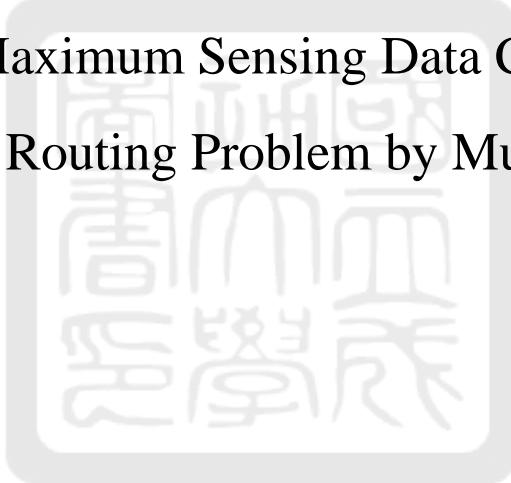


國立成功大學
資訊管理研究所
碩士論文

多台無人機最大化收集數據之
協同循環路徑規劃問題

On the Maximum Sensing Data Collection
Cyclic Path Routing Problem by Multiple UAVs



研究生 : 徐威銓
指導教授 : 王逸琳 博士

中華民國一百一十一年六月

國立成功大學

碩士論文

多台無人機最大化收集數據之協同循環路徑規劃問題

On the Maximum Sensing Data Collection Cyclic Path Routing Problem by Multiple UAVs

研究生：徐威銓

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
林立凡
周詩梵

指導教授：王逸琳

系(所)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 111 年 6 月 17 日

摘要

由於無人機可協助人類執行高難度的重複性任務，本研究探討在網路環境微弱及基站與感測站距離較遠的情況下，如何規劃多台無人機協同合作，在諸多感測站與基站間進行週期式循環繞行，定期將感測站所收集到之數據以無線傳輸接力方式，輾轉協同傳回給基站。為確保數據之時效性，本研究規範各無人機與基站的無線通訊必須在給定的循環期間內至少曾完全連通一次；在上述設定下，本研究希望能將基站接收到的感測站資料量極大化，且保持各無人機規律地循環繞行。相較於其它文獻大多僅以啟發式演算法規劃數據閒置時間極小化的單趟飛行路徑，本研究的循環繞行問題設定有更佳的資料同步時效性，可讓無人機巡邏更大的飛行範圍，更符合現實的重複性巡邏監測任務需求。

由於無線傳輸必須在通訊範圍內才能進行，如何讓無人機的循環飛行路徑能滿足在特定時刻與其它無人機及基站間能保持通訊連通，是一個極具挑戰的多載具協同循環路徑規劃問題。本研究以時空網路為基礎建構整數規劃模式，可求解出理論上最佳的協同路徑決策，但該模式之求解時間將隨維度變大而大幅增加，因此本研究也設計在貪婪演算法架構中搭配整數規劃模式為核心算法的「數學啟發式演算法」(matheuristics)，來迭代求解出最佳的無人機循環路徑。本研究一共開發了三種數學啟發式演算法：(1)「飛行範圍貪婪演算法」先使用貪婪演算法縮小原網路圖，再使用整數規劃模式求解；(2)「路徑組合貪婪演算法」使用傳統求解大型路線規劃問題常用的「列產生法」，將原來在網路上直接求解的方式改為挑出多條候選路徑，並由整數規劃模式決定那些路徑組合為最佳解；(3)「改良路徑貪婪演算法」則再進一步改良路徑組合貪婪演算法，能更有效率地挑選候選路徑，使其核心的整數規劃模式較易在短時間內解出更好的可行解。

由整數規劃模式以及演算法的求解測試結果，我們發現感測站分布較密集且離基站較近的網路較易求解；此外，小規模的網路較適合「飛行範圍貪婪演算法」求解，而大規模的網路則適合「改良路徑貪婪演算法」求解。我們亦發現當無人機的傳輸數據範圍或是連通週期達到一定上限之後，最佳解較易求得且變化不大；最後利用數值分析以及敏感度分析，可讓決策者在有限的成本下，決定無人機使用的種類以及數量，更貼近現實狀況。

關鍵詞：整數規劃、無人機、循環路徑規劃、數據閒置時間、數學啟發式演算法

On the Maximum Sensing Data Collection Cyclic Path Routing Problem by Multiple UAVs

Wei-Chuan Hsu

I-Lin Wang

Institute of Information Management, National Cheng Kung University

SUMMARY

This study investigates how to plan a cyclic path for each of multiple unmanned aerial vehicles (UAVs) to collaborate on wireless data collection periodically from sensor stations to the base station in a long-range area. In particular, a UAV flies to a sensing location to collect data and then wirelessly transfer the data to another UAV within the communication range until the data finally reaches the base station. To guarantee that the periodic amount of collected data to the base station has an acceptable latency and synchronization, we require the wireless communication between the UAVs and the base station to be fully connected at least once within the given cycle period. To this end, we propose mixed integer programming models and several effective matheuristics based on column generation techniques. The computational experiments indicate our approaches provide a good tradeoff in solution quality and efficiency and thus are suitable for practical use.

Key words: mixed integer programming, UAV, cyclic path, latency, matheuristics

INTRODUCTION

In order to protect the natural environment from pollution or over-exploitation, it is necessary to patrol and monitor the real-time status over a wide and long-range area. For example, by detecting fires or poachers in the forest, oil spills or flotsams on the sea, melting of polar ice and snow, or the flow of soil and rocks in the mountains, we could buy some time for better preparation, save more lives, or reduce more damage for the incoming or ongoing disasters. For this purpose, UAVs are the best data transmission option, since they can overcome the limitations of distance and terrain and are highly reliable and efficient. In this paper, we assume several sensor stations are deployed to collect ecological data in the area. To ensure the base station receives the ecological data from sensor stations periodically, we employ a UAV to fly along its circuit nearby a sensor station to collect data wirelessly, and then it may also transfer the data wirelessly to another UAV when they are close to each other. With careful arrangement, the data can finally be transferred to the base station via wireless transfer between UAVs. To guarantee the collected data to have an acceptable latency, i.e., sufficiently up-to-date, we require UAVs and the base station to be fully connected wirelessly at least once within the given cycle period. The major challenges to this problem come from the data synchronization requirement. In addition, so many choices of cyclic paths make the design of UAV routings difficult.

MATERIALS AND METHODS

Most related research in literature solves this problem by heuristic algorithms. In this thesis, we have designed integer programming models based on a time-space network. However, this exact optimization approach takes too much time. So we also designed three matheuristics using the greedy algorithm framework and the integer programming model as the core algorithm. In our three matheuristics, different integer programming models can be used as the core in the greedy algorithm framework to iteratively solve the best flight path. In particular, three matheuristics are developed: (1) the "Flying Range Greedy Algorithm" reduces the original network scale and then uses an integer programming model to solve it; (2) the "Path Combination Greedy Algorithm" uses the "column generation" technique that is often used in solving large route planning problems. Besides solving directly on the network, multiple candidate paths are selected to form an integer programming model, and then it determines which path combinations are the best solutions. (3) The "Improved Path Combination Greedy Algorithm" further improves the "path combination greedy algorithm" by selecting an effective set of candidate paths more efficiently, making the core integer programming model easier to get better feasible solutions in a short time.

RESULTS AND DISCUSSION

Our testings are performed on a personal computer with Windows 10, Intel Core i9-10900, 2.80GHZ Processors, and 16GB RAM. Gurobi 9.1 version is used for solving MIP. Our mathematical programming models and matheuristics are implemented in Python. From the results of the integer planning model and the matheuristics, we found networks with sensor stations with a denser distribution and closer to base stations are easier to solve. Besides, small-scale networks are suitable for the "Flying Range Greedy Algorithm," while large-scale networks are suitable for the "Improve Path Combination Greedy Algorithm." We also find that the optimal solution is easier to find and does not vary much when the UAVs' data transmission range or connectivity period reaches a specific limit. Finally, numerical and sensitivity analysis allow decision makers to decide the type and number of UAVs used at a limited cost, which is closer to the actual situation.

CONCLUSIONS

First, we propose new ways for UAVs to cooperate in transmitting data by referring to previous literature. We may arguably be the first to explore the problem of multiple UAVs collecting data together by mixed integer programming. Suppose the number of UAVs, the size of the cyclic path, the latency of the data, and the number of times each sensor station must be collected are all known. We seek the cyclic path of each UAV to maximize the total amount of data collected. Second, the sensitivity analysis of the known parameters of the mathematical model is used to test real-world problems. For example, these parameters include the types of UAVs that should be selected, the number of UAVs that should be used under cost constraints, or the types of UAVs that should be sent to perform tasks in different topologies (i.e., shape) of networks. Last, Three matheuristics are proposed to calculate good solutions in a shorter time. To shorten the overall solution time but still maintain the best solution quality, we designed and implemented three matheuristics that contain an integer programming model with a greedy framework based on column generation techniques. These matheuristics perform well for small and medium-sized networks. Among the three matheuristics, we suggest applying the flying range greedy algorithm for smaller networks, and the modified path greedy algorithm for larger networks.

誌謝

本論文得以順利完成，首先最感謝就是指導教授王逸琳老師的指導，謝謝老師在我大學時期就給我機會在老師身旁學習，讓我可以比別人有更多學習的機會與資源，在大學與研究所六年的期間，老師教導我許多知識以及做人處事的道理，無論是學術上或是生活上，恩師總是毫無保留的關懷。在此向恩師致上最誠摯的感激。Proposal 與口試期間，承蒙書審委員林清河老師、黃宇翔老師與蔡青志以及口試委員丁慶榮老師、林仁彥老師與周詩梵老師的建議與指正，使本論文更完備，感謝各位教授。

感謝書桓學長和承中學長一直以來的關心與照顧，在我兩年的研究生涯中總是主動的關心我並給予我協助，也感謝郁恩、蕎仔從碩一開始一起討論課業、準備助教課，到後面一起處理論文，都幫了我不少的忙，最後則是學弟妹彥融、凱翔、怡玲和閔樺，多謝你們在我碩二時幫我處理許多事情，讓我可以專心的撰寫論文，可以順利畢業，如果以後有問題也隨時歡迎你們來找我。祝你們的論文也能順利完成。感謝碩班這兩年有你們的陪伴，讓我的研究生涯中有許多快樂的回憶。謝謝你們！

此外，我要感謝我的家人，謝謝你們的陪伴與支持，在我遇到挫折時，因為有你們，讓我有動力可以繼續努力。還有要感謝彥童、士鈞、赫遠與明諺給予我快樂的研究生涯，謝謝你們！

最後，謹以此向所有關心我的人致上最深的謝意，並將這份成果呈獻給你們。

目錄

摘要.....	I
誌謝.....	V
表目錄.....	VIII
圖目錄.....	IX
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究範圍與限制.....	3
1.4 論文架構.....	5
第二章 文獻回顧.....	6
2.1 無人機協助收集數據問題.....	6
2.2 無人機資料連通方式相關文獻.....	7
2.3 小結.....	10
第三章 多台無人機循環路徑規劃模式.....	11
3.1 問題描述.....	11
3.2 問題假設.....	14
3.3 網路架構.....	15
3.4 數學規劃模式.....	16
3.4.1 符號定義.....	16
3.4.2 目標式.....	17
3.4.3 限制式.....	17
3.5 變數與限制式個數估算.....	21
3.6 小結.....	23
第四章 啟發式演算法.....	24
4.1 飛行範圍貪婪演算法.....	24
4.1.1 符號定義.....	26
4.1.2 目標式.....	27
4.1.3 限制式.....	27
4.2 路徑組合貪婪演算法.....	30
4.2.1 符號定義.....	35
4.2.2 目標式.....	36
4.2.3 限制式.....	36
4.3 改良路徑貪婪演算法.....	42
4.4 小結.....	45
第五章 數值分析.....	47
5.1 數學模式求解結果.....	47
5.2 飛行範圍貪婪演算法求解結果.....	55

5.3 路徑組合貪婪演算法求解結果.....	56
5.4 改良路徑貪婪演算法求解結果.....	57
5.5 感測站位置敏感度分析.....	58
第六章 結論與未來研究方向.....	62
6.1 結論與貢獻.....	62
6.2 未來研究方向.....	66
參考文獻.....	69



表目錄

表 1.1 無人機的應用	1
表 2.1 無人機相關文獻比較	9
表 3.1 估算變數與限制式個數之參數設定值	22
表 3.2 變數總個數估算	22
表 3.3 限制式總個數比較	22
表 4.1 各個模式在演算法中的功能	46
表 5.1 數學模式數值參數設定	47
表 5.2 數學模式求解結果	48
表 5.3 連通週期敏感度分析(sec)	49
表 5.4 傳輸距離敏感度分析(sec)	52
表 5.5 不同種類無人機之比較	54
表 5.6 兩種形狀網路圖比較結果	55
表 5.7 飛行範圍貪婪演算法與數學模式之比較	56
表 5.8 路徑組合貪婪演算法與數學模式之比較	57
表 5.9 改良路徑貪婪演算法與數學模式之比較	58
表 5.10 大規模網路測試結果	58
表 5.11 感測站位置敏感度分析	60

圖目錄

圖 1.1 無人機監測海面	2
圖 1.2 無人機每單位時間的飛行範圍	4
圖 1.3 無人機傳輸數據的範圍	5
圖 3.1 問題範例	13
圖 3.2 正確範例	13
圖 3.3 錯誤範例 1	14
圖 3.4 錯誤範例 2	14
圖 3.5 時空網路圖	15
圖 4.1 飛行骨幹網路圖	25
圖 4.2 飛行範圍網路圖	25
圖 4.3 路徑組合貪婪演算法流程圖	31
圖 4.4 路徑組合貪婪演算法步驟 1	32
圖 4.5 路徑組合貪婪演算法步驟 2	32
圖 4.6 路徑組合貪婪演算法步驟 3	32
圖 4.7 路徑組合貪婪演算法步驟 4	33
圖 4.8 路徑組合貪婪演算法步驟 5	33
圖 4.9 路徑與無人機步數可以分配範例	35
圖 4.10 路徑與無人機步數無法分配範例	35
圖 4.11 GM^D 會將所有候選路徑全部換新	39
圖 4.12 GM^S 會保留前一次迭代的部分候選路徑	39
圖 4.13 改良路徑貪婪演算法流程圖	44
圖 4.14 GM^N 刪除法	45
圖 5.1 連通週期敏感度分析折線圖	49
圖 5.2 $P=6$ 時無人機的飛行路線	49
圖 5.3 $P=9$ 時無人機的飛行路線	50
圖 5.4 $P=12$ 時無人機的飛行路線	50
圖 5.5 $P=6$ 時無人機的傳輸數據時刻	51
圖 5.6 $P=12$ 時無人機的傳輸數據時刻	51
圖 5.7 傳輸距離敏感度分析折線圖	52
圖 5.8 $R=1$ 時無人機的飛行路線	52
圖 5.9 $R=2$ 時無人機的飛行路線	53
圖 5.10 $R=3$ 時無人機的飛行路線	53
圖 5.11 網路 1 (左) 和網路 2 (右)	59
圖 5.12 網路 3 (左) 和網路 4 (右)	59
圖 5.13 網路 5 (左) 和網路 6 (右)	59
圖 6.1 初步初始解生成法	67

第一章 緒論

1.1 研究背景與動機

無人機最早源於第一次世界大戰之後，其廣義定義為「不需要駕駛員親自在載具上駕駛的各種遙控載具」，因其作業具有不涉及人員危險的優點，常被用來偵查、攻擊敵軍或是成為協助訓練的靶機。隨著科技的進步，無人機可應用的範圍越來越多元(如表 1.1)，其主要的優勢如下：(1)可靠度高、(2)效率性高、(3)安全性高。

表 1.1 無人機的應用

消防防救災應用	災難現場勘查、救援物資運送
建築建設	建築工地巡邏、檢查建築結構
影視娛樂	燈光秀、電影拍攝
運送物流	投擲信件、包裹配送
環境監測	偵測氣候、偵測環境險峻地區

本研究所探討的無人機應用著重於對環境資訊的監測，為了保護自然環境免於污染或過度開發，必須要時時監測其環境的即時狀態，譬如森林有否火苗或是盜獵者跡象、海面有無漏油或特殊漂流物、極地冰雪融化或山區土石流動情況等等，這些區域需要被監測的範圍不小、亦需持續一段不短的時間長度、且難以利用人工的方式去收集環境數據，此時可克服距離與地形的限制、兼具高可靠度與效率的無人機即為最佳的資料傳遞選項。

為了收集某廣闊區域的環境數據，通常會先在該區域部署數個感測站，這些感測站會獨立持續收集其自身的環境數據，但礙於感測站彼此以及與基站間的距離太遠，難以直接使用通訊網路傳遞數據資料。針對上述情境，我們必須將無人機飛行至感測站，以收集感測站的數據資料，再飛至合適地點將這些資料再依無線方式傳至其它更接近基站的無人機，如此即可以接力方式將分散各處的感測站資料藉由無人機之間的無線傳輸方式在全部無人機連通時傳遞回基站。

此外，上述的區域環境數據的收集工作通常需要持續一段足夠久的時間，亦即必須重複拜訪感測站以持續收集更新的數據。要如何讓無人機長時間地重複收集各感測站的環境數據，亦是本研究與過往研究不同的一項主要設定差別。回顧相關文獻，我們發現先前相關研究大多僅探討如何以單趟的飛行路徑將所有數據帶回基站，因此無法處理實務上需要持續收集數據的重複性任務需求。

在使用無人機收集感測資料的相關文獻中，大多會設定數據時效限制，以「基站接收到收集數據的間隔時間」（稱之為 latency）以及「感測站被收集數據的間隔時間」（稱之為 idleness）等兩個參數來代表數據的時效性，通常我們會希望該二參數不能太大，否則所收集的數據會因不夠即時而降低其可用性。本論文將設定數據時效性的上限，規範所有無人機必須在時效性上限的區間內，將數據傳回基站至少一次；為讓這些資料更具時效性，本論文更進一步規範全部無人機必須週期性地與基站完全連通，才能讓數據在保持時效性的狀態下被傳輸回基站。綜合以上諸多考量，可知本研究探討的無人機路徑規劃問題不但符合實務價值，也相當有挑戰性。



圖 1.1 無人機監測海面

1.2 研究目的

鑑於上述之動機，本研究將研究無人機路徑規劃的方法，為了使無人機可以持

續地重複在同一區域中收集數據，本論文對每台無人機規劃循環路徑，讓無人機在各自的循環路徑上收集數據，以達到持續收集數據的目的。而所規劃的每個循環路徑都必須符合週期性連通的要求，以確保數據的時效性。因此我們的數學規劃模式將以最大化無人機從感測站收集的數據量為目標式，而循環路徑及週期性連通則為限制式。

由於每架無人機的循環路徑與組合最佳化的「旅行推銷員問題」(Traveling Salesman Problem, TSP)或「車輛途程問題」(Vehicle Routing Problem, VRP)十分類似，因此我們會從 TSP 或 VRP 的整數規劃模式來設計我們的數學規劃模式，我們建立了一個完整的數學規劃模式用來找出每台無人機的循環路徑，符合週期性連通(connected)且最大化無人機從感測站收集的數據量。該模式是一個線性的混整數規劃數學模式(Mixed Integer Program, MIP)，並使用最先進的最佳化處理器(GUROBI)來求解。由於本問題較 TSP 更為困難，可預期所提出的 MIP 會隨著問題規模變大而使其求解時間大幅增長，因此除了數學規劃模式，本論文也會將設計數種啟發式演算法來求解。

本研究的目的是要設計多台無人機的路徑規劃數學模式以及啟發式演算法，目標是建立數學模式以及演算法，最大化無人機從感測站收集的數據量。本研究的數學模式以及啟發式演算法預計有以下的功能：

1. 決定每台無人機的循環路徑
2. 決定哪些時刻所有無人機與基站連通並傳輸數據回基站
3. 決定數據傳回基站時，無人機與基站連通所形成的連通網路圖
4. 最大化無人機從感測站收集的數據量

1.3 研究範圍與限制

本研究擬針對無人機群之循環路徑規劃問題設計數學模式與啟發式演算法，在使用者輸入無人機數量、與基站的連通週期(數據時效性上限)、感測站數量和位置，根據這些條件為每台無人機規劃最佳循環路徑。本研究的研究範圍與限制如下：

1. 本研究的網路圖為網格狀的網路圖。
2. 每台無人機的起點未知，但已知各台無人機之循環週期（單位時間）。
3. 每台無人機必須符合週期性連通的限制，亦即必須讓所有無人機在特定時刻同時與基站連通，以傳輸所有無人機自身收集的數據。
4. 本研究參考 Scherer & Rinner (2017, 2020a) 與 Hollinger & Singh (2010) 的測資設定，將無人機每單位時間可自該節點移動自其上下左右與斜角共 8 個鄰近節點（皆當成一個 hop），飛行範圍如圖 1.2 所示。依此設定，當無人機的傳輸距離為 2 hops 時，無人機的傳輸數據的範圍如圖 1.3。
5. 感測站的數據必須靠無人機飛至該節點方能收集，無人機不能在感測站附近節點以遠端方式收集數據；然而，無人機之間只要在通訊範圍內即可遠端互相傳輸數據。
6. 傳輸數據的時間是為 0。
7. 本研究先不考慮無人機飛行的電量限制，或可先當作我們可在無人機路徑中選一點架設充換電站，如此每走完一次循環路徑便可同時完成充換電，故各台無人機的電量不受限制。
8. 感測站無法累積收集的數據，但無人機可以累積所收集的數據，因此我們才有必要收集感測站上的數據，再將之回傳至基站。

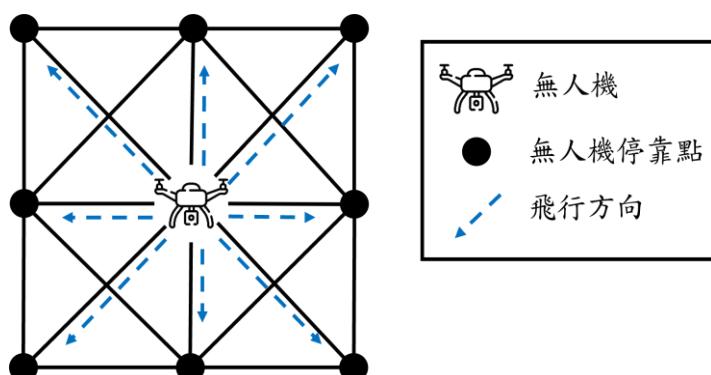


圖 1.2 無人機每單位時間的飛行範圍

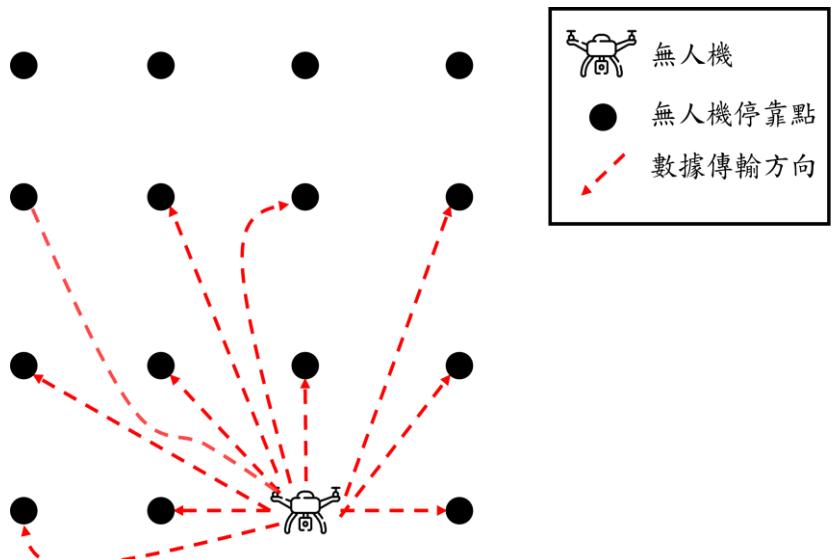


圖 1.3 無人機傳輸數據的範圍

1.4 論文架構

本論文的架構如下：第一章介紹研究背景、動機、目的以及研究範圍與限制；第二章回顧多台無人機合作傳輸數據的相關文獻，並探討過往文獻與本研究之間的差異；第三章針對無人機群之循環路徑規劃問題提出線性混整數規劃模式，求出符合週期性限制的循環路徑並最大化無人機從感測站收集的數據量並進行數值分析；第四章為數學模式數值分析與發展啟發式演算法；第五章中比較數學模式求解與啟發式演算法求解之效率；第六章中總結本論文的內容與貢獻，並提出未來可研究的方向。

