國 立 成 功 大 學

工 業 與 資 訊 管 理 研 究 所

碩 士 論 文

考慮感測數據空窗與延遲時長限制之多架無人機最多感測數據收集路線規劃問題研究

A Study of the Maximum Sensing Data Collection Routing Problem by Multiple UAVs with Sensing the Idleness and Latency Constraints

研 究 生　　　：白荷希 Baiq Zuyyina Hilyatur Rozaliya

指導教授　　　：王逸琳 教授 Prof. I-Lin Wang

共同指導教授　：Ahmad Muklason, Ph.D

中華民國　110 年 6 月 11 日

國立成功大學

碩士論文

考慮感測數據空窗與延遲時長限制之多架無人機最多感測數據收集路線規劃問題研究

A Study of the Maximum Sensing Data Collection Routing Problem by Multiple UAVs with Sensing Idleness and Latency Constraints

研究生：白荷希

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳
周詩梵
吳浩庠

指導教授：王逸琳

單位主管：翁慈宗

(單位主管是否簽章授權由各院、系(所、學位學程)自訂)

中 華 民 國 110 年 6 月 11 日

摘要

本研究探討一個多架無人機的感測數據收集路線規劃問題，為一個「合作式數據傳輸模式」，旨在指派無人機群自基地起飛，行經一或數個散在各處的感測站附近，以收集感測數據，再將收集到的數據藉由無線傳輸，以傳到其通信範圍內的另一架無人機或基地，此資料的收集與傳輸必須足設定的感測數據空窗與延遲時長限制。其中，各筆感測數據的「空窗時長」代表從同一感測站收集而得的連續兩筆數據之間的最大時間間隔長度；而「延遲時長」則代表該筆感測數據自其來源的感測站開始，直至基地接收到的總時間歷程長度，亦即類似數據「新鮮度」的概念。在滿足上述兩類時長限制的假設下，本研究擬規劃各架無人機路線，以在指定的規劃期間內，藉由合作協同傳遞資料的方式，重複執行上述任務，達成能於基地接收到最多數據的目的。

本研究以時空網路為基礎，提出一個整數規劃模型來追踪感測站所產生的各筆數據被收集和傳輸的過程。為檢驗使用無線通訊方式協同合作傳輸數據的效益，本研究亦提出一個單獨收集傳送資料的「非合作式數據傳輸模式」來對比，在該設定下，每架無人機都必須獨力收集數據，不得使用無線通訊將數據轉傳給他架無人機。實測結果顯示，合作式數據傳輸模式可以達到在更少的數據延遲下，仍能傳輸更多數據的不錯效果。

雖然本研究所提出的整數規劃模型可以求解出理論最佳解，但其求解過程十分耗時。為能在更短的時間內計算出好的解，本研究亦設計了一個直覺式的啟發式演算法，該演算法先計算出一條連接基地與所有感測站的短哈密頓迴圈，並於其上

均勻分佈無人機。由初步的數值測試結果看來，該演算法能處理較大的測資，但其求解效能仍有改善的空間。

關鍵詞：整數規劃，時空網絡，多架無人機，路線規劃，空窗，延遲

Abstract

We discuss a multi-UAV surveillance routing problem that routes a UAV fleet from a base station (i.e., depot) to repeatedly collect data from sensing locations during a specific planning horizon. The data in a sensing area is generated and ready to be collected periodically in a store-and-forward fashion. A UAV has to pass through some nodes within the communication range to collect data from a sensing location. It then carries the data and wirelessly transfers it to another UAV as long as that UAV is located within its communication range.

We seek the maximum collected data that the base station can receive during the planning horizon. Specifically, we calculate the optimal flight paths for UAVs over the space and time necessary to transfer the maximum amount of collected data. Furthermore, it is hoped that the idleness and latency, defined as the maximum length of time between consecutive data collected from a sensing location and the length of time (i.e., freshness) required for the data traveling from a sensing location to the base station, respectively, necessary to satisfy the given upper bounds.

We proposed an integer programming model on a time-space network to track how the generated surveillance data is collected and transferred. Implementation of the model is provided and compared to an uncooperative data transport model. Each UAV has to transfer the data individually to demonstrate the applicability of the proposed model and possible extensions. The results show that the cooperative data transport model can transfer more collected data with less latency than the uncooperative data transport model.

Although a mathematical model can be used to calculate an optimal solution, it is quite time-consuming. To calculate a good solution in a shorter time, we proposed a heuristic algorithm that evenly distributes UAVs on a short Hamiltonian cycle connecting the depot and all sensing locations. Computational experiments indicate that the proposed algorithm calculates good solutions and can deal with larger cases but still has room to improve in terms of its solution quality.

Keywords: Integer Programming, Time-Space Network, Multi-UAV surveillance routing, idleness, latency

Acknowledgment

First of all, I would like to thank the almighty God, Allah *Subhanahu wa ta'ala*, for His blessing, mercy, and opportunities that made it possible for me to continue my second year in the master's degree program at NCKU. I thank Him, the almighty, for the strength and guidance needed to complete this thesis.

I want to express my deepest gratitude to my advisors, Prof. I-Lin Wang (王逸琳), for accepting and trusting me as one of his students. I thank him for his assistance, guidance, dedication, patience, and immense knowledge during the time I have been completing this thesis. It was a great honor to work and study under his guidance. In addition, I also would like to thank all the committee members, Prof. Wu (吳浩庠), Prof. Chou (周詩梵), Prof. Hsieh (謝中奇), and Prof. Lyu (呂執中), who were willing to take the time to review the thesis and put forward many valuable suggestions improve it. I also want to express my sincere thanks to my advisor in ITS, Pak Ahmad Muklason, for trusting and encouraging me to continue my master study in the master's degree program at NCKU. It is great opportunity for me, and he let me explore another field to enrich my knowledge.

In addition, I would like to thank my lab mates: Rani, 胡承中, 張書桓, 沈郁恩, 李蕎伃, and 徐威銓, and my roommates: Mbak Maya, Afifah, Kiki, and Esas. Thank you for helping me whenever I struggled and helping me have an enjoyable life in Taiwan.

Finally, I want to express my gratitude to my family for their support. They have been my strongest motivation and the reason I have continued to move forward despite the difficulties I have encountered during my study. In spite of many things, they were worried about, they still supported my decision to pursue my degree in Taiwan. I feel happy and grateful for their understanding so that I could focus on my studies.

Baiq Zuyyina Hilyatur Rozaliya

Department of Industrial and Information Management

National Cheng Kung University, Tainan City, Taiwan.

June 2021

Contents

List of Figures

List of Tables

Chapter 1
INTRODUCTION

UAVs are used to continuously monitor specific locations (sensing locations) in some areas. They can send the collected data, such as images or videos, to the base station via physical contact or wireless communication. Due to its limited flight and wireless communication range, a single UAV cannot cover a large area without refueling or recharging. The use of multi-UAVs to carry the data in a store-and-forward fashion is thus considered. Assuming the entire area forms an undirected grid graph, some nodes (the base station and sensing locations) or arcs must be passed by a UAV to collect the data. These data are transferred to the base station via two types of arcs: (1) a flight arc physically carried by a UAV and (2) a communication arc carried by wireless communication. This thesis is an attempt to find an optimal route for each UAV to transport the data in a store-and-forward fashion within the planning horizon while considering the upper bounds on the idleness and latency. To elaborate further, we divide this chapter into four sections. We give the background for our research in Section 1.1. Then, in Section 1.2, we explain our research objective. Section 1.3 defines the scope of the research. Lastly, Section 1.4 provides an outline of our proposed approach.

1.1. Research Background

Recent progress in aerial robotics has led to increased interest in the use of unmanned aerial vehicles (UAVs) (Coelho et al., 2017). Several unique features of UAVs, including a wide field of view due to being small and highly mobile, low cost, fast deployment, flexibility, and zero casualties (Fu et al., 2019), make them attractive for use in several applications. Some applications that make use of UAVs include intelligence, surveillance, military reconnaissance (ISR) missions and civil tactical scenarios (Banfi et al., 2015), disaster response, and pollution source tracking (Fu et al., 2019).

In the latter case, the UAVs continuously monitor any particular location or sensing apparatus in some area and send data such as images or videos to the base station via wireless communication (Scherer & Rinner, 2020). In specific scenarios, to allow a human mission to be rapidly completed, a sensing location must be visited by a UAV as quickly as possible to make the data collected by UAVs from the sensing location arrive at the base station on a timely basis. Then, the operator or data manager obtains accurate data about the situation at the sensing location for further processing and can make a decision based on that information. Note that the area that UAVs will visit is potentially enormous, and the existing wireless communication used by UAVs has limitations in terms of range when connecting an area of interest (sensing location) to the base station. Furthermore, each UAV typically has a limited fuel or power capacity. All of these considerations make optimal UAV path planning very challenging. In response to this challenge, in this research, the use of several UAVs is considered for the mission. According to Amigoni et al. (2017), compared to a single UAV, multi-UAVs can provide significant advantages in terms of carrying out missions, such as considerable improvements in task execution efficiency, endurance, and robustness.

Some UAVs will repeatedly visit the sensing locations and send data to the base station. This scenario leads to a persistent surveillance or patrolling problem. One challenge that may occur is dealing with the time that each UAV requires to visit a sensing location and transport the data to the base station. Another challenge is finding an optimal route for each UAV in the monitoring area. There are many studies on multi-UAV surveillance, with various objectives and constraints considered. Banfi et al. (2015) attempted to minimize data transmission delay by providing a mixed integer programming (MILP) formulation and a heuristic algorithm. To ensure information propagation in area monitoring, Acevedo et al. (2013) offered a decentralized strategy to minimize the information propagation time. Considering the connection to the base station, Scherer & Rinner (2017) proposed a permanent connection to the base station via a multi-hop link with energy and communication constraints. Scherer & Rinner (2020) proposed an intermittent connection to the base station where UAVs and the base stations are assumed to be equipped with the wireless capability required to exchange data. They aimed toward optimizing idleness and data delay by determining the direction of movement and meeting points. They also considered cooperative data transportation, which eliminates detours to the base station.

In this research, cooperative data transportation is considered that allows UAVs to transfer data to the base station in a store-and-forward fashion. Each UAV does not have to travel to the base station individually to exchange data. According to Scherer & Rinner (2020), this scenario will reduce idleness. Thus, this research is aimed toward maximizing the amount of data collected while considering the idleness and latency constraints. The latency constraints will force the UAVs to collect data from the sensing location and share it with the base station within a specific time range. The idleness constraints will force the UAVs to visit the sensing location consecutively within a specific time range.

To address the aim of this research, we attempt to determine an optimal route on a given graph that will be used as a workplace for the operational movement of the UAVs to visit and collect data from all sensing locations and transfer it to the base station (Scherer & Rinner, 2020a). Determining an optimal route for each UAV to perform a mission is related to the NP-complete traveling salesman problem (TSP). The problem of transporting data from a sensing location to the base station using several UAVs is related to the shortest path problem with time windows (SPPTW). We will use the basic idea of the TSP and SPPTW as an initial solution to solve the routeing problem.

In a previous work (Scherer & Rinner, 2019), researchers provided some algorithms to determine an optimal tour for visiting all sensing locations. In the present research, a model representing the above problem is created to illustrate how differences in the number of UAVs between each iteration and network structure can affect each UAV's working time and the routing sufficient to carry out the mission while also considering the idleness and latency constraints. The model is structured based on an integer linear programming model and solved using a state-of-the-art optimization solver (GUROBI).

1.2. Research Objective

This research is aimed toward developing a mathematical model for multiple UAVs. The objective is to find an efficient model to maximize the amount of data collected while considering the upper bound of idleness and latency constraints. These mathematical models are expected to have these capabilities:

1. The ability to determine the optimal route for each UAV.
2. The ability to determine how the data will be transferred from the sensing locations to the base station.
3. The ability to track the amount of data carried by each UAV.
4. The ability to track the time needed by the UAVs to carry the data from sensing locations to the base station.

1.3. Research Scope

This research focuses on a case of multiple UAV surveillance. This problem is based on Scherer and Rinner (2019), where multi-UAVs are used to perform cooperative data transportation. To achieve the research objective, it is necessary to determine the optimal route for each UAV to visit and transfer the data from the sensing location to the base station during a given time using a given graph used as the workplace for the operational movement of the UAVs, where the mechanism for the routing of each UAV is constructed using a time-space network.

Assuming that there are two arcs in the graph: a communication arc and a flight arc, there will be two ways to transfer the data. For the communication arc, it is assumed that the data can be transferred wirelessly as long as there are 2 UAVs within their communication range. To collect the data at the sensing location, the UAV has to pass through the sensing location. Still, the data is not necessarily collected each time the UAV passes through it. We assume that the UAVs and the base station are equipped with wireless so that the UAV does not need to pass through the base station to exchange data.

1.4. Thesis Outline

This research proposal is organized as follows: In Chapter 2, we provide a literature review of the previous studies related to this research. We also provide our research contribution. Chapter 3 describes the problem definition and formulation together with the mathematical model. Chapter 4 explains the uncooperative data transport model and compares it with the proposed model and the progress of the heuristic and algorithm developed for the purposes of this research. Chapter 5 shows the computational experiments intended to measure the performance of the proposed model. Chapter 6 offers the conclusion and suggestions for future research.

1.5. Research Contribution

There are several studies related to multi-UAV surveillance where an attempt is made to minimize idleness while considering latency constraints. This research maximizes the amount of data collected while considering the idleness in multi-UAV surveillance cases and the latency or time needed to transfer the data from the sensing location to the base station.

The previous research closest to this is Scherer & Rinner (2020b), who provided a heuristic algorithm to address cooperative data transport. Thus, we propose an exact method to solve the same problem and compare it with an uncooperative case and a heuristic algorithm to ensure that the research results are optimal.

Chapter 2
LITERATURE REVIEW

The literature review is divided into three main sections; the multi UAV surveillance problem, connection requirements, and the time-space network.

