

國立成功大學
工業與資訊管理學系碩士班
碩士論文

風向鄉村型多郵差途程問題研究—以多輛無人車巡邏途
程規劃為例

On the K Windy Rural Postmen Problem with a Patrol
Routing Application by Multiple Unmanned Autonomous
Vehicles

研究生：何冠緯

指導教授：王逸琳 博士

中華民國一百零八年五月

國立成功大學
工業與資訊管理學系碩士班
碩士論文

風向鄉村型多郵差途程問題研究—以多輛無人車巡邏途
程規劃為例

On the K Windy Rural Postmen Problem with a Patrol
Routing Application by Multiple Unmanned Autonomous
Vehicles

研究生：何冠緯

指導教授：王逸琳 博士

中華民國一百零八年五月

國立成功大學

碩士論文

風向鄉村型多郵差途程問題研究—以多輛無人車巡邏
途程規劃為例

On the K Windy Rural Postmen Problem with a Patrol
Routing Application by Multiple Unmanned Autonomous
Vehicles

研究生：何冠緯

本論文業經審查及口試合格特此證明

論文考試委員：

王逸琳
吳政翰
柯秉星

指導教授：王逸琳

系(所)主管：

王惠嘉

中華民國 108 年 5 月 31 日

摘要

現行警察執行巡邏勤務之路線大多以最近鄰點法串聯巡邏箱簽到的定點巡訪方式進行，其路線大多固定可預期，且因警力有限而無法同時於多處巡邏，導致見警率不彰，治安維護效果有限。隨著人工智慧和物聯網相關技術的成熟，佈署多輛無人巡邏車搭配路徑規劃，可大幅改善上述目前警力巡邏之缺點。

本研究探討之巡邏途程問題旨在如何以最小化車隊總巡邏時間，規劃多輛無人車來巡邏某些具巡邏需求的路段。由於同一路段的正、反向(類似順、逆風)巡邏方式之成本可能不同，因此將代表巡邏範圍之網路圖稱為風向圖(Windy Graph)，而必須經過某些路段的途程規劃問題屬於鄉村型郵差途程(Rural Postman)問題，因此本研究為「風向鄉村型多郵差途程問題」。有別於過往文獻大多將巡邏路段視為無向，本研究將巡邏需求設置在有方向性的節線上，進一步可處理平行節線與單行道之一般網路圖，更能貼近現實情況。

文獻針對風向鄉村型多郵差問題大多以加入排除子迴圈限制式的分支切割演算法(Branch-and-Cut, B&C)求解，在求解過程中一遇到獨立子迴圈即產生新限制式排除之，但此方式常因持續加入的限制式過多而導致求解效率不佳。本研究則首先基於層空網路(Level-space Network)發展一個不用考慮子迴圈的混整數規劃模式，接著保留必巡之節線及相關節點，將各組上述保留之兩節點間在原網路之最短路徑濃縮成為新的節線，以此當成預處理機制來建構一個新的「限縮網路」，再進一步建構其對應的層空網路，於其上實作對應的整數規劃模式求解。

在實際進行數值測試後，我們發現整數規劃模式的求解作法將耗時過長，無法處理中或大型之網路規模，因此本研究將必須巡邏之節線編碼為一組解，利用貪婪式演算法、基因演算法、以及區域搜尋法等啟發式演算法於上述的限縮網路上求解最小化無人車隊總巡邏時間。此外，本研究亦嘗試將貪婪式演算法之求解結果，分別加入數學規劃模式與其他演算法作為初始解以加速其收斂效果。

本研究嘗試進行較完備的數學規劃模式與演算法的數值測試，在測試過不同

規模的隨機網路圖後，我們發現數學規劃模式僅可求解小規模的網路，而貪婪式演算法在極短的時間內即可求得一品質尚可的解，若再以此為初始解輸入區域搜尋法後，實驗發現其求解品質穩定，且整體表現較其他二種演算法為佳，有助於被應用在求解較具規模的實際巡邏途程規劃問題。

關鍵字：巡邏途程問題；風向鄉村型多郵差問題；無人車；整數規劃；層空網路。



On the K Windy Rural Postmen Problem with a Patrol Routing Application by Multiple Unmanned Autonomous Vehicles

Guan-Wei Ho

I-Lin Wang

Department of Industrial and Information Management

SUMMARY

With more advanced AI and IoT, unmanned autonomous vehicles can replace partial manpower for patrol routing. Current patrol routing practices usually exploit the nearest first heuristics for visit patrol boxes located in different places. Such routes take a longer time and are more predictable. Here we aim at the design of optimal patrol routes in a systematic way considering travel costs, security requirements, and fair workloads among coordinated unmanned patrol vehicles.

To be more realistic, we consider a windy graph with arcs of different lengths in different directions. This leads to a K windy rural postmen problem. In particular, we seek optimal tours for all the unmanned vehicles such that all required road segments are patrolled with minimum travel costs. We propose two integer program (IP) formulation on a level-space network. The first IP formulation can only deal with small networks. The second IP formulation is conducted on a reduced network, where we only consider the nodes adjacent to the required patrolling arcs and treat a shortest path between any two nodes as an arc in this reduced graph. This second IP formulation can solve larger cases but still cannot deal with practical cases.

We design three fast heuristics on the reduced network: Greedy algorithm, Genetic Algorithm (GA), and Local search (LS). Computational experiments indicate the initial solution obtained by GA and further improved by LS can calculate solutions of satisfactory qualities for larger problems.

Keywords: Patrol routing, Windy Rural Postmen Problem, Unmanned Autonomous Vehicle, Integer Program, Level-Space Network

INTRODUCTION

Due to the advance in the technology of the artificial intelligence (AI) and Internet of Things (IoT), unmanned autonomous vehicles can be used to conduct patrolling jobs. For example, drones and unmanned vehicles with facial recognition technology are used for street patrolling in Dubai, which help monitor public order and track suspects. If the unmanned vehicles are used in patrolling, it will not only collect and report evidence immediately, but also save police manpower effectively. Patrolling is an effective way to prevent crimes and maintain public order. To patrol over some crime hotspots in the area of jurisdiction, current practices set up patrol boxes over hotspots and these boxes are usually visited following a nearest first mechanism. In particular, after visiting one patrol box, the police may select the nearest neighboring box as the next box to visit. Such a patrolling route may take a long time and is also highly predictable, which might incur more manpower, longer mission time, and worse patrolling effects. On the other hand, by deploying and routing unmanned patrolling vehicles, we can achieve better coverage with fewer resources and shorter time, if the routes are calculated based on correct settings of mathematical analysis.

Although most literature in patrol routing does not specifically mention whether the vehicles are manned or unmanned, we think this is not the major issue to design the mathematical models or solution methods. Two classical routing problem types can be considered: (1) the Node Routing Problem (NRP) whose patrol demands are located at patrol boxes, represented by nodes in a district network, and (2) the Arc Routing Problem (ARP) whose patrol demands involve the traversal or coverage of a road segment represented by an arc (directional or not). In this thesis, we focus more on ARP with two popular variants: the Chinese Postman Problem (CPP) and the Rural Postman Problem (RPP). CPP seeks a shortest postman tour traversing all arcs at least once in the network, whereas the RPP generalizes CPP in the sense that only a subset of the arcs has to be traversed. We focus on RPP with multiple patrolling vehicles in a windy graph called K-WRPP whose arc costs may be different in different directions of the same road segment. Since general RPP and WPP are proved to be NP-hard, our problem is thus at least NP-hard as well. The objective function of our problem contains two parts: (1) to minimize the total traversing time (Min-sum), and (2) the longest route of all vehicles (Min-max). Although Min-max can balance the loading of each vehicle, in general, the Min-sum objective function makes more sense. Thus, we use the total patrol time (Min-sum) as our major objective value.

MATERIALS AND METHODS

We propose two integer programming models for calculating exact optimal solutions. Two level-space network based models M_L^K and M_{RL}^K are proposed to avoid subtours. The former model M_L^K is designed for the original network, whereas the latter model M_{RL}^K is designed on a reduced network which should be conceptually equal to the original one but with much smaller size. In the reduced network, only nodes adjacent to the required arcs to be patrolled are considered. It forms a complete graph using those nodes, and arcs between these nodes are shortest paths between these nodes in the original network. Although M_{RL}^K can solve larger cases than M_L^K , these two IP models are still too time-consuming to deal with realistic networks.

In order to calculate a good solution within a shorter time for larger networks, we also propose a nearest-first greedy algorithm that tends to schedule the unmanned vehicle of less current patrolling load to patrol its nearest unpatrolled arc. We also design a Genetic Algorithm (GA) and a Local Search Algorithm (LS) to help converge to a better solution, based on a good feasible solution calculated by the greedy algorithm. Note that all the heuristic algorithms and M_{RL}^K are designed on the reduced network.

RESULTS AND DISCUSSION

Our testing was performed on a personal computer with Windows 10, Intel Core i7-6770 3.40GHz*8 Processors, and 16GB RAM. All the solution methods are implemented in C++ language, compiled by Visual Studio 2015. Gurobi 8.1.1 was used for solving M_L^K and M_{RL}^K . Our computational experiments provide the following observations: (1) all the IPs consume too much time, and are not suitable for solving large scale networks, (2) M_{RL}^K has better performance than M_L^K , for cases of fewer required patrolling arcs, (3) our proposed algorithms are efficient and effective because the greedy algorithm usually calculates a good initial feasible solution, which then further helps improve the solution of the IPs or other heuristic algorithms to converge better, and (4) our proposed LS with initial feasible solution obtained by greedy algorithm has the best performance in our testing.

CONCLUSIONS

The unmanned vehicles patrol routing problem for crime prevention and maintenance of public order are expected to be very useful and important for future police, especially when the police manpower is not sufficient to afford to patrol over a large-scale district. We treat our patrol routing problem as a K-WRPP, propose two IP models, and three heuristic algorithms. The second IP model introduces the concept of a reduced network, and we suggest using heuristic algorithms on the reduced network

for patrolling jobs on networks of larger size with proportionally fewer required patrolling arcs. Among these heuristic algorithms, our LS algorithm has the best performance, and is suggested to deal with more practical cases. As a result, we suggest using our heuristics for practical usage in the future.



目錄

摘要.....	i
目錄.....	vii
表目錄.....	ix
圖目錄.....	x
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	1
1.3 研究目的.....	2
1.4 論文架構.....	3
第二章 文獻探討.....	4
2.1 網路圖說明.....	4
2.2 節點途程問題相關文獻.....	5
2.3 節線途程問題相關文獻.....	7
2.4 小結.....	11
第三章 無人車巡邏途程規劃問題.....	13
3.1 問題描述與假設.....	13
3.2 風向鄉村型多郵差問題之數學模式.....	14
3.2.1 參數與變數定義.....	15
3.2.2 數學模式.....	15
3.2.3 有關排除獨立(未與外界連通)子迴圈機制之進階說明.....	16
3.3 以層空網路為基礎建構之數學模式.....	17
3.3.1 層空網路.....	17
3.3.2 一般模式 M_L^K	19
3.3.3 網路縮減下之數學模式 M_{RL}^K	20
3.4 小結.....	21
第四章 多輛無人車途程規劃演算法之設計.....	23
4.1 最鄰近待巡節線優先之貪婪式演算法(Greedy Algorithm).....	23
4.2 基因演算法(Genetic Algorithm, GA)	24
4.3 區域搜尋法(Local Search, LS)	29
4.4 小結.....	32
第五章 整數規劃模式與啟發式演算法之數值測試與分析.....	33
5.1 測試網路資料.....	33
5.2 多輛無人車之巡邏途程規劃問題之測試.....	34
5.2.1 整數規劃模式比較.....	34

5.2.2	數學演算法測試.....	36
5.3	小結.....	39
第六章	結論及未來研究.....	40
6.1	結論.....	40
6.2	未來研究.....	41
參考文獻	43



表目錄

表 4.1 貪婪式演算法流程.....	23
表 4.2 基因演算法流程.....	28
表 5.1 網路結構資訊.....	33
表 5.2 混整數規劃模式之問題大小估算.....	35
表 5.3 數學模式測試(求解效率).....	35
表 5.4 數學模式測試(求解品質).....	36
表 5.5 數學演算法參數設定.....	37
表 5.6 數學演算法測試(求解效率).....	38
表 5.7 數學演算法測試(求解品質).....	38



圖目錄

圖 2.1(a) 混合圖	5
圖 2.1(b) 無向圖	5
圖 2.1(c) 有向圖	5
圖 3.1 路段示意圖	14
圖 3.2 排除子迴圈限制式之可行解	17
圖 3.3 排除獨立子迴圈限制式之不可行解	17
圖 3.4 單一無人車之層空網路圖	18
圖 3.5 多輛無人車之層空網路圖	18
圖 3.6 縮減前之網路圖 G	20
圖 3.7 縮減後之網路圖 G'	20
圖 4.1 計算目標值示意圖	24
圖 4.2 染色體示意圖	25
圖 4.3 選擇父母和交配的分割點	27
圖 4.4 進行交配與複製	27
圖 4.5 最後子代結果	27
圖 4.6 突變策略之交換法	28
圖 4.7 突變策略之插入法	28
圖 4.8 2-opt* 示意圖	30
圖 4.9 Cross exchange 示意圖	31
圖 4.10 Or-opt 示意圖	31
圖 5.1 GA 小型網路收斂測試(車數=1)	37
圖 5.2 GA 小型網路收斂測試(車數=2)	37
圖 5.3 GA 中型網路收斂測試(車數=1)	38
圖 5.4 GA 中型網路收斂測試(車數=2)	38