第二章 文獻回顧

本章中，分為兩小節，首先探討無人機協助收集數據問題的相關文獻，第二節為無人機連通限制的相關文獻，探討各類文獻中回傳數據至基站的方式，在本小節中也會提到一些無人機監測的文獻，因為無人機監測問題的概念與本研究相似，故將該類文獻放入本小節中一同探討，最後一節為文獻回顧的小結。

2.1 無人機協助收集數據問題

在過往的文獻中，每個文獻對於收集數據問題的設定與研究方法都不太一樣，但其主要目的都是為了替無人機做路徑規劃問題。首先，Alemayehu & Kim (2017)的研究提出在一區域中有許多感測站收集數據，各感測站之間會互相協助，將收集到的數據藉由接力的方式將數據傳輸回基站。由於這些感測站有電量限制，且傳輸數據以及收集數據都需要耗費感測站電量，故派出一台無人機協助收集所有感測站的數據並帶回基站，藉此延長各個感測站的電量壽命。因為數據改由無人機從各感測站收集並帶回基站，其數據延遲性可能會大幅增加，該研究的目標為找出無人機收集數據的路徑，並盡可能地降低數據延遲性；假設無人機從基站出發，需要通過各感測站並將數據帶回基站，且最小化數據延遲時間(讓無人機回到基站的時間越短越好)，其問題與 TSP 問題類似，故該研究利用演算法來處理 TSP 問題，試圖求出無人機收集數據的路徑且該路徑必須讓數據回到基站的延遲性最小。Zhan, Zeng, & Zhang (2017)的研究中，同樣是設定每個感測站本身有電量限制，為了使每個感測站的電量能以最有效率的方式被使用，該研究同樣用無人機去協助收集感測站的數據；該研究假設感測站只有在無人機距離足夠近時，才會啟動傳輸數據的模式，若沒有無人機在感測站附近時，該感測站則會進入只收集數據的省電模式。因此，該研究除了無人機收集數據的路線規劃外，還要同時顧慮每個感測站切換模式的時刻，以使得每個感測站的電量更有效率的被使用。Vera-Amaro, Rivero-Ángeles, &

Luviano-Juárez (2020)的研究中，各感測站會被分群，接者每個群中選擇一個感測站作為暫時收集數據站，該群的所有感測站會將本身收集的數據傳輸到收集站，無人機只要經由所有收集站將數據帶回基站即可；該研究同時規劃感測站分群問題、收集站選址問題和無人機路徑問題，不但可以提升整體收集數據的效率，也能使得無人機與感測站更省電。

本研究則假設無人機與感測站電量皆無虞，此乃因我們假設每台無人機的續航力皆足以負荷一個循環路徑，如此若在該路徑終點設立充換電站即可忽略無人機電量耗費的問題；此外，本研究假設感測站(sensing location; sensor station)的電量無虞，不用為其考量電力耗費，這點跟文獻中大多是收集電量較小的感測器(sensor)資料有所不同。表 2.1 為本研究與以上這些文獻之差異。

2.2 無人機資料連通方式相關文獻

本小節探討其它文獻中，無人機之間的資料連通方式與連通週期之差異，不同的連通方式以及連通週期，所有達成的目標不盡相同。無人機收集數據的文獻較少，但其與無人機巡邏問題相關，而 Scherer & Rinner (2019)指出無人機巡邏問題可以被分種兩種方式處理：一是找出每台無人機在該區域中的循環路徑，另一個則是控制以及協調無人機在給定的路徑移動。本論文與循環路徑規劃的問題相關，故將探討一些使用循環路徑規劃方式來研究無人機巡邏問題的相關文獻。

首先，Acevedo et al. (2013) 探討如何規劃多台無人機的循環巡邏路徑，藉此提高無人機巡邏的效率，將一個大型區域劃分為許多不重疊的子區域，接著派出無人機在每個子區域中規劃各自的巡邏路徑；每個子區域的循環路徑大小相同，接著將相鄰子區域的出發點位置都設置為互相對稱(即相隔 180 度)，利用這樣的設計，便能使得無人機之間傳輸數據更有效率。該研究提出了許多演算法來協助子區域的劃分，以及各子區域的循環路徑規劃。在 Flores-Luyo, Agra, Figueiredo, & Ocana (2020)的研究中，利用多台無人機加上無線傳輸數據的技術，協助將各感測站收集

的數據傳回基站，試圖在規劃期限 T 結束時，使得各感測站所留下尚未被收集的數據越少越好。該研究提出了三種數學規劃模式，分別以不同角度來規劃數學模式，第一種為時空網路的建模方式；第二種是將無人機可以觸及以及傳輸的點當成是兩種不同的決策，讓模式規劃在每個時刻無人機該做出何種決策；最後一種則是以無人機停留於節點做主體，以無人機停留於節點時的下一步決策來建構數學模式。該研究比較了這三種不同的建模方式並比較其效率，利用數學模式求出每台無人機收集數據的最佳路徑。

Pasqualetti, Durham, & Bullo (2012)將巡邏區域以不同的優先度分級，並規劃多台無人機合作將監控各個巡邏點，使得每個巡邏點的沒被巡邏的間隔時間加總最小化，利用演算法規劃無人機循環的巡邏路徑，以達到提升整體巡邏效率的目的。Banfi, Basilico, & Amigoni (2015)使用混合整數線性規劃模式和演算法來找出每台無人機的巡邏路徑，且可以同時最小化延遲時間或是閒置時間，該研究與其它文獻不同之處在於他們利用既有的通訊網路（而非藉由無人機之間組成的動態網路）將數據傳回基站。

在 Scherer & Rinner (2017)的研究中，利用無人機隨時將收集到的感測站數據回傳給基站，因此一直保持感測站與基站的連通關係，使用這種作法使得基站無時無刻都可以收到無人機所收集的數據。該研究亦考慮無人機的電量限制，使得無人機必須在電量耗盡之前返回基站充電。該研究提出演算法來最小化各個感測站被收集數據的間隔時間，並試圖找出每台無人機飛行的最短路徑，此種合作傳遞數據的方式數據延遲時間變為 0，還可以使得無人機的搜索範圍增加。此外，Scherer & Rinner (2020a)探討了許多無人機傳輸數據的問題，這些問題都與無人機永久與基站連通相關，該研究證明這些問題皆屬 NP-hard，並嘗試利用演算法來處理這些問題。接著，Scherer & Rinner (2020b)則提出與 Acevedo et al. (2013)的類似作法，只是該研究中的循環路徑大小不同。該研究必須在各個循環路徑中，找出傳輸數據的位置，並確保所有傳輸數據的位置能與基站形成一棵連通的樹。而 Scherer & Rinner

(2020c)則利用各台無人機儲存數據與傳輸數據的特性，將數據收集回基站，該研究利用演算法強制所有數據在被接收到無人機上時，必須在規定的時間內被傳輸回基站，並盡可能地讓整體任務執行的時間越短越好。Hollinger & Singh (2010)假設無人機與站必須為週期性連通，其研究設定一個固定週期，並且提前利用週期計算好每個要與基站連通的時刻，接著給予每個連通的時刻一個固定的無人機群與基站的連通圖形。該研究提供了演算法從圖與圖進行轉換之間找出所有無人機的最短移動路徑。Spirin et al. (2014)在其研究中除了考慮了數據的週期性限制，也將一些自然環境的突發狀況加入考量，不同的突發狀況有可能使得每台無人機連通的方式必須改變。Manyam, Rasmussen, Casbeer, Kalyanam, & Manickam (2017)的研究同樣是持續地收集數據，限制了每個感測站的重複拜訪間隔，利用數學模式以及演算法來規劃每台無人機的路徑。Kantaros & Zavlanos (2016)則給定所有無人機預設好的路線，利用演算法讓無人機在其各自的路線上合作傳輸數據，以使整體完成任務的時間越短越好。

表 2.1 無人機相關文獻比較

作者	電量 限制	時間 軸	連通方式	數學 模式	演算 法
Alemayehu & Kim (2017)	V	無限	無人機儲存		V
Zhan et al. (2017)	V	無限	無人機儲存		V
Vera-Amaro et al. (2020)	V	無限	無人機儲存		V
Acevedo et al. (2013)		無限	無人機週期 連通		V
Flores-Luyo et al. (2020)		無限	無人機儲存 與接力	V	
Pasqualetti et al. (2012)		無限	無人機儲存 與接力		V
Banfi et al. (2015)		無限	網路接力	V	V
Scherer & Rinner (2017)	V	無限	無人機永久 連通		V
Scherer & Rinner (2020a)		無限	無人機永久 連通		V
Scherer & Rinner (2020b)		無限	無人機週期 連通		V
Scherer & Rinner (2020c)		無限	無人機儲存		V

			與接力		
Hollinger & Singh (2010)	有限	無人機週期 連通		V	
Manyam et al. (2017)	無限	無人機儲存	V		
Kantaros & Zavlanos (2016)	無限	無人機儲存 與接力		V	
Spirin et al. (2014)	無限	無人機儲存 與接力		V	
本研究	無限	無人機週期 連通	V	V	

2.3 小結

本章回顧了過往研究相關議題的文獻，在過往的文獻中，大部分都只提出了演算法方面的研究方法來處理相關問題，鮮少有文獻提出數學規劃模式，即使提出了數學模式，其問題也相較現實簡化了許多。為了能找出最佳解，本研究嘗試對更貼近現實的問題，設計數學模式以及演算法，期望本研究的共成果可以為無人機收集數據的領域做出貢獻，下一章本研究將針對多無人機循環路徑規劃的問題，提出其對應的數學模式。

第三章 多台無人機循環路徑規劃模式

本章探討多台無人機循環路徑規劃模式，3.1 節將描述研究問題。3.1 節及 3.3 節將詳細說明本研究之假設與網路架構，為了方便數學模式的建立，我們將原始問題之網路圖架構轉成以時空網路為架構的整數規劃模式，並提出一個整數規劃模式。3.4 節將說明模式之參數、變數、目標式及限制式。

3.1 問題描述

在多台無人機循環路徑規劃問題中，我們假設已經事先知道基站與各個感測站的位置，也知道每台無人機循環路徑的大小。假設有一無向網格網路圖 $G = (N, A)$ ， N 為所有節點集合， A 為所有節線的集合，在無向網路圖上存在 K 台無人機與 n 個感測站，每台無人機 k 皆有被指派其循環路徑的長度 W_k ，目標找出一組收集所有感測站數據並在規定週期 P 內將數據傳回基站的路徑，且最大化無人機從感測站收集的數據量。在此研究中無人機可以在網路中移動、停留或是傳輸數據，所以在集合 A 中的節線大致上可以分為以下三種：

1. 無人機飛行節線：無人機可以在這些節線上移動。
2. 數據傳輸節線：無人機可以用這些節線傳輸數據。
3. 無人機停留節線：無人機利用這些節線停留在某處。

為了更清楚解釋題目，我們將以圖 3.1 展示問題，圖 3.2、圖 3.3 以及圖 3.4 展示正確以及錯誤的範例。在圖 3.1 中，我們假設一共兩台無人機 ($K = 2$)，三個感測站 ($n = 3$)、連通週期 ($P = 6$)、傳輸距離 ($R = 1$) 以及兩台無人機的循環路徑長度 ($W_1 = 4, W_2 = 8$)，在圖中有許多非感測站且非基站的節點，這些節點為無人機可能停留的位置。本研究需要在如圖 3.1 的題目之中，找出兩台無人機的循環路徑，並檢查那些循環路徑是否符合題目要求，除了在圖 3.1 中找出適合的循環路徑相當困難，本研究還要同時決定無人機的起訖點，即使是正確的循環路徑形狀，但只要起

訖點不同，都會影響答案，該情況會在圖 3.3 中詳述。接著，我們先觀察圖 3.2，圖 3.2 中有兩台無人機，左邊的無人機每 4 步為一循環路徑，而右邊的無人機則是每 8 步為一循環路徑，由圖中的紅色圓圈數字可以得知兩台無人機的循環路徑的形狀以及起訖點位置。兩台無人機都有各自的循環路徑且符合自身循環路徑的長度，接著判斷這樣的循環路徑形狀與走法順序是否符合題目要求。我們將兩台無人機的走法比較一下，左邊的無人機為($A \rightarrow B \rightarrow C \rightarrow D$)，右邊的無人機為($F \rightarrow G \rightarrow F \rightarrow E \rightarrow H \rightarrow I \rightarrow H \rightarrow E$)，從圖 3.2 中可以看出，兩台無人機會在節點 E 跟節點 D 停靠時將數據傳回基站，由兩者的走法順序可以發現，每 4 秒的時候兩台無人機會分別抵達節點 D 與節點 E 並傳送數據，比題目要求的每 6 秒至少將數據傳回基站還短。因此，在此範例中，兩台無人機的循環路徑形狀與走法順序都符合題目要求。接著觀察圖 3.3，其兩台無人機的循環路徑形狀與圖 3.2 相同，但走法順序不同，左邊無人機順序同樣為($A \rightarrow B \rightarrow C \rightarrow D$)，而右邊無人機的順序改為($G \rightarrow F \rightarrow E \rightarrow H \rightarrow I \rightarrow H \rightarrow E \rightarrow F$)。在圖 3.3 中，兩台無人機也必須在節點 D 跟節點 E 才能傳輸數據回基站，但因為走法順序的改變，使得兩台無人機永遠無法同時停在節點 D 跟節點 E。因此，圖 3.3 結果顯示即使循環路徑形狀與正確範例相同，但因為走法順序的不同，可能導致該循環路徑便成了錯誤的路徑。最後我們觀察圖 3.4，其左邊的無人機保持者相同的走法($A \rightarrow B \rightarrow C \rightarrow D$)，而右邊的無人機則選了另一種形狀的循環路徑($F \rightarrow G \rightarrow H \rightarrow I \rightarrow J \rightarrow K \rightarrow L \rightarrow E$)，觀察此兩路徑發現兩台無人機必須要每 8 秒才會分別停靠節點 D 與節點 E，並傳輸數據回基站，比題目要求的每 6 秒至少將數據傳回基站還長，故此範例也是錯誤的。從上述這些範例，我們可以了解本研究的問題以及預期目標，也能更了解本研究的困難之處。

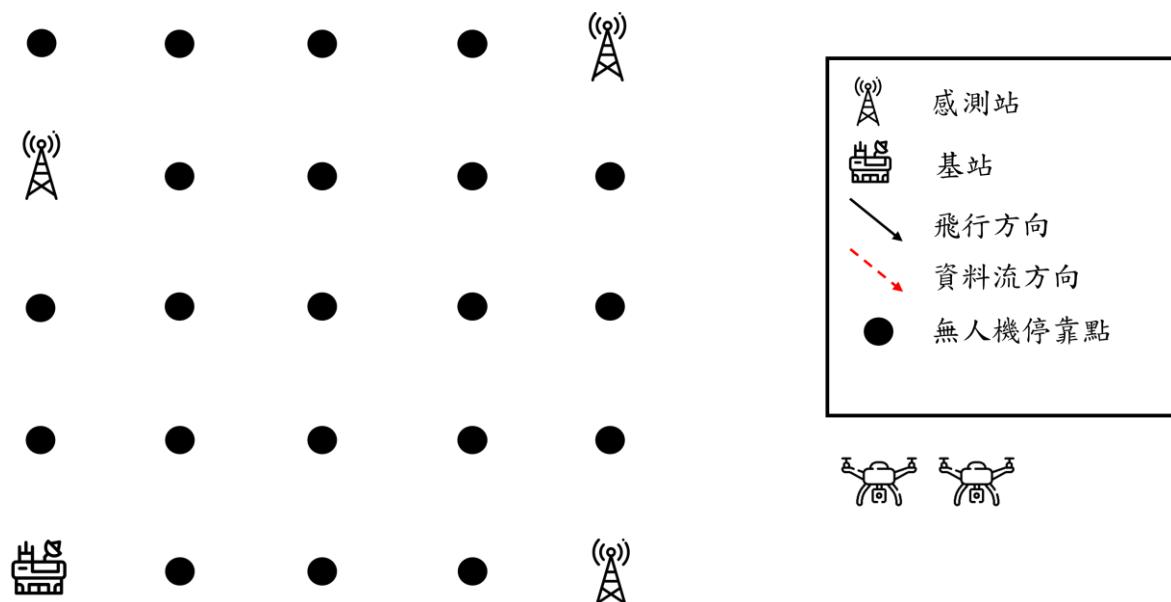


圖 3.1 問題範例

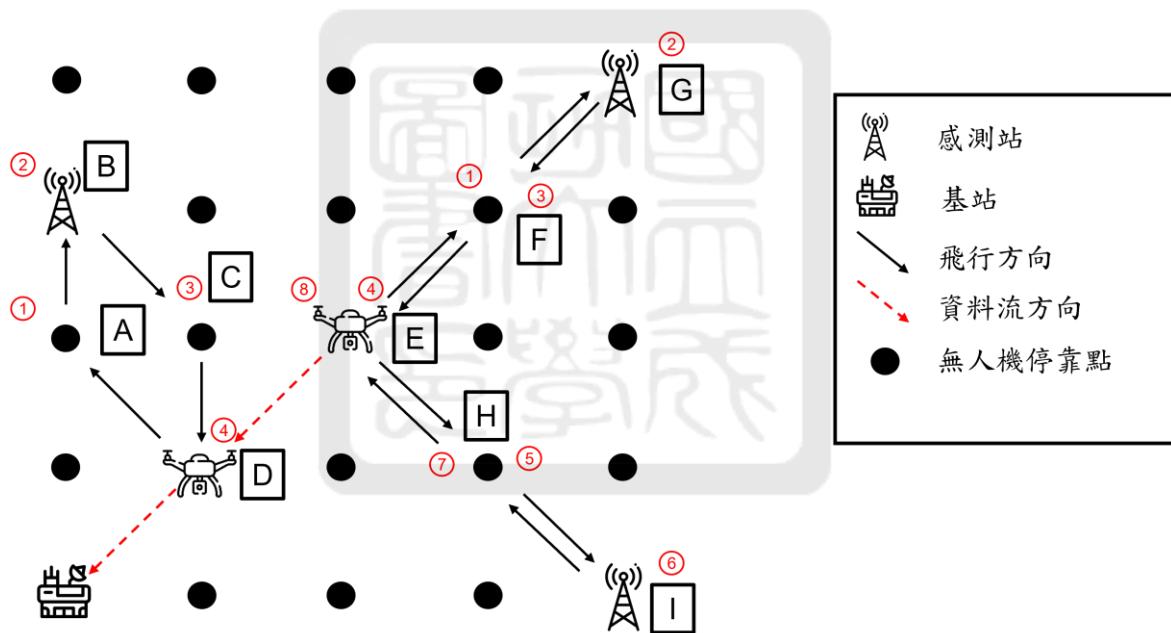


圖 3.2 正確範例

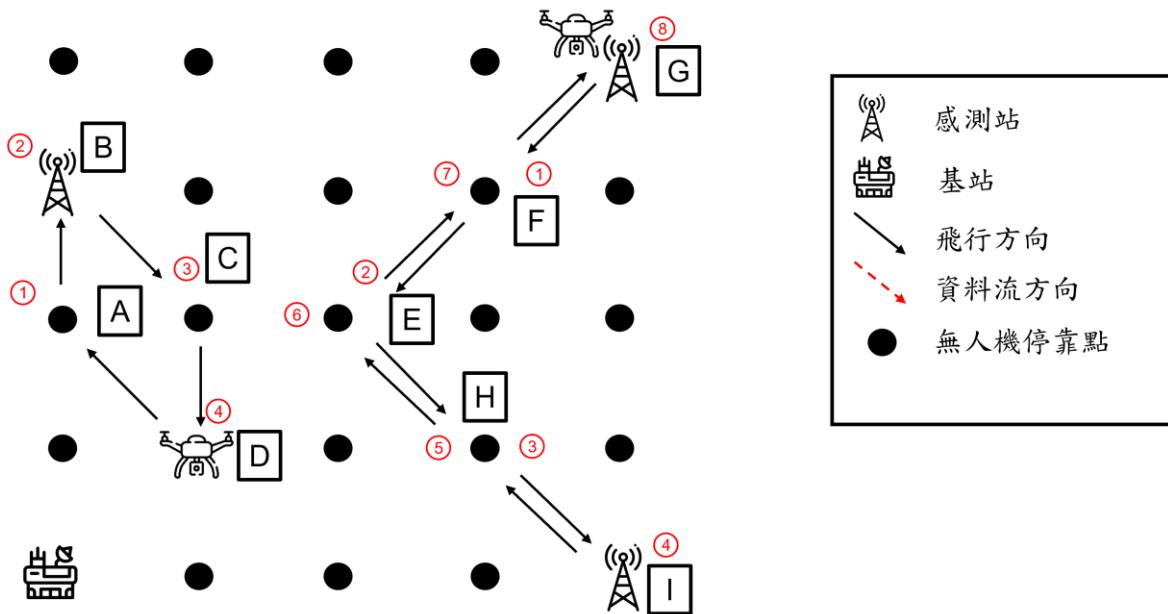


圖 3.3 錯誤範例 1

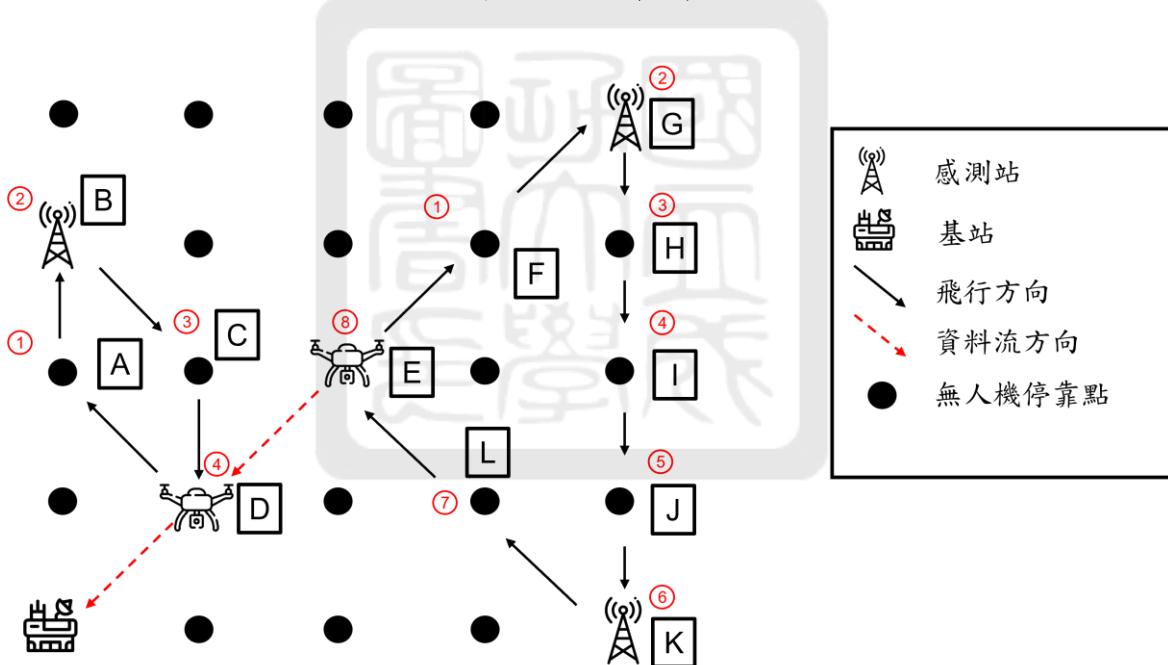


圖 3.4 錯誤範例 2

3.2 問題假設

為了適當簡化問題，本研究有以下基本假設：

1. 每台無人機的循環路徑長度是已知數值。
2. 循環路徑的起點終點必須相同。

3. 相同感測站的數據可由不同無人機收集。
4. 任何節點都可以是起點。
5. 無人機的數據承載量永遠足夠。
6. 在週期 P 之中所有無人機至少要與基站連結一次，且必須在相同時刻。
7. 循環路徑的長度可視為該台無人機的電量限制，只有在循環路徑的終點才會更換電池，更換電池的動作不消耗時間。
8. 無人機每單位時間可以朝八個方向移動或是停留在原地，雖然斜向移動距離較遠，但在本研究視為垂直和水平移動有相同的時耗。