2.1. The Multi-UAV Surveillance Problem

The multi-robot patrolling problem is related to multi-UAV persistent surveillance applications. Scherer & Rinner (2019) state that this application can be divided into determining the closest paths traversed by UAVs continuously in some area or controlling and coordinating UAVs or robots' movement along a given path. While trying to determine the closest path, some researchers have also been trying to minimize idleness. Nigam et al. (2012) provide a strategy to control UAV movement considering the idleness of cells and the distance between the UAVs and cells. Specifically optimized algorithms have been studied, where infinite horizon persistence surveillance is converted into a short horizon problem. Pasqualetti et al. (2012) provided an algorithm to determine a tour for collaborative robots within areas that consist of sensing locations with different priorities in order to provide a control rule for determining each robot's coordination in the tour to minimize idleness.

Determining the closest path while minimizing idleness is a common optimization goal. Several types of research have been attempts to minimize latency while controlling the coordination among robots in persistent surveillance cases. Banfi et al. (2015) were concerned with the problem of finding a patrolling path for each robot. They attempted to minimize delays or latencies by formulating an MILP (mixed-integer linear programming) model and proposing a heuristic algorithm, where each UAV will fly along a path that consists of the sensing location and intermediate detours to send data to the base station. This concept is similar to that proposed in this research, but in this research, delays and latencies are considered as constraints. By implementing centralized path planning problem for persistent intelligence, surveillance, and reconnaissance (PISR) routing problem, Manyam et al. (2017) formulated an MILP (mixed-integer linear programming) model and provided a heuristic algorithm to minimize the maximum number of delivery times required for all tasks while also considering a revisit period constraint. Acevedo et al. (2013) offered a decentralized algorithm for partitioning a grid area into subareas, where each robot will travel in a circular path and consider the data propagation of information among the robots, which minimizes the propagation time for information in the grid area partition.

A summary of studies related to the multi-UAV surveillance problem is provided in Table *2.1*

Table 2.1. Multi-UAV Surveillance Problem

Literature	Horizon	Latency	Method	
			Math Model	**Heuristics**
Manyam et al. (2017)	Infinite	Objective	V	V
Acevedo et al. (2013)	Finite	Objective	V	V
Banfi et al. (2015)	Finite	Objective	V	V
Nigam et al. (2012)	Infinite	Objective		V
Scherer & Rinner (2019)	Infinite	Objective		V
Scherer & Rinner (2017)	Infinite & Finite			V
Scherer & Rinner (2016)	Infinite		V	V
This Research	Finite	Constraints	V	

2.2. Connection Requirements

Connection requirements among UAVs for storing data is one area that may be considered during a mission. Several studies have been conducted considering this issue. Kuznetsov et al. (2016) proposed a two-agent robotic system in which the first agent receiving the packet will transmit it to another one as long as the first agent did not buffer and lose the packet. They try to minimize the input packet loss. Spirin & Cameron (2014) proposed limitations in communication between agents, which means that robots have to physically move to pass information. Banfi et al. (2016) offered multi-robot exploration strategies under recurrent connectivity by considering a centralized asynchronous planning framework.

Intermittent or recurrent connectivity allows each robot to be disconnected and meet at specific intervals to coordinate or exchange collected data as a connection requirement in some applications. This kind of connectivity requires determining where and when each robot should meet. Several studies on this approach used a multi-robot application, including multi-robot exploration by Banfi et al., 2018 and Wu et al. (2020) and multi-UAV persistent surveillance by Acevedo et al., (2014) and Manyam et al. (2017).

Another application was developed by Spirin et al. (2014) that considered the rate at which information was updated at a base station, where robots must regain connection with some teammates according to a policy triggered by specific events, such as discovering new information about the environment, or triggered at a specified time. Scherer & Rinner (2020b) proposes recurrent connectivity among robots by considering cooperative data transport by multi-UAVs to a single base station, which requires determining when and where each UAV should meet. This concept is similar to the present research. Kantaros et al. (2019) considered situations in which an intermittent connection is needed. Robot communication capabilities are insufficient to form reliable connected networks when the robots must move to carry out their tasks. In Kantaros' study, it was assumed that robots could communicate with each other when they meet at common locations in space.

In some scenarios, each robot must always be connected to any other teammate or base station, either directly or in a multi-hop fashion. This scenario is needed in a certain situation where real-time image streaming is important (e.g., in search and rescue) or to ensure a high degree of coordination. New plans can be computed assuming globally-shared knowledge between all robots. Scherer & Rinner (2016) proposed an offline path planning algorithm in which each UAV is always connected with the base station via a single or multi-hop link. Scherer & Rinner (2020a) considered multi-robot persistent surveillance with connectivity constraints, where robots must periodically visit sensing locations and maintain a multi-hop connection to a base station. Ponda et al. (2012) presented a cooperative distributed planning algorithm that ensures network connectivity for a team of heterogeneous agents operating in dynamic and communication-limited environments.

Grotli & Johansen (2012) defined a mixed-integer linear programming (MILP) optimization problem to solve a surveillance mission problem using multiple UAVs. To effectively transmit the data back to the base station, they considered ferrying and relaying in addition to direct transmission. Scherer & Rinner (2020b) considered robot moves on a tour, where each robot will exchange data when meeting with another robot at a neighboring time and eventually transfer the data to the base station. They present a heuristic algorithm based on the shortest path search in the tour graph to reach minimum idleness and data delay. They consider a situation where the robots did not need to connect with the base station, and neither robot had to transfer the data to the base station individually. Pasqualetti et al. (2012) considered the refresh time related to idleness and latency, where each agent will patrol some partition of the trajectory. They assumed that the agent did not connect and would be informed if any event occurred in the environment.

A summary of the research about connectivity requirements is shown in Table 2.2.

Table 2.2. Connectivity Requirement

Literature	**Horizon**	**Connectivity**	**Method**	
			Math Model	**Heuristics**
Kuznetsov et al. (2016)	finite	Store-and-forward		
Spirin & Cameron (2014)	finite	recurrent		V
Banfi et al. (2016)		recurrent	V	V
Scherer & Rinner. (2020a)	infinite	permanent		V
Scherer & Rinner (2020b)	infinite	Store-and-forward		V
Spirin et al. (2014)	Infinite	recurrent		V
Kantaros et al. (2019)	Infinite	Store-and-forward		V
Jurgen Scherer & Rinner (2016)	Infinite	Permanent		V
Ponda et al. (2012)	Infinite	Permanent		V
Grotli & Johansen. (2012)	Infinite	Store-and-forward	V	
Pasqualetti et al. (2012)	Infinite	Store-and-forward		V
This Research	Finite	Store-and-forward	V	

2.3. The Time-Space Network

In Tsai et al. (2016) considered a time-space network that has been applied and considered in logistics and has been widely used in modeling routing problems. They used a time-space network to model the practical many-to-many carpooling problem with multiple vehicles and personality types, as well as pre-matching information. According to Zhang & Kingston (2015), this has been widely adopted in the transportation literature, and the use of a time-space network has been shown to be beneficial for multi-vehicles problems.

Baldacci et al. (2011) proposed period vehicle routing problems (PVRPs) defined on a time horizon of several days that consisted of assigning appropriate combinations of delivery methods to customers. They designed a set of delivery routes for every day of the planning period. The objective was to service all customers assigned to each day with a minimum overall routing cost. The time-space network showed an individual's position in time and space. Their design is similar to the approach used in the present research, where a sensing location is treated as a customer who can be visited at any time in the planning horizon.

2.4. Summary

To find out how this study will contribute scientifically to the topic or issue of the Multi-UAV Surveillance Routing Problem, we try to find some insight from other studies related to the Multi-UAV Routing Problem. We found that there are two main focuses related to this topic:

1. Consideration about latency and idleness when dealing with Multi-UAV Problem.

   The latency is related to the make span of the vehicle's reach the target and its arrival at the depot. The idleness is related to the time between 2 consecutive visits to the target. Almost all the previous researchers consider latency and idleness as the objective of their studies. Almost all of the previous research also tried to minimize the latency while considering idleness as the constraint. In this study, we try to consider latency and idleness as the constraints and maximize the target we can visit.

2. Consideration of the connection requirements.

   The UAVs will communicate with each other during the surveillance mission (cooperative data transport). According to what we found, there are sthree types of connection can be considered. The first one is permanent connectivity, where each UAV and the depot will be fully connected during the mission. The second one is recurrent connectivity, where all UAVs or robots meet after a certain interval. The last one is store-and-forward or intermittent, where UAVs can connect in their communication range. In this study, we consider that each UAV will communicate during the mission via wireless in is a store-and-forward or intermittent way.

To track the movement of the UAVs during the mission, we consider using the Time-Space Network technique, where this technique had been performed by other studies before. We try to provide both exact and algorithm to solve this problem.

Chapter 3
MATHEMATICAL PROGRAMMING MODEL

Before defining the mathematical model, we explain our approach, provide a problem description, and discuss the characteristics to of the model.

3.1. Model Approach

As mentioned in Chapter 1, either a graph or the area used as the workplace for the operational movement of each UAV is given. The primary objective was to maximize the amount of data collected while minimizing idleness (the time between two consecutive visits at sensing locations) by determining a route for each UAV in the workplace during a given amount of time. To define a route for the UAV, it was necessary to divide the workplace area into several waypoints. The mechanism for the routing of each vehicle was constructed using a time-space network.

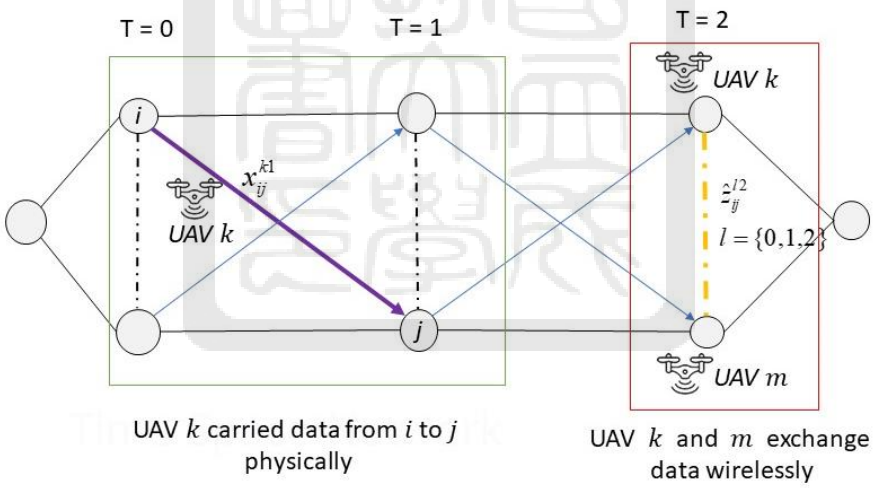


Figure 3.1. The movement of UAVs and the Data

The flow of the time-space network used to formulate UAVs' potential movements within a specific time and space is shown in Figure 3.1, where the horizontal axis represents the time duration and data collected, and the vertical axis represents each UAV's location. The nodes and arcs are two significant components in this network. A node is designed for a specific time, excluding the departure, ending (associated with the UAVs), and data collection points (associated with the data). These nodes are the dummy nodes representing the starting and ending points of UAVs and the collection data points. There are two arcs, a solid line and a broken line, describing the flight and communication arc, respectively. An arc represents an activity. A flight arc represents when UAVs carry the data from one node to another node. A communication arc represents when UAVs exchange data with each other or with the base station. Figure 3.1 illustrates the movement of data.

There are six types of arcs in this network defined as follows:

1. **Flight arc:** indicated by (1) in Figure 3.2 and representing travel from one node to another node whether the UAV carries data or not.
2. **Communication arc:** indicated by (2) in Figure 3.2 and representing data carried from one UAV to another UAV.
3. **Waiting arc:** indicated by (3) in Figure 3.2 indicating that a UAV is at the same node or waiting for a UAV at some node during the time window.
4. **Departure arc:** indicated by (4) in Figure 3.2 and representing a starting trip of each UAV from the dummy starting node to any node in the workplace.
5. **UAV return arc:** indicated by (5) in Figure 3.2 represents an ending trip of each UAV from any workplace node at the end of the planning horizon.
6. **Data collected arc:** indicated by (6) in Figure 3.2 and representing that data is transferred to the base station by UAVs at any time in the planning horizon.

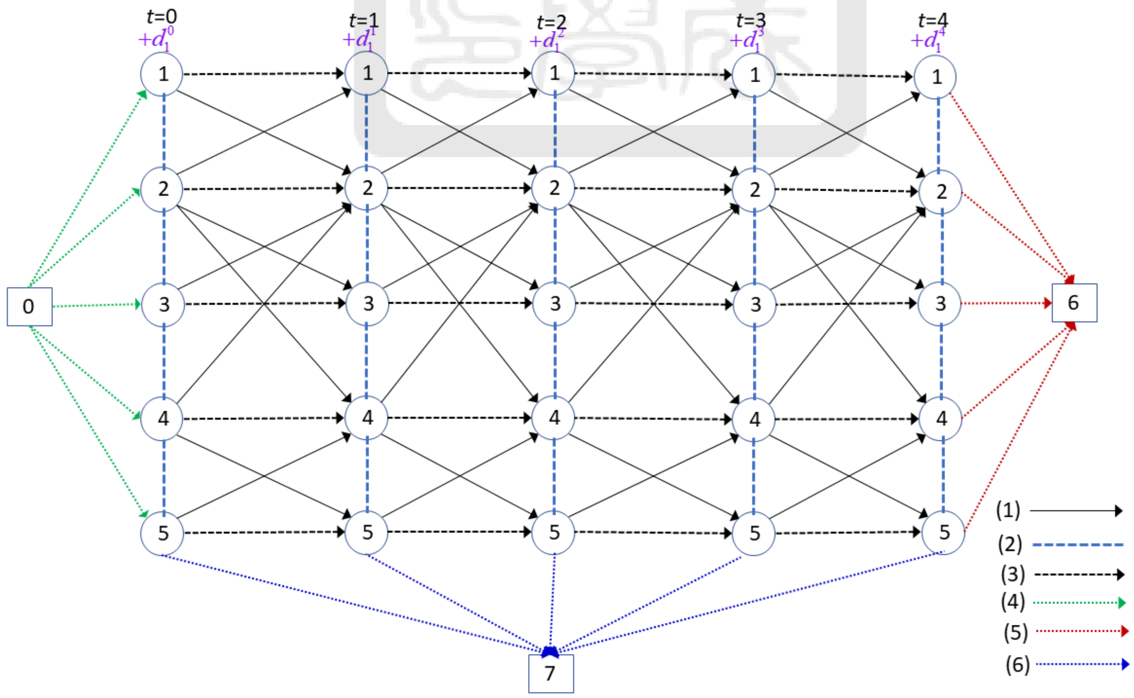


Figure 3.2. UAV Flow in the Time-space Network

Let $G=(V,A,T)$ be a complete undirected graph, where V is the set of all nodes in the workplace, including the sensing location (V^s) and the depot (V^d) nodes. A is the set of arcs in the workplace, including the flight arc (A^f), waiting arc (A^w), and communication arc A^c. T is a unit of time including flight time T^f related to the time required for the UAV to fly from one node to another node and the communication time T^c related to the time required for data transmission for every arc. Each UAV can move from one node i to another node j within time T^f as long as an arc $(i,j)\in A^f$, and two UAVs (or UAVs with the base station) can exchange data and as long as one UAV is at node i and another UAV is at node j (or base station) at the same time and $(i,j)\in A^c$.