第一章 緒論

1.1 研究背景

科技研發的最終目的在於生活應用，主要目標為提升人類生活環境品質、健康水平與社會安全福祉。隨著人工智慧和物聯網相關技術的成熟、雲端系統與無線感測快速演進，車輛載具的發展也漸漸向自動駕駛跨出一大步。雖然完全的無人駕駛汽車仍未全面商業化，部分無人駕駛車的核心技術應用，已逐漸從實驗場所進階到現實層面，其關鍵技術包括環境感知、導航定位、途程規劃和決策控制等，而搭載人工智慧技術的無人駕駛車更能夠在險峻地形執行監測任務。舉例來說，澳洲從 2017 年 9 月起已在多處泳灘派遣無人機監控海灣周遭，配合人工智慧檢測系統，即時辨識鯊魚游蹤並發送警告訊息給泳客。而無人駕駛車所帶來的好處則不僅如此，譬如其內建大範圍感知系統，將擁有超越人類有限的環境感知能力，能夠對潛在危機進行感測，反應也能較人類更為迅速。以杜拜為例，他們已使用無人機和具臉部識別技術的無人駕駛車於杜拜的街道上巡邏，幫助城市監測和跟蹤犯罪嫌疑人。目前為止，無人巡邏車已可使用機器學習演算法自行導航，而警察亦可使用電腦儀表板遠端控制巡邏車，然而這些導航或遙控應遵循的路線該如何規劃，應該遵循某些最佳化目標與準則，為整數規劃及組合最佳化的問題，而非屬於機器學習的範疇。

1.2 研究動機

無人駕駛車現今的研發技術著重於車輛本身的運動控制和感測周遭環境的能力，若再配合無線網路技術以即時與資料庫連線，對警方執行巡邏任務時將大有幫助。舉例來說，當周遭有明顯犯罪行為或存在犯罪嫌疑人，配合臉部辨識技術連結資料庫則可大幅縮短人類查找資訊或回憶的時間，能夠以最即時又有效率的方式蒐集罪證及情報，以利警方掌握犯罪情資。

現行警察之巡邏方式，主要是對於交通衝要處所及較可能發生犯罪之治安要點，設置簽到箱的定點式巡邏。其巡邏路徑大都移動至該處最近的簽到箱，然而這種路線較為單調且容易被有心人士掌握巡邏規律，因而蓄意在警察巡邏過後才進行犯罪活動。此外，更為人詬病的是巡邏人員在填寫巡簽時間可能有預簽、代簽、補簽或簽到字跡過於潦草，導致督勤人員無法直接由巡邏表得知巡邏人員有無落實巡邏勤務。若以無人巡邏車搭配巡邏途程規劃則能準確地按照已知路徑進行巡邏；若以電子系統感測無人巡邏車所在位置，即可知道其目前動態並節省巡簽時間。因此考慮犯罪易滋生路段或交通要道安排無人巡邏車執行巡邏勤務，相對於根據簽到箱的位置巡邏的傳統做法，更能廣泛地遏止犯罪發生並提升見警率，進一步將人力進行有效的管理及應用。

鑑於警力難以兼顧大範圍巡邏任務，且同時又要對周遭人事物保有相當的注意力；反之，配備資通訊系統的無人巡邏車，在其執行巡邏任務時可以電子設備對周遭錄影或監測，因此採用無人巡邏車再搭配有效的巡邏途程規劃是較為理想的作法。目前無人駕駛車配合巡邏途程規劃以執行巡邏任務的相關文獻，主要分為依巡邏需求點進行路徑安排的「節點途程問題」(Node Routing Problem, NRP)，以及依巡邏需求路段進行途程規劃的「節線途程問題」(Arc Routing Problem, ARP)兩大類，而此兩類問題大都為 NP-hard，其中 ARP 中所有節線都需經過時為「中國郵差問題」(Chinese Postman Problem, CPP)，若僅部分節線需被服務為鄉村型郵差問題(Rural Postman Problem, RPP)，若將單一郵差進行區域內送信服務改以多個郵差之形式則為多郵差問題(KCPP)與鄉村多郵差問題(KRPP)。而本研究期望以多輛無人車巡邏重要路段為出發點，進行巡邏途程規劃，達到遏止犯罪意圖之目的，可視為 KRPP 的應用。

1.3 研究目的

本研究旨在規劃以最短總旅行時間而能夠有效巡邏的多車巡邏途程組合，考

慮巡邏重點存在於「節線」上，即無人車在節線上巡邏過後便可達到有效嚇阻犯罪。為使有限的無人巡邏車發揮最大巡邏功效，依據巡邏轄區的特性、犯罪熱點以及治安死角，在巡邏轄區內所有路段可再分為「一般路段」或「巡邏重點路段」，本研究再依路段方向性將其視為具方向性的節線。為使無人車執行巡邏勤務獲得最大效益，應針對具有較高巡邏效益路段配置巡邏警力，如此才能有效防止犯罪發生，而非採取平均式巡邏分布，造成警力資源的浪費與錯置。

過往文獻中，「多郵差問題」大多將巡邏需求設置在無方向之路段節線上，只要其中一個方向有被巡邏即可達到巡邏之目的。然而，現實情況中路段通常具順逆兩向、或單行道路，因此本研究將巡邏需求設置於有方向性之節線來進行重點式巡邏。而在主要目標則分為最小化總旅行時間(Min-sum)與最小化所有車輛路線組合中最長那台車輛的巡邏時間(Min-max)兩種。其中 Min-max 雖然會將巡邏任務以較為平均的方式進行分配，但此舉可能會使無人車隊的總巡邏時間增長，進而使整體巡邏成本提高，因此我們以最小化總巡邏時間(Min-sum)為目標。

1.4 論文架構

本論文之架構如下：第二章回顧節點途程問題(NRP)與節線途程問題(ARP)等相關文獻；第三章討論由風向鄉村型多郵差問題(K Windy Rural Postmen Problem, KWRPP)延伸之無人車巡邏途程規劃問題，並提出兩種數學規劃模式；第四章介紹運用之貪婪式演算法、基因演算法與區域搜尋法；第五章為數值測試分析，比較數學規劃模式與演算法在不同網路大小下的求解表現；第六章總結目前本研究的成果及討論未來可研究方向和建議。

第二章 文獻探討

在安排巡邏的路線規劃中，通常以最小化總通行成本或時間，或最大化巡邏效益為目標，安排無人車從車輛停放的起始點開始執行巡邏任務，其被指定必須巡邏到的地方都被巡邏過後，所有無人車必須再返回到原起始點結束此次巡邏勤務。在相關文獻中分為巡邏需求設置部分是在節點或節線兩種類型，第一類「節點途程問題」(NRP)相關文獻為需求假設在節點之上，譬如最廣為人知的「車輛途程問題」(Vehicle routing problem, VRP)大多屬於此類，旨在不違反車輛容量限制下，以最短的車輛總旅程距離來滿足所有顧客點的需求，；然而，第二類的「節線途程問題」(ARP)則為在一已知的地區中，車輛從起點出發要以一條最小旅行時間(或最短距離)之路徑巡邏過此地區所有的街道，且最後要回到原出發點，其巡邏需求是設置在節線之上。

本章共分三節，第一節為說明網路圖的形式，第二節回顧需求設置在節點上的節點途程問題之相關文獻，第三節為探討節線途程問題之相關文獻及說明本研究欲探討之議題與文獻之間的相關性。

2.1 網路圖說明

在現實生活之中，警務單位會依轄區面積、地理情況劃分並在主要治安區設置巡邏箱或對於交通、經濟密集區之路段執行巡邏勤務，因此巡邏途程規劃將大大影響警勤人員進行巡邏任務之效率。警察機關巡邏之路線安排不外乎根據轄區內巡邏箱的設置點，抑或是重要路段來著手，因此可以利用網路圖來呈現簽到箱或重要路段與轄區內街道分布情形之間的相對關係。然而在數學理論的分支中，圖論以圖為研究對象，主要研究由頂點和邊構成圖形的數學方法。根據圖論中網路圖的特性和頂點與邊之間的關係，以下將說明與本研究較相關的網路圖。

假設有一混合圖(Mixed graph) $G = (V, E, A)$ ，其中 V 為頂點集合、 E 為無向邊(Undirected edges)集合、而 A 為有向邊(Directed arcs)之集合。如圖 2.1(a) 混合圖所示，：頂點 1 與頂點 2 相連的邊為無向邊；頂點 2 與頂點 3 相連的邊為有向

邊。當圖 G 中不存在無向邊($E=\emptyset$)時, G 被稱為有向圖, 如圖 2.3(c) 有向圖為例, 代表車輛要穿越圖中任何一邊時必須按照指定的方向, 即為單行道之概念, 即為點 10 和點 11 所構成的邊; 另外一方面, 當圖 G 中不存在有向邊($A=\emptyset$)則被稱為無向圖如圖 2.2(b) 無向圖, 如有車輛欲穿越圖中的邊即可雙向通行。

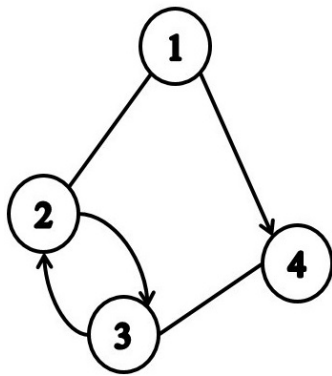


圖 2.1(a) 混合圖

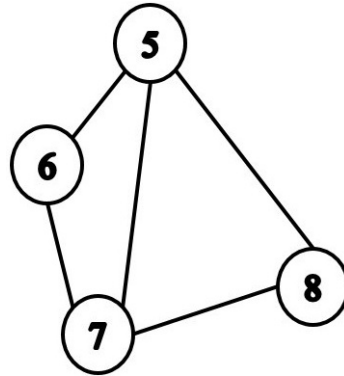


圖 2.1(b) 無向圖

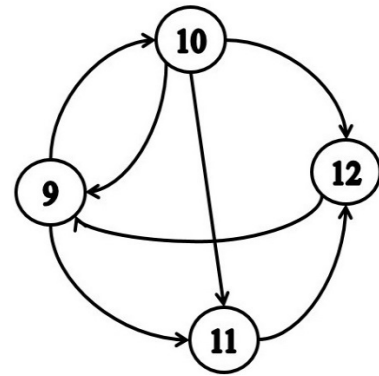


圖 2.1(c) 有向圖

一般而言, 在混合圖、無向圖及有向圖中, 車輛為了行駛頂點之間相連所形成之邊皆會存在非負的通行成本。以有向圖來說, 由於車輛通行的邊具有方向性而通行成本也會依其方向性而不同。在無向圖中, 每條邊因為不具方向性而被視為具有相同的通行成本, 但事實上, 若車輛欲穿越該無向邊, 根據行駛方向不同理應具有不同的通行成本。因為就現實情況而言, 對於同一條雙向通行的馬路可能會有順、逆風抑或是上、下坡而有不同的通行成本, 此種無向圖基於同一路段根據不同方向而有不同的通行成本的情況, 則被稱作風向圖(Windy graph)。而圖中同時具有向與無向節線的混合圖, 可放當成是風向圖的一種特例。上述所有網路圖之形式常被廣泛應用在圖論所延伸出來的經典問題, 例如: 旅行銷售員問題、車輛途程問題與中國郵差問題等。

2.2 節點途程問題相關文獻

車輛途程問題(VRP)為節點途程問題的延伸, 其起源可追溯到 Dantzig and Ramser (1959)所提出的車輛派遣問題(Vehicle Dispatching Problem), 基本的車輛

途程問題可以視為旅行銷售員問題(Traveling Salesman Problem, TSP)的延伸，最基本的問題形式為在眾多可行路線中，找出一條能服務所有顧客需求且成本最小的路徑。Bodin et al. (1983)提出討論 VRP 中許多不同的影響因素，其中與網路圖架構最具相關的主要為網路圖形式與需求設置兩部分。網路圖形式的部分包含 2.1 節中所提到的有向圖、無向圖或混合圖等，也有考慮單一或是多個起點；需求設置部分則有考慮到將需求假設在節點或是節線上，在過往 TSP 與 VRP 問題相關文獻中大多選擇將需求設置在節點上。相同文獻中指出 TSP 為一個銷售員以最短的總旅行距離為目標，從起點出發，經過每一顧客點正好一次，再回到原起點；而 VRP 以一個車隊從起點出發，在不違反車子最大容量的限制下進行配送貨物來滿足顧客端點的需求，每個顧客點都只能被服務一次且貨物都必須被配送到顧客手上，車輛在完成配送工作後返回原出發點，其主要目標為最小化所有車輛行駛路線的總長度。

Bodin et al. (1983)針對 VRP 的各式解法及策略，將常用的策略整理如下：

1. 先分群再排路線(Cluster first route second)根據所有需求(包括時窗限制、距離或容量限制)分成數群，在各分群中建立滿足目標之路徑。
2. 先排路線再分群(Route first cluster second)則先依各需求點先建構一段通過所有需求點的路徑，再依其限制分群成數段可行的小路徑。
3. 節省法與插入法(Savings/Insertion procedure)：首先產生一起始路徑，在不違反限制條件下，計算將需求點插入起始路徑或增派車輛滿足需求點之間的成本節省值，並記錄最大成本節省值作為下回合插入需求點的參考，反覆求取直到達到停止條件或產生可行之路徑。
4. 改善法與交換法(Improvement/Exchange procedure)：在已知可行解中進行節點或節線的位置交換策略，或將可行解進行分群，若成本較小時則進行交換改善，反覆運算直到成本無法改善即完成路線規劃。