3.3 網路架構

本研究所使用的無人機收集數據區域為網格網路圖(grid network)，將時間的概念加入網路圖裡，使用時空網路架構能使數學模式易於表達無人機的移動路徑。如此便有利於描述無人機每個時刻的位置，也能幫助我們判斷與基站的連通情形。如圖 3.5 所示，圖中紅色箭頭以及藍色箭頭所構築的路徑皆為無人機的路徑；此外，本時空網路圖只有設立虛擬起點，並無設立虛擬訖點，這是因為我們的無人機是持續在循環路徑上運行的，若將其連入虛擬訖點，則會造成其看起來像被打斷而非持續運行。

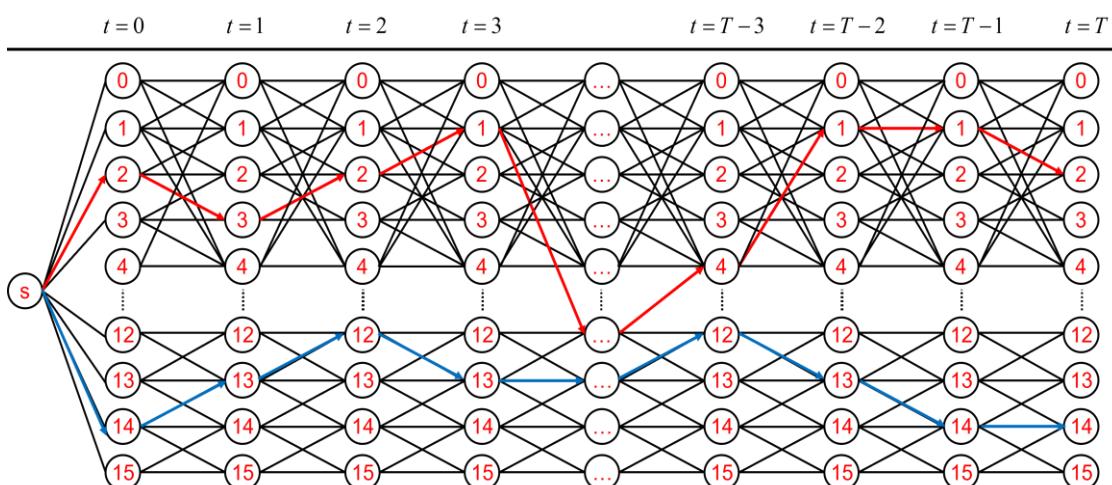


圖 3.5 時空網路圖

3.4 數學規劃模式

3.4.1 小節將介紹此次數學模式所用到之符號定義，3.4.2 小節說明目標式設定，最後 3.4.3 小節說明限制式。

3.4.1 符號定義

集合

N 所有節點之集合

N_s 感測站之集合， $N_s \subseteq N$

A 所有節線之集合

A_{GM} 所有飛行節線之集合， $A_{GM} \subseteq A$

A_{GC} 所有傳輸數據節線之集合

W 為無人機循環路徑長度之集合，一單位長度需要耗時一單位時間，因此也是無人機的時空網路的長度範圍之集合

參數

T 最大單位時間數(應為所有無人機循環路徑長度之最小公倍數)

M 極大的數

K 無人機數量

P 連通週期

R 無人機傳輸數據最長距離

變數

- x_{ijk}^t 若無人機 $k \in K$ 在時刻 $t \in W_k$ 經過飛行節線 $(i, j) \in A_{GM}$ 為 1；反之為 0
- y_{ijk}^t 若無人機 $k \in K$ 在時刻 $t \in T$ 經過傳輸數據節線 $(i, j) \in A_{GM}$ 為 1；反之為 0
- u_{ijk}^t 若無人機 $k \in K$ 的數據在時刻 $t \in T$ 經過傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0
- w_{ij}^t 若在時刻 $t \in T$ 可以被使用傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0
- z_n^t 若在時刻 $t \in T$ 無人機所在的節點 $n \in N$ 為 1；反之為 0
- p^t 若在時刻 $t \in T$ 所有無人機需要與基站連通為 1；反之為 0

$MinC$ 所有感測站被收集數據次數的下限

3.4.2目標式

本研究的目標式為最大化無人機從感測站收集的數據量，並要符合無人機循環路徑的特性以及與基站週期性連通的限制。

$$\text{Maximize } MinC \quad (3.4.1)$$

3.4.3限制式

本研究考量無人機之循環路徑規劃，無人機會持續在感測站之間移動，協助基站收集數據。

流量守恆限制式：

限制式(3.4.2)為描述無人機 k 在每個時刻 ($1 \leq t \leq W_k$) 只能通過最多一條 arc 。

$$\sum_{(i,j) \in A_{GM}} x_{ijk}^t \leq 1 \quad \forall k \in K, 1 \leq t \leq W_k \quad (3.4.2)$$

限制式(3.4.3)為每個感測站都至少會被收集數據一次。

$$\sum_{k \in K} \sum_{1 \leq t \leq W_k} \sum_{(i,j) \in A_{GM}} x_{ijk}^t \geq 1 \quad \forall i \in N_S \quad (3.4.3)$$

限制式(3.4.4)為表示無人機 k 每個時刻 ($1 \leq t \leq W_k$) 之間必須對每個節點都會保持流量守恆。

$$\sum_{(i,j) \in A_{GM}} x_{ijk}^t = \sum_{(i',j') \in A_{GM}} x_{i'j'k}^{t+1} \quad \forall k \in K, 1 \leq t < W_k, j, i' \in N, j = i' \quad (3.4.4)$$

循環路徑限制式：

限制式(3.4.5)為限制每台無人機個別的出發點以及終點要在同一個節點上。

$$\sum_{(i,j) \in A_{GM}} x_{ijk}^{W_k} = \sum_{(i',j') \in A_{GM}} x_{i'j'k}^1 \quad \forall k \in K, j, i' \in N, j = i' \quad (3.4.5)$$

限制式(3.4.6)為每台無人機循環路徑的步數必須要等於一開始給定的 W 集合。

$$\sum_{t \in W_k} \sum_{(i,j) \in A_{GM}} x_{ijk}^t = W_k \quad \forall k \in K \quad (3.4.6)$$

決策變數關係限制式：

限制式(3.4.7)(3.4.8)(3.4.9)(3.4.10)為 x_{ijk}^t 與 y_{ijk}^t 的關係限制式， x_{ijk}^t 代表無人機 $k \in K$ 在時刻 $t \in W_k$ 經過飛行節線 $(i,j) \in A_{GM}$ 為 1；反之為 0，而 y_{ijk}^t 無人機 $k \in K$ 在時刻 $t \in T$ 經過飛行節線 $(i,j) \in A_{GM}$ 為 1；反之為 0。因為 T 為 W 集合的最小公倍數，所以每台無人機的循環路徑會在整體時間軸上循環多次。在(3.4.7)(3.4.9)限制式中， M 是一個夠大的整數，其對限制式的約束力受的 x_{ijk}^t 值影響，若 x_{ijk}^t 和 y_{ijk}^t 為 1，將使限制式(3.4.7)(3.4.9)強迫要將相對應的 y_{ijk}^t 和 $y_{ijk}^{t'}$ 設定為 1；反之若 x_{ijk}^t 和 y_{ijk}^t 為 0，則限制式(3.4.7)(3.4.9)將沒有任何影響；利用這樣的特性，就可以將 x_{ijk}^t 決策變數的值放入全部 y_{ijk}^t 決策變數之中。

$$x_{ijk}^t \leq M \cdot y_{ijk}^t \quad \forall k \in K, 1 \leq t \leq W_k, (i, j) \in A_{GM} \quad (3.4.7)$$

$$x_{ijk}^t \geq y_{ijk}^t \quad \forall k \in K, 1 \leq t \leq W_k, (i, j) \in A_{GM} \quad (3.4.8)$$

$$y_{ijk}^t \leq M \cdot y_{ijk}^{t'} \quad \forall k \in K, 1 \leq t \leq W_k, W_k < t' \leq T, (i, j) \in A_{GM} \quad (3.4.9)$$

$$y_{ijk}^t \geq y_{ijk}^{t'} \quad \forall k \in K, 1 \leq t \leq W_k, W_k < t' \leq T, (i, j) \in A_{GM} \quad (3.4.10)$$

無人機停靠限制式：

限制式(3.4.11)(3.4.12)為 y_{ijk}^t 和 z_j^t 的關係限制式，利用 y_{ijk}^t 決策變數就可以得知在整體時間窗的每個時刻有哪些節點是有無人機的，與前面相同，利用 M 來幫助建立限制式。

$$\sum_{k \in K} \sum_{(i, j) \in A_{GM}} y_{ijk}^t \leq M \cdot z_j^t \quad \forall 1 \leq t \leq T, j \in N, j \neq 0 \quad (3.4.11)$$

$$\sum_{k \in K} \sum_{(i, j) \in A_{GM}} y_{ijk}^t \geq z_j^t \quad \forall 1 \leq t \leq T, j \in N, j \neq 0 \quad (3.4.12)$$

限制式(3.4.13)確保只要在時刻 t 有除基站以外的值為 1，則基站的 z_0^t 必須為 1，會有這樣的限制式是為了讓未來判斷時刻 t 是否有與基站連通時會使用到。

$$z_0^t \geq z_j^t \quad \forall 1 \leq t \leq T, j \in N, j \neq 0 \quad (3.4.13)$$

傳輸數據節線限制式：

限制式(3.4.14)(3.4.15)(3.4.16)(3.4.17)為 z_i^t 、 z_j^t 與 w_{ij}^t 的關係限制式，限制式(3.4.14)(3.4.15)(3.4.16)為一組，這三條限制式旨在確定節線 (i, j) 存在於 A_{GC} 傳輸數據網路中，且 $i \neq j$ 以及 $i \neq 0$ (i 不是基站)的情況下建立的限制式。限制式(3.4.15)的邏輯是在確定 $z_i^t = 1$ 以及 $z_j^t = 1$ (在時刻 t 節點 i 和節點 j 上有無人機)時，則 $w_{ij}^t = 1$ (在時刻 t 時節線 (i, j) 可以被用來傳輸數據)；限制式(3.4.15)(3.4.16)則表示若在時刻 t 時節點 i 或節點 j 上其中之一沒有無人機的話，則在時刻 t 時節線 (i, j) 不可以被用

來傳輸數據，導致 w_{ij}^t 必須為 0；限制式(3.4.17)代表若不是在以上情況的所有節線 (i, j) ，應該都不能被拿來傳輸數據。

$$w_{ij}^t + M \cdot (1 - z_i^t) + M \cdot (1 - z_j^t) \geq 1 \quad \forall 1 \leq t \leq T, (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (3.4.14)$$

$$z_i^t \geq w_{ij}^t \quad \forall 1 \leq t \leq T, (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (3.4.15)$$

$$z_j^t \geq w_{ij}^t \quad \forall 1 \leq t \leq T, (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (3.4.16)$$

$$w_{ij}^t = 0 \quad \forall 1 \leq t \leq T, (i, j) \in A_{GC}, i = j \vee i = 0 \quad (3.4.17)$$

傳輸時刻限制式：

限制式(3.4.18) 為在每個週期 P 內至少要與基站連通一次。

$$\sum_{t \leq t' < (t+P)} p^{t'} \quad \forall 1 \leq t \leq T - P + 1 \quad (3.4.18)$$

週期性傳輸數據限制式：

限制式(3.4.19)為 w_{ij}^t 和 u_{ijk}^t 的關係限制式，決策變數 u_{ijk}^t 代表無人機 $k \in K$ 的數據在時刻 $t \in T$ 經過傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0。未來會利用 u_{ijk}^t 所形成的網路圖來判斷在此時刻是否有與基站連通，所以若 $w_{ij}^t = 0$ 的話，那代表該時刻節線 (i, j) 不可以被用來傳輸數據，因此該時刻的 u_{ijk}^t 就不可能為 1。

$$w_{ij}^t \geq u_{ijk}^t \quad \forall k \in K, 1 \leq t < T, (i, j) \in A_{GC} \quad (3.4.19)$$

限制式(3.4.20) 確定期時刻 t 必須要與基站形成連通網路時，若節點 i 有無人機，則該無人機必須從該節點 i 傳輸數據到其它節點，強迫有無人機的節點送出流量。

$$M \cdot (1 - p^t) + \sum_{k \in K} \sum_{(i, j) \in A_{GC}} u_{ijk}^t \geq z_i^t \quad \forall 1 \leq t \leq T, i \in N, i \neq 0 \quad (3.4.20)$$

限制式(3.4.21) 代表無人機 k 每個時刻 ($1 \leq t \leq T$) 對每個節點都要保持流量守恆，同樣是在確定時刻 t 必須要與基站形成連通網路時才需要限制。

$$M \cdot (1 - p^t) + \sum_{(i,j) \in A_{GC}} u_{ijk}^t \geq \sum_{(i',j') \in A_{GC}} u_{i'jk}^t \quad \forall k \in K, 1 \leq t \leq T, i, j' \in N, i = j' \quad (3.4.21)$$

限制式(3.4.22)確定時刻 t 必須要與基站形成連通網路時，無人機 k 在同一條節線中只能選擇一種飛行方向(節線 (i, j) 或是節線 (j, i))，此限制式是為了避免流量在同一條節線來回流動，造成子迴圈。

$$u_{ijk}^t + u_{jik}^t \leq M \cdot (1 - p^t) + 1 \quad \forall k \in K, 1 \leq t \leq T, (i, j) \in A_{GC} \quad (3.4.22)$$

限制式(3.4.23)代表當確定該時刻 t 必須要與基站形成連通網路時，必須有一個流量流入基站。

$$\sum_{(i,j) \in A_{GC}} u_{ijk}^t = p^t \quad \forall k \in K, 1 \leq t \leq T, j = 0 \quad (3.4.23)$$

最少收集次數限制式：

限制式(3.4.24)可以使所有感測站被收集數據次數都大於等於 $MinC$ 。

$$\sum_{1 \leq t \leq T} z_i^t \geq MinC \quad \forall i \in N_S \quad (3.4.24)$$

決策變數範圍限制式：

限制式(3.4.25)至限制式(3.4.29)則限制了變數的範圍。

$$x_{ijk}^t \in \{0,1\} \quad \forall (i,j) \in A, k \in K, 0 \leq t \leq W_k \quad (3.4.25)$$

$$y_{ijk}^t, u_{ijk}^t \in \{0,1\} \quad \forall (i,j) \in A, k \in K, 0 \leq t \leq T \quad (3.4.26)$$

$$w_{ij}^t \in \{0,1\} \quad \forall (i,j) \in A, 0 \leq t \leq T \quad (3.4.27)$$

$$z_n^t \in \{0,1\} \quad \forall n \in N, 0 \leq t \leq T \quad (3.4.28)$$

$$p^t \in \{0,1\} \quad \forall 0 \leq t \leq T \quad (3.4.29)$$

3.5 變數與限制式個數估算

本小節進行本研究的數學模式變數與限制式個數估算，估算的例子為 5×5 的網格圖，共有 144 條飛行節線、25 條停留節線、361 條傳輸數據節線、5 個感測站、

3 台無人機、傳輸距離為 2 以及連通週期為 6，無人機的循環路徑大小分別為 ($W_1 = 9, W_2 = 4, W_3 = 12$)，循環路徑大小即為各台無人機之時空網路圖時長定，參數相關設定如表 3.1 所示。

表 3.1 估算變數與限制式個數之參數設定值

節點 個數	飛行 節線數	傳輸 節線數	無人機 數量	個別網路時長	總時 長	傳輸 距離	連通 週期
$ N $	$ A_{GM} $	$ A_{GC} $	$ K $	W_k	T	R	P
25	169	361	3	$W_1 = 9, W_2 = 4, W_3 = 12$	36	2	6

表 3.2 變數總個數估算

本研究		
Variable	Calculation	Number
x_{ijk}^t	$ A_{GM} \cdot \sum_{k=1}^3 W_k$	4225
y_{ijk}^t	$ A_{GM} \cdot K \cdot T$	18252
u_{ijk}^t	$ A_{GC} \cdot K \cdot T$	38988
w_{ij}^t	$ A_{GC} \cdot T$	12996
z_n^t	$ N \cdot T$	900
p^t	T	36
變數總個數		75397

表 3.3 限制式總個數比較

本研究		
Constraint	Calculation	Number
3.4.2	$\sum_{k=1}^3 W_k$	36
3.4.3	$ N_s $	5
3.4.4	$\sum_{k=1}^3 W_k$	33
3.4.5	$ K $	3
3.4.6	$ K $	3
3.4.7	$ A_{GM} \cdot \sum_{k=1}^3 W_k$	4225
3.4.8	$ A_{GM} \cdot \sum_{k=1}^3 W_k$	4225
3.4.9	$ A_{GM} \cdot \sum_{k=1}^3 (T - W_k)$	14027

3.4.10	$ A_{GM} \cdot \sum_{k=1}^3 (T - W_k)$	14027
3.4.11	$ N \cdot T$	900
3.4.12	$ N \cdot T$	900
3.4.13	$ N \cdot T$	900
3.4.14	$(A_{GC} - 33) \cdot T$	11808
3.4.15	$(A_{GC} - 33) \cdot T$	11808
3.4.16	$(A_{GC} - 33) \cdot T$	11808
3.4.17	$33 \cdot T$	1188
3.4.18	$T - P + 1$	31
3.4.19	$ A_{GC} \cdot K \cdot T$	38988
3.4.20	$ N \cdot T$	900
3.4.21	$ N \cdot K \cdot T$	2700
3.4.22	$ A_{GC} \cdot K \cdot T$	38988
3.4.23	$ K \cdot T$	108
3.4.24	$ N_s $	5
限制式總個數		157616

在表 3.3 中限制式(3.4.14)-(3.4.17)中出現了數字 33，該數字代表扣除無人機將數據傳給自己的 25 條節線，以及扣除從基站(節點 0)向外傳輸數據的節線，因為本模式是使用無人機是否能將數據傳回基站來判斷無人機群使否成功與基站連通，所以在設定這些限制式時需要扣除基站向外傳輸數據的節線。

3.6 小結

本章使用時空網路圖的概念，針對多無人機之循環路徑規劃，提出一個數學模式，由於使用時空網路圖建模，若是節點數量變多，則會導致求解時間過久。如果要將本研究應用於實際例子，則需要設計更有效率的啟發式演算法，以更快得到近似最佳或品質尚可接受的解。雖然數學模式的求解速度較慢，但其求出的答案皆為最佳解，所以可以藉由觀察數學式所的答案來改善演算法，盡可能的使演算法有機會收斂以得到最佳解，以開發出良好的啟發式演算法。

第四章 啟發式演算法

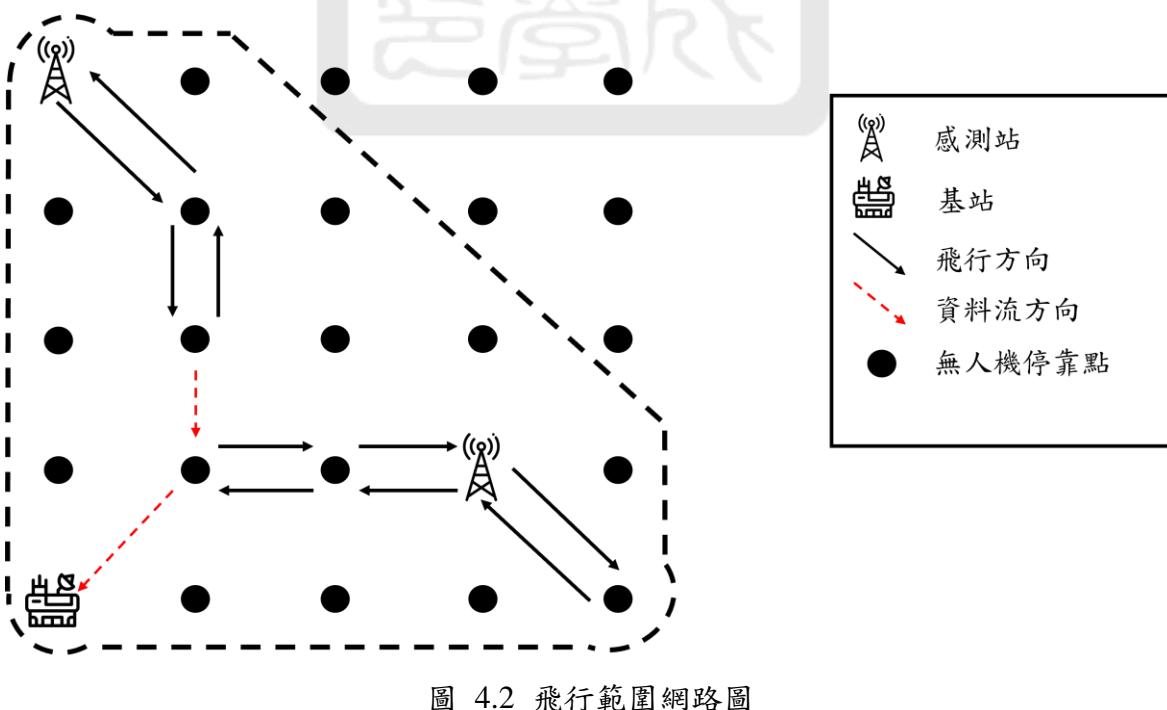
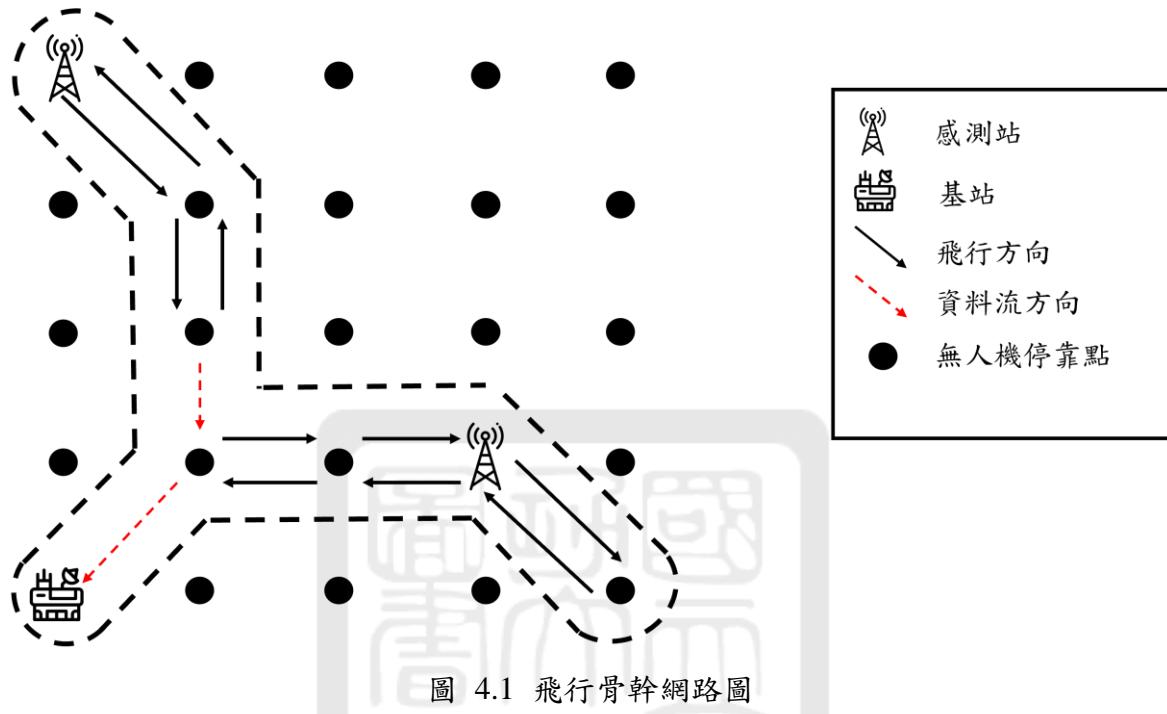
由於第三章提出之數學模式主要以時空網路作為建模的主軸，該模式對於大規模的問題的求解速度太慢，導致無法在合理時間內解出最佳解，因此本章節將嘗試開發啟發式演算法求解。然而本問題的變數之間具有太過複雜的牽連關係，若單純使用一般演算法處理的話，通常仍無法有效地改善其解的品質，因此本研究嘗試發展利用簡化後的混整數規劃模式(MIP)為貪婪演算法的核心求解機制，將數學模式搭配在演算法的架構中，設計三類「數學啟發式演算法」，較能針對規模更大的網路更快求出品質不錯的解，以下將一一描述三類數學演算法之開發概念與過程。