The problem is given by a set of sensing locations at different workplace nodes that contain some data to be captured by UAVs at any time on the planning horizon. d_i^l represents the amount of data captured by UAVs at sensing locations i from a data source l associated with time *t*. Each UAV will fly along the arc during the mission, and in order to send the data captured at a sensing location to the base station in a store-and-forward fashion, every time 2 UAVs arrive at node i and j at the same time, and $(i,j)\in A^c$, they will exchange data. $\hat{z}_{ij}^{lt}$ represents the amount of data generated or captured at the data source l and transferred by one UAV at node i to another UAV at node j at the same time t.

To represent the route assignment, we introduce a binary variable; first, we identify the presence of UAV k, where v_i^{kt} represents that there is UAV k passing through node i at time t. If the UAVs are beginning to move, there will be a variable x_{ij}^{kt} that will equal 1 if there is any UAV k beginning to fly at time t along arc $(i,j)\in A^f$. To find and track the data carried by a UAV when it is flying along an arc $(i,j)\in A^f$, we present variable $\overline{y}_{ij}^{klt}$, which will equal 1 if UAVs are carrying the data generated or captured from a data source l along an arc $(i,j)\in A^f$ time t. The amount of data carried by the UAVs along an arc $(i,j)\in A^f$ is represented by $\overline{z}_{ij}^{klt}$. This research is aimed toward maximizing the amount of data transferred by the UAVs to the base station at any time in the planning horizon. To store the event when any UAV transfers data to the base station, we present a variable h_i^{lt} that will equal 1 if any amount of data is transferred by some UAV to the base station. Otherwise, it will equal 0.

3.2. Problem Assumption

The assumptions introduced to develop the model are given as follows:

1. The flight time of a vehicle along a solid arc takes 1 unit of time. However, wireless communication can transfer data along a dashed arc in 0 units of time if the other end has a vehicle or facility (depot) to receive it.
2. Each UAV has unlimited storage capacity to collect the data, so each UAV can collect and receive as much data as possible.
3. The sensing location and depot point in the workplace are known beforehand (and remain fixed).
4. A UAV has to pass through the sensing location to collect the data, but data is not necessarily collected each time the UAV passes through it.
5. The UAVs and the base station are equipped with wireless communication so that the UAV does not need to pass through the base station to exchange data.
6. Each UAV has to start from the depot at the beginning of the mission and return to the depot by time *T*.
7. The data captured by each UAV at a specific time during the mission is not the accumulated data from the previous time but real-time data from the sensing location at that time.

3.3. Mathematics Model

3.3.1. Notation

The notations that are being used for the mathematic model are listed in Table 3.1

Table 3.1. Notations for Mathematical Formulation in Cooperative Data Transport Case

Parameters	
d_i^l	amount of data generated on a sensor node i at time (epoch) $l = 0,1,\ldots,T$.
T	$= \{0,1,\ldots,T\}$: Set of working time t
K	**N**umber of UAVs use

LT	The latency upper bound
ID	The idleness upper bound
Sets	
V	$=\{i : i = 1, 2, ..., n\}$: Set of nodes that a UAV may pass through
V^s	$=\{i \in V : i$ is a sensor node$\}$: Set of sensor nodes
V^d	$=\{i \in V : i$ is a depot node$\}$: Set of depot nodes
A^f	$=\{(i, j):$ a flight solid arc in G from $i \in V$ to $j \in V$ by a UAV
A^w	$=\{(i, i):$ a waiting solid arc in G at $i \in V$ by a UAV
A^c	$=\{(i, j):$ a communication dashed arc in G from $i \in V$ to $j \in V$ by wireless connection
$\bar{V}$	$=\{0, n+1, n+2\}$: Set of dummy nodes, where 0 & n+1 are the vehicle source & sink, and n+2 is the data sink
A_0^{fo}	$=\{(0, i) : i \in V\}$: Set of dummy flight source arcs outgoing from 0
A_{n+1}^{fi}	$=\{(i, n+1) : i \in V\}$: Set of dummy flight sink arcs entering $n+1$
A_{n+2}^{ci}	$=\{(i_t, n+2) : i_t \in V^d$ is the depot node at time $t = 0, 1, ..., T\}$: Set of dummy communication sink arcs entering $n+2$
A_i^{fo}	$=\{(i, j) \in A^v$, with travel time $\bar{t}_{ij}\}$: Set of all flight arcs outgoing from $i \in V$
A_j^{fi}	$=\{(i, j) \in A^v$, with travel time $\bar{t}_{ji}\}$: Set of all flight arcs entering to $j \in V$
A_i^{fw}	$=\{(i, i)$ for $i \in V$, with 1 unit waiting time$\}$: Set of the flight waiting arc (i, i)
A_i^{co}	$=\{(i, j) \in A^c$, with 0 transfer time$\}$: Set of all communication arcs outgoing from $i \in V$
A_j^{ci}	$=\{(i, j) \in A^c$, with 0 transfer time$\}$: Set of all communication arcs entering to $j \in V$
Decision Variables	
x_{ij}^{kt}	whether (1:Y; 0:N) the vehicle k starts from $i \in V$ along a flight arc $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0, 1, 2, ..., T$
x_{0j}^{k0}	whether (1:Y; 0:N) the vehicle k starts from dummy source 0 to $j \in V$ along a dummy arc $(0, j) \in A_0^{fo}$
$x_{i,n+1}^{k,\lvert T\rvert}$	whether (1:Y; 0:N) the vehicle k starts from $i \in V$ to dummy sink $n+1$ along a dummy arc $(i, n+1) \in A_{n+1}^{fi}$

v_i^{kt}	whether (1:Y; 0:N) the vehicle k passes through a node $i \in V$ at time $t = 0,1,2,...,T$
$\bar{y}_{ij}^{klt}$	whether (1:Y; 0:N) the data generated at time $l \le t$ starts to be carried by vehicle k along $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0,...,T-1$
$\bar{z}_{ij}^{klt}$	amount of data generated at time $l \le t$ starts to be carried by vehicle k along an arc $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0,...,T-1$
$\hat{z}_{ij}^{lt}$	amount of data generated at time $l \le t$ flows along $(i, j) \in A_i^{co} \cup A_{n+2}^{ci}$ at time $t = 0,...,T$
h_i^{lt}	whether (1:Y; 0:N) the data generated at time $l \le t$ flows along $(i, j) \in A_{n+2}^{ci}$ at time $t = 0,...,T$

3.3.2. Formulation

There will be a given set of sensing locations visited by some UAV at most once to capture the data. In this case, each UAV may not be required to visit the sensing location and transfer data to the base station. The objective function maximizes the amount of data transferred by UAVs to the base station at any time on the planning horizon, as shown in Equation (3.1).

$$Maximize \sum_{l=0}^{T} \sum_{t=l}^{T} \sum_{i \in V^d} \hat{z}_{i,n+2}^{lt} \tag{3.1}$$

Vehicle Constraints

The constraint shown in Equation (3.2) ensures that each UAV is flying from the dummy source node to exactly one node j in the workplace.

$$\sum_{(0,j) \in A_0^{fo}} x_{0j}^{k0} = 1 \qquad \forall k \in K,\ \forall i \in V,\ t = 0,1,2...,T \tag{3.2}$$

The constraint shown in Equation (3.3) ensures that each UAV will go back to sink at the end of the planning horizon from exactly one node i in the workplace.

$$\sum_{(i,n+1) \in A_{n+1}^{fi}} x_{i,n+1}^{k,T+1} = 1 \qquad \forall k \in K,\ \forall i \in V,\ t = 0,1,2,...,T,T+1 \tag{3.3}$$

The constraint shown in equations (3.4) – (3.5) defines the flow balance of each UAV from a dummy source node to any node j in the workplace and the flow balance of each UAV from any node i in the workplace to the dummy sink, respectively.

$$\sum_{(i,j)\in A_i^{fo}\cup A_i^{wo}} x_{ij}^{k0} = x_{0i}^{k,0} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,...,T \tag{3.4}$$

$$x_{i,n+1}^{k,T+1} = \sum_{(j,i)\in A_i^{fi}\cup A_i^{wi}} x_{ji}^{k,T-\bar{t}_{ji}} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,....,T,T+1 \tag{3.5}$$

To ensure that any UAV flies along an arc $(0,i)\in A^f$ from a dummy source to any node i in the workplace, a UAV must pass through node i simultaneously. This constraint is shown in Equation (3.6).

$$v_i^{k0} = x_{0i}^{k,0} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,....,T \tag{3.6}$$

If there is a UAV flying along the arc $(i,n+1)\in A^f$ (i.e., from a node i in the workplace to the dummy sink), this means that some UAV passed through node i at the end of the planning horizon, as shown in Equation (3.7).

$$x_{i,n+1}^{k,T+1} = v_i^{k,T} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,...,T,T+1 \tag{3.7}$$

Equation (3.8) shows the flow balance constraints of each UAV in the workplace.

$$\sum_{(i,j)\in A_i^{fo}\cup A_i^{wo}} x_{ij}^{kt} = \sum_{(j,i)\in A_i^{fi}\cup A_i^{wi}} x_{ji}^{k,t-\bar{t}_{ji}} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,...,T \tag{3.8}$$

To ensure whether there is a UAV flying along an arc $(i,j)\in A^f$ from node i to node j in the workplace, there must be a UAV that passes through node i at the same time. This constraint is shown in Equation (3.9).

$$\sum_{(i,j)\in A_i^{fo}\cup A_i^{wo}} x_{ij}^{kt} = v_i^{kt} \qquad \forall k\in K,\ \forall i\in V,\ t=0,1,2,...,T \tag{3.9}$$

To ensure that there is a vehicle *k* passing through node i and *j* at the beginning of period t to the end of period t respectively so that the vehicle can fly along an arc $(i,j)\in A^f$ at period t, this constraint is shown in Equation (3.10).

$$x_{ij}^{kt} \le v_i^{kt} + v_j^{k,t+\bar{t}_{ij}} - 1 \qquad \forall k\in K,\ \forall (i,j)\in A_i^{fo}\cup A_i^{wo},\ t=0,1,2,...,T \tag{3.10}$$

The constraints in equations (3.11) – (3.12) ensure that each sensing location must be visited at least twice during a mission carried out by the UAVs.

$$\sum_{k\in K}\sum_{(i,j)\in A^{fi}\cup A^{fw}}\sum_{t=0,1,2,\ldots,T} x_{ij}^{kt} >= 2 \qquad \forall i \in V^s \tag{3.11}$$

$$\sum_{k\in K}\sum_{t=0,1,2,\ldots,T} v_i^{kt} >= 2 \qquad \forall i \in V^s \tag{3.12}$$

Data Constraints

The constraint shown in Equation (3.13) is the flow balance constraint for data flow in the workplace.

$$\sum_{(i,j)\in A_i^{co}} \hat{z}_{ij}^{lt} + \sum_{(i,j)\in A_i^{fo}\cup A_i^{fw}} \overline{z}_{ij}^{klt} - \sum_{(j,i)\in A_i^{ci}} \hat{z}_{ij}^{lt} - \sum_{k\in K}\sum_{(j,i)\in A_i^{fi}\cup A_i^{fw}} \overline{z}_{ji}^{kl,t-\bar{t}_{ji}} \le d_i^l \tag{3.13}$$
$$\forall l = 0,1,2,\ldots T,\ \forall t \ge l,\ \forall i \in V \setminus V^d$$

The constraint shown in Equation (3.14) is the flow balance constraint for data flow from depot to the data sink.

$$\sum_{(i,j)\in A_{n+2}^{ci}} \hat{z}_{ij}^{lt} + \sum_{k\in K}\sum_{(i,j)\in A_i^{fi}\cup A_i^{fw}} \overline{z}_{ij}^{klt} - \sum_{(j,i)\in A_i^{ci}} \hat{z}_{ij}^{lt} - \sum_{k\in K}\sum_{(j,i)\in A_i^{fi}\cup A_i^{fw}} \overline{z}_{ji}^{kl,t-\bar{t}_{ji}} \tag{3.14}$$
$$\le d_i^l \qquad \forall l = 0,1,2,\ldots T,\ \forall t \ge l,\ \forall i \in V^d$$

In order to transfer the data to the data sink, there must be a UAV passing through the depot at any time in the planning horizon. The constraint shown in Equation (3.15) ensures this.