Chang and Chen (2007)針對定義在無向圖之具時窗(Time Window, TW)限制的車輛途程問題利用基因演算法(Genetic Algorithm, GA)進行求解，問題中除了

原本車輛已具備之負載量限制之外，同時在每條節線上加上時間成本，並限制某些節點在某段給定的時間區段內必須被服務。其中，Koskosidis et al. (1992)依據時窗限制的寬鬆程度又可分成硬時窗(Hard time window)和軟時窗(Soft time window)兩種形式。硬時窗限制車輛必須在需求點的時窗內到達，若是早於限制之時窗下界到達，則必須等待到時窗所限制之下界時間需求點才會接受服務；若超過時窗限制的上限時間，則會拒絕車輛所提供的服務。軟時窗與硬時窗最大的不同在於對時窗限制採取較寬鬆的政策，不論車輛對於服務點的時窗限制分別是早到或遲到都改以懲罰值來取代硬時窗相對應的等待或拒絕服務之策略，並將懲罰值納入目標中加以考慮。自車輛途程問題被提出後，文獻已發展出許多的求解方法，一般而言主要分為最佳化方法及啟發式演算法，但隨著網路規模擴大進而使求解時間急遽上升，求得最佳解較為困難，文獻多以啟發式演算法快速求得一可行解後，再試圖改善該可行解以逼近最佳解。

2.3 節線途程問題相關文獻

節線途程問題(ARP)探討網路圖中存在某些節線需備經過或是服務，而經過或服務節線存在相對應之成本，目標為求出涵蓋所有需被服務之節線的最短迴圈，且使總成本最小。當所有節線皆需被經過時，此即為中國郵差問題(CPP)；而若只部分節線需被經過，則為鄉村型郵差問題(RPP)。中國郵差問題由 Guan (1962)所提出，目標為在一連通的網路圖內找到一條最短的封閉迴圈，且此路徑必須經過圖中所有邊至少一次，此一路徑若在特別的圖（譬如所有節點之相鄰節線個數為偶數之無向圖，或所有節點之進入與連出之節線個數相同之有向圖）將與著名的七橋問題所求的歐拉迴圈(Euler Circuit)相同，其中歐拉迴圈表示在一網路圖中找到一條從起點出發經過圖中所有邊剛好一次後再回到原出發點的路徑。然而，考量到現實層面的情況，除了必須經過網路圖內所有的邊之外，還必須考慮網路圖中定義的邊有向與否。假設 CPP 分別被定義在有向圖和無向圖，則被稱為有

向的 CPP (Directed CPP, DCP) 與無向的 CPP (Undirected CPP, UCPP), 若 CPP 被定義在一混合圖 $G = (V, E, A)$, 目標為以最小路徑經過所有有向邊及無向邊, 此問題被稱為混向郵差問題 (Mixed Chinese Postman Problem, MCPP)。若假設 $G = (V, E)$ 為一風向圖, 為了在此圖中找出一條以最小成本涵蓋所有邊的路徑, 這類型問題則被稱為風向郵差問題 (Windy Postman Problem, WPP)。上述所有定義在不同網路圖之 CPP 皆是為了找出單一車輛之最小成本巡邏路徑, 但是當網路規模變大, 僅由一部車輛進行巡邏任務所花費的時間會非常龐大而沒有效率, 所以多郵差問題(KCPP) 即為在同一時間將網路圖中所有待巡視之路段分配給多部車輛來完成巡邏街道的任務, 此一作法是較為理想的部署。然而在問題複雜度方面, Edmonds and Johnson (1973)證明單純的 DCP 在網路圖中若為強相連 (Strongly connected)與 UCPP 都可以在多項式時間內進行求解, 而 MCPP 和 WPP 的難度則至少為 NP-hard。再者, KCPP 若被定義在單純無向圖和單純有向圖的情況下, 分別由 Assad et al. (1987)和 Pearn (1994) 證明可在多項式時間內以求得最佳解, 而多載具的 K-MCPP 及 K-WPP 問題仍至少為 NP-hard 問題。

根據巡邏車輛的數目, 可以將以下文獻分為兩類, 一為以單一車輛規劃其路線進行服務的單郵差問題; 二為利用多台車輛來對所有街道進行途程規劃的多郵差問題。另外一方面, 在郵差問題中若以目標式區分成三種形式, 第一種為較為常見之最小化總旅行時間(或成本)的 Min-sum, 在單一車輛(郵差)或多個車輛情況下均存在, 其應用廣泛包含資源抄表路徑安排、路面積雪清除作業、廢棄物收集路線規劃、橋梁及建築物骨架檢查; 第二種 Min-max 則是在於多台車輛的情況下, 最小化所有車輛中耗時最長車輛的巡邏時間(亦可視為通行成本或通行距離), 此目標式主要用來平衡各車輛之工作負荷; 而第三種則是所有路段上皆存在通行成本, 但在特定路段上進行服務才具有可獲得的收益, 目標為求取最大化總收益(服務收益扣除通行成本)的路線規劃問題。

風向郵差問題(WPP)首先由 Minieka (1979)所提出。在該研究中指出在 CPP 中的每一邊通行成本之對稱性的假設是較不貼近現實, 例如當車輛欲通過某一路

段時，必然會有上坡與下坡、順風或逆風，因此根據不同行駛方向而應有不同的通行成本。Grötschel and Win (1992)對於風向郵差問題提出一個整數線性規劃模式，其求解目標為最小化總通行成本，亦針對數學模式所形成之多面體空間進行研究及提出切割平面演算法得以求解中等規模的網路圖。此外，當通行成本 $c_{ij} = c_{ji}$ 時，即可將定義在無向圖之 CPP 視為 WPP 的特例；當有向邊 (i, j) 具通行成本 c 可將此有向邊轉換為無向邊，並使原本不存在另一方向的有向邊設定其為 $c_{ji} = \infty$ ，因此可將 DCP 視為 WPP 的特例；相同地，MCP 也可被視為 WPP 之特例。綜合上述之特性，先前提到的 WPP 數學模式及演算法也被用來求解 DCP、UCP 及 MCP。

Malandraki and Daskin (1993)提出最大收益中國郵差問題(Maximum Benefit Chinese Postman Problem, MBCP)，在通行路段的同時不僅存在通行成本外也可以透過服務該路段獲得收益，並加入預算限制式以多目標的方式求取最大化途程淨收益和最小化總通行成本，將此問題定義在有向的網路上並提出分支界限演算法(Branch-and-bound, B&B)。既而 Pearn and Wang (2003)針對相同問題定義在無向的網路上，提出一個混合應用最小生成樹(Minimum Spanning Tree)及匹配演算法(Matching algorithm)的啟發式演算法進行求解。以上兩篇文獻皆假設具有服務收益的路段在每次被經過時都可以獲得（而非多次經過但僅計算第一次收益）。

而鄉村型郵差問題(RPP)則以最少旅行時間為目標，探訪過網路圖中所給定必須走的路段至少一次，再回到原出發點完成一個封閉迴圈。若將給定必走的路段集合定義為網路圖中所有路段，此特殊之 RPP 即為 CPP，因此可將 CPP 視為 RPP 的一種特例。依車輛數目之多寡，可將文獻分為鄉村型郵差問題及鄉村型多郵差問題(KRPP)。

RPP 首先由 Orloff (1974)提出，此乃因為在現實生活中針對網路圖中所有路段進行服務的 CPP 相對較少見，只需要服務特定路段的 RPP 更貼近實際情況。Lenstra and Rinnooy Kan (1976)證明除非在給定的無向圖中必須被服務的路段都

相連的情況下可以在多項式時間內求解，否則鄉村型郵差問題至少為 NP-hard 的複雜問題，其中亦包含定義在有向圖、混合圖及風向圖的 DRPP、MRPP 和 WRPP，而再進一步延伸的鄉村型多郵差問題也都可以被推廣為困難問題。

Feillet et al. (2005) 以最大化淨收益為目標，求解定義在有向圖上具利益之節線旅行問題(Directed Profitable Arc Tour Problem, DPATP)，除了限制節線上利益存在的次數外，更針對每台車最大的總通行距離進行限制，利用一不具容量限制的車隊進行途程規劃，特別的是在此問題中，每台車並不存在出發節點與結束節點，意即每台車可以在任一點開始或結束服務。

Moreira et al. (2007)以製造高度精密工具的公司為背景，為了找出在給定的圓形板上進行切割多個預定形狀的最短切割路徑問題。上述問題有幾個特徵，如：被切割的圓形板被懸吊在空中，切出預定形狀的半成品會脫離板體而掉落在收集盒裡等特性。另外，切割過程是由一條穿過圓形板的電銅線進行，切割過程不會離開切割的表面直到工作結束再回到原出發點。由於被切割的圓形板會隨著切割的進行而漸漸改變形狀，而被定義成動態鄉村型郵差問題(Dynamic Rural Postman Problem)並提出以節點路徑的演算法進行近似求解。

Aráoz et al. (2009)將私有化鄉村型郵差問題(Privatized rural postman problem)更名為「獎勵收集鄉村型郵差問題」(Prize-collecting Rural Postman Problem, PRPP)，定義在已有起點的無向網路圖上，車輛在拜訪圖中所有邊的過程中，每次行經某邊，均必須支付該條邊相對應之過路費成本；而服務該條邊也可以得到利潤，但此利潤僅會在第一次行經時獲得，往後再次經過便不存在。目標是為了找到從起點出發可以得到最大收益(利潤-成本)的迴圈，並針對此問題提出兩階段演算法。

Benavent et al. (2009)針對多車輛之風向鄉村型郵差問題(MM K-WRPP)，目標為最小化所有車輛中通行時間最長的車輛旅行時間，提出一數學模型，並結合基於多個求解起始程序之演算法(Multi-start algorithm)、變動鄰域尋優法(Variable Neighborhood Descent)以及迭代區域搜尋法(Iterated Local Search)發展啟發式演算法以求解大規模之 MM K-WRPP 問題，這些演算法在處理一定規模的網路圖時，

可於合理求解時間內得到不錯的可行解。

Archetti et al. (2010) 研究在無向圖上的節線途程問題，無向圖中的每一條邊皆有其通行時間，欲利用一具有容量限制的車隊來服務有利可圖的邊集合，而這些邊同時存在服務利益與服務需求，且每台車輛都具有最大服務時間限制，利用分支價格演算法(Branch-and-Price, B&P)及多種啟發式演算法來求解出車隊最大化的總收益路線。Archetti et al. (2014) 求解有向圖上的鄉村型郵差問題，其中透過服務某些路段來獲取利益，利用單一不具有容量限制的車輛來進行服務，以最大化服務收益扣除總通行成本的淨收益為目標，提出一個結合禁忌搜尋演算法及整數線性規劃模式的混合演算法來求解。

Black et al. (2013) 提出依時變化的獲利性節線途程問題(Time Dependent Profitable Arc Routing Problem, TDPARP)，其中依時變化表示在旅行某節線時依據一天時刻的不同而有不同的通行時間。並將此問題定義在有向圖上，建構一數學規劃模式及利用變動鄰域搜尋法(Variable Neighborhood Search, VNS)和禁忌搜尋法兩種啟發式演算法來求解。

雖然途程問題依需求設置可區分為節點途程問題(NRP)與節線途程問題(ARP)，然而 Pearn et al. (1987)以節點取代必須經過的路段，將具容量限制的節線途程問題(CARP)轉換成具容量限制的車輛途程問題(CVRP)。而 Baldacci and Maniezzo (2006)與 Foulds, L. et al. (2015)皆利用較少節點數取代必須經過路段，將 CARP 轉換成 CVRP，進一步縮小網路圖規模，分別利用分支切割演算法(B&C)與分支切割價格演算法(Branch-and-Cut-and-Prize, B&C&P)求解轉換後的網路圖。

2.4 小結

綜合上述文獻回顧，我們可以發現針對節點與節線途程相關問題發展歷史都相當悠久，有許多相關的變形問題，多數學者也相繼提出許多數學模式與演算法來求解此兩類經典問題，但在處理大規模之網路圖時，數學模式仍會花費過長時

間或無法求得一可行解的情況。在過往文獻中，學者多以排除子迴圈方式建構數學模式，其做法具一定的求解複雜度，也會使求解時間大幅增加。因此本研究嘗試以層空網路建構數學模式，將避免出現子迴圈的情況。此外，相關文獻大都著重在求解時間的縮短或嘗試處理較大規模的網路圖，較少延伸原問題。



第三章 無人車巡邏途程規劃問題

本章共分四節，第一節描述風向鄉村型郵差問題及其假設，第二節則探討 Benavent et al. (2009) 之混整數規劃模式 $M_{B\&C}^K$ ，第三節描述本研究以層空網路所建構之混整數規劃模式 M_L^K ，以及針對原網路圖起點與必須巡邏節線之起訖點，透過對兩兩節點間求取最短路徑以建構一縮減網路，接著以數學模式 M_{RL}^K 求解之。