4.1 飛行範圍貪婪演算法

在本研究的貪婪演算法中，其核心概念為將原來複雜的問題，拆解為幾個子問題，將個別的子問題發展為數學規劃模式，再利用貪婪演算法的架構結合這些數學規劃模式來找出最佳解。觀察第三章數學模式的求解時間以及變數和限制式估算，可以看出影響限制式個數的多寡通常都是網路的大小，因此我們先將網路縮小來觀察求解，並希望貪婪演算法能夠在很短的時間內求解出最佳解。

由於縮小網路的方法有很多種，為了能準確地將最佳解仍被縮小的網路所涵蓋，本研究建立了另一個數學規劃模式，選取網路中的某些節線來建構無人機循環路徑所構成的子網路，讓該子網路涵蓋所有的感測站、基站以及為了連通這些節點的某些節點與節線。直觀上來看，就是先僅考慮資料傳輸的連通性、忽略週期性連通的限制，利用該數學規劃模式建構出可以傳輸數據回基站的循環路徑所構成的子網路。雖然此子網路尚未考量準確的週期性限制，卻可大幅縮小網路並以最低限度的條件符合週期性限制。本研究採用兩種方法來將第一階段模式所找出的路徑變成網路：(1)只將模式得到的路徑上的節點以及節線加入網路，在本研究稱為「飛行骨幹網路」，如圖 4.1 所示；(2)以飛行骨幹網路為基礎，再將該網路周圍於通訊範

園內的節點與節線也一併加入，構成「飛行範圍網路」，如圖 4.2 所示。本研究將此數學規劃模式作為「飛行範圍貪婪演算法」的第一階段。第二階段則在數學規劃模式該縮小網路之後，將其當成第三章的主要模式之輸入網路來求解。



以下模式為此飛行範圍貪婪演算法的第一階段的數學模式。

4.1.1 符號定義

集合

N 所有節點之集合

N_s 感測站之集合， $N_s \subseteq N$

A 所有節線之集合

A_{GM} 所有飛行節線之集合， $A_{GM} \subseteq A$

A_{GC} 所有傳輸數據節線之集合

W 為無人機循環路徑長度之集合，一單位長度需要耗時一單位時間，因此也是無人機的時空網路的長度範圍之集合

參數

M 極大的數

K 無人機數量

R 無人機傳輸數據最長距離

L 無人機傳輸數據總層(Level)數

變數

x_{ijk}^t 若無人機 $k \in K$ 在時刻 $t \in W_k$ 經過飛行節線 $(i, j) \in A_{GM}$ 為 1；反之為 0

u_{ij}^l 若在 level l 使用傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0

w_{ij} 若可以使用 $(i, j) \in A_{GC}$ 節線為 1；反之為 0

z_i 若在無人機停留在節點 $i \in N$ 為 1；反之為 0

4.1.2 目標式

本研究的目標式為最大化無人機從感測站收集的數據量，並要符合無人機循環路徑的特性。

$$\text{Maximize } \sum_{k \in K} \sum_{1 \leq t \leq W_k} \sum_{(i, j) \in A_{GM}, i \in N_s} x_{ijk}^t \quad (4.1.1)$$

4.1.3 限制式

流量守恆限制式：

限制式(4.1.2)為描述無人機 k 在每個時刻 $(1 \leq t \leq W_k)$ 都要通過一條 arc 。

$$\sum_{(i, j) \in A_{GM}} x_{ijk}^t = 1 \quad \forall k \in K, 1 \leq t \leq W_k \quad (4.1.2)$$

限制式(4.1.3)為每個感測站都至少會被收集數據一次。

$$\sum_{k \in K} \sum_{1 \leq t \leq W_k} \sum_{(i, j) \in A_{GM}} x_{ijk}^t \geq 1 \quad \forall i \in N_S \quad (4.1.3)$$

限制式(4.1.4)為表示無人機 k 每個時刻 $(1 \leq t \leq W_k)$ 之間必須對每個節點都會保持流量守恆。

$$\sum_{(i, j) \in A_{GM}} x_{ijk}^t = \sum_{(i', j') \in A_{GM}} x_{i'j'k}^{t+1} \quad \forall k \in K, 1 \leq t < W_k, j, i' \in N, j = i' \quad (4.1.4)$$

循環路徑限制式：

限制式(4.1.5)為限制每台無人機個別的出發點以及終點要在同一個節點上。

$$\sum_{(i, j) \in A_{GM}} x_{ijk}^{W_k} = \sum_{(i', j') \in A_{GM}} x_{i'j'k}^1 \quad \forall k \in K, j, i' \in N, j = i' \quad (4.1.5)$$

決策變數關係限制式：

限制式(4.1.6)(4.1.7)為 x_{ijk}^1 與 z_i 的關係限制式， x_{ijk}^1 代表無人機 $k \in K$ 在時刻 1 經過飛行節線 $(i, j) \in A_{GM}$ 為 1；反之為 0，而 z_i 無人機停留在節點 $i \in N$ 為 1；反之為 0，由於前面的限制式，可以利用 x_{ijk}^t 決策變數得知每台無人機經過哪些點，因此利用 M 的技巧來設置限制式(4.1.6)(4.1.7)，當無人機 k 通過 i 點時，限制式(4.1.6)會強迫 z_i 變成 1，限制式(4.1.7)會無限制力，反之，當無人機 k 沒有通過 i 點時，限制式(4.1.7)會強迫 z_i 變成 0，限制式(4.1.6)會無限制力；利用這樣的特性，就可以所有無人機路徑的起點放入變數 z_i ，利用 z_i 做後面的傳輸數據節線限制式。限制式(4.1.8)則代表基站永遠可以接收數據。

$$\sum_{k \in K} \sum_{(i', j') \in A_{GM}} x_{i' j' k}^1 \leq M \cdot z_i \quad \forall i = j', i \in N \quad (4.1.6)$$

$$M \cdot \sum_{k \in K} \sum_{(i', j') \in A_{GM}} x_{i' j' k}^1 \geq z_i \quad \forall i = j', i \in N \quad (4.1.7)$$

$$z_0 = 1 \quad (4.1.8)$$

傳輸數據節線限制式：

限制式(4.1.9)(4.1.10)(4.1.11)(4.1.12)為 z_i 、 z_j 與 w_{ij} 的關係限制式，限制式(4.1.9)(4.1.10)(4.1.11)為一組，這三條限制式是在確定節線 (i, j) 存在於 A_{GC} 傳輸數據網路中，且 $i \neq j$ 以及 $i \neq 0$ (i 不是基站)的情況下建立的限制式。限制式(4.1.9)的邏輯是在確定 $z_i = 1$ 以及 $z_j = 1$ (節點 i 和節點 j 上有無人機)時，則 $w_{ij} = 1$ (節線 (i, j) 可以被用來傳輸數據)；限制式(4.1.10)(4.1.11)則表示節點 i 或節點 j 上其中之一沒有無人機，則節線 (i, j) 不可以被用來傳輸數據，導致 w_{ij} 必須為 0；限制式(4.1.12)代表若不是在以上情況的所有節線 (i, j) 都不能被拿來傳輸數據。

$$w_{ij} + M \cdot (1 - z_i) + M \cdot (1 - z_j) \geq 1 \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.1.9)$$

$$z_i \geq w_{ij} \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.1.10)$$

$$z_j \geq w_{ij} \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.1.11)$$

$$w_{ij}=0 \quad \forall (i, j) \in A_{GC}, i = j \vee i = 0 \quad (4.1.12)$$

傳輸數據限制式：

限制式(4.1.13)為 w_{ij} 和 u_{ij}^l 的關係限制式，決策變數 u_{ij}^l 代表的數據在 level l 使用傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0。未來會利用 u_{ij}^l 所形成的網路圖來判斷在此時刻是否有與基站連通，所以若 $w_{ij} = 0$ 的話那代表該節線 (i, j) 不可以被用來傳輸數據，那同樣該節線的 u_{ij}^l 就不可能為 1。

$$w_{ij} \geq u_{ij}^l \quad \forall 1 \leq l \leq L, (i, j) \in A_{GC} \quad (4.1.13)$$

限制式(4.1.14)為每個 level l 只能選擇一條節線來傳輸數據。

$$\sum_{(i, j) \in A_{GC}} u_{ij}^l = 1 \quad \forall 1 \leq l \leq L \quad (4.1.14)$$

限制式(4.1.15)為每個 level l 必須保持數據傳輸的流量守恆。

$$\sum_{(i, j) \in A_{GC}} u_{ij}^l = \sum_{(i', j') \in A_{GC}} u_{i'j'}^{l+1} \quad \forall 1 \leq l < L, i', j' \in N, i' = j \quad (4.1.15)$$

限制式(4.1.16)確定與基站形成連通網路時，無人機只能在同一條節線中只能選擇一種飛行方向(節線 (i, j) 或是節線 (j, i))，此限制式是為了避免流量在同一條節線來回流動，造成子迴圈。

$$u_{ij}^l + u_{ji}^l \leq 1 \quad \forall 1 \leq l \leq L, (i, j) \in A_{GC} \quad (4.1.16)$$

限制式(4.1.17)代表當確定 t 必須要與基站形成連通網路時，必須有一個流量流入基站。

$$\sum_{(i, j) \in A_{GC}} u_{ijk}^L = 1 \quad \forall j = 0 \quad (4.1.17)$$

決策變數範圍限制式：

限制式(4.1.18)至限制式(4.1.21)則限制了變數的範圍。

$$x_{ijk}^t \in \{0,1\} \quad \forall (i,j) \in A, k \in K, 0 \leq t \leq W_k \quad (4.1.18)$$

$$u_{ij}^l \in \{0,1\} \quad \forall (i,j) \in A, 1 \leq l \leq L \quad (4.1.19)$$

$$w_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (4.1.20)$$

$$z_i \in \{0,1\} \quad \forall i \in N \quad (4.1.21)$$

經由第一階段模式得到路徑之後，藉由飛行骨幹網路以及飛行範圍網路得到縮小網路之後，將第一階段所得到的網路圖放入第二階段的模式中，期望能求出最佳解且可以提升了求解效率，本演算法更詳細的數值分析可見第五章。

4.2 路徑組合貪婪演算法

「飛行範圍貪婪演算法」利用數學規劃模式將原網路變成縮小的子網路，再將該子網路放入下一階段求解，此法的缺點在於難以產生出多樣性的縮小子網路，也就是說即使我們輸入的節點與節線略有不同，數學模式為了保證連通性，幾乎都會求解出一樣的縮小子網路。為了改善此問題，本研究開發出新的「路徑組合貪婪演算法」，其流程圖如圖 4.3 所示。在該演算法中，步驟(GM_1)代表第一階段的數學模式，(GM_2)代表第二階段的數學模式，(GM_3)代表第三階段的數學模式。而每個步驟的過程將另外個別展示於圖 4.4~圖 4.8。

其中， P 代表 GM_2 的候選路徑集合， P 代表 GM_3 一次迭代的候選路徑集合， \bar{P} 代表最佳路徑集合， T 代表迭代的次數，本演算法可以依據網路規模的大小，針對以上的參數進行修改加以調整結果：

1. 利用 BFS 演算法在原網路中找出不同長度循環路徑，並將這些路徑存放全路徑清單(AP)，做為第一階段模式(GM_1)的候選路徑。

2. 用第一階段模式(GM_1)來求出候選路徑，這些路徑會進入第二階段候選路徑，並將選擇的路徑從(AP)中刪除，其所選出的路徑必須達到一定數量才能供第二階段模式(GM_2)選出無人機的路徑組合。
3. 用第二階段模式(GM_2)，從候選路徑中選出無人機各自的路徑，該無人機路徑的組合便會成為第三階段模式(GM_3)的候選路徑組合。
4. 將步驟2和步驟3重複數次，以讓第三階段模式(GM_3)有足夠的候選組合可以選擇最佳解。
5. 利用第三階段模式(GM_3)選出最佳的路徑組合。

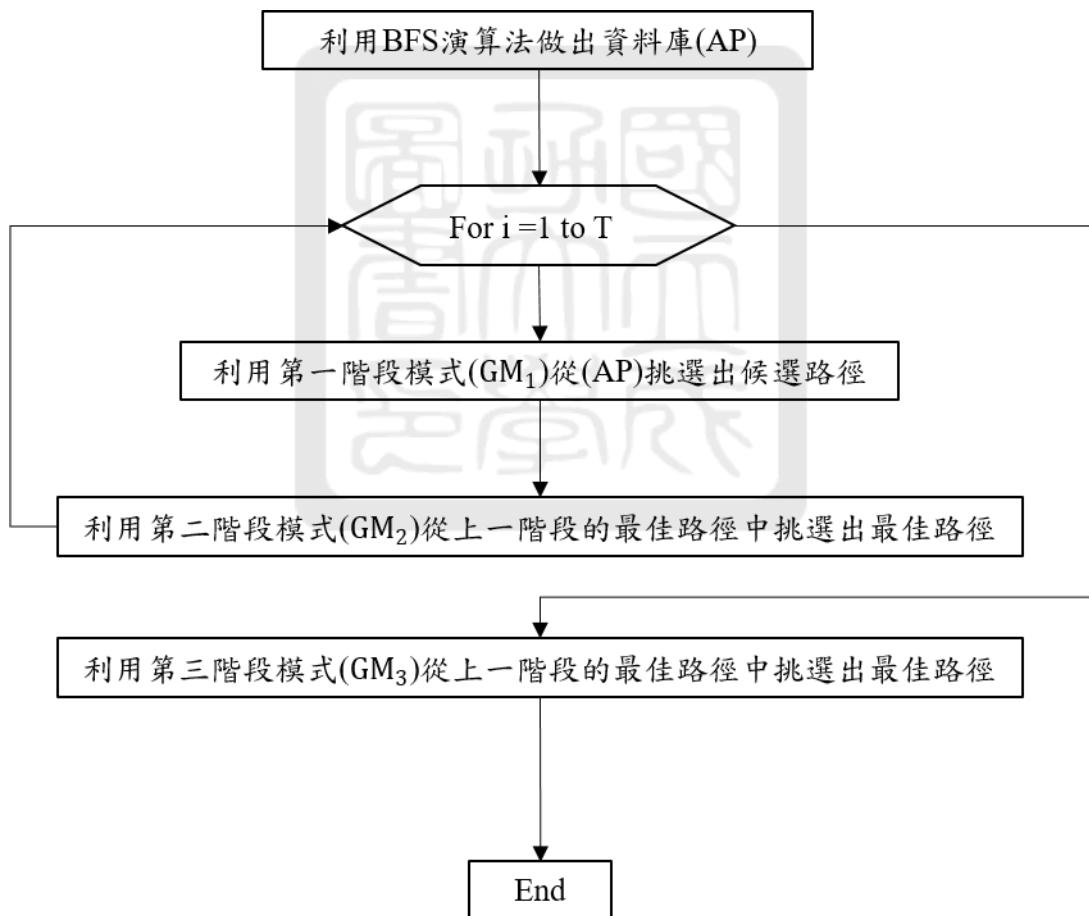


圖 4.3 路徑組合貪婪演算法流程圖

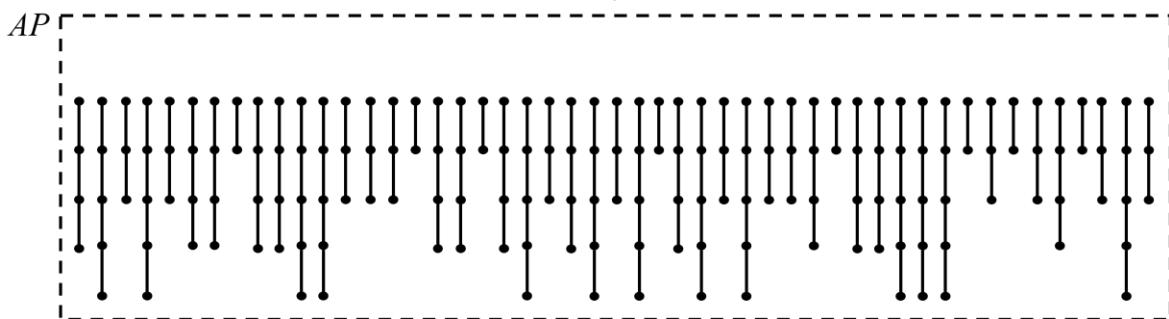
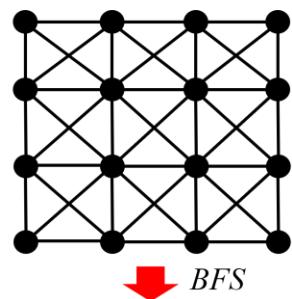


圖 4.4 路徑組合貪婪演算法步驟 1

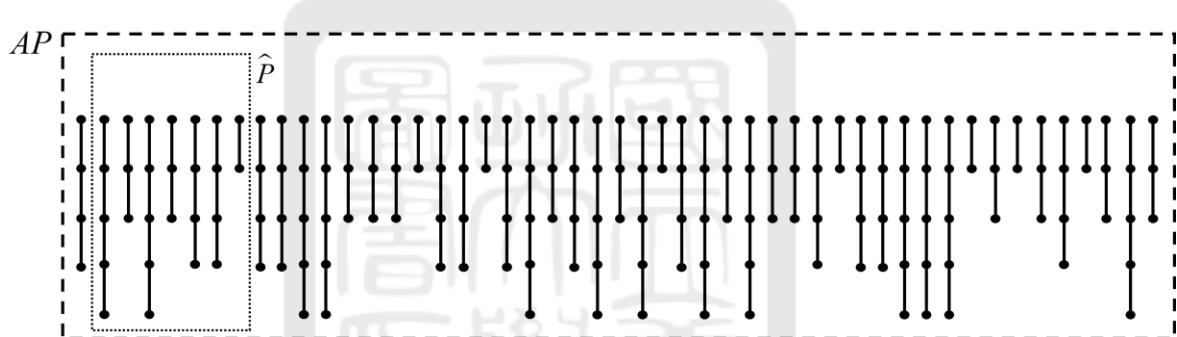


圖 4.5 路徑組合貪婪演算法步驟 2

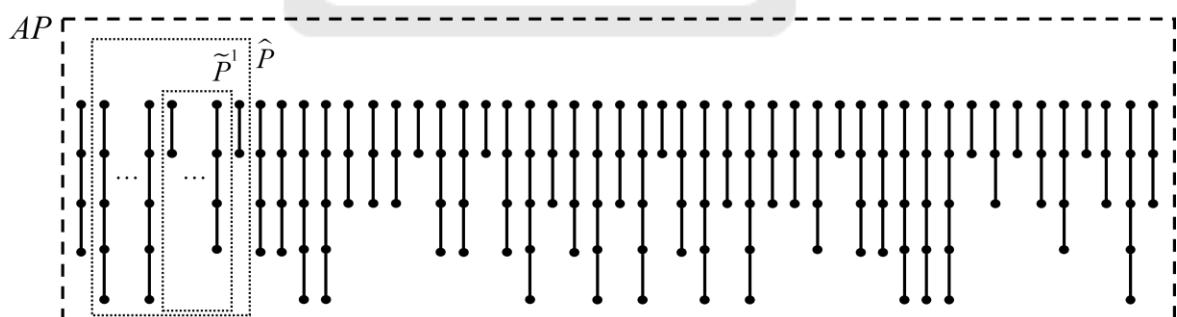


圖 4.6 路徑組合貪婪演算法步驟 3

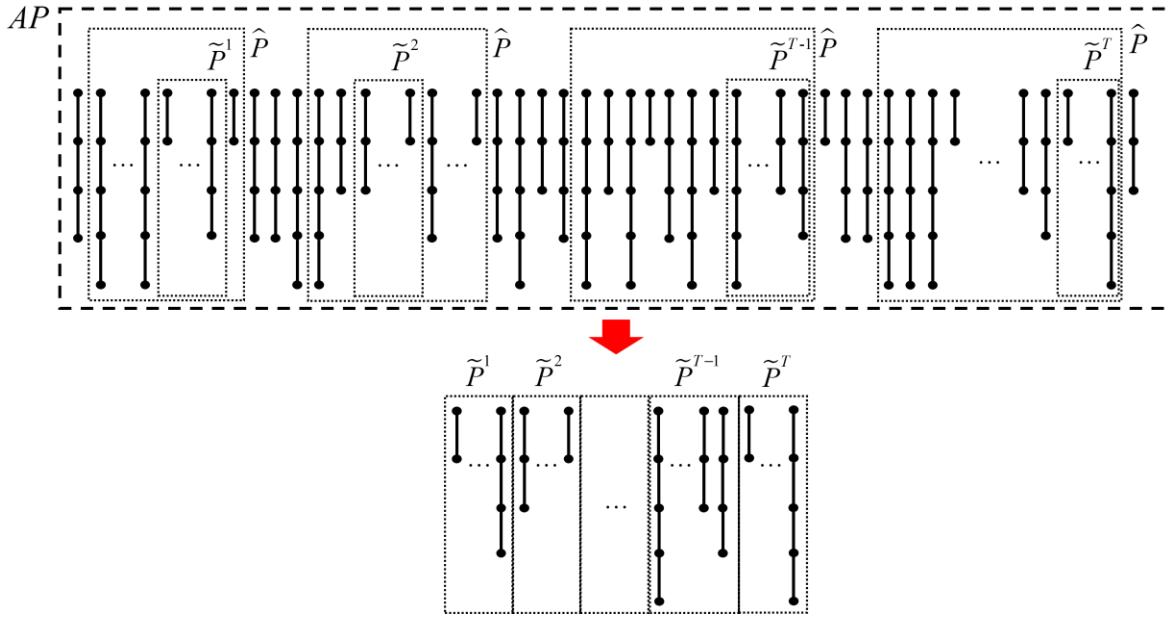


圖 4.7 路徑組合貪婪演算法步驟 4

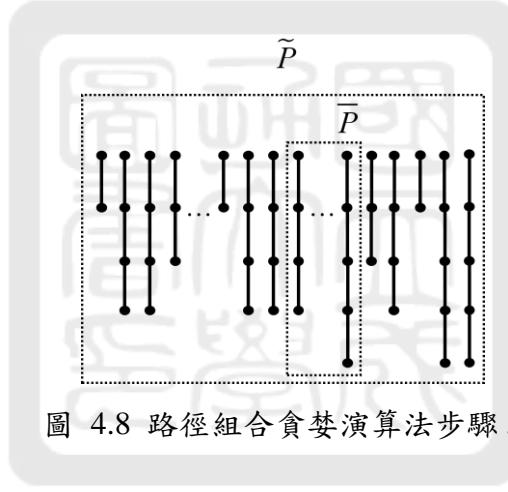


圖 4.8 路徑組合貪婪演算法步驟 5

步驟 2 的路徑是直接用 *BFS* 演算法將一定長度以內的路徑都放入路徑資料庫當中，故當網路規模較大或是感測站分布的位置較極端時，會導致整個路徑資料庫規模太大，導致整個演算法無法求解。因此若遇到網路規模較大或是感測站分布較極端的網路時，會針對 *BFS* 演算法進行調整，讓演算法不會一次將所有路徑找出，而是通過一些範圍的限制找出部分的路徑來求解，只針對部分可能性較高的路徑求解。所以 *BFS* 演算法是可以針對不同的網路而做些微的調整，利用這些調整不但可以加速整體的求解時間，有時也會因為這些調整而提升解的品質。即使這樣做沒辦法一次考量所有路徑，但是為了讓演算法可以運行所做必要的衡量(trade-off)。

除此之外，該演算法主要考慮無人機所造訪的空間資訊（亦即節點與節線），暫先不考慮其時間資訊（譬如在某節點停留），所以先不將無人機停留在原地的路徑放入 AP 當中，而讓無人機是否停留在原地的決策再由後面步驟的模式來判斷。

GM_1 必須從整個路徑資料庫 AP 選出適當的候選路徑給 GM_2 ，由於本研究要最大化無人機從感測站收集的數據量，所以直觀上來看無人機本身的路徑一定是經過越多感測站越好。因此 GM_1 的主要目標為其選出的候選路徑經過的感測站越多越好，限制式則是規範這些選出的路徑至少要包括各個感測站各一次，以及所選出的路徑必須要與基站形成一個連通的網路。