$$\sum_{l\le t} \hat{z}_{in+2}^{lt} \le M \sum_{k\in K} v_i^{kt} \qquad \forall (i,j) \in A_{n+2}^{ci},\ \forall i \in V^d,\ \forall t = 0,1,2,\ldots,T \tag{3.15}$$

The constraints in Equation (3.16) – (3.18) trigger $\overline{y}_{ij}^{klt}$ = 1 when data is generated at time $l \le t$ along flight arc $(i,j) \in A^f$ by some k; otherwise $\overline{y}_{ij}^{klt}$ = 0.

$$\overline{z}_{ij}^{klt} \ge \epsilon \overline{y}_{ij}^{klt} \qquad \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V,\ \forall l \le t, \forall t = 0,1,2,\ldots T \tag{3.16}$$

$$\overline{z}_{ij}^{klt} \le M \overline{y}_{ij}^{klt} \qquad \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V,\ \forall l \le t, \forall t = 0,1,2,\ldots,T \tag{3.17}$$

$$\overline{y}_{ij}^{klt} \le M x_{ij}^{kt} \qquad \forall t = 0,1,2,\ldots,T,\ \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V \tag{3.18}$$

The constraints in equations (3.19) – (3.20) trigger $\hat{z}_{ij}^{lt} > 0$ and ensure that there is a vehicle at node i and j at time t if data is transferred via communication arc $(i, j) \in A^c$; otherwise $\hat{z}_{ij}^{lt} = 0$.

$$\sum_{l \le t} \hat{z}_{ij}^{lt} \le M \sum_{k \in K} v_i^{kt} \qquad \forall (i, j) \in A_i^{co},\ \forall i \in V,\ \forall t = 0,1,2,...,T \tag{3.19}$$

$$\sum_{l \le t} \hat{z}_{ij}^{lt} \le M \sum_{k \in K} v_j^{kt} \qquad \forall (i, j) \in A_i^{co},\ \forall i \in V,\ \forall t = 0,1,2,...,T \tag{3.20}$$

Latency Constraints

Equations (3.21) – (3.23) are latency constraints. Equations (3.21) – (3.22) trigger = 1 when there is an amount of data transferred to a data sink; otherwise $h_i^{lt} = 0$. Equation (3.23) ensures that the time length between capturing (from the sensing location) and transferring data (to the depot) is no longer than some specific latency bound.

$$\hat{z}_{i,n+2}^{lt} \ge \epsilon h_i^{lt} \qquad \forall l \le t, \forall t = 0,1,2,...,T,\ \forall (i, j) \in A_{i,n+2}^{ci} \tag{3.21}$$

$$\hat{z}_{i,n+2}^{lt} \ge \mathrm{M} h_i^{lt} \qquad \forall l \le t, \forall t = 0,1,2,...,T,\ \forall (i, j) \in A_{i,n+2}^{ci} \tag{3.22}$$

$$LT \ge (t - l) h_i^{lt} \qquad \forall l \le t, \forall t = 0,1,2,...,T,\ \forall i \in V \tag{3.23}$$

Idleness Constraint

The constraint in Equation (3.24) ensures that some number of consecutive visits by UAVs to the sensing location is not greater than the idleness bound.

$$\max_{t'=0,,,.t} \sum_{i \in V^s} \sum_{k \in K} t' v_i^{kt'} - \max_{t'=0,,,.t-1} \sum_{i \in V^s} \sum_{k \in K} t' v_i^{kt'} \le ID \quad \forall t = 0,1,2,...,T \tag{3.24}$$

The constraints in equations (3.25) - (3.29) define the variable constraints.

$$v_i^{kt} \in \{0,1\} \qquad \forall k \in K,\ \forall i \in V,\ \forall t = 0,1,2,...,T \tag{3.25}$$

$$x_{ij}^{kt} \in \{0,1\} \qquad \forall k \in K,\ \forall (i, j) \in A_i^{fo} \cup A_i^{fi} \cup A_i^{fw},\ \forall t = 0,1,2,...,T \tag{3.26}$$

$$\bar{y}_{ij}^{klt} \in \{0,1\} \qquad \forall k \in K,\ \forall (i,j) \in A_i^{fo} \cup A_i^{fi} \cup A_i^{fw},\ \forall t = 0,1,2,\ldots,T \tag{3.27}$$

$$\bar{z}_{ij}^{klt} \in \mathbb{R}^{+} \qquad \forall (i,j) \in A_i^{fo} \cup A_i^{fi} \cup A_i^{fw},\ \forall t = 0,1,2,\ldots,T \tag{3.28}$$

$$\hat{z}_{ij}^{lt} \in \mathbb{R}^{+} \qquad \forall (i,j) \in A_i^{co} \cup A_i^{ci},\ \forall t = 0,1,2,\ldots,T \tag{3.29}$$

3.4. Summary

We provide the exact solution to the multi-UAV surveillance routing problem where each UAV will communicate during the mission (cooperative data transport) by conducting the Integer Programming (IP) model. There are some approaches and assumptions that we elaborate on in this chapter before conducting the IP model. We conduct the IP model based on the Time-Space Network technique to track the movement of UAVs in space and time.

We determine there are 5 parameters for this model. Our objective function is to maximize the amount of data collected in the depot or the base station. The constraints that we conduct are divided into 3 types: (1) vehicle constraints which are related to the movement of UAVs; (2) data flow constraints which are the constraints that are related to the movement of data carried or transferred by UAVs; (3) Latency and Idleness constraints.

Chapter 4
DESIGN OF UNCOOPERATIVE DATA TRANSPORT MODEL AND HEURISTIC ALGORITHM

The proposed model is aimed toward collecting the maximum amount of data during a predetermined time *T* while also considering the idleness and latency constraints. The model uses a wireless connection to allow data to be exchanged between UAVs. We compared the proposed model and the uncooperative data transport model to determine the applicability of the proposed model and to compare the model performance when each UAV has to transfer the data individually without considering any wireless communication between UAVs.

4.1. Uncooperative Data Transport Model

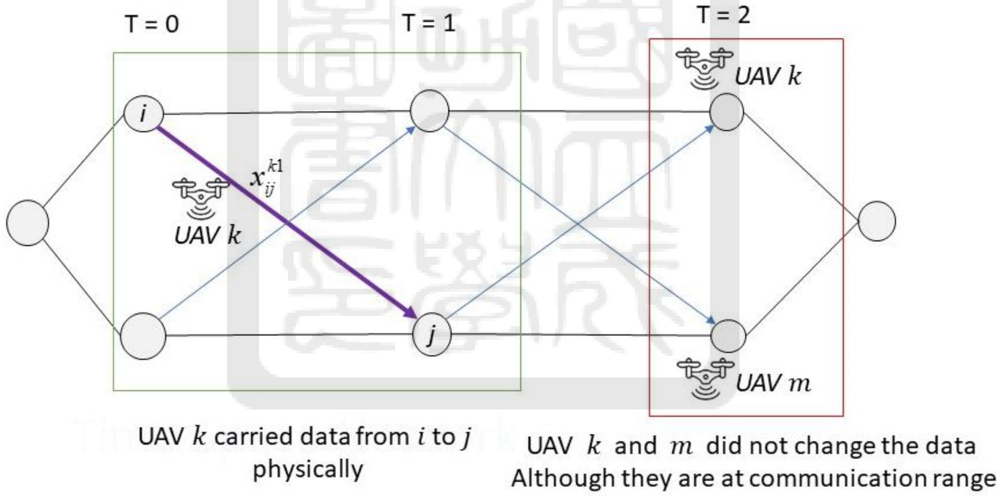


Figure 4.1. Time-Space Network for Uncooperative Data Transport

When comparing these two models, it was found that the cooperative case was always much better than the uncooperative case in terms of the amount of data collected. However, this came with the price of taking a longer processing time. Therefore, developing a heuristic and algorithm to shorten the computational time is necessary, especially to deal with large instances.

This chapter explains the uncooperative data transport model and the heuristics and algorithms developed to solve the multi-UAV surveillance routing problem. Section 4.1. describes the design of the uncooperative data transport model, and Section 4.2. discusses the developed heuristics and algorithms.

We describe uncooperative data transport as a case where each UAV used for a surveillance mission does not use wireless communication to exchange data. Each UAV collects data at the sensing location and transfer it to the base station individually. A comparison is made with the uncooperative data transport case to determine the effects of wireless communication between UAVs when carrying out surveillance missions on our proposed model. To make the comparison fair, we used a time-space network approach to model this case.

Figure 4.1 shows the time-space network for the uncooperative data transport case. As shown in Figure 4.1, the only arc considered in this case was the solid arc. This solid arc means that each UAV must transfer and carry the data from the sensing location to the base station individually. They cannot exchange the data with one another even though both UAVs are within their communication range.

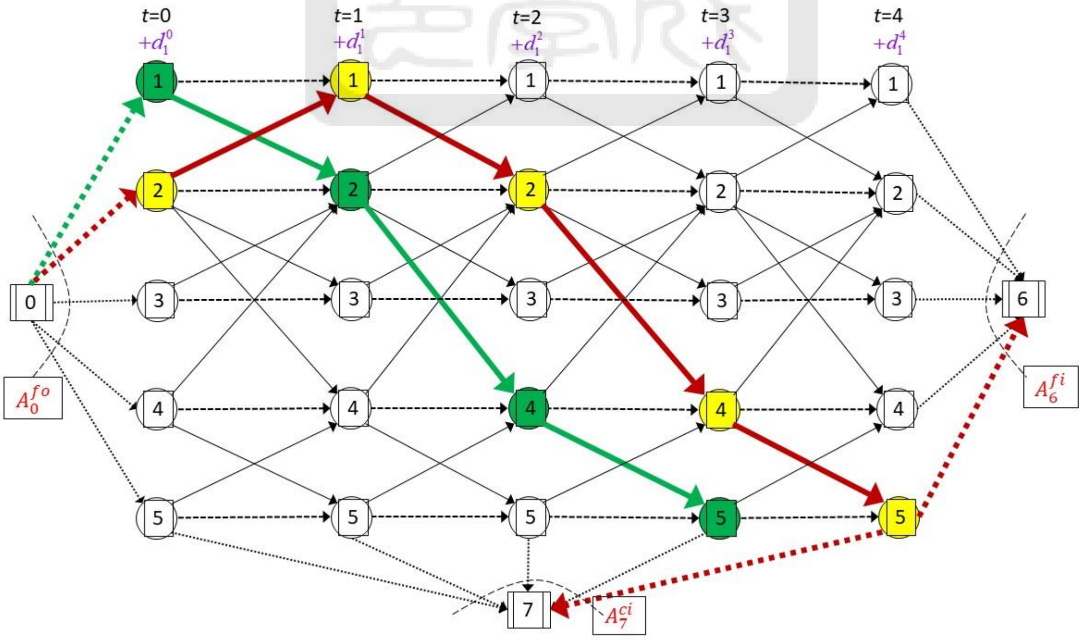


Figure 4.2. The Movement of Data in the Time-Space Network for the Uncooperative Case

Figure 4.2 shows an illustration of the UAVs and data movement in uncooperative data transport cases. As shown in Figure 4.2, there are 2 UAVs used for the mission, and we treat node 1 and node 2 as the sensing locations and node 5 as the depot or base station. Each UAV visits the sensing locations to collect the data. The planning horizon carries and transfers the data to the base station individually, without any communication between the UAVs.

The assumption, objective function, variable, vehicle, latency, and idleness constraints are the same as those of the cooperative data transport model. However, there are some differences in the data constraints.

4.1.1. Notation

All the notation used for the uncooperative data case are the same those used in the cooperative data case. The only difference is that this case did not consider the notations related to wireless communication, such as A_i^{co} A_j^{ci} and $\hat{z}_{ij}^{lt}$ in the workplace.

Table.4.1. Notations for Mathematical Model in Uncooperative Cases

Parameters	
d_i^l	amounts of data is generated on a sensor node i at time (epoch) $l = 0,1,\ldots,T$.
T	$= \{0,1,...,T\}$: Set of working time t
K	**N**umber of UAVs use
Sets	
V	$= \{i : i = 1, 2, ..., n\}$: Set of nodes that a UAV may pass
V^s	$= \{i \in V : i$ is a sensor node$\}$: Set of sensor nodes
V^d	$= \{i \in V : i$ is a depot node$\}$: Set of depot nodes
A^f	$= \{(i, j):$ a flight solid arc in G from $i \in V$ to $j \in V$ by a UAV
A^w	$= \{(i,i):$ a waiting solid arc in G at $i \in V$ by a UAV
$\bar{V}$	$= \{0, n+1, n+2\}$: Set of dummy nodes, where 0 & n+1 are the vehicle source & sink, and n+2 is the data sink
A_0^{fo}	$= \{(0,i) : i \in V\}$: Set of dummy flight source arcs outgoing from 0
A_{n+1}^{fi}	$= \{(i, n+1) : i \in V\}$: Set of dummy flight sink arcs entering $n+1$

A_{n+2}^{ci}	$=\{(i_t, n+2): i_t \in V^d$ is the depot node at time $t = 0,1,...,T\}$: Set of dummy communication sink arcs entering $n+2$
A_i^{fo}	$=\{(i, j) \in A^v$, with travel time $\overline{t}_{ij}\}$: Set of all flight arcs outgoing from $i \in V$
A_j^{fi}	$=\{(i, j) \in A^v$, with travel time $\overline{t}_{ji}\}$: Set of all flight arcs entering to $j \in V$
A_i^{fw}	$=\{(i,i)$ for $i \in V$, with 1 unit waiting time$\}$: Set of the flight waiting arc (i,i)
Decision Variables	
x_{ij}^{kt}	whether (1:Y; 0:N) the vehicle k start from $i \in V$ along a flight arc $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0,1,2,...,T$
x_{0j}^{k0}	whether (1:Y; 0:N) the vehicle k start from dummy source 0 to $j \in V$ along a dummy arc $(0, j) \in A_0^{fo}$
$x_{i,n+1}^{k,\|T\|}$	whether (1:Y; 0:N) the vehicle k start from $i \in V$ to dummy sink $n+1$ along a dummy arc $(i, n+1) \in A_{n+1}^{fi}$
v_i^{kt}	whether (1:Y; 0:N) the vehicle k passes through a node $i \in V$ at time $t = 0,1,2,...,T$
$\overline{y}_{ij}^{klt}$	whether (1:Y; 0:N) the data generated at time $l \le t$ starts to be carried by vehicle k along $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0,...,T-1$
$\overline{z}_{ij}^{klt}$	amount of data generated at time $l \le t$ starts to be carried by vehicle k along an arc $(i, j) \in A_i^{fo} \cup A_i^{fw}$ at time $t = 0,...,T-1$
$\hat{z}_{in+2}^{lt}$	amount of data collected at time $l \le t$ to data sink $(i, j) \in A_{n+2}^{ci}$ at time $t = 0,...,T$
h_i^{lt}	whether (1:Y; 0:N) the data generated at time $l \le t$ flows along $(i, j) \in A_{n+2}^{ci}$ at time $t = 0,...,T$

4.1.2. Mathematical Model

The objective function is the same as the cooperative case: to maximize the amount of data transferred by UAVs to the base station at any time on the planning horizon, and shown before in Equation (3.1).

