演化式演算法的視角可調之無人機路徑規劃。

中華大學資訊工程碩士論文。取自 <https://hdl.handle.net/11296/qvcnrg>  
陳彥瑋（2020）。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。國立成功大學工業與資訊管理碩士論文。取自 <http://etds.lib.ncku.edu.tw/html/>

無人機送貨趨勢全球市場規模展望（DIGITIMES 整理），取自 <https://digitimes.com.tw/tech/showimg.asp?source=&filename=548634-1-K92NX.jpg&Sourcetype=1&newskey=548634>

黃慧雯（民 108 年 2 月 20 日）。屏東燈會 Intel 神助攻 無人機排出 TAIWAN 閃耀天空。中時電子報。民 108 年 8 月 5 日，取自 <https://www.chinatimes.com/realtimenews/20190220001470-260412>

蔡成鴻（2020）。考慮電量耗損之無人機群飛展演最佳群飛路徑規劃研究。國立成功大學工業與資訊管理在職專班碩士論文。取自 <http://etds.lib.ncku.edu.tw/html/>

劉禹慶 (民 108 年 4 月 10 日)。澎湖花火節 22 場次主打無人機表演。自由時報。民 108 年 8 月 5 日，取自  
<https://news.ltn.com.tw/news/local/paper/1280418>



## 英文

- Chen, Y.K., Wang, Y.C., Chen, X.Y., Han, J.X.(2018).Research on the Flight Path of the Drones Based on the 0-1 Linear Programming Model for the Light Show. *International Conference on Mechanical, Electronic and Information Technology*, 172-175. doi: 10.12783/dtetr/icmeit2018/23404
- He, L., Bai, P., Liang, X., Zhang, J., Wang, W.(2018).Feedback formation control of UAV swarm with multiple implicit leaders. *Aerospace Science and technology*, 72, 327-334.doi: 10.1016/j.ast.2017.11.020
- Huang, C. X., Lan, Y. S., Liu, Y. C., Zhou, W., Pei, H. B., Yang, L. Z., & Peng, Y. H. (2018). A New Dynamic Path Planning Approach for Unmanned Aerial Vehicles. *Complexity*, 17,doi:10.1155/2020/6549572.
- Luo, Z., Liu, Z., & Shi, J. (2017). A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle, *Sensors*, vol. 17, no. 5, pp. 1-17.
- Majd, A., Loni, M., Sahebi, G., & Daneshthalab, M. (2020). Improving motion safety and efficiency of intelligent autonomous swarm of drones. *Drones*, 4(3), 48, doi: 10.3390/drones4030048
- Mor, A., & Speranza, M. G. (2020). Vehicle routing problems over time: a survey. *4OR*. doi:10.1007/s10288-020-00433-2.
- Oz, I., Topcuoglu, H. R., & Ermis, M. (2013). A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles.

*Simulation-Transactions of the Society for Modeling and Simulation International*, 89(8), 903-920.

Prats, X., Delgado, L., Ramirez, J., Royo, P., & Pastor, E. (2012). Requirements, issues, and challenges for sense and avoid in unmanned aircraft systems. *Journal of aircraft*, 49(3), 677-687, doi: 10.2514/1.C031606

Ropero, F., Muñoz, P., & R-Moreno, M. D. (2019). Terra: A path planning algorithm for cooperative ugv-uav exploration. *Engineering Applications of Artificial Intelligence*, 78, 260–272.

Schneider, M., & Drexl, M. (2017). A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1-2), 389–414. doi:10.1007/s10479-017-2509-0

Sun, H., Qi, J., Wu, C., Wang, M.(2020).Path Planning for Dense Drone Formation Based on Modified Artificial Potential Field.*Proceeding of the 39<sup>th</sup> Chinese Control Conference*. 4658-4664. doi:10.23919/CCC50068.2020.9189345

Wang, X., Yadav, V., & Balakrishnan, S. N. (2007). Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Transactions on control systems technology*, 15(4), 672-679, doi: 10.1109/TCST.2007.899191

Yilmaz, O., Yakici, E., & Karatas, M. (2019). A UAV location and routing problem with spatio-temporal synchronization constraints solved by ant colony optimization. *Journal of Heuristics*, 25(4-5), 673-701.

Yu, K., Budhiraja, A. K., Buebel, S., & Tokekar, P. (2019). Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Journal of Field Robotics*, 36(3), 602-616.