3.1 問題描述與假設

本研究之風向鄉村型多郵差問題可描述如下，在一個待巡邏的區域中所有街道順、逆向行駛的旅行時間可能不同的情況下，巡邏轄區存在一個起點停放所有被指派巡邏任務的無人車，欲巡視過所有治安重點路段後全數無人車都必須再回到原起點即完成此次巡邏勤務，其中由兩個相連頂點所構成的路段可以雙向通行但不同方向有相對應不同的通行時間(成本)，欲求解出一組車輛巡邏途程組合，在此車隊路徑組合中以最小化車隊總巡邏時間為目標，決定車隊中每一車輛之巡邏路徑。

本研究中假設所有無人巡邏車均集中停放在起點處，車隊數量為已知且車輛之配備、性能及外觀等條件並無差異，僅每輛無人車都有其代表編號作為識別，每台車僅會被交付一條巡邏路徑，開始執行巡邏勤務後無人車若同時在相同頂點交會並不會交換進行他車後續之巡邏路徑，每一次巡邏作業皆始於起點，依序巡訪必須巡邏之重點路段後會立刻返回至原出發起點，等候下一次巡邏任務指派。

本研究假設待巡邏之所轄區域內所有街道如何相連皆為給定，街道分為一般道路及必須巡邏的重點節線，其中不論道路形式都為兩個頂點所構成且得以雙向通行，但無人巡邏車在行駛任一路段不同方向時可能會存在不同旅行時間，以圖 3.1 為例。

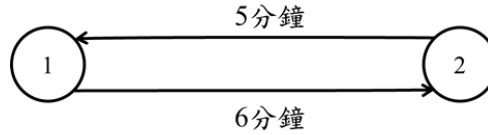


圖 3.1 路段示意圖

為適當簡化模式的發展與界定模式的使用限制，本研究有下列基本假設：

1. 僅考慮單一停車場所並設其為起點，所有無人巡邏車皆從起點出發，完成巡邏作業後無人巡邏車需返回至停車地點。
2. 巡邏轄區並非完全圖(Complete graph)，圖中任兩點所構成之通道相連情形皆為已知。
3. 假設必須巡邏節線之優先性相同，僅考慮巡邏之優先順序。
4. 無人巡邏車車隊數量為已知且車輛條件並無差異，且每台車輛都有其編號。
5. 無人巡邏車從場址出發時續航力無限，在巡邏途中不需中途補充能源。
6. 無人車巡邏不受到交通狀況、天氣與人為因素等外在緊急因素所影響，且巡邏時一定會將目前的工作完成後才會進行下一個工作，不會有工作到一半終止去尋訪其他節線(即 non-preemptive)。

3.2 風向鄉村型多郵差問題之數學模式

根據前一節之問題描述，在求解風向鄉村型多郵差問題時必須考慮到街道相連情形、車輛及街道通行時間等因素，本節首先討論 Benavent et al. (2009)提出之混整數規劃模式 $M_{B\&C}^K$ (其中 K 代表多車，B&C 代表 Branch-and-Cut)。

3.2.1 參數與變數定義

集合

V 節點之集合，節點1 $\in V$ 代表起點

E 所有路段構成之集合， $(i, j) \in E$

E_R 所有必須巡邏的路段構成之集合， $E_R \subseteq E$

$\delta(S)$ 路段分割子集合， $\delta(S) = \{(i, j) \in E : i \in S, j \in V \setminus S\}$

$\delta_R(S)$ 包含必須巡邏路段的邊分割子集合

參數

K 車隊數量上限

C_{ij} 路段 $e = (i, j)$ 對應之通行時間

變數

x_{ij}^k 車輛 k 從 i 到 j 經過之次數， $x_{ij}^k, x_{ji}^k \geq 0$ and integer

$y_{ij}^k = \begin{cases} 1, & \text{路段 } e = (i, j) \text{ 被車輛 } k \text{ 巡邏} \\ 0, & \text{otherwise} \end{cases}$

3.2.2 數學模式

目標為求取最小之最大車輛的總通行時間，如式子(3.2.1)所示。

$$\text{Minimize } z \quad (3.2.1)$$

限制式(3.2.2)為最小化最大車輛總通行時間。

$$\sum_{e \in E} (c_{ij} x_{ij}^k + c_{ji} x_{ji}^k) \leq z, \quad \forall k = 1, \dots, K \quad (3.2.2)$$

式子(3.2.3)為必須巡邏之重點路段需被一台車輛所巡邏過至少一次。

$$\sum_k y_e^k = 1, \quad \forall e \in E_R \quad (3.2.3)$$

限制式(3.2.4)為車輛通行不等式，不論車輛行駛方向為 i 到 j 或 j 到 i 皆代表曾巡邏過路段 (i, j) 。

$$x_{ij}^k + x_{ji}^k \geq y_e^k, \forall e \in E_R \text{ and } \forall k = 1, \dots, K \quad (3.2.4)$$

限制式(3.2.5)為流量守恆限制式。

$$\sum_{(i,j) \in \delta(i)} (x_{ij}^k - x_{ji}^k) = 0, \forall i \in V \text{ and } \forall k \quad (3.2.5)$$

限制式(3.2.6)為連接不等式，限制車輛之通行路徑區段都會與起點相連。

$$\begin{aligned} x^k(\delta(S)) &\geq 2y_e^k, \forall S \subset V \setminus \{1\}, \\ &\forall e \in E_R(S), \\ &\forall k = 1, \dots, K \end{aligned} \quad (3.2.6)$$

限制式(3.2.7)為「排除獨立子迴圈」(Isolated sub-tour elimination) 限制式，其中 $\delta_R(S)$ 為包含有必須巡邏路段之內部迴圈的節點集合， $|\delta_R(S)|$ 為集合內的節點個數。

$$\sum_{k=1}^K x^k(\delta_R(S)) \geq |\delta_R(S)| + 1, \text{ such that } |\delta_R(S)| \text{ is odd.} \quad (3.2.7)$$

可以利用此整數規劃模式 $M_{B\&C}^K$ 求得在多台車輛時，如何派遣車輛並且決定通行節線順序，最小化所有車輛中巡邏時間最長那台車的巡邏時間。而限制式(3.2.6)和(3.2.7)常利用分支切割演算法(B&C)在每次求解過程得到可行解時，檢查是否此解中有子迴圈未與起點相連接，若有此情況要進行排除。排除子迴圈的步驟在每次得到可行解就要檢查一次子迴圈是否與起點相連，求解過程稍嫌繁複，進而耗費許多時間。基於以上，本研究以階層式概念進行建構整數規劃模式，避免 B&C 步驟繁複大大影響求解時間。

3.2.3 有關排除獨立(未與外界連通)子迴圈機制之進階說明

車輛途程問題相關文獻之數學模式中經常會探討排除獨立子迴圈的限制式，限制式(3.2.8)即為 Fisher and Jaikumar (1981)所提出之排除子迴圈限制式(Sub-tour elimination constraint)，其中 P 為某些不包含起始來源點 S 之節線集合。

$$\begin{aligned} \sum_{i,j \in P, i \neq j} x_{ij} &\leq |P| - 1, P \subseteq \{1, \dots, n\} / S \\ &, 2 \leq |P| \leq n - 1 \end{aligned} \quad (3.2.8)$$

式子(3.2.8)主要是為了將節點集合 P 限制住，使子迴圈完全不會出現在求出的解之中。如果以一台車的旅行推銷員問題為例，求出的路徑只能如圖 3.2 所表示的單一迴圈，而不可出現如圖 3.3 所表示的兩連結迴圈，子迴圈的節點集合為 P ，即節點 B、C、D 所構成的集合，而子迴圈之外的節點為 \bar{P} ，若使用式子(3.2.8)則子迴圈內含三條節線及三個節點，可以得到 $3 > 3 - 1 = 2$ 的結果則違反限制式(3.2.8)，所以在求出的解中會把此種可能清除。

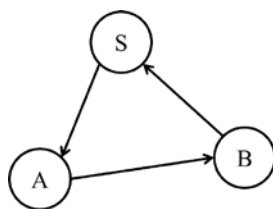


圖 3.2 排除子迴圈限制式之可行解

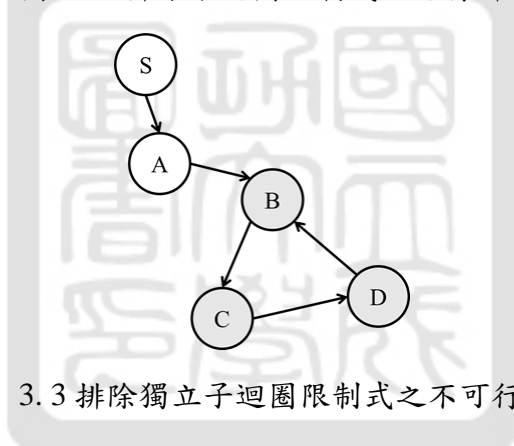


圖 3.3 排除獨立子迴圈限制式之不可行解

3.3 以層空網路為基礎建構之數學模式

根據 3.1 節之問題描述與假設，並在 3.2 節討論 Benavent et al. (2009)之混整數規劃模式的求解步驟繁複及困難後，本研究採取層空網路圖(Level-Space network)的概念建構數學模式，階層表示所有無人巡邏車僅會往後一層的相連節點通行（亦即一條節線的概念），直到待巡邏路段都涵蓋則結束巡邏任務，不會出現回流情形。透過此混整數規劃模式可進行無人車隊中每台車輛巡邏途程規劃

3.3.1 層空網路

本研究以層空網路建構數學模式，其單一無人車之層空網路圖如圖 3.4 所

示，點 0 為虛擬節點，節點 1 代表無人車開始執行巡邏勤務的網路圖起點，其他節點則為待巡邏網路圖原有節點，虛線代表車輛移動節線；實線為必須巡邏節線如圖中之節線(3,2)，其中 l 為階層數之引數，若將無人車數量增加，則多輛無人車之層空網路圖如圖 3.5， k 為車輛數之引數，每一層都代表一台無人車可能巡邏的網路，總共 K 層，每一台無人車都會由虛擬起點 0 出發巡邏過所有必須巡邏之節線後再回到虛擬起點 0，即完成巡邏任務。

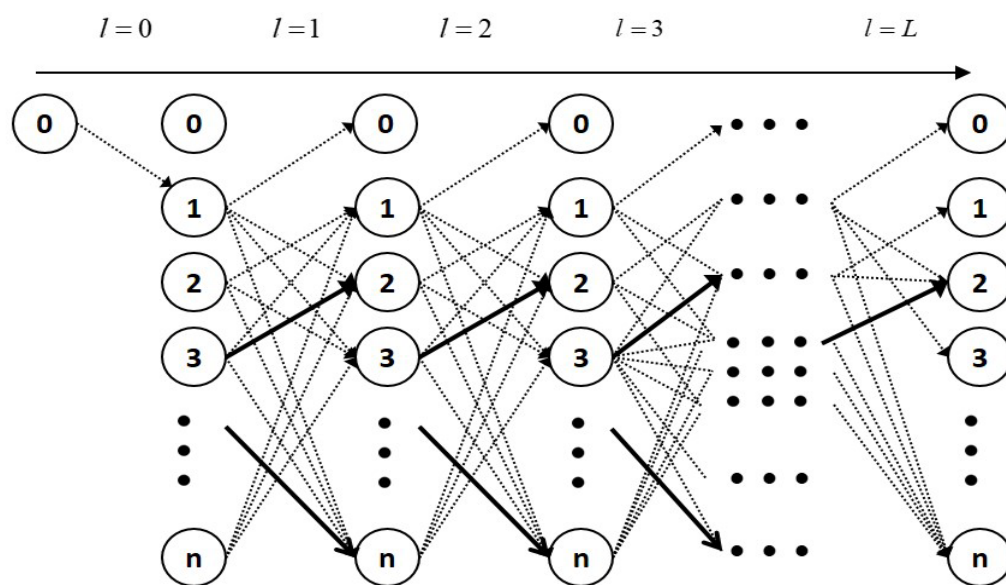


圖 3.4 單一無人車之層空網路圖

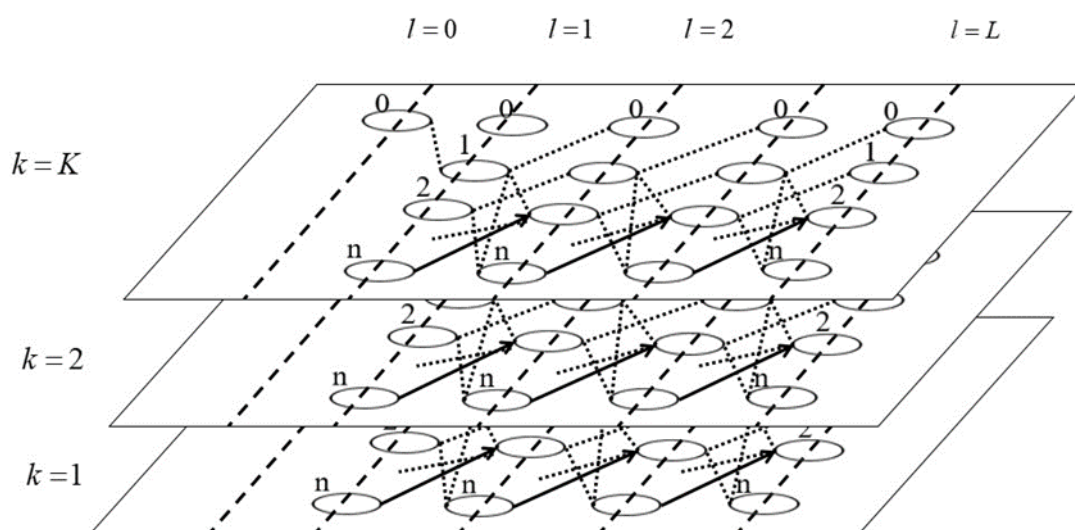


圖 3.5 多輛無人車之層空網路圖

3.3.2 一般模式 M_L^K

集合

N 所有節點之集合，包含虛擬起點 0

N_R 所有必須巡邏節線起訖兩點構成的集合， $N_R \subseteq N$

A_R 所有必須巡邏節線構成的集合， $A_R \subseteq A$

A 所有節線之集合，包含虛擬起點與網路圖之已知起點相連之節線，編號為 0 與 $a+1$ ，代表起始與結束， $a \in A$

參數

L 最大階層數

K 無人巡邏車數量

C_a 節線 a 之巡邏所需時間

M 極大的數

變數

$$x_a^{k,l} = \begin{cases} 1, & \text{節線 } a \text{ 被車輛 } k \text{ 在階層 } l \text{ 所巡邏過} \\ 0, & \text{otherwise} \end{cases}$$