All the constraints related to vehicle movement in the uncooperative case are the same as those for the cooperative case and are shown in equations (3.2) – (3.12). However, there are some differences in the data movement constraints. Equations (4.1) – (4.6) show the data movement constraints for the uncooperative cases.

Data Constraints for the Uncooperative Data Transport Cases

The constraint in Equation (4.1) is the flow balance for data in the workplace, in which the only data flow allowed is that carried by UAVs.

$$\sum_{k\in K}\sum_{(i,j)\in A_i^{fo}\cup A_i^{fw}} \bar{z}_{ij}^{klt} - \sum_{k\in K}\sum_{(j,i)\in A_i^{fi}\cup A_i^{fw}} \bar{z}_{ji}^{kl,t-\bar{t}_{ji}} \le d_i^l \qquad (4.1)$$
$$\forall l = 0,...T, \forall t \ge l, \forall i \in V \setminus V^d$$

The constraint in Equation (4.2) is the flow balance for data from the workplace to the data sink. Here, we consider the use of $\hat{z}_{ij}^{lt}$, which represents wireless communication. We assume that it will be transferred to the data sink wirelessly whenever data reach the depot.

$$\sum_{(i,j)\in A_{n+2}^{ci}} \hat{z}_{ij}^{lt} + \sum_{k\in K}\sum_{(i,j)\in A_i^{fi}\cup A_i^{fw}} \bar{z}_{ij}^{klt} - \sum_{k\in K}\sum_{(j,i)\in A_i^{fi}\cup A_i^{fw}} \bar{z}_{ji}^{kl,t-\bar{t}_{ji}} \le d_i^l \qquad (4.2)$$
$$\forall l = 0,1,2,...T,\ \forall t \ge l,\ \forall i \in V^d$$

The constraint shown in Equation (4.3) ensures that a UAV passes through the depot to transfer the data to the data sink at any time on the planning horizon.

$$\sum_{l\le t} \hat{z}_{in+2}^{lt} \le M \sum_{k\in K} v_i^{kt} \qquad \forall (i,j) \in A_{n+2}^{ci},\ \forall i \in V^d,\ \forall t = 0,1,2,...,T \qquad (4.3)$$

The constraints in equations (4.4) – (4.6) trigger $\bar{y}_{ij}^{klt}$ = 1 when data is generated at time $l \le t$ along flight arc $(i, j) \in A^f$ by some k; otherwise $\bar{y}_{ij}^{klt}$ = 0. These constraints align the data with the UAV's movement and ensure that there is a UAV carrying the data for every movement of data in the workplace.

$$\bar{z}_{ij}^{klt} \ge \epsilon \bar{y}_{ij}^{klt} \qquad \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V,\ \forall l \le t, \forall t = 0,1,2,...T \qquad (4.4)$$

$$\bar{z}_{ij}^{klt} \le M \bar{y}_{ij}^{klt} \qquad \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V,\ \forall l \le t, \forall t = 0,1,2,...,T \qquad (4.5)$$

$$\bar{y}_{ij}^{klt} \le M x_{ij}^{kt} \qquad \forall t = 0,1,2,...,T,\ \forall (i,j) \in A_i^{fo} \cup A_i^{fw},\ \forall k, \forall i \in V \qquad (4.6)$$

4.2. The Heuristic for Multi-UAV Surveillance Routing Problem

Although the mathematical model can generate an optimal solution, it can require a long period of computational time depending on the graph size and variables. Our mathematical model also experiences this problem. In this section, we propose a heuristic approach by developing algorithms for the model. The algorithm is divided into three phases. The first phase is intended to find the shortest path from the base station to all sensing locations; the second phase is intended to find the k-subcircuits from the shortest path according to the number of UAVs used, followed by the synchronization algorithm, where we attempt to synchronize each UAV's flying schedule in each circuit. Figure 4.3 shows the flowchart of these three phases.

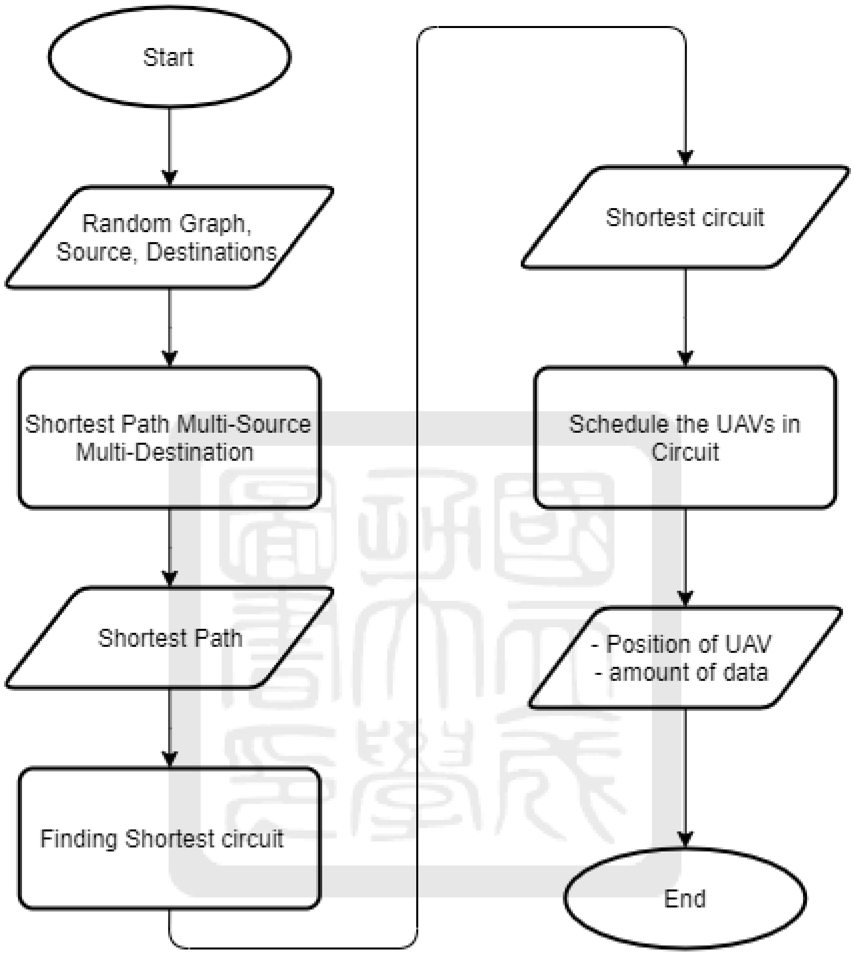


Figure 4.3. Heuristic Framework

1. Phase 1 – Find the shortest path from the base station to all sensing locations

This phase was aimed toward finding the shortest path from the base station to all sensing locations. We used the Dijkstra algorithm as a basis for finding the shortest path from the source to the destination. The shortest path obtained from this phase was later the input for the second phase.

The first step in this phase is finding the closest sensing location to the base station. The nearest sensing location from the base station is in turn used as the next source. Then, from this new source, a search is made to find the shortest path to all remaining sensing locations, and the closest sensing location to the new source is used as the subsequent new source. This procedure is repeated until only the last sensing location remains.

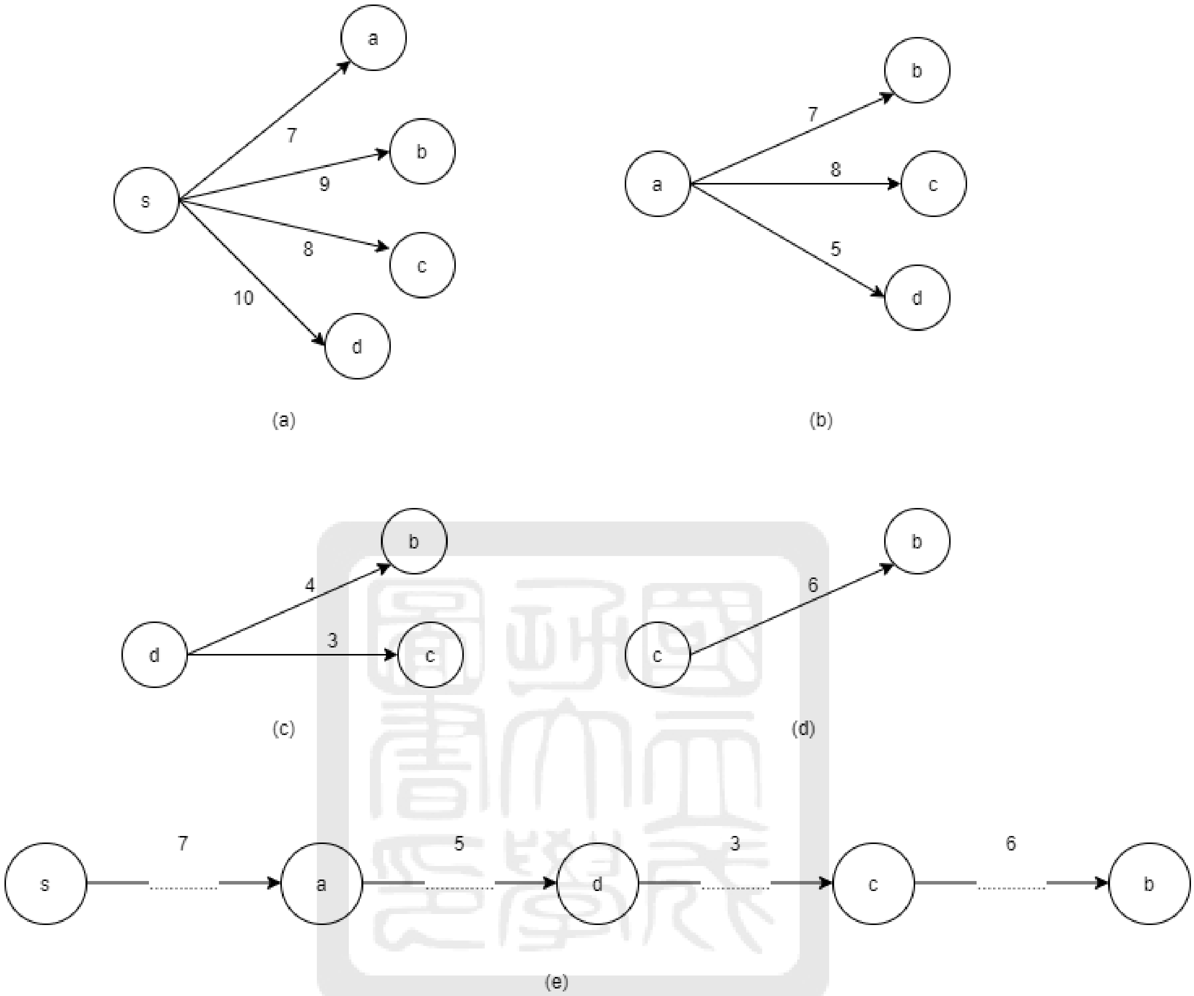


Figure 4.4. Shortest Path Multi-Source Multi-Destination

Figure 4.4 illustrates the procedure to find the shortest path from the base station to all of the sensing locations. Figure 4.4 (a) is the first step in this procedure. Here, an attempt is made to find the shortest or the minimum distance from the base station (V^d) to the sensing locations (V^s). After finding the closest one, where, in this example, sensing location *a* is the closest one and is considered to be the next source in the algorithm. We treat the remaining sensing locations (b, c, d in the example) as the destination using the same procedure. The shortest path that this algorithm finds is shown in Figure 4.4 (e). Table 4.2 shows the detailed pseudocode to find the shortest path from the base station to all sensing locations.

We use the single-source single-destination Dijkstra algorithm to find the nearest destination sensing location to the base station. Table 4.3 shows the detailed pseudocode for the single-source single-destination algorithm.