為了盡可能地讓 GM_2 可以求出可行解，必須以最低限度的標準來盡可能地滿足週期性限制，然而週期性限制的最低標準即為無人機的路徑必須要與基站連通，除此之外，至少要有 $K/2$ 條路徑的長度是小於等於最小無人機步數的，這是因為當這些路徑進入 GM_2 時， GM_2 會從當中選擇與無人機數量相同的路徑派發給每台無人機，而無人機則必須按照配發的路徑上的節點順序一一通過每個節點，但是否停留在某些節點則由 GM_2 自行判斷。

以圖 4.9 為例，假設模式所選出的路徑長度為 4，而無人機的步數為 6，則該無人機除了必須通過所選路徑的 4 個節點之外，還必須選擇在某些節點停留 2 步，這樣才能滿足該無人機在所選的路徑上走了 6 步，先固定無人機能走的範圍，但讓數學模式可以決定如何在有限範圍中規劃路徑。反之，在圖 4.10 的例子中，假設模式所選出的路徑長度為 4，而無人機的步數為 3，則該無人機由於步數不夠，便無法通過模式所選出的路徑上的 4 個節點，會導致 GM_2 解不出可行解。因此，必須在 GM_1 加入至少要選擇數條比無人機最小步數還要更短的路徑之限制式，才可以避免 GM_2 找不到可行解。上述例子說明了本研究先固定無人機的路徑空間，但不固定其時間的作法。

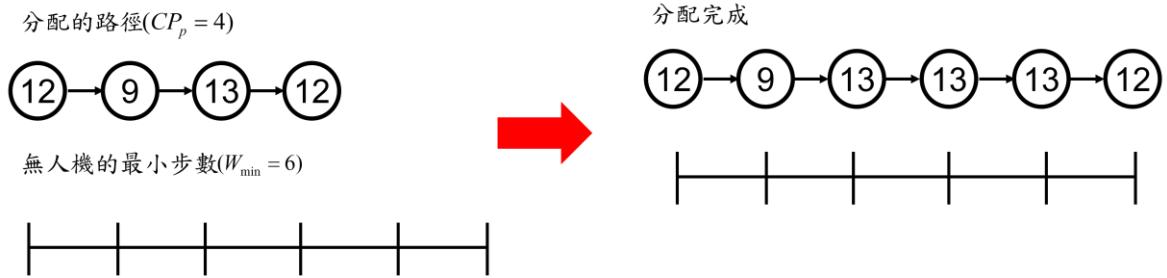


圖 4.9 路徑與無人機步數可以分配範例 ($CP_p = 4, W_{\min} = 6$)

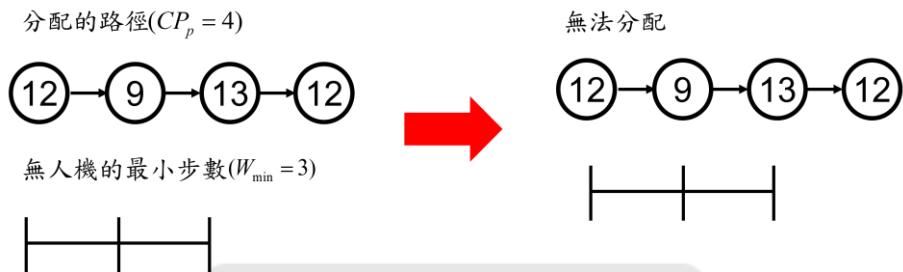


圖 4.10 路徑與無人機步數無法分配範例 ($CP_p = 4, W_{\min} = 3$)

說明完 GM_1 的用處以及架構後，以下將說明其數學模式。

4.2.1 符號定義

集合

N 所有節點之集合

N_s 感測站之集合， $N_s \subseteq N$

AP 所有路徑之集合

A_{GC} 所有傳輸數據節線之集合

參數

M 極大的數

num 選擇路徑的數量

W_{\min} 無人機的最小步數

變數

x_k 若路徑 $k \in AP$ 有被選為 1；反之為 0

y_{kl} 若路徑 $k \in AP$ 的節點 $l \in N$ 有被經過為 1；反之為 0

u_{ij}^l 若第 $l \in N$ 個數據流量使用傳輸數據節線 $(i, j) \in A_{GC}$ 為 1；反之為 0

w_{ij} 若可以使用節線 $(i, j) \in A_{GC}$ 傳輸數據流量為 1；反之為 0

z_l 若有無人機停留在節點 $l \in N$ 為 1；反之為 0

4.2.2 目標式

本研究的目標式為最大化無人機從感測站收集的數據量，並要符合無人機循環路徑的特性。

$$\text{Maximize } \sum_{l \in N_s} z_l \quad (4.2.1)$$

4.2.3 限制式

限制式(4.2.2)為被選到的路徑數量要等於規定的所選路徑數量。

$$\sum_{k \in AP} x_k = num \quad (4.2.2)$$

限制式(4.2.3)為所選的路徑當中，比無人機最小步數還要小的路徑數量至少要有 $K/2$ 條。

$$\sum_{k \in AP} x_k \geq \frac{K}{2} \quad \forall |AP_k| \leq W_{\min} \quad (4.2.3)$$

限制式(4.2.4)為路徑 $k \in AP$ 必須被選，那路徑 $k \in AP$ 上的 $l \in N$ 節點才有可能被經過。

$$M \cdot x_k \geq \sum_{l \in N} y_{kl} \quad \forall k \in AP \quad (4.2.4)$$

限制式(4.2.5)為當節點 $l \in N$ 出現在路徑 $k \in AP$ 時，該點的 y_{kl} 的值必等於 x_k 的值。

$$x_k = y_{kl} \quad \forall k \in AP, l \in N, if \quad l \in AP_k \quad (4.2.5)$$

限制式(4.2.6)會強迫每個感測站至少被經過一次。

$$\sum_{k \in AP} y_{kl} \geq 1 \quad \forall l \in N_s \quad (4.2.6)$$

限制式(4.2.7)(4.2.8)為 y_{kl} 與 z_l 的關係限制式， y_{kl} 代表路徑 $k \in AP$ 的節點 $l \in N$ 有被經過為 1；反之為 0，而 z_l 代表無人機停留在節點 $l \in N$ 為 1；反之為 0，由於前面的限制式，可以利用 y_{kl} 決策變數得知每台無人機經過哪些點，因此利用 M 的技巧來設置限制式(4.2.7)(4.2.8)，當無人機通過 l 點時，限制式(4.2.7)會強迫 z_l 變成 1，限制式(4.2.8)會無限制力，反之，當無人機沒有通過 l 點時，限制式(4.2.8)強迫 z_l 變成 0，限制式(4.2.7)會無限制力；利用這樣的特性，就可以所有無人機路徑的起點放入變數 z_l ，利用 z_l 做後面的傳輸數據節線限制式。

$$\sum_{k \in AP} y_{kl} \leq M \cdot z_l \quad \forall l \in N \quad (4.2.7)$$

$$M \cdot \sum_{k \in AP} y_{kl} \geq z_l \quad \forall l \in N \quad (4.2.8)$$

限制式(4.2.9)(4.2.10)(4.2.11)(4.2.12)為 z_i 、 z_j 與 w_{ij} 的關係限制式，限制式(4.2.9)(4.2.10)(4.2.11)為一組，這三條限制式是在確定節線 (i, j) 存在於 A_{GC} 傳輸數據網路中，且 $i \neq j$ 以及 $i \neq 0$ (i 不是基站)的情況下建立的限制式，限制式(4.2.9)的邏輯是在確定 $z_i = 1$ 以及 $z_j = 1$ (節點 i 和節點 j 上有無人機)時，則 $w_{ij} = 1$ (節線 (i, j) 可以被用來傳輸數據)，限制式(4.2.10)(4.2.11)則表示節點 i 或節點 j 上其中之一沒有

無人機，則節線 (i, j) 不可以被用來傳輸數據，導致 w_{ij} 必須為 0，限制式(4.2.12)代表傳輸數據回激戰的結線皆可使用，這樣才能繼續設立後續的限制式，限制式(4.2.13)代表若不是在以上情況的所有節線 (i, j) 都不能被拿來傳輸數據。

$$w_{ij} + M \cdot (1 - z_i) + M \cdot (1 - z_j) \geq 1 \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.2.9)$$

$$z_i \geq w_{ij} \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.2.10)$$

$$z_j \geq w_{ij} \quad \forall (i, j) \in A_{GC}, i \neq j, i \neq 0 \quad (4.2.11)$$

$$w_{i0} = 1 \quad \forall (i, 0) \in A_{GC} \quad (4.2.12)$$

$$w_{ij} = 0 \quad \forall (i, j) \in A_{GC}, i = j \vee i = 0 \quad (4.2.13)$$

限制式(4.2.14)為 w_{ij} 和 u_{ij}^l 的關係限制式，決策變數 u_{ij}^l 代表第 $l \in N$ 個數據流量若使用傳輸數據節線 $(i, j) \in A_{GC}$ ，則為 1；反之為 0。未來會利用 u_{ij}^l 所形成的網路圖來判斷選取的路徑是否有與基站連通，所以若 $w_{ij} = 0$ 的話那代表該節線 (i, j) 不可以被用來傳輸數據，那同樣該節線的 u_{ij}^l 就不可能為 1。

$$w_{ij} \geq u_{ij}^l \quad \forall 1 \leq l \leq L, (i, j) \in A_{GC} \quad (4.2.14)$$

限制式(4.2.15)為若 l 節點有無人機停靠過，則該節點必須發送一個流量回基站，故基站收到的流量總合為無人機有停靠的點的總數。

$$\sum_{l \in N} \sum_{(i, 0) \in A_{GC}} u_{i0}^l = \sum_{l \in N} z_l \quad (4.2.15)$$

限制式(4.2.16) (4.2.17)為若 l 節點有無人機停靠過，則在第 l 個路徑的節點 l 必須發送一個流量回基站，而在第 l' 個路徑的其它節點必須維持流量守恆。

$$\sum_{(i, j) \in A_{GC}} u_{ij}^l + z_{l'} = \sum_{(i', j') \in A_{GC}} u_{ij'}^{l'} \quad \forall l \in N, l' \in N, l = l', j = j' \quad (4.2.16)$$

$$\sum_{(i, j) \in A_{GC}} u_{ij}^l = \sum_{(i', j') \in A_{GC}} u_{ij'}^{l'} \quad \forall l \in N, l' \in N, l \neq l', j = j' \quad (4.2.17)$$

以上為 GM_1 的架構，選出的候選路徑會進入 GM_2 當中，步驟 2 選出 GM_2 的候選路徑之後，便會將所選出的候選路徑從 AP 中刪除，。

第 3 步驟中，當 GM_2 從候選路徑中選擇路徑時，便會有些路徑沒有被選中，有兩種處理方式：第一種為被選中的路徑將會進入第二階段，然而沒被選中的路徑則會直接刪除；第二種方法則是將沒被選中的路保留下來，並將其用在下一次步驟 3 的選擇之中。

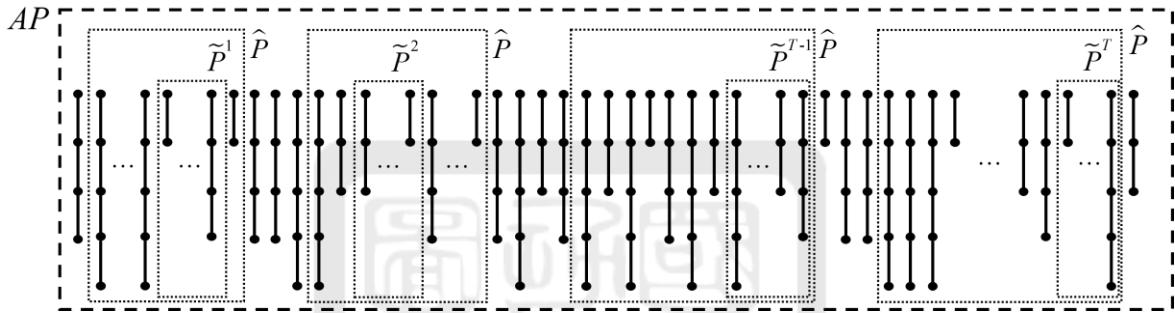


圖 4.11 GM^D 會將所有候選路徑全部換新

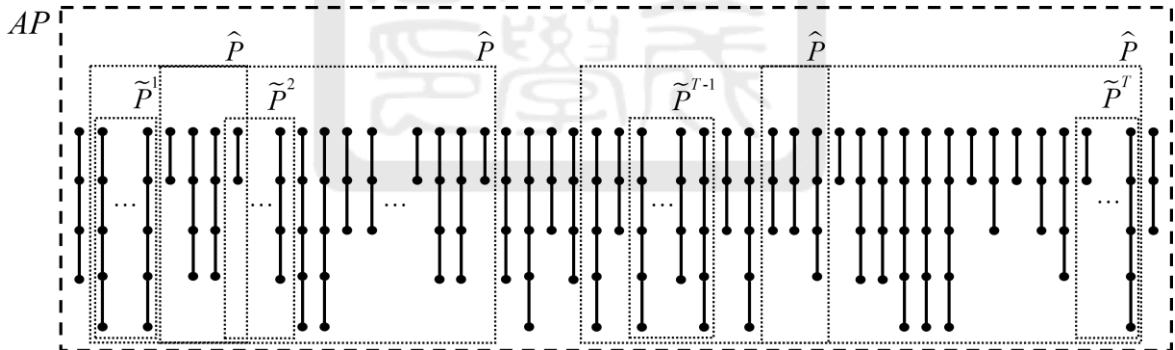


圖 4.12 GM^S 會保留前一次迭代的部分候選路徑

舉例來說，假設在問題中有三台無人機，而步驟 3 的候選路徑為十條路徑，那步驟 3 就會從十條候選路徑中選擇三條路徑放入第三階段模式的候選路徑中；此時剩餘的七條路徑會先被儲存起來。接著，在步驟 2 中重新從 (AP) 選擇三條路徑，此時步驟 3 就會將新選的三條跟原來儲存的七條重新組合而成新的十條候選路

徑，並繼續執行步驟 3。因為有上述兩種選法，故未來會以 GM^D (第一種方法)和 GM^S (第二種方法)來區別這兩種方法。圖 4.11 和圖 4.12 展示了這兩種刪除方法。

說明完兩種方法差異後，無論選了哪種方法，步驟 3 都必須利用 GM_2 來選出候選路徑給 GM_3 ，然而 GM_2 和 GM_3 兩者的目標式以及限制式都相同，且該模式是由原來第三章所提的數學模式改造之後，將它當成 GM_2 和 GM_3 的數學模式。為了將原模式從網路尋找路徑改成從候選路徑中，選擇較好的路徑給每台無人機，本研究修改了第三章中的 x_{ijk}^t 以及 y_{ijk}^t 決策變數，以及新增一個決策變數 b_k^p ，因為要知道哪台無人機選了哪一條路徑，所以必須幫這兩個決策變數新增一個候選路徑的維度。

本研究稱候選路徑的集合為 CP ，故這兩個決策變數必須增加一個上標 $p \in CP$ ，也就是新增了選擇哪條候選路徑的上標，以下為更新後的三個決策變數：

- x_{ijk}^{tp} 若無人機 $k \in K$ 選擇路徑 $p \in CP$ 在時刻 $t \in W_k$ 經過飛行節線 $(i, j) \in A_{GM}$ 為 1；
反之為 0
- y_{ijk}^{tp} 若無人機 $k \in K$ 選擇路徑 $p \in CP$ 在時刻 $t \in T$ 經過傳輸數據節線 $(i, j) \in A_{GM}$ 為 1；
反之為 0
- b_k^p 若無人機 $k \in K$ 選擇路徑 $p \in CP$ 為 1；反之為 0

由於決策變數更改的緣故，導致限制式(3.4.2)~(3.4.12)都受影響，將由新的 12 條限制式(4.2.18)~(4.2.27)來取代，以下為更新後的限制式說明。

流量守恆限制式：

限制式(4.2.18)描述每台無人機 k 都必須選擇一條路徑。

$$\sum_{p \in CP} b_k^p = 1 \quad \forall k \in K \quad (4.2.18)$$

限制式(4.2.19)描述無人機 k 如果選定了某一條路徑，則必須按照順序經過該路徑上所有的節點。

$$\sum_{1 \leq t \leq W_k} x_{ijk}^{tp} \geq b_k^p \quad \forall k \in K, p \in CP, (i, j) \in CP_p \quad (4.2.19)$$

限制式(4.2.20)和(4.2.21)為表示無人機 k 若選擇了路徑 p ，則每個時刻

$(1 \leq t \leq W_k)$ 都會通過一條節線，反之，若無人機 k 沒有選擇路徑 p ，則每個時刻不會通過路徑 p 上的任何節線。

$$\sum_{(i,j) \in CP_p} x_{ijk}^{tp} \leq b_k^p \quad \forall k \in K, p \in CP, 1 \leq t \leq W_k \quad (4.2.20)$$

$$\sum_{(i,j) \in CP_p} x_{ijk}^{tp} + M \cdot (1 - b_k^p) \geq 1 \quad \forall k \in K, p \in CP, 1 \leq t \leq W_k \quad (4.2.21)$$

限制式(4.2.22)為表示無人機 k 每個時刻 $(1 \leq t \leq W_k)$ 之間必須對每個節點都會保持流量守恆。

$$\sum_{(i,j) \in CP_p} x_{ijk}^{tp} = \sum_{(i',j') \in CP_p} x_{i'j'k}^{t+1,p} \quad \forall k \in K, p \in CP, 1 \leq t < W_k, j, i' \in N, j = i' \quad (4.2.22)$$

循環路徑限制式：

限制式(4.2.23)為每台無人機的起訖點相同。

$$\sum_{(i,j) \in CP_p} x_{ijk}^{W_k p} = \sum_{(i',j') \in CP_p} x_{i'j'k}^{1,p} \quad \forall k \in K, j, i' \in N, j = i' \quad (4.2.23)$$

決策變數關係限制式：

限制式(4.2.24)(4.2.25)為 x_{ijk}^{tp} 與 y_{ijk}^{tp} 的關係限制式， x_{ijk}^t 若無人機 $k \in K$ 選擇路徑 $p \in CP$ 在時刻 $t \in W_k$ 經過飛行節線 $(i, j) \in A_{GM}$ 為 1；反之為 0，而 y_{ijk}^{tp} 若無人機 $k \in K$ 選擇路徑 $p \in CP$ 在時刻 $t \in T$ 經過傳輸數據節線 $(i, j) \in A_{GM}$ 為 1；反之為 0，因為 T 為 W 集合的最小公倍數，所以每台無人機的的循環路徑會在整體時間軸上循環多次，因此要將 x_{ijk}^t 決策變數的值放入全部 y_{ijk}^t 決策變數之中，才能用後面的限制式來做週期性限制的限制式。

$$x_{ijk}^{tp} = y_{ijk}^{tp} \quad \forall k \in K, 1 \leq t \leq W_k, (i, j) \in CP_p \quad (4.2.24)$$

$$y_{ijk}^{tp} = y_{ijk}^{t,p} \quad \forall k \in K, 1 \leq t \leq W_k, W_k < t \leq T, (i, j) \in CP_p \quad (4.2.25)$$

無人機停靠限制式：

限制式(4.2.26)(4.2.27)為 y_{ijk}^t 和 z_j^t 的關係限制式，利用 y_{ijk}^t 決策變數就可以得知在整體時間窗的每個時刻有哪些節點是有無人機的，與前面相同，利用 M 來幫助建立限制式。

$$\sum_{k \in K} \sum_{(i,j) \in CP_p} y_{ijk}^{tp} \leq M \cdot z_j^t \quad \forall p \in CP, 1 \leq t \leq T, j \in N, j \neq 0 \quad (4.2.26)$$

$$\sum_{k \in K} \sum_{(i,j) \in CP_p} y_{ijk}^{tp} \geq z_j^t \quad \forall p \in CP, 1 \leq t \leq T, j \in N, j \neq 0 \quad (4.2.27)$$

修改完以上限制式之後，便可將第三章的數學模式轉變為 GM_2 以及 GM_3 的數學模式。本演算法經過測試後，整體得到的解答較二階段貪婪演算法還好，但在步驟 2 及步驟 3 時，其模式求解的時間表現不穩定，推測可能是 GM_1 所求出的候選路徑品質不穩定，以及 GM_2 和 GM_3 模式的求解時間依然太長，且 GM_2 模式是需要迭代數次的，故更需要將 GM_2 模式改良成更快的模式，才能有效提升整體演算法的表現，下一小節便會介紹如何改良路徑貪婪演算法。

4.3 改良路徑貪婪演算法

在觀察上一小節的路徑組合貪婪演算法後，發現了兩個問題：首先， GM_1 所求出的候選路徑整體品質不穩定，其原因是 GM_1 利用最大化無人機從感測站收集的數據量，也就是說 GM_1 所選出的路徑當中，有越多感測站越好。雖然這樣選有機會選到最佳解，但其缺點為會選到許多同質性的路徑。舉例來說，假設在某個例子中，路徑資料庫 AP 當中，有一些路徑經過了三個同樣的感測站，這些路徑之間的差別可能是路徑中的某個點，例如: $1 \rightarrow 2 \rightarrow 6 \rightarrow 8 \rightarrow 9 \rightarrow 3 \rightarrow 1$ 中，感測站為 6、8 和 3，那 $1 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 9 \rightarrow 3 \rightarrow 1$ 就可以稱作另一條路徑。我們可以發現這兩條路徑的長度、經過感測站數量都相同，唯一的不同是他們分別經過 2 以及經過 5 兩個不同的點，這樣的話這兩條路徑的同質性就相當高。假設 GM_1 要選出十條路徑，而上面中舉例的同質性路徑有七條，那七條便會都被選入候選路徑中， GM_1 則會利用剩下可選的三條，來滿足限制式的其它限制，像是所選路徑要經過所有感測站以及

所選出的路徑中有幾條要比無人機的最小步數還要小。

為了處理這個問題，本研究修改了路徑組合貪婪演算法中 GM_1 選出候選路徑的數量，將原來隨機的整數，改成與無人機相同的數量。這樣的修改將會強制只用無人機數量的路徑來滿足經過所有感測站的限制，因此可降低其所選擇的路徑同質性，保持候選路徑的品質。但由於這樣每次選擇的路徑將變少，所以必須將 GM_1 選出候選路徑的步驟重複幾次，以選出足夠多可以讓 GM_2 挑選的候選路徑組合。

上一小節路徑組合貪婪演算法的第二個問題則是 GM_2 的求解速度太慢，因此必須藉由改良 GM_2 來提升整體演算法的求解速度。在上一節中的路徑組合貪婪演算法中，其步驟 3 時利用 GM_2 來求解出最好的無人機路徑組合，我們測試得知週期性限制的處理時間大約佔了整個 GM_2 的 99% 的時間，因此在本章節中，我們先將原 GM_2 的週期性限制全部刪除，將其改不含週期性限制的 GM_2' ，則其求解速度可大幅提升。也因此可將 GM_1 選出路徑的步驟重複數次，讓 GM_2' 可以一次從大量的路徑中選擇最好的組合。

上述改良作法的好處是可以一次從大量路徑中篩選組合，但如果 GM_1 一直用相同的目標式最大化無人機從感測站收集的數據量，仍會導致其所選擇的最佳路徑組合的組成相似性過高，差異多樣性不夠。因此本研究針對 GM_1 開發其它三種選擇標準，並測試這些選擇標準中何者表現最佳：第一種便是最大化無人機從感測站收集的數據量；第二種則是將所有 GM_1 的候選路徑隨機賦予一個權重，並將目標式改成將權重的總和最大化，這樣做便可以達到隨機選擇的目的；第三種則是將權重改為每條路徑感測站在整段路徑中的比例，目標式改為最大化比例總和，第五章將會分析這三種的選擇標準對整體演算法的影響。

以下為改良路徑貪婪演算法的步驟以及流程圖(圖 4.13):

1. 利用 BFS 演算法在原網路中找出不同長度循環路徑，並將這些路徑存放全路徑清單 (AP)，做為第一階段模式 (GM_1) 的候選路徑。

2. 用第一階段模式(GM_1)來求出候選路徑，這些路徑會進入第二階段候選路徑，並將選擇的路徑從(AP)中刪除，
3. 重複步驟2數次，使得其所選出的路徑達到一定數量供第二階段模式(GM_2)選出無人機的路徑組合。
4. 用第二階段模式(GM_2)，從候選路徑中選出無人機各自的路徑，該無人機路徑的組合便會成為第三階段模式(GM_3)的候選路徑組合。
5. 將步驟3重複數次，以讓第三階段模式(GM_3)有足夠的候選組合可以選擇最佳解。
6. 利用第三階段模式(GM_3)選出最佳的路徑組合。