數學模式

在已知各路段之不同方向的節線 a 之通行時間 C_a ，目標為最小化巡邏車隊的總巡邏時間，因此目標如式(3.3.1)所示。

$$\text{Minimize } \sum_{k=1}^K \sum_{l=0}^L C_a x_a^{k,l} \quad (3.3.1)$$

限制式(3.3.2)針對無人車尋訪所有必須巡邏節線集合 A_R 至少一次。

$$\sum_{k=1}^K \sum_{l=1}^L x_a^{k,l} \geq 1, \forall a \in A_R \quad (3.3.2)$$

限制式(3.3.3)規定所有車輛必須從階層 0 之虛擬起點出發到網路圖的起點。

$$x_0^{k,0} = 1, \forall k = 1, \dots, K \quad (3.3.3)$$

限制式(3.3.4)表示所有車輛在階層結束並完成巡邏後須回到虛擬起點。

$$\sum_{l=1}^L x_{a+1}^{k,l} = 1, \forall k = 1, \dots, K \quad (3.3.4)$$

限制式(3.3.5)、(3.3.6)為流量守恒限制式，包含一般節點與網路圖起點。

$$\sum_{\substack{a \in A \\ \text{tail}(a)=i}} x_a^{k,l} - \sum_{\substack{a \in A \\ \text{head}(a)=i}} x_a^{k,l-1} = 0, \forall l = 1, \dots, L, \forall k = 1, \dots, K \quad (3.3.5)$$

$$\sum_{\substack{a \in A \\ \text{tail}(a)=\text{depot}}} x_a^{k,1} + x_{a+1}^{k,1} - x_0^{k,0} = 0, \forall k = 1, \dots, K \quad (3.3.6)$$

$$\sum_{\substack{a \in A \\ \text{tail}(a)=\text{depot}}} x_a^{k,l} + x_{a+1}^{k,l} - \sum_{\substack{a \in A \\ \text{head}(a)=\text{depot}}} x_a^{k,l-1} = 0, \forall l = 2, \dots, L, \forall k = 1, \dots, K$$

3.3.3 網路縮減下之數學模式 M_{RL}^K

根據 3.1 節的問題描述，本研究旨在以最小化車隊總巡邏時間派遣無人車巡邏過所有必須巡邏節線。在給定一待巡邏網路中可將節點分為必須巡邏節線的起訖點、網路圖起點以及其他一般節點，本研究透過 Dijkstra 演算法針對網路圖起點與所有必須巡邏節線之起訖點求取兩點間之最短路徑來縮減網路，即可將兩點之間的一般節點與節線縮減成一條路徑(Path)，並將這些最短路徑以及原網路圖上所有必須巡邏節線一同建構為新的待巡邏網路圖 $G' = (N_R, A')$ ，必經節點集合 N_R 包含網路圖起點與必須巡邏節線之起訖點， A' 為原網路圖上所有必須巡邏節線和 N_R 中兩兩節點間的最短路徑，如圖 3.6 網路縮減至圖 3.7 網路，目標著重在於巡邏過必須巡邏之節線即可。

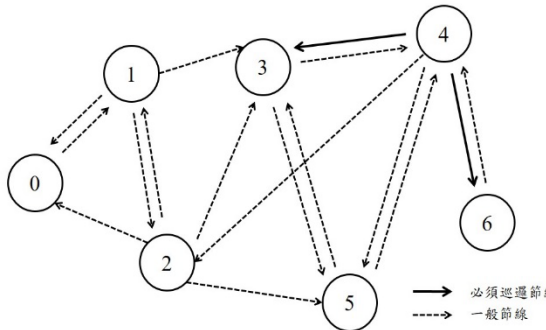


圖 3.6 縮減前之網路圖 G

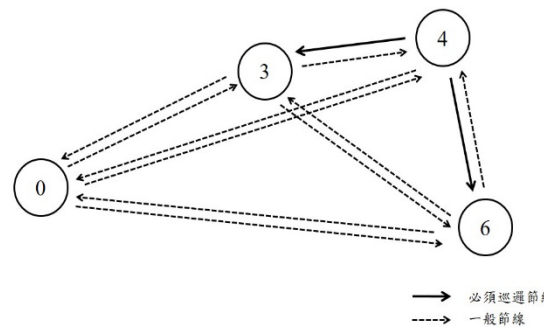


圖 3.7 縮減後之網路圖 G'

此模式藉由修改一般模式 M_L^K 而得，保留 3.3.2 節的限制式(3.3.1)~(3.3.6)為限制流量守恒與必須巡邏節線皆要被巡邏，並新增新的變數與限制式。以下新增新的符號定義：

集合

A' 所有必須巡邏節線之起訖點與網路圖起點兩點間最短路徑構成之集合

變數

$$y_{a,r}^{k,l} = \begin{cases} 1, & \text{路徑 } a \text{ 中含有必須巡邏節線 } r \text{ 被無人車 } k \text{ 在階層 } l \text{ 巡邏過} \\ 0, & \text{otherwise} \end{cases}$$

參數

$C_{a'}$ 經縮減網路之巡邏路徑 a' 所需時間

限制式

限制式(3.3.7)為路徑 a 若被巡邏，路徑上的必須巡邏節線 r 也會被巡邏過。

$$x_a^{k,l} = y_{a,r}^{k,l}, \forall a \in A', \forall k = 1, \dots, K, \forall l = 1, \dots, L \quad (3.3.7)$$

限制式(3.3.8)限制所有必須巡邏節線皆要被巡邏過。

$$\sum_k \sum_{l=1}^L y_{a,r}^{k,l} \geq 1, \forall r \in A_R \quad (3.3.8)$$

在經過網路縮減後，本小節之數學模式適用於必經節點集合較少的情況，因為限制式(3.3.7)是根據 N_R 以及節線數來限制必須巡邏節線一定要被巡邏至少一次，因此本模式必須產生許多限制式導致求解效率不彰，故建議採用此模式求解時應檢查 N_R 之多寡再進行求解，後續第五章數值分析會再驗證上述說法。

3.4 小結

本章首先介紹 Benavent et al. (2009)之數學模式並說明其求解過程繁複，在

每一次求解過程中找到可行解即要檢查是否有子迴圈存在，若有則要加入排除子迴圈限制式。因此我們以層空網路建構混整數規劃模式求解風向鄉村型多郵差問題可以避免子迴圈產生，且現實情況中存在單行道或平行節線，因此調整為以節線為必須巡邏重點。但由於混整數規劃變數與限制式過多，無法針對中大型更貼近現實的網路以有效的時間內求得最佳解。因此，下一章節將會針對無人車巡邏途程規劃問題提出較有效率的數種求解演算法。



第四章 多輛無人車途程規劃演算法之設計

在第三章以數學規劃模式為基礎利用求解軟體求取較大規模之網路，需耗時甚久，因此為了提升求解效率，本章節提出數種啟發式演算法進行求解，在短時間內可獲得品質不錯的可行解。

4.1 最鄰近待巡節線優先之貪婪式演算法(Greedy Algorithm)

貪婪式演算法為一種短視近利的演算法，在搜尋最佳解的過程中，找尋每一步驟中最有利的解，以期快速求取品質較佳的結果。根據 13.1 節問題描述，本研究之求解重點在於要尋訪過所有必須巡邏節線，利用當下時間巡邏工作負荷較少的無人車去巡邏，以達到車隊內負荷較為平均且較少總巡邏時間。演算法流程如下表所示，其中 s 為網路圖起點、 A_R 為必須巡邏節線之集合、 K 為無人車數量。

表 4.1 貪婪式演算法流程

Greedy Algorithm
1: function Greedy(s, A_R, K)
2: while A_R is not empty do
3: find $k \in K$ with minimum patrol load
4: find the shortest path from current node to the nearest source of $a \in A_R$
5: assign the nearest a to patrol load and delete a from A_R
6: end while
7: end function

此演算法會先找出所有無人車巡邏任務最少的無人車優先指派，計算所有必須巡邏節線的起點與該無人車所在節點之最短路徑，取最小者之必須巡邏節線指派給該無人車，並將指派過的節線從必須巡邏節線之集合 A_R 中刪除，持續以上步

驟直到所有必須巡邏節線皆被指派完畢。在指派結束後，本研究問題之目標值計算方法以圖 4.1 計算目標值示意圖進行說明，首先儲存網路圖起點與必須巡邏節線的起訖兩點，將網路圖起點與節線 1 之起點 3 的最短距離、節線 1 的巡邏時間、節點 8 至節點 6 之最短距離等依序計算後加總，而得最後的目標值。而求取兩兩節點間的最短路徑可藉由現存的最短路徑演算法達成，本研究以 Dijkstra 演算法來計算最短路徑。由於以貪婪式演算法所求解的並不一定為全域最佳解，因此可將此解作為 3.3.2 節與 3.3.3 節中數學模式之初始解，以期提高數學模式之求解效率。

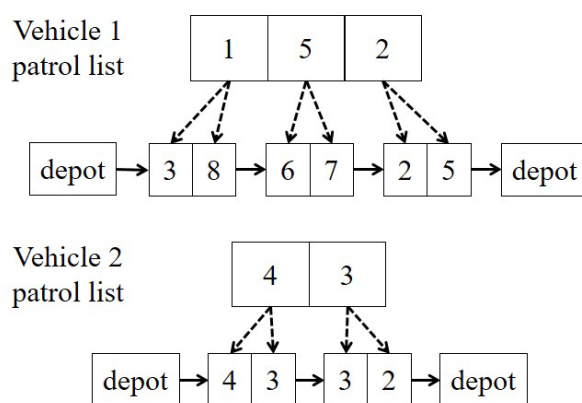


圖 4.1 計算目標值示意圖

4.2 基因演算法(Genetic Algorithm, GA)

在演算法設計上，除了貪婪式演算法外，路徑規劃問題常會使用基因演算法進行求解，而 GA 的概念源於進化生物學，在族群繁殖的過程中，兩個體會經由交配與基因突變進而產出差異的後代個體，當個體間具有差異，所具有的染色體也會不同，透過競爭後適應度低的個體將會逐步被淘汰，隨著迭代的次數不斷增加，最後可能會進化出適應值最高的全域最佳解。本研究參考 Karakatič and Podgorelec (2015)與 Lo et al. (2018)等以 GA 求解多旅行銷售員問題或車輛途程問題，實作其交配與變異的方法。

編碼 (encode)

每一條染色體皆可以代表為一組解，每台無人車皆有其需負責進行巡邏的節線編號，我們可以一陣列儲存之。若現在車隊共有兩台無人車且有必須巡邏節線五條須被巡邏，可將一條染色體以圖 4.2 表示之，以此例而言，無人車 1 從網路圖起點出發後會依序巡邏節線 1、5、2。以此類推，藉由此方法可以方便表示一組可行解，不須利用傳統的二元組合來進行編碼，最後則將此兩段染色體片段進行組合並在不同無人車的任務間插入一個負數(如圖中的 -1)代表為任務切割點，最後一段可行的染色體如圖 4.2 右半部所示。

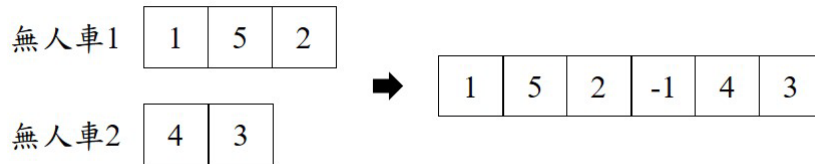


圖 4.2 染色體示意圖

適應函式 (fitness function)

透過計算適應函式我們可以求出該染色體的目標值，本研究問題為最小化車隊總旅行時間，故值愈小表示愈接近最佳解，在無人車依序巡邏完後，可以透過 4.1 節介紹的目標值計算方法分別計算無人車的巡邏時間，並加總而得到目標值。接著藉由下列介紹的交配以及突變策略，使得求出的適應值逐漸收斂變小。

挑選策略 (selection operator)