Table 4.2. Pseudocode for Multi-Source Multi-Destination Algorithm

Multi-Source Multi-Destination Algorithm

Input: Graph: G (V, A,), $V^d \in V$, $V^s \in V$

Output: A path from V^d to V^s and shortest path among destination nodes

```
L1, L2, L3
While len(V^s) > 0
    For v in V^s:
        nodes_path: = singleSourceSingleDestinationAlgorithm(G, V^d,v)
        For i in range(len(nodes_path)):
            length_path: = length_path + G[nodes_path [i], nodes_path[i + 1]]
            Add L2->col1: = nodes_path[i], L2->col2: = T[v]
        End For
        Add L1->col1: = V^s[v], L1->col2: = length_path
    End For
    L1 sorting according to length_path
    e_Dest_Node: = L1->col1
    For m in range(len(L2)):
        IF L2[m]->col2== e_Dest_Node
            Add L3->col: = L2[m]->col1
        End IF
    End For
    S: = e_Dest_Node
    Remove the e_Dest_Node value from T
End While
Return L3
```

Table 4.3. Pseudocode for Single-Source Single-Destination

Single-Source Single-Destination Algorithm

Input: Graph: G (V, A,), $V^d \in V$, $v \in V^s$

Output: Shortest Path from Base station to a sensing location

```
function singleSourceSingleDestinationAlgorithm (Graph, source, destination):
    for each vertex v in Graph:
        dist[v] := infinity
        previous[v] := undefined
    dist[source] := 0
```

```
    Q := the set of all nodes in Graph
    while Q is not empty:
        u := node in Q with smallest dist[ ]
        remove u from Q
        for each neighbor v of u:
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v]
                dist[v] := alt
                previous[v] := u
    return previous[ ]
```

2. Find the Shortest Circuit

Table 4.4. Pseudocode for Finding the Shortest Circuit

Shortest Circuit

Input: Graph: G (V, A,), Shortest Path, V^d, Source node
Output: Shortest Path Circuit

```
Function isSafe (shortestPath, v)
    For each vertex in shortestPath:
        If vertex == v:
            return false
function Shortest Circuit (Graph, shortest path, source, destination):
    for each vertex v in Graph:
        If isSafe == True:
            dist[v] := infinity
            previous[v] := undefined
    dist[source] := 0
    Q := the set of all nodes in Graph
    while Q is not empty:
        u := node in Q with smallest dist[ ]
        remove u from Q
        for each neighbor v of u:
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v]
                dist[v] := alt
                previous[v] := u
```

```
return previous[ ]
```

After finding the shortest path from the base station to all sensing locations, in this phase, an attempt was made to find the shortest circuit by developing a procedure based on the Hamilton circuit and the shortest path idea. The idea here is to find the shortest path that links the last node from the previous phase to the depot while excluding the nodes and arcs used. We treated the last node from the shortest path as the source node and the depot as the destination node. Then, we attempted to calculate the smallest distance from the node to the depot. The Hamilton circuit idea here was used to exclude the nodes and edges that had been used in the shortest path from the previous phase.

3. Finding the Schedule of the UAVs in the Circuit

Table 4.5. Pseudocode for Finding the Position of UAVs in Circuits

Find Position of UAVs in Circuits

Input: Shortest Circuit, *K* Number of UAVs, *T*
Output: Position of Each UAVs

```
Position_at_time_0 = V^d
For each t in T:
    Curr_Position = Position_at_t [t]
    Next_poss = Assign()
    For each k in K:
        goal_next = Next_position
        Curr_Position[k] = goal_next
    Position_at_t[t+1] = Curr_Position
```

After finding the shortest circuit, we attempted to schedule the UAV routes in the circuit by defining the position of each UAV for planning horizon *T*. We attempted to determine whether the circuit obtained from the previous path could be traversed with *K* UAVs before scheduling. In this study, we determined the minimum number of UAVs that can be traversed along the circuit by multiplying the length of the circuit two times. This determination was aimed toward avoiding violating the idleness constraints given in the model.

The scheduling was done by calculating the position of each UAV at each time point, starting from the base station over a finite horizon of *T*. The idea was to make each UAV travel along the circuit where one node can only be passed through by 1 UAV at each time point. This procedure avoids collision. Figure 4.4 shows an illustration of how we scheduled the flight of each UAV. Table 4.5 shows the pseudocode used to find the position of each UAV at each time point in the circuit.

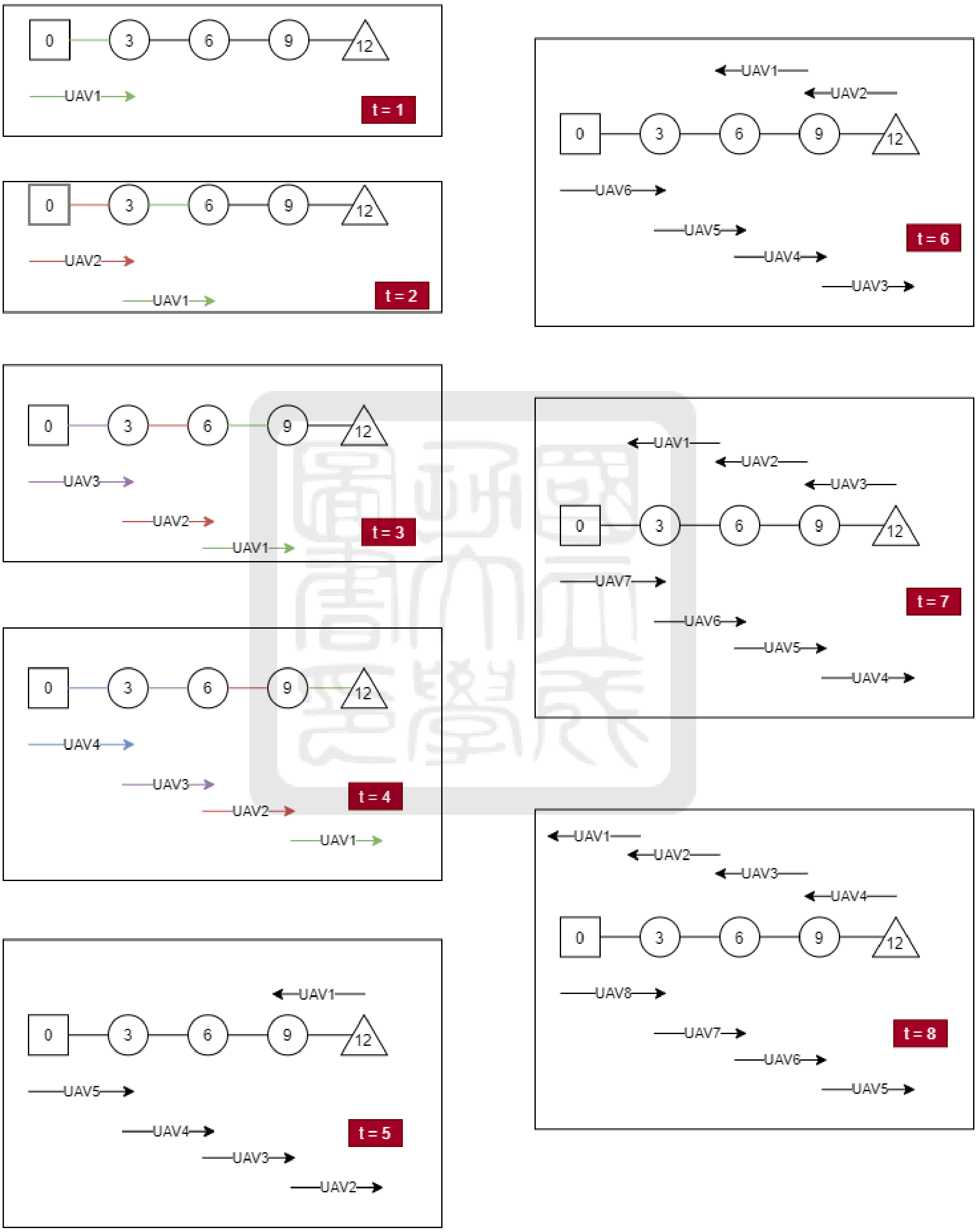


Figure 4.5. Scheduling Illustration

After finding the position of each UAV at each time point, we calculated the amount of data that could be collected using this scheduling. The calculation counts how many times each UAV reaches the depot after T = shortest circuit weight.

4.3. Summary

To know how communication among UAVs can affect our model mentioned in chapter 3, we compare it with uncooperative data transport. In this case, there is no communication made among UAVs during the mission. We provide the integer programming model for the case of uncooperative data transport.

We also provide the heuristic algorithm to deal with the computational time problem in our proposed model. This heuristic is to find the shortest circuit that links the base station or the depot with all the sensing locations. We use the idea of Dijkstra's shortest path algorithm to build this heuristic. Our current heuristic can only deal with the idleness constraints of our proposed model but still cannot directly deal with the changing of the idleness upper bound.

Chapter 5
COMPUTATIONAL EXPERIMENTS

This chapter explains the computational experiments for the proposed model as well as the computations used for the comparison model, the uncooperative data transport model. This chapter starts with Section 5.1 to explain the technical specifications for computational testing. Section 5.2 explains the settings of the experimental cases randomly generated to assess the proposed and comparison model. The experimental results are provided in Section 5.3; the results of the experimental comparison are provided in Section 5.4, and a summary of the results is given in Section 5.5.

5.1. Technical Specifications

The computational experiments were conducted on a personal computer, with the specification for the PC and the software as shown in Table 5.1.

Table 5.1. PC and software specifications

PC specifications	
Operating System	Microsoft Windows 10 1x64 based
Processor	Intel i7-8700, 3.2GHz
RAM	8.00 GB
Software specifications	
Python	Version 3.6.5
Anaconda	Version 4.9.2
Gurobi	Version 9.0.0

5.2. Settings for Experimental Cases

For the experimental process, we divided the cases, cooperative and uncooperative, into 5 scenarios according to the parameters. The parameters and scenario settings for this experiment are shown in Table 5.2. The first scenario was used to understand the impact of the length of the planning horizon on the computational time for both cases. Scenario 2 was used to understand the effect of the different upper bounds for the given latencies. Scenario 3 was used to understand the impact of the different upper bounds for idleness. Scenario 4 was used to analyze the effects of the number of UAVs used in the model. Scenario 5 was used to understand the impact of the network structure on the model, where we used two different sizes of indirect graphs that consisted of 15 nodes and 27 nodes and treated three nodes for each graph as sensing locations. Figure 5.1 shows the graphs used in this experiment. We limited the computational time for both models to 3,600 seconds. Thus, if the computational time exceeded 3,600 seconds, the calculation was terminated, and the current solution was determined to be feasible but not optimal.

Table 5.2. The Scenarios and Parameter Settings for the Model Comparison

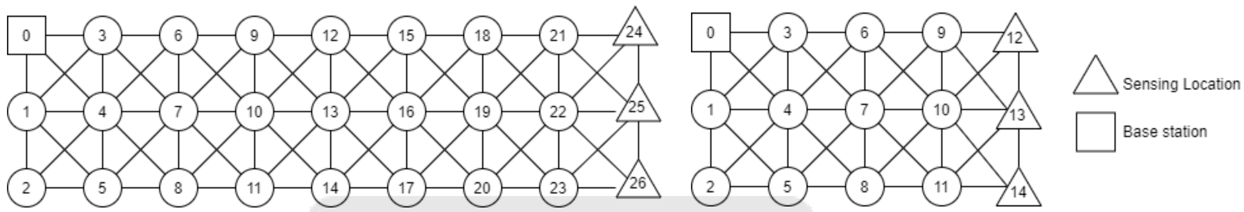
Scenarios	Codes	Nodes	Number of UAV	working time	Latency	Idleness
1	S1_1		10	15	7	1
	S1_2		10	20	7	1
2	S2_1	15	10	20	3	1
	S2_2			20	5	1
	S2_3			20	7	1
	S2_4			20	100	1
3	S3_1	15	10	20	7	1
	S3_2			20	7	3
	S3_3			20	7	5
	S3_4			20	7	8
4	S4_1	15	10	20	7	1
	S4_2		8	20	7	1
	S4_3		6	20	7	1
5	S5_1	15	10	20	7	1
	S5_2	27	10	30	11	1

The latency upper bounds in scenarios 1, 3, 4, and 5 were obtained by calculating the shortest path to all sensing locations from the base station for each graph. The shortest path was calculating using a multi-source multi-destination Dijkstra algorithm.

The value of the upper bound of the latencies given in Scenario 2 were the smaller number for the longest shortest path from all sensing locations to the base station, the longest shortest path from all sensing locations to the base station as calculated using the single-source single-destination Dijkstra algorithm, the shortest path to all sensing locations from the base station calculated using the multi-source multi-destination Dijkstra algorithm,. Table 5.3 shows the shortest path obtained.

Table 5.3. Shortest Path for the Latency Constraints

Nodes	Single-source Single-destination	Multi-source Multi Destination
15	5	7
27	9	11



(a) (b)

Figure 5.1. The Graphs used for the Experiment.
(a). Graph with 27 Nodes; (b). Graph with 15 Nodes

Table 5.4 shows the experimental settings used when comparing the model with the algorithm.

Table 5.4. The Scenarios and Parameters Settings for the Model and the Algorithm Comparison

Scenarios	Codes	Nodes	Number of UAVs	Working Time	Latency	Idleness
	S6_1			15		1
	S6_2			20		1
	S6_3			21		1
6	S6_4	15	10	25	7	1
	S6_5			27		1
	S6_6			30		1
	S6_7			35		1

5.3. Experimental Results

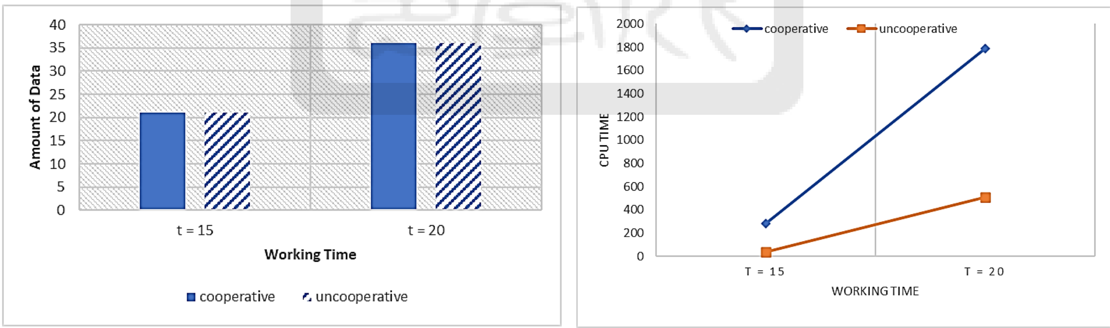
This section is divided into two subsections detailing the evaluation made to determine the amount of data collected and the computational time necessary for each case.