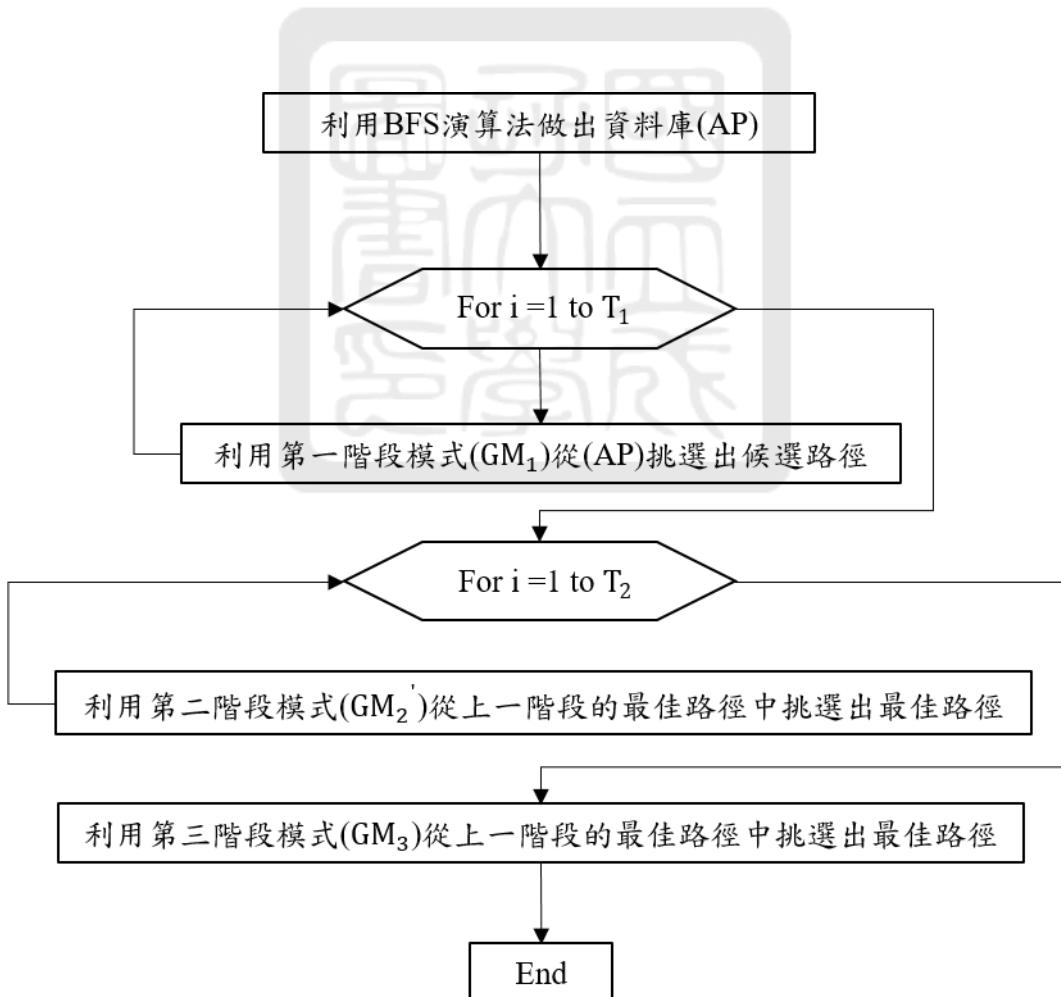


圖 4.13 改良路徑貪婪演算法流程圖

除了改動以上流程步驟以及 GM_1 和 GM_2 ，另外也將上一小節中的路徑刪除方法做些改良。上一小節中若是 GM^D 的刪法，便會刪掉許多沒有被選上的候選路徑，但這些路徑中有可能隱藏著最佳解的路徑，因為藉由第三章模式的求解結果會發現，無人機最終選擇的路徑中通常都覆蓋蠻多的感測站，故可以推測最佳解較有可能會是覆蓋較多感測站的路徑所組成。故採用 GM^D 有可能會刪掉太多可能表現好的路徑，但若採用 GM^S 則是會保留太多類型相似的路徑，導致結果的表現沒辦法有新的突破。因此在改良的路徑組合貪婪演算法中，本研究也有提出一種採用了 GM^D 以及 GM^S 的優勢而開發的折衷刪除法 GM^N ，下圖 4.14 為 GM^N 的方法展示：

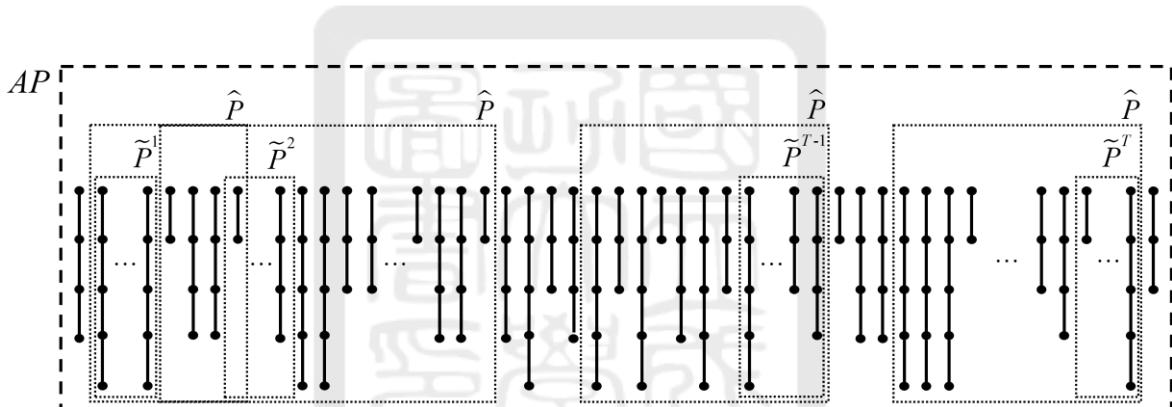


圖 4.14 GM^N 刪除法

在前 $T/2$ 迭代中都採用 GM^D 的刪除方式，而在後 $T/2$ 迭代中則採用 GM^S 的刪除方式，在改良這些地方後，便得到了改良後的路徑組合貪婪演算法，該演算法的表現也比前面兩者的表現還好，所有演算法與模式的比較將會在第五章中做詳盡的數值分析。

4.4 小結

說明完本研究所有數學啟發式演算法後，由於演算法當中有許多模式，每個模

式的功能各不相同，為了能更快速的分辨各個數學模式在其演算法的作用，便設計表 4.1，以下為表 4.1:

表 4.1 各個模式在演算法中的功能

數學模式	模式功能
GM_1	從路徑資料庫 AP 選出適合的路徑組合
GM_2	從適合的路徑組合中，挑選出最佳解
GM_3	從適合的路徑組合中，挑選出最佳解
GM_2'	限制式放鬆的 GM_2 ，可以用更快的速度取得解
GM^D	在步驟 3 中，每次迭代都會有落選的路徑，將落選路徑直接刪除
GM^S	在步驟 3 中，每次迭代會有落選的路徑，將落選路徑保留至下次迭代中，成為候選路徑
GM^N	假設一共 n 次迭代，那前 $n/2$ 次採用 GM^D ，後 $n/2$ 次採用 GM^S ，將兩種方法混合使用

第五章 數值分析

本章將對第三章提出之數學模式以及第四章的演算法進行測試與分析，包括求解大小、速度及品質進行比較。本研究測試環境為 Windows 10 作業系統，搭配 Intel Core i9-10900，2.80GHZ 處理器，與 8G 記憶體。數學模式利用 Gurobi 9.3 版求解。

5.1 數學模式求解結果

本小節針對第三章所提出數學模式進行測試，表 5.1 為數學模式在不同網路圖及參數設定，每種網路規模的例子都會產生 5 個網路圖，這 5 個網路圖的差異在於其感測站位置都不相同，並將求解時間的上限設為 7200 秒(2 小時)。表 5.2 展示 15 個網路例子的求解時間，在 4×4 的網路規模時，平均求解時間為 54.87 秒； 5×5 的網路規模時，平均求解時間為 950.19 秒； 6×6 的網路規模時，有一組網路求解時間超過 7200 秒時還無法求解。從求解時間上觀察，可以看出隨著網路規模上升，求解時間也跟著上升，推斷原因為數學模式是基於時空網路圖為架構進行建模，時空網路圖的大小影響數學模式的求解速度甚巨，一旦節點數量變多，時空網路的大小就會增加，造成求解時間大幅增加。另外可以明顯的看出無論是 4×4 、 5×5 或是 6×6 規模的網路，求解的標準差都蠻大的，代表著在相同規模的網路中，不同的感測站地點，對求解時間影響很大。所以我們將這些觀察納入考量，藉著這些資訊來開發啟發式演算法，期望利用啟發式演算法來處理規模較大的網路。

表 5.1 數學模式數值參數設定

	節點 數量 (N)	感測站數 量 (SN)	無線傳輸距 離 (R)	無人機 數量 (K)	週期性限 制 (P)
Grid_Network_1.1	16	3	1	2	5
Grid_Network_1.2	25	5	2	3	6
Grid_Network_1.3	36	5	2	2	8

表 5.2 數學模式求解結果

<i>N</i>	<i>SN</i>	<i>R</i>	<i>K</i>	<i>P</i>	<i>CPU Time(s)</i>	<i>Obj</i>	<i>Status</i>
16	3	1	2	5	12.92	56	<i>Optimal</i>
16	3	1	2	5	52.48	48	<i>Optimal</i>
16	3	1	2	5	2.45	72	<i>Optimal</i>
16	3	1	2	5	192.73	48	<i>Optimal</i>
16	3	1	2	5	13.80	48	<i>Optimal</i>
25	5	2	3	6	1056.49	96	<i>Optimal</i>
25	5	2	3	6	2219.36	102	<i>Optimal</i>
25	5	2	3	6	212.55	102	<i>Optimal</i>
25	5	2	3	6	131.20	102	<i>Optimal</i>
25	5	2	3	6	1131.39	96	<i>Optimal</i>
36	5	2	2	8	2778.96	92	<i>Optimal</i>
36	5	2	2	8	779.63	126	<i>Optimal</i>
36	5	2	2	8	5216.84	58	<i>Optimal</i>
36	5	2	2	8	3075.27	64	<i>Optimal</i>
36	5	2	2	8	7200	--	<i>Time_limit</i>

由上述表格中可以發現，當網路規模越大，整體的求解時間就會越久。除此之外也可以看出，在不同的例子中，即使網路規模相同的情況下，由於感測站位置的差異，每個例子的求解時間也相差很多，所以可以知道，感測站位置會對求解時間有很大的影響。

除了這些例子之外，本小節也有針對一些參數做敏感度分析，觀察哪種參數因子的變動，會影響大幅數學模式的求解時間。表 5.3 為連通週期的敏感度分析，圖 5.1 為其折線圖，表中使用 5×5 的網路當作範例，在其它參數相同只有連通週期變動的情況下求解，發現連通週期與求解時間有密切的關係。連通週期越大，則模式限制越寬鬆，自然會使得整體求解時間變短。圖 5.2、圖 5.3 和圖 5.4 分別代表三種不同連通週期的飛行路線，從這些飛行路線可以觀察出，當連通週期越長時，無人機更傾向在各個感測站上方收集數據，進而導致整體收集的數據量變多。接著利用圖 5.5 以及圖 5.6 展示出在不同的連通週期下，數據被傳回基站的時刻。可以看出當連通週期較短時，基站收集到數據的時間間隔變短，代表數據的更新速度很快，數據可以維持在較新的狀態來反應整體環境的狀況。相反的，若是連通週期拉長或是甚至不規定有連通週期，則基站的數據更新速度會慢很多，導致整體數據較舊。若將較舊的數據當作決策使用的依據，會無法做出良好的決策。在過往的文獻中，

會利用不同的方式或是強迫一直保持連通來傳輸數據回基站；相較於本研究傳輸數據的方法，過往文獻的方法通常都會導致其收集數據的範圍較小，或是數據的更新速度較慢。基於上述考量，本研究期望能在有限的週期性限制下來傳輸數據，以維持數據的更新速度；為達此目的，必須在開發演算法的過程中，將週期性限制的部分視為核心，針對其做處理才能使演算法的整體求解時間與求解品質提升。

表 5.3 連通週期敏感度分析(sec)

	<i>CPU Time(s)</i>	<i>Obj</i>
$P=6$	2219.36	102
$P=9$	427.02	104
$P=12$	11.07	105



圖 5.1 連通週期敏感度分析折線圖

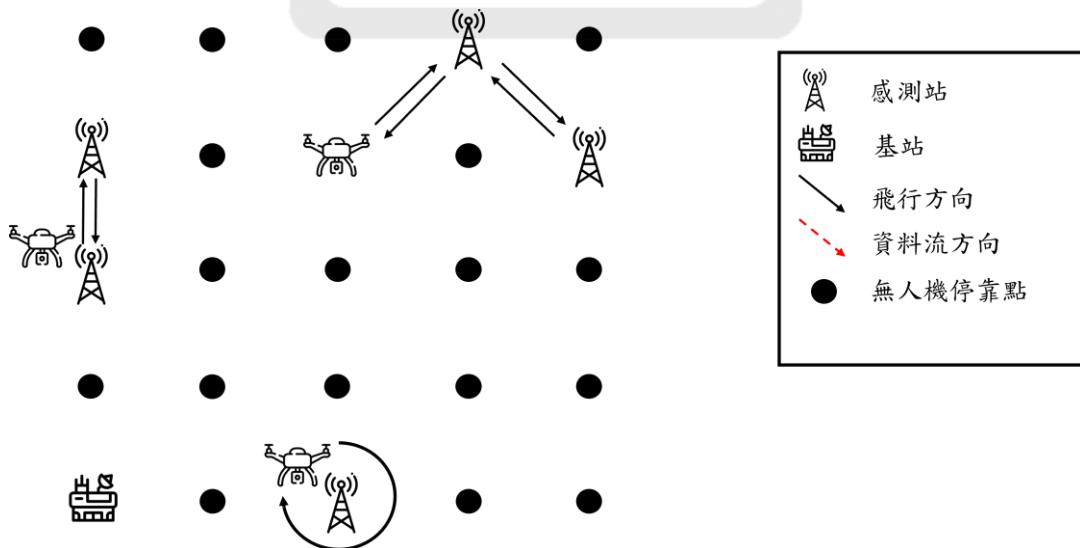


圖 5.2 $P=6$ 時無人機的飛行路線

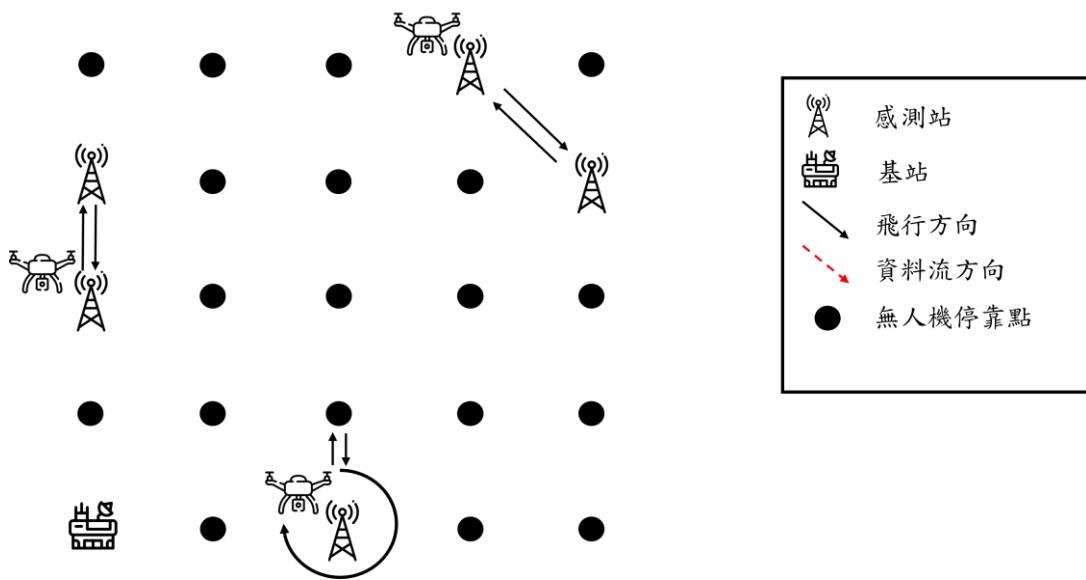


圖 5.3 $P=9$ 時無人機的飛行路線

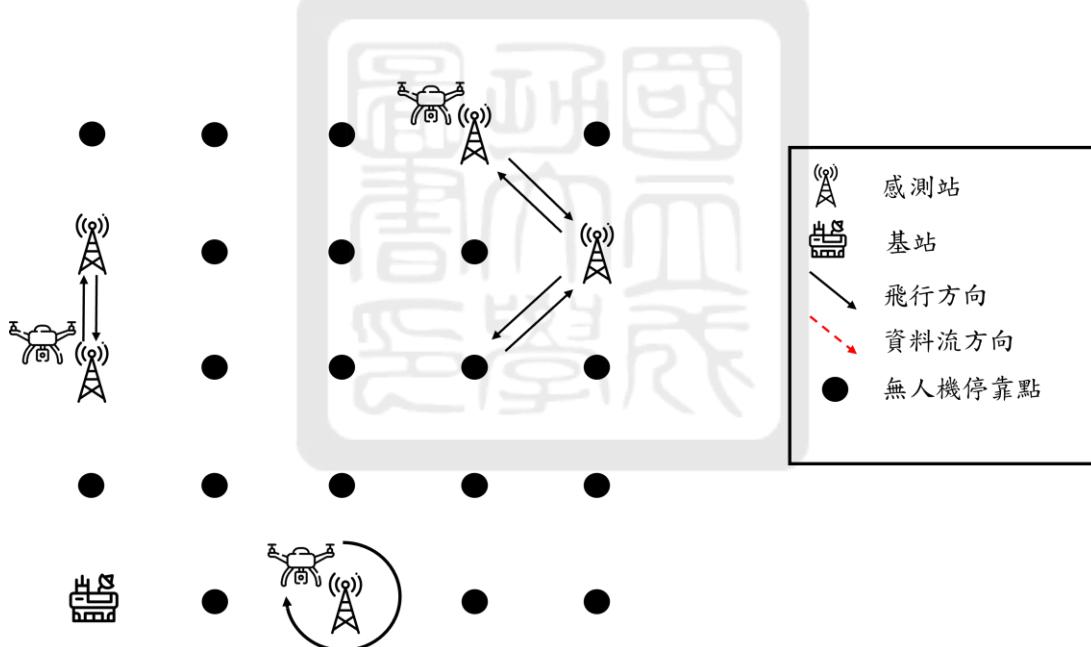


圖 5.4 $P=12$ 時無人機的飛行路線

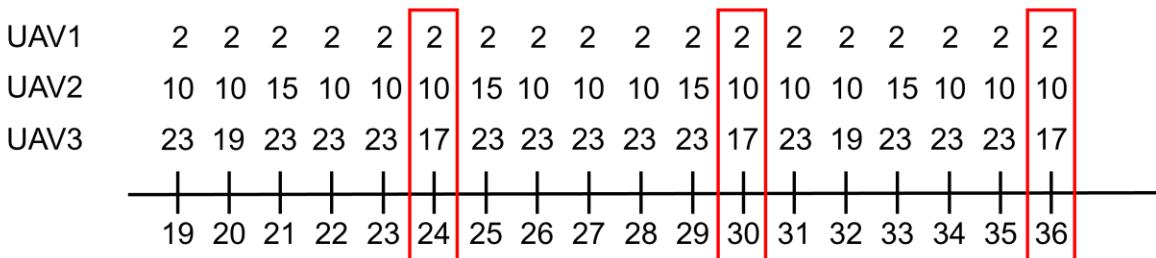
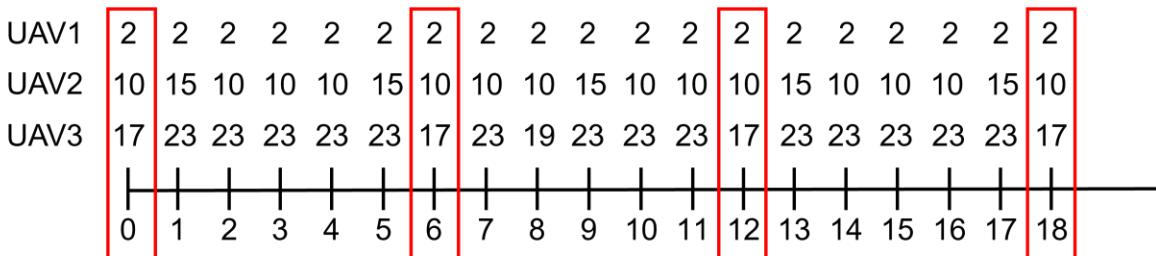


圖 5.5 $P=6$ 時無人機的傳輸數據時刻

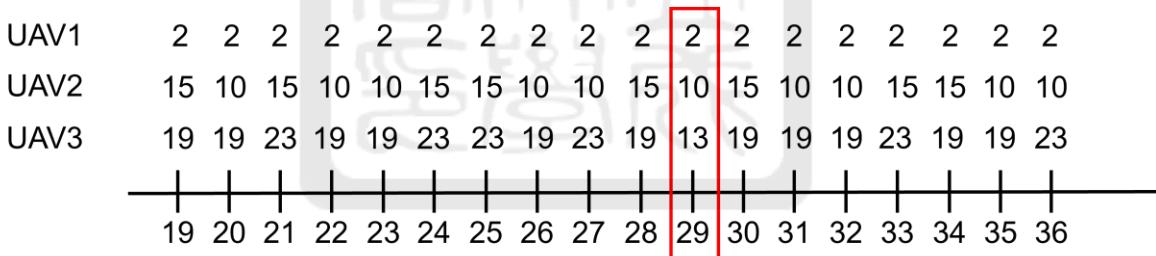
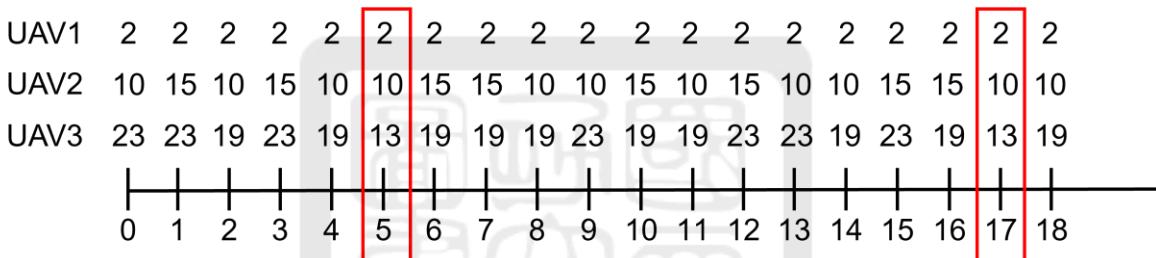


圖 5.6 $P=12$ 時無人機的傳輸數據時刻

表 5.4 則是傳輸距離的敏感度分析，圖 5.7 為其折線圖，可以發現數據傳輸距離變大時對於整體求解時間並無太大影響，但當傳輸數據變小時，整體的求解時間卻大幅上升。從圖 5.8、圖 5.9 和圖 5.10 可以發現，當可傳輸數據的距離較遠時，對整體求解時間影響不大，無人機可以不離感測站很遠就能傳輸數據。在 R 達到一定上限時，其最佳解就會固定，並不再改變，這是因為每台無人機的傳輸距離足夠遠，因此每台無人機在網路中的任何點時，都可以隨時將自身的所收集的數據傳輸

回基站。在達到傳輸距離上限前，其收集的總數據量會隨著傳輸距離越遠而更多。

表 5.4 的例子即 $R=2$ 為上限，故 $R=2$ 和 $R=3$ 的最佳解相同；而 $R=1$ 時，由於限制較多，其傳輸數據距離太短導致其需要花更多時間來找出最佳解，且該最佳解的數值也會相較傳輸距離更遠的情況還差。

表 5.4 傳輸距離敏感度分析(sec)