在每次迭代的一開始都會進行染色體的挑選，本研究的挑選策略則則是使用輪盤法的方式，即當一組可行解的目標值品質越好時，其能被選進此次迭代的候選染色體的機率就會越高，以下介紹其方法。

假設目前有 A_R 條染色體，其適應值分別為 $E(C_1), E(C_2), \dots, E(C_{A_R})$ 。

1. 將所有的適應值取其倒數： $E(C_i) = \frac{1}{E(C_i)} \quad \forall i = 1, \dots, A_R$ 。
2. 計算所有適應值的加總： $E_{total} = \sum_{i=1}^{A_R} E(C_i)$ 。
3. 計算各染色體被選到的機率： $P_i = E(C_i) / E_{total}$ 。
4. 利用累積機率的方式，每條染色體的累積機率為 Q_i 。
5. 最後產生隨機亂數 x ，若 x 為 $Q_i \leq x \leq Q_{i+1}$ ，則選擇染色體 Q_i 。

其中步驟 1 的計算方式之所以要取倒數的原因為本研究的目標問題為最小化問題，為了讓目標值愈小的染色體具較高挑選機率故以倒數表示。從步驟中可以看出，在經過挑選策略後，有較佳適應值的染色體將會有較高機率被選擇，而較差的染色體也不會被完全排除，因在後續的迭代中仍有可能會得到較佳的適應值。

交配策略 (crossover operator)

在交配策略中，因為每條必須巡邏節線只會被一組無人車隊巡邏，所以利用以下方法來確保交配後為可行解。一開始先隨機挑選兩條染色體當作父母，如圖 4.3 所示，並在染色體上隨機選取一個切割點，圖中代表以第三條節線為切割點。

接著將切割點後的節線進行交換策略，只要插入的節線並不存在於切割點之前，就進行插入；若存在則不插入，如圖 4.4，由於節線 4 與象徵任務切割點的負數並不存在切割點前，因此插入相對應的位置。剩餘未被指派的節線則被隨機插入空位中，如圖 4.5。

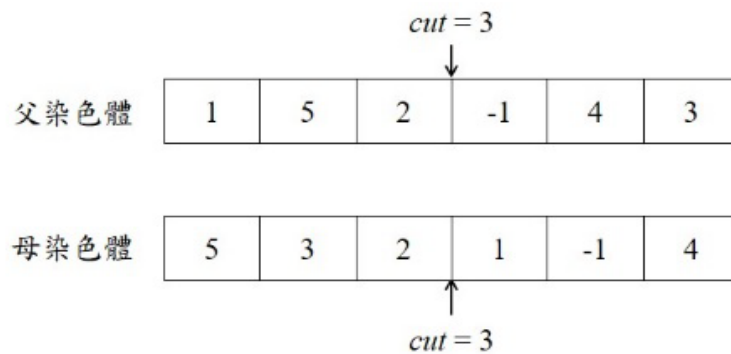


圖 4.3 選擇父母和交配的分割點

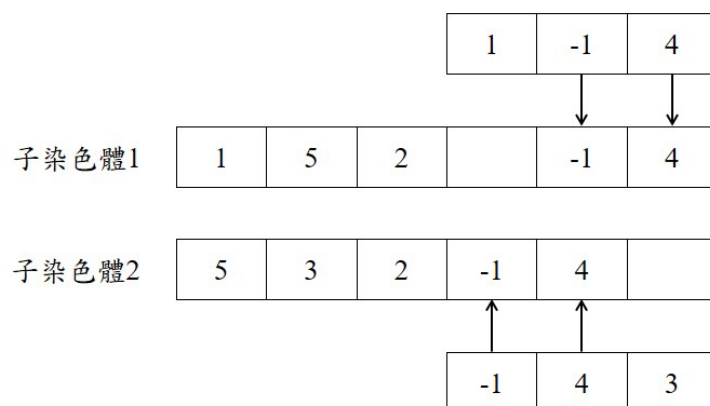


圖 4.4 進行交配與複製

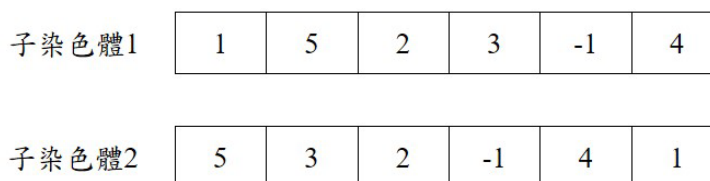


圖 4.5 最後子代結果

突變策略 (mutation operator)

突變策略是為了避免基因演算法落入區域最佳解，所以使產出的可行解有機率性的改變，以下為幾個常見的突變策略：

突變策略 1：交換法

隨機選擇染色體上兩個位置，將位置上的基因或基因區段進行互換，如圖 4.6，將節線 3 與節線 5 位置上的基因互相交換。

突變策略 2：插入法

隨機挑選染色體上兩個位置，將挑到第一個位置上的節線插入第二個位置上，如圖 4.7 所示，將節線 5 插到節線 3 的位置後，其他的節線依序往前遞補。

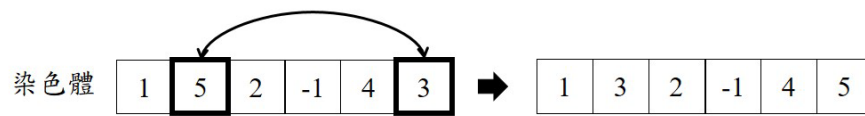


圖 4.6 突變策略之交換法

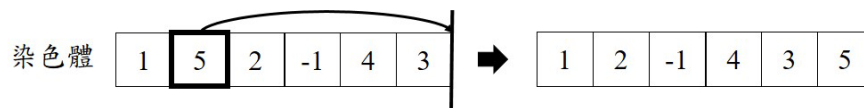


圖 4.7 突變策略之插入法

綜合以上可將基因演算法表示如表 4.2，其中 M 為每回合初始的染色體個數， P_c 為每次迭代進行交配的機率， P_m 為染色體突變的機率。

表 4.2 基因演算法流程

Genetic Algorithm	
1:	function GA(M, P_c, P_m)
2:	Generate initial population(M random chromosome)
3:	while stopping criteria not satisfied do
4:	evaluate each chromosome in population
5:	select M chromosome and copy them to new population
6:	probability P_c to do crossover and copy to population
7:	probability P_m for each chromosome to mutate
8:	end while
9:	end function

4.3 區域搜尋法(Local Search, LS)

區域搜尋法是一種常使用於車輛途程問題與旅行銷售員問題中的啟發式演算法，其概念為若存在一組初始解時開始向周邊的鄰域解進行搜尋，假如有表現更好的解則移動至該解，並從該組解繼續搜尋其鄰域解；否則返回當前之解繼續搜索，本研究參考 Hashimoto and Yagiura (2008)提出的區域搜尋法，該文獻利用 2-opt*、cross exchange 和 Or-opt 方法來搜尋鄰域解，同時為了使搜尋更有效率，對於每個節點僅搜尋其附近的節點。但因本研究探討的問題是以節線為主，因此將該方法修改後套用至本研究問題。

編碼方式

由於本研究以節線為探討對象，延續 4.2 節之編碼方式以節線編號進行編碼，每組可行解中仍會包含所有必須巡邏節線，並以二維陣列來儲存，進行完所有步驟後再該二維陣列進行組合，最後計算該組解之目標值。

鄰域列表(Neighbor List)

為了更有效率的在鄰域進行搜尋，針對每一條節線分別建立鄰域列表記錄每節線附近的節線，假若兩條路徑中的部分節線要進行交換時，其中判斷交換程序是否成立的基準為交換而相連的兩條節線是否存在鄰域列表中，若其中之一存在則進行交換，不然則否。因此我們在讀取完網路圖資訊後，就會建立所有節線的鄰域列表，用以記錄每條節線距離最短的 n_{list} 條節線。

2-opt* 演算法

2-opt* 演算法由 Potvin and Bengio (1996)提出，為 Lin (1965)針對旅行銷售員問題提出 2-opt 之變形。其概念為將 2 台無人車的路徑各切割成兩部分，接著將 2 台無人車的後半部分路徑進行交換，而演算法僅針對前半部的最後一條節線與後半部的第一條節線，若有在鄰域列表才能進行交換。

如圖 4.8 所示，針對無人車 k 與無人車 $k+1$ 之路徑進行交換，為了便於說

明，我們將具方向的節線以圓形進行代替，而圖中方形代表網路圖起點。首先可將無人車 k 的路徑拆成兩部分，第 i 條節線前與 $i+1$ 條節線後，同理無人車 $k+1$ 之路徑也能拆成兩部分，接下來判斷第 i 條節線與第 $j+1$ 條節線或第 j 條節線與第 $i+1$ 條節線有無存在鄰域列表，若有則交換並搜尋，若否則不交換。此方法的鄰域解數量為 $O(mn_{nlist})$ ， m 為所有節線的數量。為了順利進行交換程序，目前針對具最多巡邏時間的路徑與最少巡邏時間的路徑優先進行交換，其交換可使每台無人車被安排的巡邏任務較為平均。

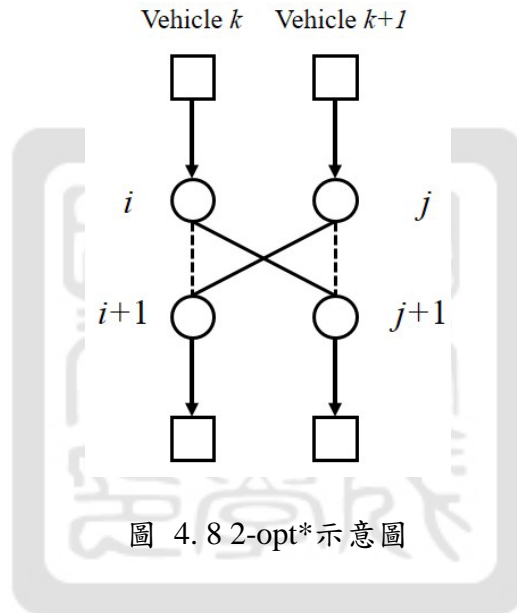


圖 4.8 2-opt* 示意圖

Cross exchange 演算法

Cross exchange 演算法由 Taillard *et al.* (1997) 提出，概念是將 2 台無人車的路徑各移除一段長度最多為 L^{cross} 的路徑，然後將此兩條路徑彼此進行交換，同樣地，為了交換程序的效率，僅針對存在於鄰域列表的節線進行交換，如圖 4.9 所示，針對無人車 k 與無人車 $k+1$ 之路徑進行交換，先挑選節線 i 之後長度至多 L^{cross} 的路徑與節線 j 之後長度至多 L^{cross} 的路徑，判斷節線 i 與節線 $j+1$ 或節線 j 與節線 $i+1$ 是否有在鄰域列表內，若有則交換並搜尋，並求解新路徑 P 的目標值 $E(P)$ ，若目標值小於全域解 $E(GBest)$ 則取代全域解，並從新的路徑 P 再執行 Cross exchange 演算法；否則不進行交換程序。此方法的鄰域解數量為 $O((L^{cross})^2 mn_{nlist})$ 。

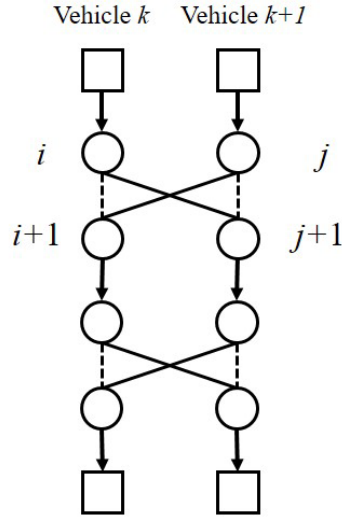


圖 4.9 Cross exchange 示意圖

Or-opt 演算法

Or-opt 演算法由 Reiter et al. (1965) 提出，其概念為移除一台無人車中路徑長度至多為 $L_{\text{path}}^{\text{intra}}$ 的路徑，插入到距離原移除位置至多 $L_{\text{ins}}^{\text{intra}}$ 的位置，如圖 4.10 所示，從節線 i 移除長度為 $L_{\text{path}}^{\text{intra}}$ 的路徑，將此路徑插入位置 $i + L_{\text{ins}}^{\text{intra}}$ ，並計算該次程序是否有使目標值更好，若有則進行此次交換程序並從交換後的新路徑開始尋找鄰域解；若未能改善目標值，則從原先之路徑繼續搜尋鄰域解。而此方法的鄰域解數量為 $O(L_{\text{path}}^{\text{intra}} L_{\text{ins}}^{\text{intra}} m)$ 。