5.3.1. Comparison of the Cooperative and Uncooperative Data Transport Cases

To evaluate the model's performance, as mentioned in Chapter 3, each scenario was run on each model with the different parameter settings according to Table 5.2. Each experiment was considered alone in an individual run for the first step. All scenarios were solved in 3,600 seconds, irrespective of whether a feasible or optimal solution was found. This time limit was used because increasing the 3,600 seconds limit would not lead to any significant improvement or because it would take too long to obtain any significant improvement.

1. Scenario 1 – Impact of the length of the planning horizon

We tried to determine the effect of the length of the planning horizon on the computational time required to generate the optimal solution for each model. Figure 5.2 (a) shows the amount of data collected by the two cases with planning horizons of different lengths. It can be seen that both cases produced the same amount of data. This shows that the planning horizon did not lead to differences in the amount of data collected in either case.



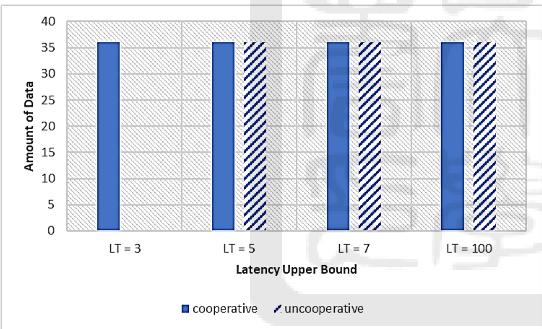
(a) (b)

Figure 5.2. Scenario 1 Results of the Comparison of the Cooperative and Uncooperative Models

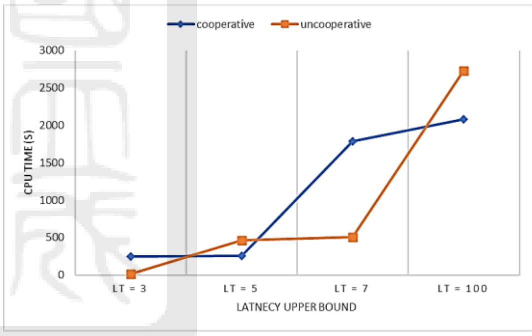
Figure 5.2 (b) shows a graph of the computational time required for both cases to solve the problem. It can be seen from the figure that there is a significant increase in the amount of computational time required if the length of the time horizon is increased, or it can be said that the computational time is directly proportional to the length of the time horizon in both cases. We concluded that the cooperative case requires a longer computational time than the uncooperative case in this scenario.

2. Scenario 2 – Impact of different upper bounds on the latency constraint

In the second scenario, we attempted to determine the effects of latency upper bound on the amount of data collected and the computational time needed in both cases to generate the optimal solution. Figure 5.3 (a) shows the amount of data collected in both cases with a different latency upper bound. The figure shows that the amount of data collected in both cases remained the same for all problems in the scenario, except the problem with LT = 3, where the uncooperative case could not collect any amount of data due to the latency constraints. We thus concluded that if the latency upper bound given is equal to or larger than the shortest distance from the base station to all sensing locations, the same amount of data could be collected in both cases, or, it can be said that if the latency constraints are always guaranteed, then both cases could generate the same optimal solution.



(a)



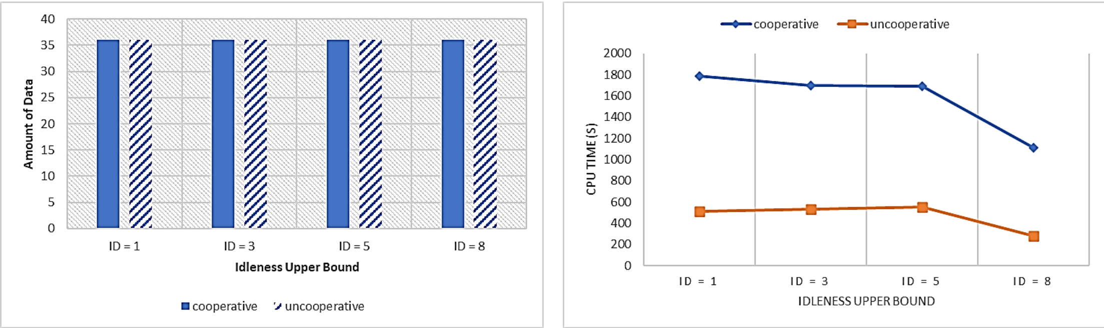
(b)

Figure 5.3. Scenario 2 Results of the Comparison of the Cooperative and Uncooperative Models

Figure 5.3 (b) shows the computational time required for both cases for Scenario 2. The figure shows that the computational time was directly proportional to the latency upper bound in both cases. In some problems, the computational time for the cooperative cases was longer than the uncooperative one, and vice versa. From the figure, it could not be concluded that the cooperative case for this scenario required more computational time than the uncooperative case.

3. Scenario 3 – Impact of different upper bounds on the idleness constraint

In this scenario, we tried to understand the effect of the upper bound of idleness on the computational time required in both cases. Figure 5.4 (a) shows the amount of data collected in both cases. It can be seen that the amount of data collected was not affected by the upper bound of idleness, and in both cases, the same amount of data for all problems in this scenario.



(a) (b)

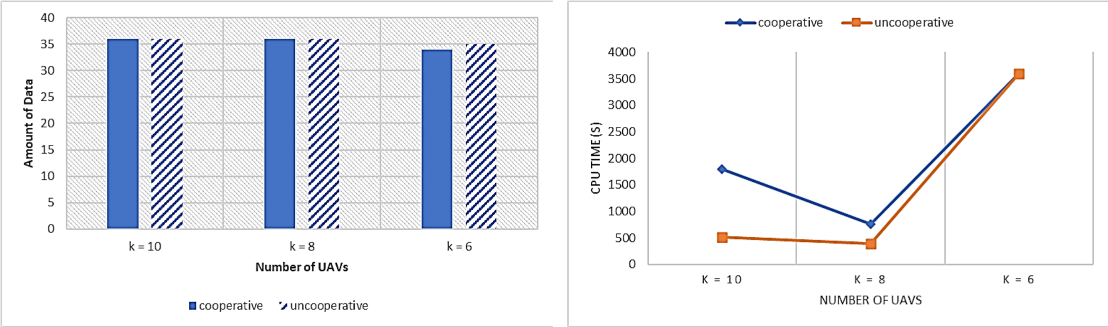
Figure 5.4. Scenario 3 Results of the Comparison of the Cooperative and Uncooperative Models

Figure 5.4 (b) shows the computational time required by both cases to generate the solution. The figure shows that the computational time required for both cases was affected by the upper bound of idleness. Still, we could not conclude that the computational time was directly proportional to the upper bound of idleness because both cases exhibited different patterns. The computational time for the cooperative case was inversely proportional to the upper bound of idleness, and it fluctuated for the uncooperative cases.

4. Scenario 4 – Impact of the use of a different number of UAVs

In this scenario, we attempted to understand the effect of a different number of UAVs used to carry out the mission on the amount of data and the computational time required in both cases.

Figure 5.4 (a) shows the amount of data collected in both cases. From the figure, we can see that the amount of data collected with different numbers of UAVs remained the same. This means that the number of UAVs did not affect the amount of data collected during the mission by time 0. We also concluded that both cases collected the same amount of data, meaning that the number of UAVs also did not affect the difference in the amount of data collected in the two cases.



(a) (b)

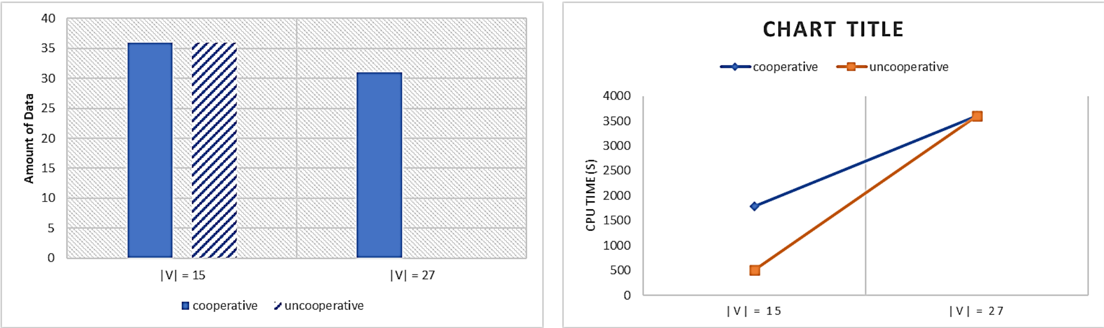
Figure 5.5. Scenario 4 Results of the Comparison of the Cooperative and Uncooperative Models

Figure 5.4 (b) shows the computational time required by both cases. The figure shows that the computational time for both cases was not directly proportional to the number of UAVs used. It can be seen that the computational time fluctuated, and for this scenario, a smaller number of UAVs used led to a longer computational time. Figure 5.4 (b) shows that when K = 6, the computational time for both cases exceeded the time limit of 3,600 seconds.

5. Scenario 5 – Impact of the graph size

In this scenario, we attempted to understand the impact of the size of the graph on both models. Figure 5.6 (a) shows the optimal solution generated by both models using the GUROBI solver and Figure 5.6 (b) shows the computational time needed to generate the optimal solution. From the figures, it can be seen that when the nodes |V| = 15, both models were able to give the optimal solution within the specified time limit (3,600 seconds). When the nodes |V| = 27, neither model could generate the optimal solution within the specified time limit.

Table 5.5 shows all the experimental results for the comparison of the cooperative and uncooperative cases. For a further analysis, we provided the optimality gap for both models. The models' optimality gaps were obtained from a mixed-integer programming (MIP) gap provided by GUROBI. Both models had almost the same optimal solution values in all scenarios with varied parameter settings taken from the table.



(a) (b)

Figure 5.6. Scenario 4 Results of the Comparison of the Cooperative and Uncooperative Models

Table 5.5. Results of the Comparison of the Cooperative and Uncooperative Data Transport Cases

		Cooperative			Uncooperative		
Scenarios	**Codes**	**Data Collected**	**CPU TIME (s)**	**Gap**	**Data Collected**	**CPU TIME (s)**	**Gap**
1	S1_1	21	285.58	0.00%	21	38.11	0.00%
	S1_2	36	1789.54	0.00%	36	508.65	0.00%
2	S2_1	36	244.94	0.00%	0	10	0.00%
	S2_2	36	253.67	0.00%	36	468.13	0.00%
	S2_3	36	1789.54	0.00%	36	508.65	0.00%
	S2_4	36	2077.39	0.00%	36	2726.22	0.00%
3	S3_1	36	1789.54	0.00%	36	508.65	0.00%
	S3_2	36	1694.81	0.00%	36	533.79	0.00%
	S3_3	36	1691.37	0.00%	36	549.43	0.00%
	S3_4	36	1113.03	0.00%	36	279.52	0.00%
4	S4_1	36	1789.54	0.00%	36	508.65	0.00%
	S4_2	36	756.19	0.00%	36	387.85	0.00%
	S4_3	34	3600.9	5.88%	35	3600.71	2.86%
5	S5_1	36	1789.54	0.00%	36	508.65	0.00%
	S5_2	31	3600.39	35%	N/A	3600.46	N/A

5.3.2. Comparison of the Models and the Proposed Algorithm

To analyze the performance of both models and the algorithm, we compared three parameters: computational time, the optimality gap, and whether the algorithm could find an optimal solution or not. Table.5.6 shows all the experiment results. The table shows that both models were able to obtain an optimal solution, where *T* is lower than 30 units of time within the time limit (3,600 seconds). For cases where T was more than 30 units, neither model could find the optimal solution within the time limit (3,600 seconds). Meanwhile, the proposed algorithm could find an optimal solution in less than 10 seconds. We calculated the optimality gap of the solution obtained for further analysis. The optimality gaps for both models were obtained from a mixed-integer programming (MIP) gap provided by GUROBI. Meanwhile, to calculate the optimality gap for the algorithm, we compared the results obtained for cooperative cases from GUROBI and the algorithm. The formulation for the algorithm optimality gap is as follows:

$$OptGap(\%)_{(c)} = \left| \left((Algorithm_{result} - GUROBI_{cooperative\_result}) / GUROBI_{cooperative\_result} \right) 100\% \right| \quad (5.1)$$

Table 5.6. Experimental Results for Both Models with GUROBI and the Proposed Algorithm

		Cooperative			Uncooperative				Algorithm		
Codes	**T**	**Data**	**CPU Time**	**Gap (G)**	**Data**	**CPU Time**	**Gap (G)**	**Gap (C)**	**Data**	**CPU Time**	**Gap (C)**
S6_1	15	21	285.58	0.00%	21	38.11	0.00%	0.00%	18	3.6	14.29%
S6_2	20	36	1789.5	0.00%	36	508.65	0.00%	0.00%	33	5.3	8.33%
S6_3	21	39	1989.1	0.00%	39	841.9	0.00%	0.00%	36	5.5	7.69%
S6_4	25	51	2456.5	0.00%	51	1160	0.00%	0.00%	48	5.9	5.88%
S6_5	27	57	2653.3	0.00%	57	1361.5	0.00%	0.00%	54	7.01	5.26%
S6_6	30	66	2970	0.00%	66	1799.9	0.00%	0.00%	63	8.4	4.55%
S6_7	35	67	3600	19.14%	56	3600	44.64%	16.41%	78	10.1	16.42%

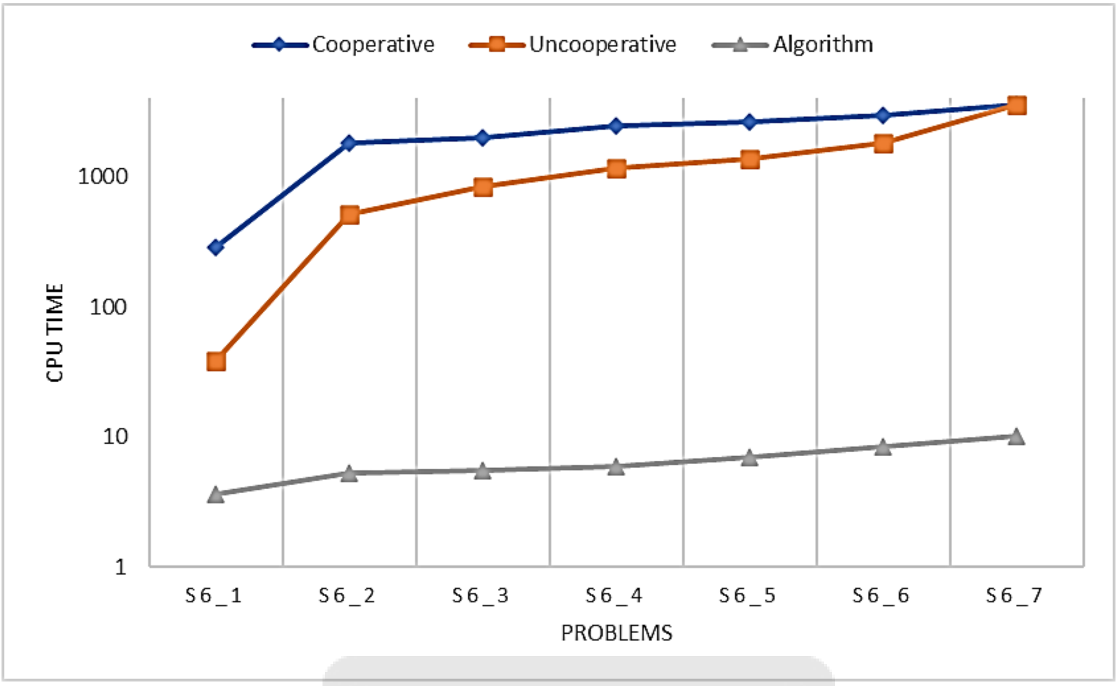


Figure 5.7. Comparison of Computational Time for Both Models and the Proposed Algorithm.