	CPU Time(s)	Obj
$R=1$	4413.63	84
$R=2$	131.20	102
$R=3$	212.06	102

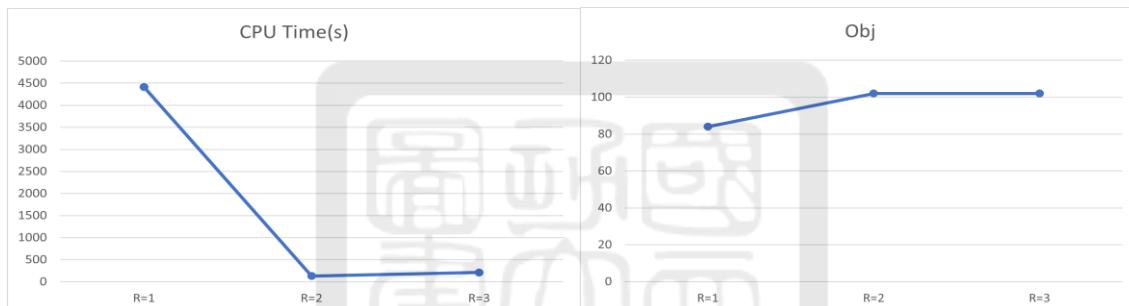


圖 5.7 傳輸距離敏感度分析折線圖

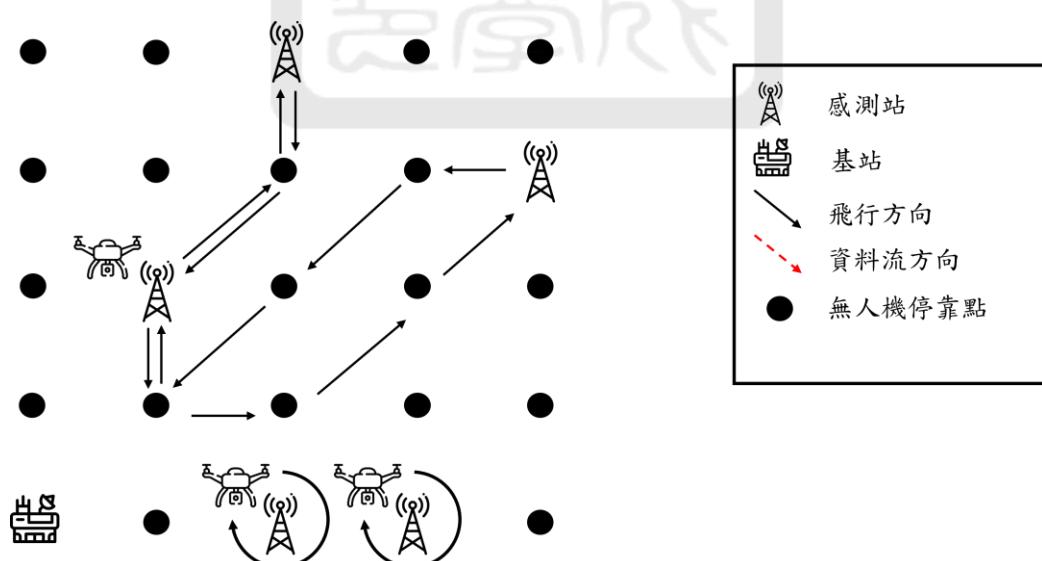


圖 5.8 $R=1$ 時無人機的飛行路線

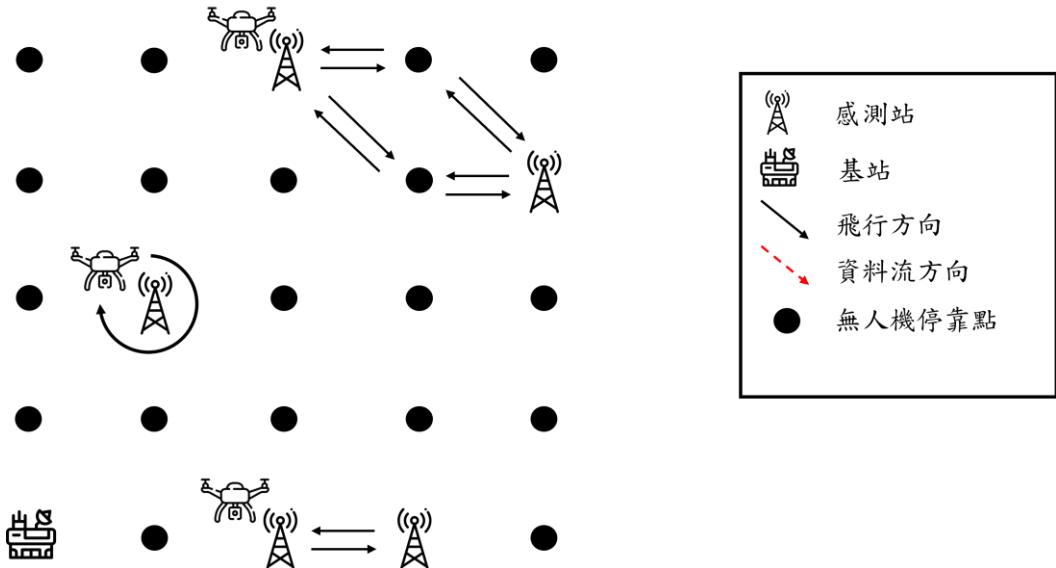


圖 5.9 $R = 2$ 時無人機的飛行路線

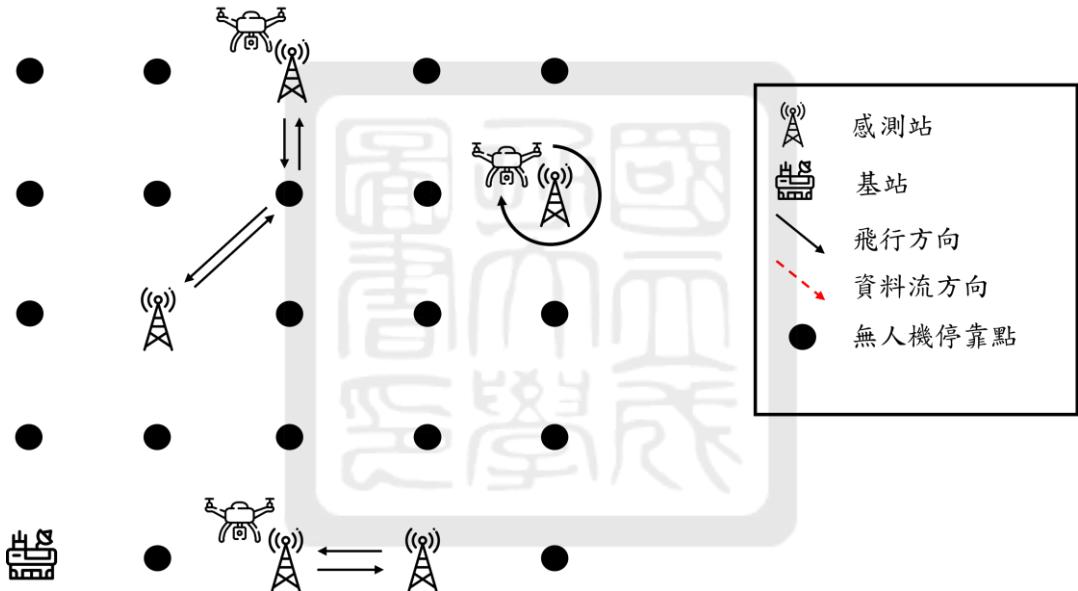


圖 5.10 $R = 3$ 時無人機的飛行路線

藉由傳輸數據距離的敏感度分析，我們可以將其延伸為對不同種類的無人機做敏感度分析，這樣的敏感度分析可以用於實際情況決策時使用。在現今科技發達的社會中，有各式各樣的無人機，這些無人機的可傳輸數據範圍不盡相同，他們之間的價格也有一定的差距，因此可以針對這方面利用數學模式做敏感度分析，藉此判斷在一定的成本上限下，應該使用哪種無人機來執行收集數據的任務。舉例來說，表 5.5 列出不同種類之無人機的價格、無人機可傳輸數據的範圍、每種無人機的使

用數量以及收集的總數據量，從表中可看出使用越多無人機，其所收集的數據越多，但由於不同的成本上限，其最佳決策有可能轉用不同數量的不同類型的無人機。譬如若預算上限只有 40，那最好的方案就是使用四台第一類無人機（總成本剛好為 40），但此方案的模式求解時間需要 6700 多秒才可以得到收集數據 126 次的最佳解；若實務上必須在 5 分鐘(300 秒)內做派遣的決策，此時或可先使用兩台第二類無人機的方案，因為該方案可在 180 多秒內求得目標值為收集數據 60 次的最佳解，因此可在 5 分鐘內先以此方案應付短期的需求，待最佳方案求解完成(6700 秒之後)再伺機轉以最佳方案的派遣方式繼續作業。

表 5.5 不同種類無人機之比較

Type	SN	R	K	P	Cost of 1 Uav	CPU Time(s)	Obj	Status
<i>Uav1</i>	5	1	1	6	10	--	--	Infeasible
<i>Uav1</i>	5	1	2	6	10	849.19	48	Optimal
<i>Uav1</i>	5	1	3	6	10	4413.63	84	Optimal
<i>Uav1</i>	5	1	4	6	10	6783.15	126	Optimal
<i>Uav2</i>	5	2	1	6	20	9.06	21	Optimal
<i>Uav2</i>	5	2	2	6	20	181.16	60	Optimal
<i>Uav2</i>	5	2	3	6	20	131.20	102	Optimal
<i>Uav3</i>	5	3	1	6	30	19.53	21	Optimal
<i>Uav3</i>	5	3	2	6	30	237.36	60	Optimal
<i>Uav3</i>	5	3	3	6	30	212.06	102	Optimal

除了將數學模式應用在實際無人機種類選擇的分析上，本研究也分析相同的節點數量，但在不同形狀的網路中對整體求解時間的影響。在現實環境中，網路的形狀很有可能不是正方形，有可能較為狹長的形狀，為了探討形狀間的差異，本研究用一個 2×8 的網路，與同樣擁有 16 個節點的 4×4 網路，使用同樣的無人機收集數據並加以比較。表 5.6 為兩種不同形狀的網路的比較結果，從表中觀察可以看出，該模式求解結果受形狀的影響非常大。由於將網路改成 2×8 的形狀，若將傳輸數據距離維持在 $R=1$ ，那會導致所有例子都沒有可行解。這是因為形狀改變成狹長形狀的 2×8 網路時，會有許多感測站以及節點離基站相當遠，若只用兩台無人機且 $R=1$ 時，無法同時將數據送回基站且又符合週期性限制；因此，我們將 $R=1$ 改成

$R=2$ 再重新進行測試，重新測試之後發現依然有部分例子解不出來，而就算有解出來的例子，其解也相較原來正方形的網路圖還差上許多。因此，本研究推測無人機在狹長的網路途中極難滿足週期性限制，可能原因在於為了達成週期性的連通限制而無法多收集數據，故相較其它節點數相同但形狀較寬的網路圖來說，狹長形網路的解會差上許多。因此，在狹長的網路中，建議使用數據傳輸距離較長的無人機來收集數據，將週期性連通限制的部分化簡，才能更有效地收集數據。

表 5.6 兩種形狀網路圖比較結果

Shape	N	SN	R	K	P	CPU Time(s)	Obj	Status
2×8	16	3	1	2	5	--	--	Infeasible
2×8	16	3	2	2	5	--	--	Infeasible
4×4	16	3	1	2	5	12.92	56	Optimal
2×8	16	3	1	2	5	--	--	Infeasible
2×8	16	3	2	2	5	--	--	Infeasible
4×4	16	3	1	2	5	52.48	48	Optimal
2×8	16	3	1	2	5	--	--	Infeasible
2×8	16	3	2	2	5	222.45	29	Optimal
4×4	16	3	1	2	5	2.45	72	Optimal
2×8	16	3	1	2	5	--	--	Infeasible
2×8	16	3	2	2	5	191.00	20	Optimal
4×4	16	3	1	2	5	192.73	48	Optimal
2×8	16	3	1	2	5	--	--	Infeasible
2×8	16	3	2	2	5	--	--	Infeasible
4×4	16	3	1	2	5	13.80	48	Optimal

5.2 飛行範圍貪婪演算法求解結果

本小節針對第四章 4.1 小節提出的飛行範圍模式搭配演算法的架構，故在求解時間快速的例子中，並不會有太大改善的作用，因此在本小節中測試的網路為上一小節求解時間超過 1800 秒的網路，期望本研究開發的「飛行範圍貪婪演算法」可以提升求解效率，並觀察該演算法的求解品質。表 5.7 為飛行範圍貪婪演算法的測試結果，顯示飛行範圍貪婪演算法有時無法準確地求出最佳解，又以「飛行骨幹網路方法」較常發生這種情況，推測可能是因為縮小網路限縮太多而導致二階段模式

找不出最佳解；而飛行範圍網路方法的表現亦不甚穩定，在某些網路中提升了求解效率也找出最佳解，但在某些例子中卻無法求出最佳解，也只稍微提升了一些速度。從表看出大部分案例的求解速度都有提升，但是其提升後的速度還是要 1000 多秒；有些甚至還是超過了 1800 秒，並沒有完全達到本研究設計演算法預期能求解較快的目的。本研究希望可以開發出能大幅提升求解速度與品質的作法，在飛行範圍貪婪演算法中，求解品質大部分都很好，但時間方面還是沒什麼突破，推測是因為即使利用飛行骨幹或是飛行範圍來限縮原網路圖，但其縮小幅度不夠導致其模式大小並沒有縮小多少，才無法大幅地改善求解時間。故飛行範圍貪婪演算法較適合用在規模較小的網路當中，在小規模的網路中，該演算法可以比原模式更快找出最佳解，同時兼顧求解速度與品質。

表 5.7 飛行範圍貪婪演算法與數學模式之比較

N	SN	R	K	P	M*		飛行骨幹			飛行範圍		
					CPU Time(s)	Obj	CPU Time(s)	Obj	Gap (%)	CPU Time(s)	Obj	Gap (%)
25	5	2	3	6	2219.36	102	Infeasible	--	--	713.28	102	0
36	5	2	2	8	2778.96	92	Infeasible	--	--	1398.81	78	15.21
36	5	2	2	8	5216.84	58	Infeasible	--	--	330.34	58	0
36	5	2	2	8	3075.27	64	Infeasible	--	--	1856.56	64	0
36	5	2	2	8	7200	--	Infeasible	--	--	6615.69	138	--

5.3 路徑組合貪婪演算法求解結果

本小節針對第四章 4.2 小節提出的路就組合貪婪演算法進行測試，採用 5.2 小節測試飛行範圍貪婪演算法相同的測資，並將這些結果與原數學模式的結果做比較。表 5.8 為路徑組合貪婪演算法的測試結果，從表中可以看出路徑組合貪婪演算法的時間求解方面不太穩定，有時較數學模式還快，但有時卻比較慢。推測可能在 GM_2 的階段耗時太久，雖然只要從特定的路徑組合中選出符合週期性連通限制的最佳組合， GM_2 的求解時間一定比原模式還快速；但該演算法必須重複該步驟數次，導致整體求解的時間有時會比原模式還要長。然而解的部分，雖然提供所有可

能的路徑組合給演算法挑選，但演算法很難剛好選出最好的組合，只能依照過往的求解經驗將 GM_1 的目標式設為覆蓋最多感測站的路徑，故路徑組合貪婪演算法的解很難達到最佳解。從表中可以看出該演算法得到的解與原模式最佳解的 Gap 值平均都落在 25%，其實差距不小，由於本演算法的表現不是很好，所以本研究再進一步開發了改良路徑貪婪演算法。

表 5.8 路徑組合貪婪演算法與數學模式之比較

N	SN	R	K	P	M^*		GM^D			GM^S		
					CPU Time(s)	Obj	CPU Time(s)	Obj	Gap (%)	CPU Time(s)	Obj	Gap (%)
25	5	2	3	6	2219.36	102	1836.52	85	16.66	1796.78	76	25.49
36	5	2	2	8	2778.96	92	3204.36	63	31.52	2963.45	60	34.78
36	5	2	2	8	5216.84	58	4951.98	44	24.13	4672.82	39	32.75
36	5	2	2	8	3075.27	64	3286.21	50	21.87	3324.15	43	32.81
36	5	2	2	8	7200	--	5716.66	108	--	5967.14	105	--

5.4 改良路徑貪婪演算法求解結果

本小節針對第四章 4.3 小節提出的改良路徑貪婪演算法進行測試，採用 5.2 小節測試飛行範圍貪婪演算法相同的測資，並將這些結果與原數學模式的結果做比較。表 5.9 為改良路徑貪婪演算法的測試結果，從結果中可以發現，改良路徑貪婪演算法的求解時間相較於原模式已經縮短許多，且求出的解與最佳解的 Gap 值，相較於改良的路徑組合貪婪演算法皆有大幅的改善。在測試完三種演算法後，本研究挑選飛行範圍貪婪演算法與改良路徑貪婪演算法來測試更大的網路，表 5.10 為其測試結果，該表包含 10×10 、 15×15 和 20×20 等三種不同規模的網路。經過測試發現在小規模網路中表現良好的，飛行範圍貪婪演算法在 10×10 的網路規模時就沒辦法求出解；改良組合演算法針對同例雖然有解，但是因為網路規模較大，所以路徑資料庫 AP 會有相當多的路徑。為了能讓改良組合演算法可處理此 10×10 的網路，我們修改 BFS 演算法使其能一次求解較少條路徑（因為若直接一次將所有長度的路徑放入資料庫中，會導致演算法無法求解），由於將全部路徑分裝在許多資料庫，

所以沒辦法準確地讓所有資料庫當中的路徑相互組合比較，也導致其求解結果距離最佳解有一定的差距。

表 5.9 改良路徑貪婪演算法與數學模式之比較

N	SN	R	K	P	M*		GM ^N		
					CPU Time(s)	Obj	CPU Time(s)	Obj	Gap (%)
25	5	2	3	6	2219.36	102	197.30	102	0
36	5	2	2	8	2778.96	92	830.07	82	10.86
36	5	2	2	8	5216.84	58	298.36	53	8.62
36	5	2	2	8	3075.27	64	423.68	60	6.66
36	5	2	2	8	7200	--	3859.06	125	--

表 5.10 大規模網路測試結果

N	SN	R	K	P	飛行範圍貪婪演 算法		GM ^N	
					CPU Time(s)	Obj	CPU Time(s)	Obj
100	12	2	8	7	--	--	15673.68	396
255	18	2	12	10	--	--	57128.16	571
400	24	2	16	13	--	--	83652.93	863

5.5 感測站位置敏感度分析

由於在 5.1 小節針對數學模式進行數值分析時，從測試結果中得知，不同的感測站位置對整體求解的時間影響很大，因此以下便針對感測站位置做一系列的測試，期望在不同的感測站位置分布時，能使用相對應的解法來提升求解效率以及品質。由於在上述測試的過程中，感測站位置的分布較為均勻，所以在這次的測試中，會挑選一些較極端的例子來進行測試，像是感測站位置高度集中或是高度分散等情況。在下面的表格中，一共有六種網路，六種網路都是相同的 6×6 規模以幾兩台無人機，這些網路的不同之處在於其感測站分布的情形，以下為測試的網路圖片：

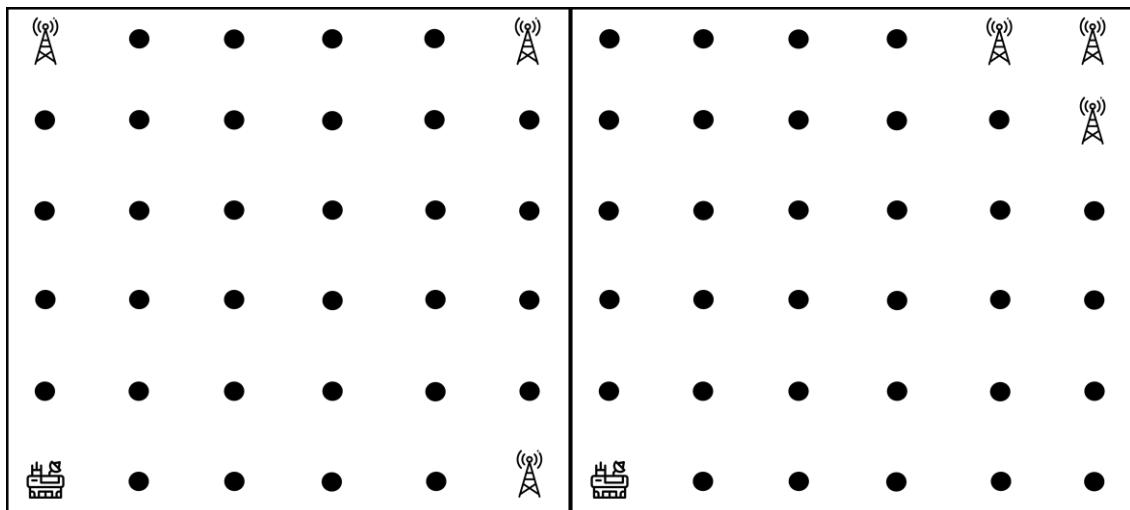


圖 5.11 網路 1 (左) 和網路 2 (右)

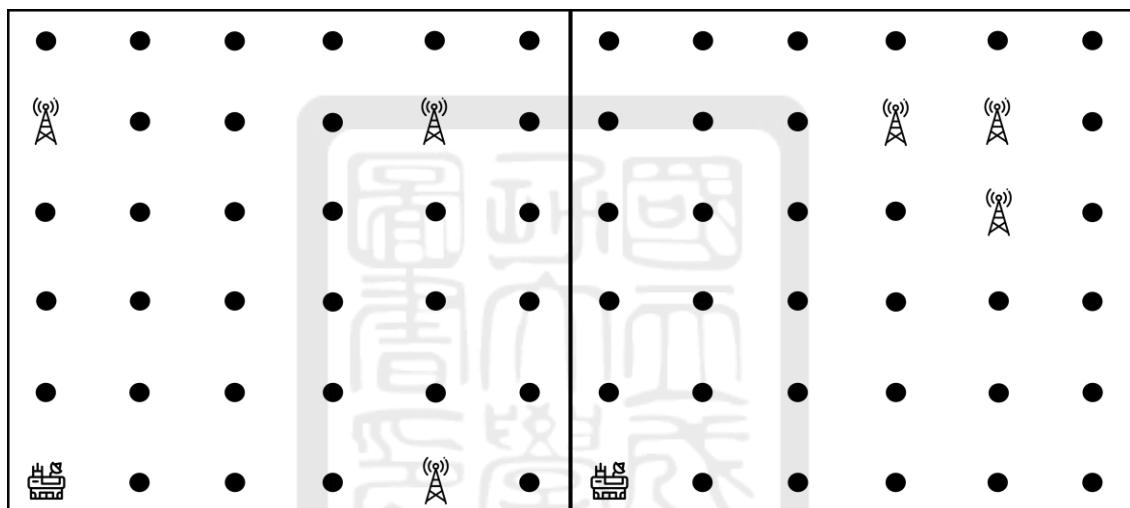


圖 5.12 網路 3 (左) 和網路 4 (右)

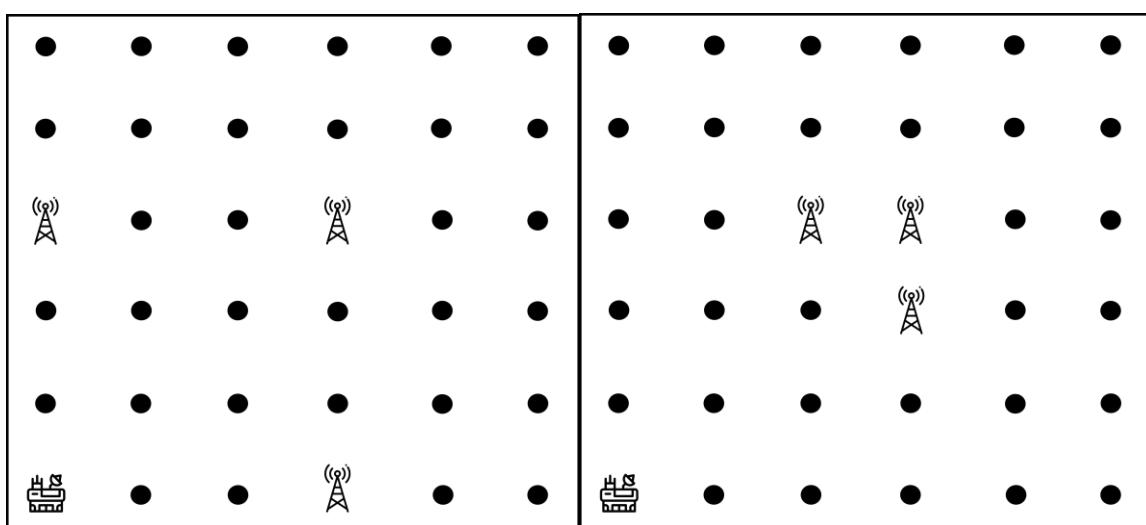


圖 5.13 網路 5 (左) 和網路 6 (右)