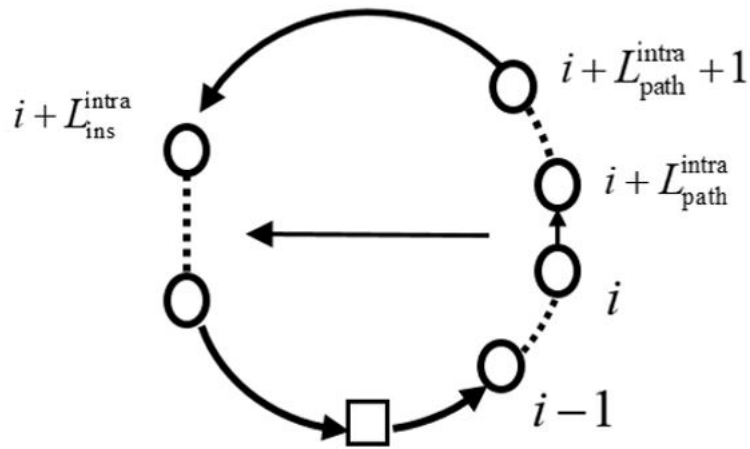


圖 4.10 Or-opt 示意圖

在介紹完上述演算法後，說明區域搜尋法的步驟如下：演算法由外而內分別是 cross exchange、2-opt*和 Or-opt，由最內層的 Or-opt 開始函式，每當執行完 Or-opt 函式，就會執行一次 2-opt*，然後繼續執行 Or-opt，直到 2-opt*執行完，同樣地，當 2-opt*執行完，就會執行更外層的 Cross exchange，若有目標值更佳的解出現，立即接受該解，並重新從 Or-opt 函式開始，直到滿足終止條件。

4.4 小結

儘管求解數學規劃模式在小例子中會求得最佳解，但往往問題隨著規模擴大，求解時間會大幅增加，求解品質會逐漸下降。為求在短時間內得到品質高的可行解，演算法的實現將是必要的，本章首先提出將解利用編碼的方式呈現，我們主要想知道必須巡邏節線的巡訪順序，而詳細節線與節線間如何連接則是利用最短路徑演算法而得。本章提出利用兩種方式進行求解，第一種是利用貪婪式演算法，在面臨選擇下個拜訪節線時挑選距離目前節點最近的節線優先巡邏，而做出當下最有利的決策，能快速地提供一個品質不錯的可行解。第二種是基因演算法，透過染色體間的交配與突變，不斷迭代找到較佳的組合，而區域搜尋法除針對內部路線的重組優化外，更在多台無人車的路徑間交換部分巡邏節線來進行優化，為了增加交換的效率，我們僅對於鄰域節線交換，當達到終止條件時即完成搜尋。

第五章 整數規劃模式與啟發式演算法之數值測試與分析

此章節將會針對本研究所建構的混整數規劃模式及演算法進行測試與分析，本研究的測試環境為 Windows 10 作業系統，搭配 Intel Core i7-6770，3.40GHz*8 處理器與 16G 記憶體，以 Microsoft Visual Studio 2015 為程式語言介面，使用 C++ 程式語言撰寫程式，並利用最佳化軟體 Gurobi，版本為 8.1.1 求解，求解時間設定為 2 小時(7200 秒)，以下說明測試網路參數設定，接著針對不同的參數設定與網路大小等不同狀況分析數據。

5.1 測試網路資料

在現實的資料中，因為待巡邏網路通常如同現實地圖而沒有特殊結構，所以在網路圖的測試使用隨機產生的方式為主。本研究使用 Cherkassky et al. (1999)和 Goldberg and Silverstein (1997)兩篇求解最短路徑的論文中所使用的資料產生套件(<http://www.avglab.com/andrew/soft.html>)來產生隨機網路圖，此套件提供的 sp_rand 可以根據給定的節點數 n 與節線數 m 來隨機產生網路圖，再利用隨機亂數給定不重複的必須巡邏節線編號以補足給定的必須巡邏節線數量 m_R ，其中我們針對必須巡邏節線的兩端點加上網路圖起點所構成的必經節點數 n_R 進行統計，以利後續分析，最後決定無人車的數量 K 後，針對不同網路規模來進行求解。

以下針對隨機網路測試小型、中型與大型的網路規模，且每個規模皆有 5 個例子，詳細資訊如表 5.1。

表 5.1 網路結構資訊

網路名稱	節點數 n	節線數 m	必須巡邏節線數 m_R	必經之節點 n_R
$random_s$	30	120	[10, 18]	[9, 18]
$random_m$	60	240	[28,30]	[37, 41]
$random_l$	90	360	[45,46]	[53, 62]

5.2 多輛無人車之巡邏途程規劃問題之測試

關於多輛無人車之巡邏途程規劃問題的測試，首先我們將針對整數規劃模式與縮減網路之數學模式進行測試，接著再比較數學啟發式演算法之求解效率與求解品質。

5.2.1 整數規劃模式比較

本小節針對以基本模型的數學模式 M_L^K 與 M_{RL}^K 和分別加入初始解之模式 \bar{M}_L^K 與 \bar{M}_{RL}^K 進行測試，初始解的來源為 4.1 節之貪婪式演算法之求解結果。表 5.2 為模式大小推估，以 *big-O* 估算模式的變數及限制式的個數，可看出模式 M_L^K 的大小主要取決於節線數 m 、無人車隊中最長路徑之節線數 L 與無人車數量 K ，而 L 的大小主要受到 m 與 K 的影響，當 m 愈大或 K 愈小則 L 會愈大。而模式 M_{RL}^K 則是透過網路限縮，新的網路將由必經節點所建構，因此模式的大小除了取決於 L 、 K ，主要還是受到必經節點數 n_r 所影響。

表 5.3 為比較模式 M_L^K 、 M_{RL}^K 、 \bar{M}_L^K 、 \bar{M}_{RL}^K 的求解表現，其中，Num 為該組網路測試資料數；CPU time 為數學模式的求解時間；Gap 是利用最佳化軟體 Gurobi 之測試結果(求解時限為 7200 秒)，計算方式如下：

$$Gap = \frac{UB - LB}{UB} \times 100\% \text{。} UB \text{ 及 } LB \text{ 為 Gurobi 所求得的上、下界。}$$

階層數的設定一開始皆按照表 5.3 中所設，一旦無法求解(模式不可行)時，在程式內部自動增加階層數再次進行求解，直到模式所得為可行解。由表 5.3 可以發現當階層數 L 增加時，求解時間會大幅上升。而加入初始解的模型，在收斂的速度上明顯較未加入初始解的模型來的有效。另外在相同的網路規模下，若巡邏節線兩端點與網路圖起點之總數 n_r 較大時，會使得模式 M_{RL}^K 與 \bar{M}_{RL}^K 之求解效率

較佳；這是因為在進行縮減網路時，僅保留 n_R 內的節點進行求解節點到節點之最短路徑；反之若 n_R^2 大於 m 時，會導致 M_{RL}^K 的網路規模明顯大於 M_L^K ，間接使得求解時間也會大幅增加。

表 5.2 混整數規劃模式之問題大小估算

建構網路	數學模式	二元變數個數	實數變數個數	限制式個數
層空	M_L^K	$O(mLK)$	-	$O(mLK)$
	M_{RL}^K	$O(n_R^2LK)$	-	$O(m_R n_R^2LK)$

表 5.3 數學模式測試(求解效率)

Network	K	L	Num	M_L^K		M_{RL}^K		\bar{M}_L^K		\bar{M}_{RL}^K	
				Num opt	CPU Times	Num opt	CPU Times	Num opt	CPU Times	Num opt	CPU Times
$random_s$	1	50	5	5	0	5	7.2	5	10.2	5	2.8
	2	40	5	5	0	5	26	5	37.6	5	14
	4	30	5	5	0	5	7.2	5	72.4	5	35.6
$random_m$	1	90	5	5	1736.6	5	6101	5	1790.6	5	4834.4
	2	75	5	3	6049	5	7200	5	6380.4	5	7200
	4	60	5	0	7200	3	4815	0	7200	3	7200
$random_l$	1	120	5	0	-	0	-	0	-	0	-
	2	105	5	0	-	0	-	0	-	0	-
	4	90	5	0	-	0	-	0	-	0	-

註：-表示該組網路的測試資料，模式在求解時間上限(7200 秒)內無法求得最佳解。

表 5.4 數學模式測試(求解品質)

Network	K	L	Num	M_L^K		M_{RL}^K		\bar{M}_L^K		\bar{M}_{RL}^K	
				Num opt	Gap (%)	Num opt	Gap (%)	Num opt	Gap (%)	Num opt	Gap (%)
$random_s$	1	50	5	5	0.0	5	0.0	5	0.0	5	0.0
	2	40	5	5	0.0	5	0.0	5	0.0	5	0.0
	4	30	5	5	0.0	5	0.0	5	0.0	5	0.0
$random_m$	1	90	5	5	0.0	5	0.0	5	0.0	5	0.0
	2	75	5	3	2.1	5	0.0	5	0.0	5	0.0
	4	60	5	0	2.9	3	2.8	0	2.5	3	4.1
$random_l$	1	120	5	0	2.8	0	11.8	0	2.2	0	12.0
	2	105	5	0	8.4	0	17.0	0	7.6	0	16.8
	4	90	5	0	-	0	-	0	-	0	-

註：-表示該組網路的測試資料，模式在數次增加階層數後求解模式仍不可行。

5.2.2 數學演算法測試

在演算法的部分，針對貪婪式演算法、基因演算法與區域搜尋法進行測試，表 5.4 為演算法之參數設定，在 GA 的部分，初始的染色體總量 M 為 10 條，交配的機率 P_c 為 80%、突變的機率 P_m 為 20%，此部分若將突變機率提高，會越接近亂數隨機決定，所以設定 20% 使其有機會可跳出區域最佳解，但也不會過於隨機。針對迭代次數的設定測試，圖 5.1 到圖 5.4 分別測試中小型的網路在無人車數在 1 與 2 台時求解一萬次迭代的結果。可以發現，整體來看幾乎是在迭代到 4000 次就可差不多收斂，較難再找到更好的解。所以之後的測試皆使用 4000 次的迭代次數求解小型的隨機網路。而 LS 的部分，參數設定是參考 Hashimoto and Yagiura (2008) 的設定，鄰域解個數 n_{list} 為 10、cross exchange 演算法中交換最大長度 L^{cross} 為 3、Or-opt 演算法中移除最多 L_{path}^{intra} 條節線插入鄰近 L_{ins}^{intra} 為 3 的位置，且 LS 在進行搜尋時若超過 100 次沒進步即停止。

表 5.5 數學演算法參數設定

GA				LS			
P_c	P_m	M	Iteration	n_{list}	L^{cross}	L_{path}^{intra}	L_{ins}^{intra}
80%	20%	10	4000	10	3	3	3

在演算法求解的部分，我們分別對 3 個方法進行測試，測試的標準依據數學模型求得的最佳解(由於大型網路圖模型無法得到最佳解，因此我們以 \overline{LS} 求得的解作為比較基準)，由表 5.6 數學演算法測試(求解效率)和表 5.7 可看出， LS 在短時間內就可以獲得品質不錯的可行解，但在車輛數為 1 時表現較差，其原因可能為容易陷入區域解中而無法跳脫。另外在 GA 中花費較多時間在處理交配、突變與挑選的過程，但所獲得的求解品質也並無優於 Greedy 和 LS ，然而當我們將 Greedy 當作初始解輸入 GA 以及 LS 中，我們以 \overline{GA} 和 \overline{LS} 代表之，在 \overline{GA} 中雖然稍微減慢收斂速度，求解品質卻有所提升，然而在交配過程中則會將明顯較佳的初始解犧牲，整體而言，基因演算法在解的品質上與 LS 和 Greedy 有著相當大的落差，這部分可能需要改善基因演算法的交配與變異方法。

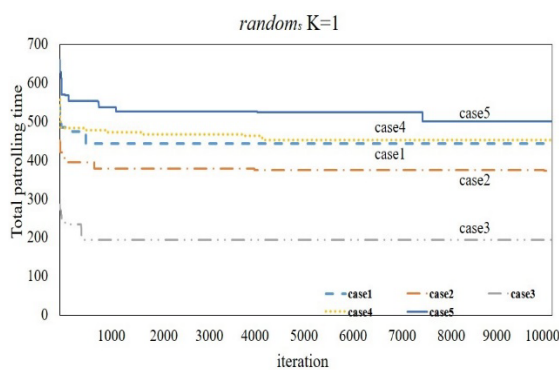


圖 5.1 GA 小型網路收斂測試(車數=1)

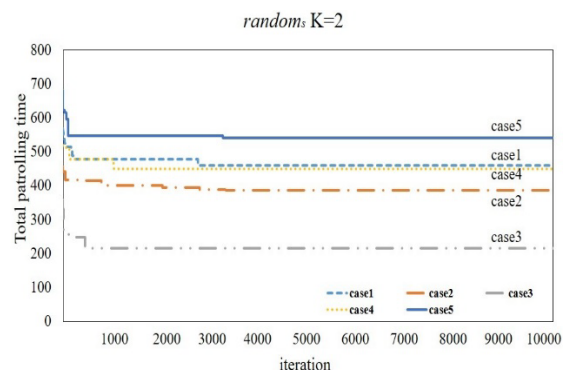


圖 5.2 GA 小型網路收斂測試(車數=2)

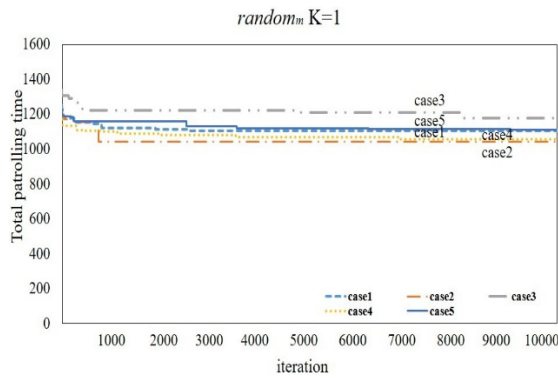


圖 5.3 GA 中型網路收斂測試(車數=1)