5.4. Summary

To determine the effect of the wireless connection used by UAVs while carrying out their mission, we attempted to compare our proposed cooperative data transport model with the uncooperative data transport model, which does not consider the use of a wireless connection among UAVs. There were five scenarios used for this comparison.

Based on the model comparison results, neither case was affected by scenarios 2, 3, and 4 in terms of the amount of data collected. The results showed that the amount of data collected remained the same in all scenarios with different parameter settings. In Scenario 2, we found that both cases will always collect the same amount of data if the latency constraints are guaranteed or if the latency upper bound is equal to or larger than the shortest distance from the base station to the sensing location. In a case where the latency upper bound is smaller than the shortest distance from the base station to the sensing location, the uncooperative cases cannot collect any data due to violating the latency constraints.

Using scenarios 1 and 5, we attempted to determine how significant the effect of increasing the amount of working time and the size of the graph is in both cases, respectively.

From the comparison results, we found that using a GUROBI solver in both scenarios 1 and 5 significantly affected the computational time in both models. We found that the cooperative data transport model requires a longer computational time than the uncooperative case in both scenarios.

The following experiment was conducted to determine how well the proposed algorithm handled the computational time problem in scenarios 1 and 5. The experiment was carried out by comparing the results of the models and algorithm with different working time settings. We found that neither models could find an optimal solution when the amount of working time exceeded 30 units within the specified computational time bound (3,600 seconds). On the other hand, the proposed algorithm was able to provide a solution in less than 10 seconds. However, the proposed algorithm cannot deal with the latency constraints. In addition, before implementing the algorithm, we had to ensure that the idleness constraints were always guaranteed.

Chapter 6
CONCLUSION AND SUGGESTIONS FOR FUTURE RESEARCH

6.1. Conclusion

We focus on discussing a multi-UAV surveillance routing problem. An integer programming model for this problem was proposed that routes a fleet of UAVs from a base station to repeatedly collect data from sensing locations through cooperative data transport while ensuring latency and idleness constraints during the planning horizon. Using cooperative data transport, the multi-UAVs used to carry out surveillance missions work together to transfer the collected data to the depot in a store-and-forward fashion. To track the UAV movement, we use a time-space network technique that stores the trajectory of each UAV over time and space.

To determine the influence of the UAV cooperation and the wireless data transfer on the UAVs, we designed an uncooperative data transport model for the purpose of comparison. This uncooperative model considers a situation without wireless communication, which means each UAV has to collect and transfer the data from the sensing location to the base station individually. We conducted computational experiments for both cooperative and uncooperative data transport cases. From the comparison results, it was found that, in terms of the amount of data collected, both cases can produce the same result in the following two situations: (1) The latency upper bound given is equal to or larger than the closest distance from the base station to all sensing locations, or (2) the latency constraints are always guaranteed. In terms of computational time, both cases are affected by the working time and the graph size.

Dealing with the computational time problem in both cases, we attempted to perform some algorithms. The algorithm was divided into 3 phases: The first one finds a route from the base station to all sensing locations consecutively using the shortest path algorithm; the second phase connects the route back to the base to form a circuit, and the last one schedules each UAV to move along the circuit.

We experiment to analyze the performance of the proposed algorithm by comparing the results obtained by both models with our algorithm. The results show that neither model could find an optimal solution when the amount of working time used was more than 30 units within the specified computational time bound (3,600 seconds). The proposed algorithm, however, was able to provide a solution in less than 10 seconds.

6.2. Future Research

There are some related issues worthy of future investigation. A list of our suggestions based on our findings is provided as follows:

1. Consideration of capacity and energy consumption:
   We only focused on how wireless connections can affect the amount of data collected by UAVs during a mission while considering the latency and the idleness. Thus, we assumed that each UAV has unlimited capacity and energy to carry out the mission. However, in some instances, to provide a more realistic setting, it is possible to consider the use of the capacity and energy consumption for each UAV.
2. Consideration of the environment:
   We did not consider the environment in which a fleet of UAVs will move during a mission (e.g., buildings, wind, weather, etc.). However, considering the real-world environment for UAV movement will make the problem more realistic and match a real-world scenario.
3. Design a more effective algorithm that guarantees idleness:
   Our proposed method resolves the idleness constraints indirectly. Although we expect the route constructed using the shortest paths will indirectly help shorten idleness, this approach mixes the data collection and transfer purposes. It would be better to have an algorithm that can also directly deal with the idleness guarantee.
4. Design an algorithm that can deal with both idleness and latency:
   The larger the network becomes, the longer the computational time will be. Thus, we suggest investigating a design with a more efficient solution framework. For example, a genetic algorithm could be used to deal with both latency and idleness. The idea of developing this genetic algorithm based on our problem set would be to treat the UAV route as a chromosome. To satisfy the latency and idleness constraints, we would track the encoding in each chromosome, which might represent the node sequence passed by UAVs. However, we expect some problems would need to be addressed for performing a genetic algorithm, including the following:

a. Deciding whether each chromosome will have a fixed or variable chromosome length:
   For each decision, there would have to be some considerations. If the fixed number of genes for each chromosome were to be decided, it would be very likely that some chromosomes would not end at the base station, or perhaps the last gene of the chromosome would not be the gene representing the depot node. Otherwise, if we consider each chromosome to have a different number of genes based on each UAV's route, this would raise some problems when we deal with mutations and crossover in the genetic algorithm.
b. Deciding the type of graph that will be used to represent the area movement of UAVs:
   Determining the type of graph to be used in this problem matters whether we will use the complete graph or not. If we consider the graph to be incomplete, after the mutation or crossover, the node (i.e., gene) sequence in some chromosome may violate the adjacency relationships. In that case, we would need to further fix the chromosome to deal with its feasibility.
c. Penalty for an infeasible chromosome:
   When we encounter an infeasible chromosome, there may be more than one way to set the associated penalties. However, the method for setting them up is not clear.

References

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. https://doi.org/10.1007/s10846-013-9938-z

Acevedo, J. J., Arrue, B. C., Diaz-Banez, J. M., Ventura, I., Maza, I., & Ollero, A. (2013). Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 565–574. https://doi.org/10.1109/ICUAS.2013.6564734

Amigoni, F., Banfi, J., & Basilico, N. (2017). Multirobot Exploration of Communication-Restricted Environments: A Survey. *IEEE Intelligent Systems*, *32*(6), 48–57. https://doi.org/10.1109/MIS.2017.4531226

Baldacci, R., Bartolini, E., Mingozzi, A., & Valletta, A. (2011). An Exact Algorithm for the Period Routing Problem. *Operations Research*, *59*, 228–241. https://doi.org/10.1287/opre.1100.0875

Banfi, J., Basilico, N., & Amigoni, F. (2015). Minimizing communication latency in multirobot situation-aware patrolling. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 616–622. https://doi.org/10.1109/IROS.2015.7353436

Banfi, J., Li, A. Q., Basilico, N., Rekleitis, I., & Amigoni, F. (2016). Asynchronous multirobot exploration under recurrent connectivity constraints. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 5491–5498. https://doi.org/10.1109/ICRA.2016.7487763

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. https://doi.org/10.1007/s10514-017-9652-y

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Haghnazar K., R., Zuidema, D., Lima, M. S. F., & da Costa, A. R. (2017). A multi-objective green UAV routing problem. *Computers & Operations Research*, *88*, 306–315. https://doi.org/10.1016/j.cor.2017.04.011

Fu, Z., Chen, Y., Ding, Y., & He, D. (2019). Pollution Source Localization Based on Multi-UAV Cooperative Communication. *IEEE Access*, *7*, 29304–29312. https://doi.org/10.1109/ACCESS.2019.2900475

Grotli, E. I., & Johansen, T. A. (2012). Path- and data transmission planning for cooperating UAVs in delay tolerant network. *2012 IEEE Globecom Workshops*, 1568–1573. https://doi.org/10.1109/GLOCOMW.2012.6477819

Kantaros, Y., Guo, M., & Zavlanos, M. (2019). *Temporal Logic Task Planning and Intermittent Connectivity Control of Mobile Robot Networks*. https://doi.org/10.1109/TAC.2019.2893161

Kuznetsov, N. A., Myasnikov, D. V., & Semenikhin, K. V. (2016). Optimal control of data transmission in a mobile two-agent robotic system. *Journal of Communications Technology and Electronics*, *61*(12), 1456–1465. https://doi.org/10.1134/S1064226916120159

Manyam, S. G., Rasmussen, S., Casbeer, D. W., Kalyanam, K., & Manickam, S. (2017). Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 573–580. https://doi.org/10.1109/ICUAS.2017.7991314

Nigam, N., Bieniawski, S., Kroo, I., & Vian, J. (2012). Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results. *IEEE Transactions on Control Systems Technology - IEEE TRANS CONTROL SYST TECHN*, *20*, 1236–1251. https://doi.org/10.1109/TCST.2011.2167331

Pasqualetti, F., Durham, J. W., & Bullo, F. (2012). Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms. *Robotics, IEEE Transactions On*, *28*, 1181–1188. https://doi.org/10.1109/TRO.2012.2201293

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. https://doi.org/10.1109/TRO.2011.2179580

Ponda, S. S., Johnson, L. B., Kopeikin, A. N., Choi, H., & How, J. P. (2012). Distributed Planning Strategies to Ensure Network Connectivity for Dynamic Heterogeneous Teams. *IEEE Journal on Selected Areas in Communications*, *30*(5), 861–869. https://doi.org/10.1109/JSAC.2012.120603

Scherer, J., & Rinner, B. (2019). Persistent Multi-UAV Surveillance with Data Latency Constraints. *ArXiv:1907.01205 [Cs, Eess]*. http://arxiv.org/abs/1907.01205

Scherer, J., & Rinner, B. (2020a). Multi-Robot Persistent Surveillance With Connectivity Constraints. *IEEE Access*, *8*, 15093–15109. https://doi.org/10.1109/ACCESS.2020.2967650

Scherer, J., & Rinner, B. (2020b). Multi-UAV Surveillance With Minimum Information Idleness and Latency Constraints. *IEEE Robotics and Automation Letters*, *5*(3), 4812–4819. https://doi.org/10.1109/LRA.2020.3003884

Scherer, J., & Rinner, B. (2020). Multi-Robot Patrolling with Sensing Idleness and Data Delay Objectives. *Journal of Intelligent & Robotic Systems*, *99*(3–4), 949–967. https://doi.org/10.1007/s10846-020-01156-6

Scherer, J., & Rinner, B. (2016). Persistent multi-UAV surveillance with energy and communication constraints. *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 1225–1230. https://doi.org/10.1109/COASE.2016.7743546

Scherer, J., & Rinner, B. (2017). Short and full horizon motion planning for persistent multi-UAV surveillance with energy and communication constraints. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 230–235. https://doi.org/10.1109/IROS.2017.8202162

Spirin, V., & Cameron, S. (2014). Rendezvous through obstacles in multi-agent exploration. *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, 1–6. https://doi.org/10.1109/SSRR.2014.7017656

Spirin, Victor, Cameron, S., & de Hoog, J. (2014). Time Preference for Information in Multi-agent Exploration with Limited Communication. In A. Natraj, S. Cameron, C. Melhuish, & M. Witkowski (Eds.), *Towards Autonomous Robotic Systems* (pp. 34–45). Springer. https://doi.org/10.1007/978-3-662-43645-5_5

Tsai, F.-M., Lu, C.-C., & Chang, Y.-M. (2016). A network model for solving the yard truck routing and scheduling problem. *The International Journal of Logistics Management*, *27*(2), 353–370. https://doi.org/10.1108/IJLM-09-2014-0158

Wu, Y., Ren, X., Zhou, H., Wang, Y., & Yi, X. (2020). A Survey on Multi-Robot Coordination in Electromagnetic Adversarial Environment: Challenges and Techniques. *IEEE Access*, *8*, 53484–53497. https://doi.org/10.1109/ACCESS.2020.2981408

Zhang, M., & Kingston, D. (2015). Time-space network based exact models for periodical monitoring routing problem. *2015 American Control Conference (ACC)*, 5264–5269. https://doi.org/10.1109/ACC.2015.7172161