這些圖片可以分為三組，其中圖 5.11 中的網路 1 和網路 2 為同一組，網路 1 為感測站分散的情況，網路 2 為感測站密集的情況，兩個網路的共通點為其感測站與基站的距離都是相同的，距離為 5。而圖 5.12 中的網路 3 與網路 4 被分為同一組，網路 3 為感測站分散的情況，網路 4 為感測站密集的情況，這兩個網路與基站的距離縮短為 4。最後則是圖 5.13 中的網路 5 與網路 6，網路 5 為感測站分散的情況，網路 6 為感測站密集的情況，這兩個網路與基站的距離縮短為 3，每個網路組別的差別在於感測站與基站間的距離，每個組別都有一個分散的以及密集的網路。接著利用本研究的演算法求解這六個網路，其結果顯示於表 5.11：

表 5.11 感測站位置敏感度分析

	M^*		飛行範圍貪婪演算法			GM^N		
	CPU $Time(s)$	Obj	CPU $Time(s)$	Obj	Gap (%)	CPU $Time(s)$	Obj	Gap (%)
網路 1	2611.66	24	1572.35	24	0	983.96	24	0
網路 2	>10000	--	>10000	--	--	732.82	56	--
網路 3	1017.95	98	135.81	98	0	543.44	98	0
網路 4	796.27	110	191.83	110	0	277.44	110	0
網路 5	1706.31	112	79.85	112	0	225.27	112	0
網路 6	152.37	130	17.59	130	0	195.49	130	0

由表 5.11 結果可觀察數學模式以及不同演算法在六種網路中的表現，其中網路 1、3、5 都是屬於感測站分散的網路，網路 2、4、6 為感測站集中的網路。在實作敏感度分析之前，我們預期感測站集中的網路的求解時間應該要比感測站分散的網路的求解時間還要短，因為感測站集中代表無人機可以輕易地在保持週期性連通的同時來收集感測站的數據，應該相較分散的感測站還要更容易找出解。然而，實測之後卻發現結果與預期不同：在用數學模式求解的情況下，可以看出感測站分散的網路 1 其求解時間相較感測站集中的網路 2 還要短，而用飛行範圍貪婪演算法求解也同樣是感測站分散的網路 1 和網路 3 的求解時間相較感測站集中的網路 2 和網路 4 還要短，此兩種作法的測試結果無法證明感測站的集中與分散對求解時間是否有

一定的影響；然而，改良路徑貪婪演算法的結果卻可看出求解感測站分散的網路的確會比感測站集中的網路還快，此與先前的推測相同。

不同的求解方法的時間耗費會有如此差異，原因在於求解方法的設計本質不同。數學模式以及飛行範圍貪婪演算法都是基於網路上直接規劃路徑，故感測站的集中或是分散對求解時間的影響較小，反而是整體的網路規模對兩者的影響最直接。而改良路徑貪婪演算法是將路徑都放入路徑資料庫中，所以當感測站集中時，放入路徑資料庫的路徑會比較少，因為可以用較短的路徑通過所有感測站。然而，當感測站分散時，必須要用較長的路徑才能通過所有感測站；而路徑資料庫是將一定長度以內的路徑都放入資料庫中，所以需要的路徑長度越長，整個路徑資料庫的規模會越大，導致求解時間越長。當求解時間太長時，就會使用先前提到用的方法將路徑限縮在網路較有可能有解的部分，以部分的路徑來找出最佳解。

另外，本敏感度分析的網路也可從距離的遠近來觀察分析。如同先前所述六個網路一共可以被分為三組，其主要差別在於感測站距離基站的距離長度。從表格中觀察可以看出無論是哪種求解方法，距離較遠的網路 1 和網路 2 的求解時間最久，接著是網路 3 和網路 4，網路 5 和網路 6 則是最快的。由此可以發現感測站與基站的距離大小是會影響整體的求解時間的，當感測站距離基站較遠時，求解時間較長；當基站與感測站距離較短時，其求解時間較短。

綜合以上的觀察，感測站的位置對於整體求解時間的影響真的很大，在同一規模的網路中，有些感測站的位置會導致無法求出可行解，有些感測站的位置則是能很容易的求出可行解。在極端網路中，不同的演算法各有其優劣，但都比原來的數學模式還好，此結果與先前的發現雷同：在小規模的網路適合使用飛行範圍貪婪演算法，而大規模的網路依然要利用改良路徑貪婪演算法來求解，除此之外，也發現在極端的網路中，可以將改良路徑貪婪演算法一開始用 *BFS* 演算法所選擇的路徑限制在某一區域，這樣的作法不但可以提升求解時間，也能讓該演算法更容易找出最佳解。

第六章 結論與未來研究方向

6.1 結論與貢獻

由於科技越來越發達，無人機所能做的事情也越來越多，由於其具高可靠度、效率、與安全性，十分適合被用來執行大範圍的重複性區域監測巡邏等任務。本研究即以此類問題為主要探討的主題，亦即針對座落於大範圍中各區域的感測站，派遣無人機前去收集其感測資料再回傳給基站，我們希望能顧及資料的新鮮度與時效性，且希望能在規律的週期內在基站收到各感測站的資料。

因為探索的範圍較大，所以收集環境數據的感測站都會離基站較遠；加上通常自然環境中網路訊號都會比較微弱，因此感測站難以直接將數據以無線通訊直接傳回基站。此外，監控自然環境需要投入長時間，才能從持續收集的數據看出環境變化。故本研究期望能找出多台無人機的循環路徑，使得這些無人機可以持續地在各自的循環路徑上收集數據，並規定所有無人機必須在時效性上限的區間內，週期性地與基站完全連通，讓數據在保持時效性的狀態下被傳輸回基站。綜合以上這些因素使得本研究無人機的路徑規劃問題相當有挑戰性。

本研究在第二章中，對於無人機收集數據的問題，首先回顧了過往無人機收集數據的相關問題，從中歸納出各種無人機在收集數據時遇到的困難點，並嘗試在本研究中解決這些困難點。過往的相關研究大多著重在降低「數據延遲性」(latency)，所謂的數據延遲性就是當一份數據從感測站被取出時，需經過多久的時間，才能將無人機攜帶的數據帶回基站。因為數據都有時效性，所收集到的數據若超過其時效將被視為過期，不適合被用來當決策的參考，因此處理數據延遲性是無人機收集數據必須處理的一項核心問題，本研究也會針對數據延遲性以及數據時效性的問題去做處理。除了從這些文獻中觀察出數據延遲性的問題，在各種文獻中，無人機的合作方式以及收集數據的方法也都不盡相同，較常見的方式為無人機各自收集數據，在遇到另一台無人機時將自身所收集的數據傳輸給另一台無人機，該台無人機在將

那些數據帶回基站。然而這樣的連通方式會使得數據延遲量變大，因為數據是由接力的方式慢慢傳輸回基站，萬一接收數據的無人機中途必須去某感測站收集數據，便會使得原來在該台無人機上的數據延遲量變大。針對上述缺點，本研究採用週期性連通傳輸數據的方式，亦即在某些時刻，所有無人機會與基站形成一個傳輸數據連通的網路圖，當該網路連通完成時，所有的無人機會將自身的數據全部即時傳至基站，使得基站可以一次接收所有無人機所收集的數據。使用這種方式不但可以處理數據時效性的問題，也能使得數據延遲性大幅降低，使得本研究可以將整個研究問題的目標放在極大化整體的數據總收集量。

另外，部分文獻會進一步考量無人機的電量消耗，由於本研究探討長期的週期性循環路徑，所以本研究先假設無人機的續航力足夠，才能著重於通訊連通的飛行路徑規劃上。但若真的需要將續航力列入後續研究的考量，我們建議可在所求解出來的這些循環路徑上再建構充換電站，至於該在哪些節點上建構充換電站這個議題，我們建議可將之視為一個的 facility location 問題去處理。

本研究在第三章探討多台無人機協同最大化收集數據之協同循環路徑規劃問題，試圖找出無人機的循環路徑，這是因為通常執行監測環境的任務應該為永久或是長時間的。過往的研究中大多使用演算法來處理此類型的複雜問題，較少使用數學模式來求解；而那些使用了數學模式的文獻，也大多再將題目大幅簡化才推導其數學模式，這跟本研究自始即針對完整的問題來設計完整的混整數數學規劃模式，並以 Gurobi 求解無人機的循環路徑作法有所不同。本研究提出的數學規劃模式已考量了數據延遲性跟數據時效性，也能明確地提供每台無人機在每個時刻所停留的位置，並已平均分配每個感測站被收集數據的次數，不會讓無人機只收集離基站較近的感測站數據，使得整體數據偏差在某一區域而已。相較於相關文獻大多僅用演算法求解設定更簡化的問題，本研究是第一個利用混整數規劃模式來處理設定較完備複雜的問題。然而，也因為問題過於複雜，當無人機可以停靠的節點數增加時，整體網路規模會變大，勢必會導致求解時間大幅增加。

接著，為了處理規模較大的問題，本研究嘗試開發可兼具效率與效能的「數學啟發式演算法」，在第四章中提出了三種以貪婪演算法為架構、整數規劃模式為求解核心的數學啟發式演算法。第一種演算法為「飛行範圍貪婪演算法」，該演算法主要發想的求解邏輯是先從整個網路中篩選出重點飛行範圍，再用第三章的數學模式來求解。因為輸入的網路變小了，整體的求解時間亦將變短；然而，測試結果顯示此法在一開始利用數學模式限縮網路範圍時常常會將整個網路過度限縮，導致連可行解都經常找不出來；後來我們曾試圖將限縮的範圍變成較寬鬆，如此雖然能找出可行解，但其整體求解時間並沒有縮短很多。第二種「路徑組合貪婪演算法」則不像「飛行範圍貪婪演算法」那般使用數學模式來限縮飛行範圍，而是師法路徑規劃常使用的「列產生法」來試圖產生許多可能的候選路徑，接著從這些路徑中嘗試找出最佳解，藉由不同的準則來挑選候選路徑，這些準則都是由第三章的數學模式求解結果來判斷的，該演算法雖然已經將整體可以選擇的網路縮小，但是由於還要計算排列組合，從所有候選路徑中挑選最好的組合，一次迭代的時間會比第三章的數學模式快上許多，但是必須多方比對路徑組合，執行多次迭代才能找出較好的路徑組合，所以整體迭代完的時間有時候會比原來第三章的數學模式還要久。由於第二種演算法的求解表現與求解時間不穩定，本研究改良了路徑組合貪婪演算法來當作第三種演算法「改良路徑貪婪演算法」，為了提升求解效率，將該第二種演算法用來挑選最佳的路徑組合的數學規劃模式限制稍微放鬆，使其能更快地找出最佳路徑組合。在此三種演算法中，改良路徑貪婪演算法的求解時間最快，雖然不保證可找出最佳解，但其整體的求解效率以及品質已經較前兩種演算法有大幅改善。

第五章記錄了第三章的數學模式以及第四章演算法的數值分析結果，也針對數學模式做了參數的敏感度分析。從敏感度分析中得知，週期性連通的週期長短，對數學模式整體的求解時間有很大的影響，若週期性連通的週期越短，則整體求解時間相對較久，且能收集的數據也比較少；反之若是週期變長，則能讓無人機收集更多數據，而不是花費大多數的時間在傳輸數據。另外，本研究在敏感度分析中實測

了本數學模式若是應用到實際案例中，在一定的成本上限中，該如何去選擇無人機的種類以及數量，能讓整個數據的收集量最大化。此外，我們也通過網路形狀的敏感度分析發現在狹長形狀的網路中，必須要使用較多或是傳輸數據較廣的無人機來執行任務。最後，我們也讓表現較好的飛行範圍貪婪演算法和改良路徑貪婪演算法測試較大的網路，實測結果可以看出飛行範圍貪婪演算法的表現較差，可能是因為當網路實在太大時，即使僅考慮已縮小飛行範圍的網路，其規模對數學模式來說還是太大；而改良路徑貪婪演算法的表現雖然好一些，但整體求解時間還是明顯上升，因此本研究的數學啟發式演算法針對處理大規模的網路，尚有不少的改善空間。

綜合上述，本研究目前貢獻如下：

1. 提出多台無人機最大化收集數據之協同循環路徑規劃問題，參考過往的文獻，提出一些新的無人機合作傳輸數據的方式，首度以混整數規劃模式探討多台無人機協同收集數據之問題。在已知所有無人機的數量、循環路徑大小、數據的時效性，以及各個感測站必須被收集數據的次數的限制下，來決定所有無人機的循環路徑以最大化收集的總數據量，除確保了數據的時效性以外，也盡可能地降低數據延遲性。
2. 對數學模式的各項已知參數實測敏感度分析，利用敏感度分析實測現實環境中的問題，例如在成本限制下，應該選擇的無人機種類以及使用的數量，或是在不同形狀的網路應該派出何種無人機執行任務，將此模式應用在一些可能的實際決策當中。
3. 提出三種數學啟發式演算法來處理多台無人機最大化收集數據之協同循環路徑規劃問題，為了提升整體的求解速率並維持最佳解的品質，嘗試以數學模式搭配貪婪演算法來實作，並成功地在中小規模的網路中，能以快速的求解時間求出好的解。在三種演算法中，若網路規模小，則利用飛行範圍貪婪演算法可以得到較好的解；若網路規模較大則比較適合利用改良路徑貪婪演算法來求解。

6.2 未來研究方向

到目前為止，本研究雖然對多台無人機最大化收集數據之協同循環路徑規劃問題做整體探討，但仍有未臻完善之處，以下列舉幾個未來可延伸之議題：

1. **自然環境因素：**本研究雖然有針對無人機種類以及網路形狀做敏感度分析，但是尚未考慮無人機群在任務期間移動的環境變化（例如，建築物、風、天氣等）。這些天氣因素可能會導致無人機在執行任務的期間遇上困難，例如下雨時無人機的數據傳輸距離應該更短一些；或是在強風時，無人機會較慢抵達下一個停靠點；抑或是一些建築物阻隔了無人機的傳輸路徑以及規範的禁航區等等，這些環境因素的變化現階段皆可以藉由調整本研究的參數以列入考量。本研究目前先著重於探討理想網路（無突發的自然環境變化）中無人機循環路徑的規劃作法，而考量環境變化該如何做較好，可當成未來研究的選項。
2. **設計更好的數學模式：**在現在數學模式的設計當中，因為無法克服讓每台無人機該如何自行決定其循環路徑的大小，所以需要給定每台無人機的移動步數，並強制總體時間窗必須為無人機步數的最小公倍數，如此才較能滿足本研究設定的週期性循環的限制。然而，上述做法將會限縮無人機的路徑可能，未來希望可以開發出不用限縮無人機移動步數與總體時間窗呈倍數關係的數學規劃模式或演算法。
3. **在一般啟發式演算法中搜尋初始解與鄰近新解的方法：**本研究目前只開發數學模式搭配贪婪演算法的數學啟發式演算法，而不是傳統較常被使用的啟發式演算法。主要原因之一是本研究的複雜度過高，很難讓初始解或是新的鄰近解能滿足通訊網路連通與資料同步的排程與路徑可行性。以尋找初始解來說，現階段可能的作法為，先找出所有可以與基站形成傳輸數據通道的無人機位置，而這些通道會形成一個以基站為起點的樹，再來要根據數據時效性隨機選擇要傳輸數據的時刻，接著還要再從那棵傳輸數據通道的樹中隨機選擇與傳輸時刻

相對應的傳輸數據通道，這樣配對下來就可以知道哪些時刻，無人機必須在哪些特定的點互相配合並傳輸數據回基站，接著再幫每台無人機選擇其所負責的感測站，該台無人機便可以用 *BFS* 演算法尋找兩點之間的最短路徑，藉此找出每台無人機的循環路徑。

舉例來說，如圖 6.1，假設現在有一個 5×5 的網路，且有三台無人機，假設其中一號無人機負責收集感測站 5 和 12 的數據，且也選定傳輸數據的時刻，假設在無人機連通傳輸數據時，三台無人機選擇傳輸數據的通道為(14→7→1)和(13→6→1)，從這兩個通道各挑一個節點(譬如節點 7、6)給一號無人機，則一號無人機必須在節點 5、12、7、6 上利用 *BFS* 演算法造出一條循環路徑，該路徑即可當一號無人機的初始路徑。上述所提的方式只是一個大略的架構，因為即使找出該無人機必須通過的感測站點，也找出了路徑，但還是不能保證三台無人機的路徑可以在可以在數據時效性內一直將數據傳回基站。為了達成這些目的，在找出初始路徑時還會有許多限制好讓無人機各自的路徑可以互相配合，故找出初始路徑以及鄰近路徑解並不是只要找出節點分配就能完成，還必須讓無人機各自的路徑配合傳輸數據。

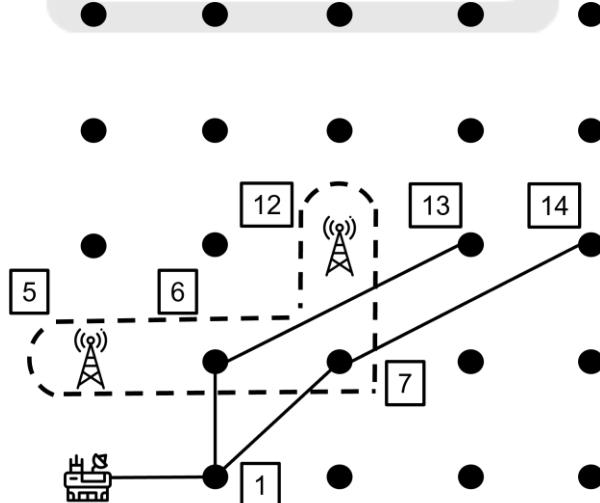


圖 6.1 初步初始解生成法

在尋找鄰近解方面，由於本研究的無人機是可以停在原地的，這將導致每次無人機可以移動的選項多達 9 種。假設要找出一條循環路徑長度為 10 的鄰近路徑，那就有 9^{10} 種路徑的組合。假設有三台無人機，那這三台無人機所選的路徑之間還要相互配合以滿足週期性限制。在這麼多選項中，無論是初始解或是鄰近解的尋找都相當困難。若隨意挑選初始解，如何保證可行？就算可行，可當用初始解找尋鄰近解時，同樣由於可以選擇的鄰近解太多，以及鄰近解必須要與其它路徑相互配合，導致可能的組合太多，在大規模的情況下，很容易導致演算法在選定初始解後就停留在區域最佳解。也就是說，光是地理位置的路徑要顧到可連通之外，在時間上還要達成資料傳輸的同步化，且還要保持資料時效性，要能同時滿足這些要求的初始解其實已經很難達成，遑論之後若採取一般啟發式演算法常用的「稍微調整」（譬如鄰域搜尋法、禁忌搜尋法、基因演算法、模擬退火法等）再收斂的作法，那將導致調整之後的路徑與時間上很容易就難以配合再調回來。因此若將這種微調方式完全交由演算法去「盲目」地調整，極可能無法收斂，甚至連「可行解」都不易計算而得，故本研究目前仍持續在探討較有效率尋找初始解與新鄰近解的方法。

4. **發展更有效之啟發式演算法：**本研究的改良路徑貪婪演算法在遇到大規模的網路時，可能會倚賴人工的方式觀察該網路中可能有最佳解的區域，提前將 BFS 搜尋路徑的範圍縮小，減少演算法求解的路徑數量，才能讓整體演算法順利運行。目前的作法只是一個折衷的作法，未來希望能將此機制以更具體的系統化方式處理。另外本研究希望未來能在開發完上述第三點尋找初始解與鄰近解的方法之後，能將該方法套入諸如鄰域搜尋法、禁忌搜尋法、基因演算法、模擬退火法等各種演算法，比較各個演算法之間的差異，再試圖分析各個演算法的優缺點並加以改良，期能開發出能快速解出本研究問題的啟發式演算法。

參考文獻

- Acevedo, J. J., Arrue, B. C., Diaz-Banez, J. M., Ventura, I., Maza, I., & Ollero, A. (2013, May). Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications. In 2013 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 565-574). IEEE.
- Acevedo, J. J., Arrue, B. C., Diaz-Bañez, J. M., Ventura, I., Maza, I., & Ollero, A. (2014). One-to-one coordination algorithm for decentralized area partition in surveillance missions with a team of aerial robots. *Journal of Intelligent & Robotic Systems*, 74(1), 269-285.
- Alemayehu, T. S., & Kim, J. H. (2017). Efficient nearest neighbor heuristic TSP algorithms for reducing data acquisition latency of UAV relay WSN. *Wireless Personal Communications*, 95(3), 3271-3285.
- Banfi, J., Basilico, N., & Amigoni, F. (2015, September). Minimizing communication latency in multirobot situation-aware patrolling. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 616-622). IEEE.
- Banfi, J., Li, A. Q., Basilico, N., Rekleitis, I., & Amigoni, F. (2016, May). Asynchronous multirobot exploration under recurrent connectivity constraints. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5491-5498). IEEE.
- Banfi, J., Quattrini Li, A., Rekleitis, I., Amigoni, F., & Basilico, N. (2018). Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots*, 42(4), 875-894.
- Flores-Luyo, L., Agra, A., Figueiredo, R., & Ocaña, E. (2020). Mixed integer formulations for a routing problem with information collection in wireless networks. *European Journal of Operational Research*, 280(2), 621-638.
- Hollinger, G., & Singh, S. (2010, May). Multi-robot coordination with periodic connectivity. In 2010 IEEE International Conference on Robotics and Automation (pp. 4457-4462). IEEE.
- Kantaros, Y., & Zavlanos, M. M. (2016). Distributed intermittent connectivity control of mobile robot networks. *IEEE Transactions on Automatic Control*, 62(7), 3109-3121.
- Manyam, S. G., Rasmussen, S., Casbeer, D. W., Kalyanam, K., & Manickam, S. (2017, June). Multi-UAV routing for persistent intelligence surveillance &

- reconnaissance missions. In 2017 international conference on unmanned aircraft systems (ICUAS) (pp. 573-580). IEEE.
- Pasqualetti, F., Durham, J. W., & Bullo, F. (2012). Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. *IEEE Transactions on Robotics*, 28(5), 1181-1188.
- Pasqualetti, F., Franchi, A., & Bullo, F. (2012). On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 28(3), 592-606.
- Scherer, J., & Rinner, B. (2017, September). Short and full horizon motion planning for persistent multi-UAV surveillance with energy and communication constraints. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 230-235). IEEE.
- Scherer, J., & Rinner, B. (2019). Persistent multi-UAV surveillance with data latency constraints. *arXiv preprint arXiv:1907.01205*.
- Scherer, J., & Rinner, B. (2020a). Multi-robot persistent surveillance with connectivity constraints. *IEEE Access*, 8, 15093-15109.
- Scherer, J., & Rinner, B. (2020b). Multi-UAV surveillance with minimum information idleness and latency constraints. *IEEE Robotics and Automation Letters*, 5(3), 4812-4819.
- Scherer, J., & Rinner, B. (2020c). Multi-robot patrolling with sensing idleness and data delay objectives. *Journal of Intelligent & Robotic Systems*, 99(3), 949-967.
- Spirin, V., & Cameron, S. (2014, October). Rendezvous through obstacles in multi-agent exploration. In 2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014) (pp. 1-6). IEEE.
- Spirin, V., Cameron, S., & Hoog, J. D. (2013, August). Time preference for information in multi-agent exploration with limited communication. In Conference Towards Autonomous Robotic Systems (pp. 34-45). Springer, Berlin, Heidelberg.
- Vera-Amaro, R., Rivero-Ángeles, M. E., & Luviano-Juárez, A. (2020). Data collection schemes for animal monitoring using WSNs-assisted by UAVs: WSNs-oriented or UAV-oriented. *Sensors*, 20(1), 262.
- Zhan, C., Zeng, Y., & Zhang, R. (2017). Energy-efficient data collection in UAV enabled wireless sensor network. *IEEE Wireless Communications Letters*, 7(3), 328-331.