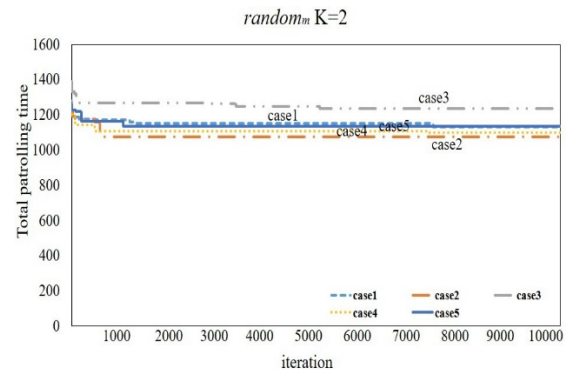


圖 5.4 GA 中型網路收斂測試(車數=2)

表 5.6 數學演算法測試(求解效率)

Network	K	Num	Greedy	GA	\overline{GA}	LS	\overline{LS}
			CPU time(s)	CPU time(s)	CPU time(s)	CPU time(s)	CPU time(s)
$random_s$	1	5	0.01	3.5	5.9	0.1	0.2
	2	5	0.01	29.7	49.3	0.4	0.3
	4	5	0.01	85.7	142.8	0.1	0.1
$random_m$	1	5	0.09	15.7	15.8	0.6	1.0
	2	5	0.09	184.8	185.1	13.2	7.3
	4	5	0.09	584.1	581.7	0.3	0.9
$random_l$	1	5	0.25	32.8	33.0	2.4	3.0
	2	5	0.26	464.3	465.4	150.1	54.8
	4	5	0.26	1297.5	1305.3	16.1	3.4

表 5.7 數學演算法測試(求解品質)

Network	K	Num	Greedy	GA	\overline{GA}	LS	\overline{LS}
			Gap(%)	Gap(%)	Gap(%)	Gap(%)	Gap(%)
$random_s$	1	5	16.6	13.1	14.3	18.5	6.0
	2	5	26.0	15.3	17.8	7.7	3.8
	4	5	36.1	46.5	19.7	3.7	3.4
$random_m$	1	5	21.1	45.4	17.0	30.3	11.7
	2	5	23.3	52.5	16.5	6.4	2.9
	4	5	28.3	53.6	27.3	3.2	2.3
$random_l$	1	5	0.0	45.8	1.0	26.8	0.0
	2	5	25.5	40.8	8.5	1.3	0.0
	4	5	1.1	39.0	1.1	3.3	0.0

註：由於在大型網路中模式無法求得最佳解，因此以 \overline{LS} 求解結果為比較基準。

5.3 小結

一開始介紹測試網路的參數設定，以 *big-O* 計算數學模式變數與限制式的數量，並估算模式大小，接著測試數學規劃模式的求解效率與品質，並加入演算法求得之初始解進入模型中進行比較。接下來針對 3 種數學演算法進行數值測試，可以發現貪婪式演算法可以在不同規模的問題中都可以快速求得不錯的可行解，而基因演算法在加入初始解後雖能提升求解品質但仍有許多進步空間，而區域搜尋法在不同的規模中都能快速地求解出不錯的可行解。



第六章 結論及未來研究

6.1 結論

本研究探討當警力難以兼顧大範圍巡邏任務時，該如何以系統性的學理建模以及有效率的求解演算法設計，以規劃出最佳的多台巡邏車路線安排。如此即可透過搭配資通訊系統的無人巡邏車來分擔巡邏勤務，以達到有效預防犯罪及儘快針對不法情資進行蒐證的效果。過往在巡邏重點路段時，通常僅將巡邏需求的耗時與效益簡化成設置在該路段的任一方向即可，然而本論文將此簡化的假設延伸成各路段的巡邏需求、效益與通行時間可依順逆方向而不同的 KWRPP，因此更符合實際，並更具挑戰性與困難性。以下陳列本論文具體貢獻，並於 6.2 小節建議未來可嘗試的研究議題：

具體貢獻：

1. 過去文獻常將具有巡邏需求的路段任一方向有被巡邏過即代表巡邏效益可以擴大至雙向，而本研究考慮巡邏需求會依方向將路段轉換成必須巡邏之節線，可進一步處理平行節線(parallel arc)與單行道之網路圖，更能貼近現實情況。(第二章、第三章)
2. 提出以「層空網路」建構之數學規劃模式，利用 Level-space 的方式展開網路，階層代表節線的巡邏順序，每台無人車在其階層內相連節線而自成一路徑，共 K 個層空網路。此方法求解效率不佳，且受最大階層數 L 和無人車數量 K 影響。(第 3.3 小節)
3. 提出以最短路徑演算法對網路圖進行預處理，僅由必經節點來建構網路圖，相鄰節點間藉由最短路徑相連，接著以「層空網路」來建構數學規劃模式。此方法對於必經節點比例較低的網路圖有較佳的求解表現，但求解效率仍會受到最大階層數 L 與無人車數量 K 影響。(第 3.3 小節)
4. 提出貪婪式演算法(Greedy)、基因演算法(GA)及區域搜尋法(LS)。其中，貪

婪式演算法會在無人車在選擇巡邏節線時，會考慮當下所有待巡節線中挑選距離最近的必須巡邏節線，並以當下車隊中負擔最少的無人車優先指派，進而將規劃出所有無人車的巡邏路徑。基因演算法則利用交配與突變來迭代出品質較佳的可行解；而區域搜尋法針對鄰域進行搜尋，一旦搜尋至較佳的鄰域解，移動至該解後繼續搜尋，直到達成終止條件。(第 4 章)

5. 將貪婪式演算法所求可行解作為初始解，輸入數學規劃模式、基因演算法與區域搜尋法中進行求解，經過迭代後得到品質較佳的結果。數值測試結果顯示，我們最推薦以初始解為基礎再利用區域搜尋法進一步收斂的方法，因其求解表現最佳，可在短時間內求解出品質較佳的可行解。(第 5 章)

6.2 未來研究

若以本研究探討的巡邏途程規劃問題為基礎，我們發現尚有許多相關研究問題值得進一步深究，茲列舉部分如下：

1. 可持續精進及發展本研究之數學模式與演算法：

由於本研究之數學模式無法求解中與大規模網路，由層空網路產生限縮節線的方式，目前保留必經節點所建構的縮減網路圖為完全圖，或可嘗試將此完全圖再進一步縮減，例如：透過貪婪機制進行單一節點 i 挑選前 l 短節線巡邏時間 c_{ij} 的連出節線(outgoing arc)。但此方法在 l 太小的情形下有可能會找不到可行解；而 l 太大時，又可能會使求解時間過久。

2. 考慮時窗限制

在現實中每條路段可能在不同時段具不同的巡邏效益，某些地區在特定時間內比較容易滋生犯罪情事，或是當交通尖峰時刻會使路段的通行時間上升，導致無人車巡邏該路段需要較高的巡邏時間，在未來研究可將此因素列入考量而更能貼近真實的情況，同樣地，這方面的做法可參考時窗限制的鄉村型多郵差問題(Time-Dependent Rural Postman Problem, TDRPP)。

3. 考慮巡邏路線的變動彈性

當一路段發生緊急即時事件時，需要警察或無人巡邏車到該處巡訪或追緝犯人，為了使距離該現場最近的無人車最慢何時抵達目標節線，應可考慮無人車巡邏路線的變動彈性，以提高更多必須巡邏節線的巡邏效益。

4. 考慮無人巡邏車的能源消耗

本研究假設無人車在停放場址內會將能源補足，指派巡邏任務後出發進行巡邏勤務，而該無人車隊之滿載能源足以負荷一次巡邏勤務，但現實情況中，無人車的能源耗損可能難以負擔多次巡邏指派，而可能在巡邏途中進行能源的補充，因此也許以巡邏時間來推估能源消耗的過程，假如巡邏途中將要耗盡能源時先行前往補給，並將補給時間納入總巡邏時間中是較為符合現實的作法。

5. 考慮巡邏路線不重複性

根據 Sherman and Weisburd (1995)美國明尼阿波利斯的犯罪熱點巡邏實驗，可知犯罪並非平均分布在城市的各個區域，而是顯著聚集在一些小的熱點區域。針對這些犯罪熱點有效投注警力搭配隨機不可預期的無人車共同執行巡邏計畫，考慮空間與時間的不重複性，不僅能提高見警率，更能改善治安與預防犯罪。

參考文獻

- 張棟(2017 年 7 月 3 日)。迪拜招募最新機器警察，配面部識別技術及無人機。雷鋒網。取自 <https://www.leiphone.com/>
- Aráoz, J., Fernández, E., & Meza, O. (2009). Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, 196(3), 886-896.
- Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2010). The undirected capacitated arc routing problem with profits. *Computers & Operations Research*, 37(11), 1860-1869.
- Archetti, C., Guastaroba, G., & Speranza, M. G. (2014). An ILP-refined tabu search for the directed profitable rural postman problem. *Discrete Applied Mathematics*, 163, 3-16.
- Assad, A. A., Pearn, W. L., & Golden, B. L. (1987). The capacitated Chinese postman problem: Lower bounds and solvable cases. *American Journal of Mathematical and Management Sciences*, 7(1-2), 63-88.
- Baldacci, R., & Maniezzo, V. (2006). Exact methods based on node-routing formulations for undirected arc-routing problems. *Networks*, 47(1), 52-60.
- Benavent, E., Corberán, A., Plana, I., & Sanchis, J. M. (2009). Min-Max K-vehicles windy rural postman problem. *Networks: An International Journal*, 54(4), 216-226.
- Black, D., Eglese, R., & Wøhlk, S. (2013). The time-dependent prize-collecting arc routing problem. *Computers & Operations Research*, 40(2), 526-535.
- Bodin, L. D., Golden, B. L., Assad, A., & Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, 10(2), 63-211.

- Chang, Y., & Chen, L. (2007). Solve the vehicle routing problem with time windows via a genetic algorithm. *Discrete and continuous dynamical systems supplement*, 240-249.
- Cherkassky, B. V., Goldberg, A. V., & Silverstein, C. (1999). Buckets, heaps, lists, and monotone priority queues. *SIAM Journal on Computing*, 28(4), 1326-1346.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- Feillet, D., Dejax, P., & Gendreau, M. (2005). The profitable arc tour problem: solution with a branch-and-price algorithm. *Transportation Science*, 39(4), 539-552.
- Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2), 109-124.
- Foulds, L., Longo, H., & Martins, J. (2015). A compact transformation of arc routing problems into node routing problems. *Annals of Operations Research*, 226(1), 177-200.
- Goldberg, A. V., & Silverstein, C. (1997). Implementations of Dijkstra's algorithm based on multi-level buckets. In *Network optimization* (pp. 292-327). Springer, Berlin, Heidelberg.
- Grötschel, M., & Win, Z. (1992). A cutting plane algorithm for the windy postman problem. *Mathematical Programming*, 55(1-3), 339-358.
- Guan, M. (1962). Graphic programming using odd and even points. *Chinese Math.*, 1, 237-277.
- Hashimoto, H., & Yagiura, M. (2008, March). A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. In *European Conference on Evolutionary Computation in Combinatorial Optimization* (pp. 254-265). Springer, Berlin, Heidelberg.

- Karakatič, S., & Podgorelec, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, 27, 519-532.
- Koskosidis, Y. A., Powell, W. B., & Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation science*, 26(2), 69-85.
- Lenstra, J. K., & Rinnooy Kan, A. H. G. (1976). On general routing problems. *Networks*, 6(3), 273-280.
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10), 2245-2269.
- Lo, K. M., Yi, W. Y., Wong, P. K., Leung, K. S., Leung, Y., & Mak, S. T. (2018). A genetic algorithm with new local operators for multiple traveling salesman problems. *International Journal of Computational Intelligence Systems*, 11(1), 692-705.
- Malandraki, C., & Daskin, M. S. (1993). The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. *European Journal of Operational Research*, 65(2), 218-234.
- Mei-Ko, K. (1962). Graphic programming using odd or even points. *Chinese Math.*, 1, 273-277.
- Minieka, E. (1979). The Chinese postman problem for mixed networks. *Management Science*, 25(7), 643-648.
- Moreira, L. M., Oliveira, J. F., Gomes, A. M., & Ferreira, J. S. (2007). Heuristics for a dynamic rural postman problem. *Computers & Operations Research*, 34(11), 3281-3294.
- Orloff, C. S. (1974). A fundamental problem in vehicle routing. *Networks*, 4(1), 35-64.
- Pearn, W. L. (1994). Solvable cases of the k-person Chinese postman problem. *Operations Research Letters*, 16(4), 241-244.

- Pearn, W. L., & Wang, K. H. (2003). On the maximum benefit Chinese postman problem. *Omega*, 31(4), 269-273.
- Potvin, J. Y., Kervahut, T., Garcia, B. L., & Rousseau, J. M. (1996). The vehicle routing problem with time windows part I: tabu search. *INFORMS Journal on Computing*, 8(2), 158-164.
- Reiter, S., & Sherman, G. (1965). Discrete optimizing. *Journal of the Society for Industrial and Applied Mathematics*, 13(3), 864-889.
- Sherman, L. W., & Weisburd, D. (1995). General deterrent effects of police patrol in crime “hot spots”: A randomized, controlled trial. *Justice quarterly*, 12(4), 625-648.
- Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2), 170-186.

