Analytics-Driven Routing of Inspection Crews and Aerial Sensors for Post-Disaster Damage Assessment

by

Andrew C. Lee

B.S., United States Military Academy (2003)
S.M., Massachusetts Institute of Technology (2012)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author .....
Department of Civil and Environmental Engineering
May 17, 2019

Signature redacted

Certified by .....
Saurabh Amin
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Heidi Nepf
Donald and Martha Harleman Professor
of Civil and Environmental Engineering
Chair, Graduate Program Committee

Analytics-Driven Routing of Inspection Crews and Aerial Sensors for Post-Disaster Damage Assessment

by

Andrew C. Lee

Submitted to the Department of Civil and Environmental Engineering
on May 17, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation

Abstract

Infrastructure networks such as natural gas pipelines and water systems are prone to failures from natural disasters, which result in huge societal and economic losses. To minimize these losses, inspection crews must rapidly identify failures (e.g., pipeline bursts, waterway blockages). However, infrastructure agencies often incur high costs and delays due to limited resources and diagnostic uncertainty about locations and types of failures. This thesis presents an analytics-driven network inspection approach that leverages data from fixed sensors and Unmanned Aerial Systems (UAS) to reduce diagnostic uncertainty, and determines optimal routing strategies for both ground crews and UAS.

In our approach, the network is partitioned into smaller regions (subnetworks) based on the monitoring range of fixed sensors. We use sensor data and relevant physical features to assign priority inspection levels and predict failure rates for these subnetworks. We then leverage UAS to localize failures and incrementally update failure rates. The overall inspection is based on two routing problems: the Aerial Sensor Inspection Problem (ASIP), which guides UAS-based inspection of subnetworks; and the Prioritized Inspection Routing Problem (PIRP), which integrates pre-solved ASIP times and failure rates to determine crew routing strategies.

For pipeline network inspection, we consider a set of monitoring locations that enable modeling of UAS platform and infrastructure topology constraints, and determine feasibility of UAS routes. To solve the ASIP for realistic situations, we propose an efficient set-cover-based heuristic. We show how to obtain crew routing strategies for large-scale network inspection by integrating ASIP solutions into the PIRP, and solving the resulting Mixed Integer Programming (MIP) problem.

For drainage network inspection, we find that post-storm fixed sensor alerts are strongly correlated to the extent of damage in corresponding subnetworks. We present two formulations of PIRP: an adaptive stochastic dynamic program that considers prediction intervals

of failure rates; and a non-adaptive certainty equivalent MIP that only accounts for mean failure rates. Solutions to these problems allow us to evaluate the value of integrating sensor data into inspection operations. We demonstrate the benefits of our approach using real data on network failures and inspections following Hurricane Harvey in 2017.

Thesis Supervisor: Saurabh Amin
Title: Associate Professor of Civil and Environmental Engineering

Acknowledgments

First and foremost, I would like to express my sincere appreciation and immense gratitude to my advisor and mentor Professor Saurabh Amin for his relentless support of my Ph.D study and related research. He provided what seemed to be a bottomless supply of motivation, inspiration, and encouragement throughout this endeavor. His rare combination of technical expertise, research experience, and empathy guided me throughout the writing of this thesis, while at the same time empowering me with the flexibility to reserve precious time with my family. I also thank his wife for supporting him and his children throughout the hundreds of hours he must have logged trying to guide me.

I would also like to thank the rest of my thesis committee. Professor Cynthia Barnhart and Professor Arnold Barnett helped me to view this research from alternative perspectives and to consider several different extensions. Professor Barnhart has been my advocate and stalwart supporter throughout my graduate school experience, and I will always admire her leadership as well as her exceptional combination of academic and emotional intelligence. The generous time she set aside to meet with me, revise my work, and to guide me, despite the demands of her role as Chancellor, speaks volumes to her character. Professor Barnett has been a great role model for me since I took his Urban Operations Research class. In particular, his outstanding teaching and writing style is something that I will always aspire to achieve.

I am so thankful for my friends in the Resilient Infrastructure Networks Lab. In particular, I am grateful for Dr. Mathieu Dahan, whose friendship and expert counsel in professional writing and mathematical notation helped me immensely in journal and conference proceedings. He provided honest and valuable feedback that had a multiplicative impact on the quality of this research. I thank the rest of my labmates, Dr. Jeffrey Liu, Dr. Li Jin, Dr. Devendra Shelar, Manxi Wu, and Derek Chang, for their valuable input during our group presentation meetings.

This thesis work also benefited from my collaboration with individuals outside of MIT. Andrew Weinert from Lincoln Labs, whose research fortunately aligned well with mine, provided the requisite technical background on unmanned aerial systems. I am so very grateful for my interactions with Jeremy Justice, Bruce Davidson, and Andrew Orlando from the Harris County Flood Control District. Their assistance in the data gathering process and their domain-specific knowledge of hydrology and drainage network inspection operations were instrumental in understanding the practical impacts to be gained from this research.

Above all, I would like to dedicate this work to my wife (aka soulmate and best friend), for her unwavering support throughout this entire process, which includes taking care of my children during a preceding Afghanistan deployment. I will always be thankful for her selflessness and sacrifice; this PhD is as much hers as it is mine. I thank my son and daughter (aka precious gifts from God) for providing me with the requisite comedic relief from my

studies, and the impetus to schedule my time wisely to complete my requirements.

Of course, this PhD would not have been possible without the financial support from the US Army, the support of the senior faculty in the Department of Mathematical Sciences at the US Military Academy, and the support of the faculty and staff in the Interdepartmental Transportation Program at MIT.

Last but not least, I would like to thank God and my parents for always leading me towards the right decisions in the most critical intersections of my life, and for their steadfast love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the frequency and severity of natural and man-made disasters have increased, resulting in significant societal and economic losses. In the U.S., the 2017 hurricane season was the costliest on record, totaling $306.2 billion in damages [61]. The 2018 season was not far behind at $91 billion, the fourth highest in recent years [61]. Moreover, the 2018 wildfires in California destroyed nearly 14,000 homes and causing 85 deaths, the most in over a century [52]. Furthermore, a 2018 cyber-attack on government systems in Atlanta impacted over 6 million people by disabling basic municipal functions, costing the city nearly $10 million [22, 59].

These occurrences reveal the particularly vulnerable state of critical infrastructure networks such as electric, transportation, gas, and water, to disaster-induced failures. For example, vital corridors of a transportation network can be disconnected by debris or damage from a hurricane, impeding emergency response and rescue efforts. For oil pipeline networks, the recent "Protecting our Infrastructure of Pipelines and Enhancing Safety (PIPES) Act of 2016" indicates the level of increasing legislation and regulatory pressure by the government for added protection from failures [63]. For open-channel drainage networks, debris blockages can lead to adverse consequences such as levee structural failures, limited access to clean drinking water, or prolonged flood risks.

In promoting disaster mitigation and coordination of response efforts, the Federal Emergency Management Agency (FEMA) prescribes four phases: mitigation, preparedness, response, and recovery. Of these, the recovery phase often incurs the highest costs for the aforementioned network agencies. In fact, FEMA's public assistance funding program, which helps states to clear debris and rebuild infrastructure, had eight of its most expensive years in the last decade [65]. To reduce recovery costs for a network agency, inspections are an important part of the response phase, constituting a critical bottleneck for efficient recovery as shown in Figure 1-1. Inspection operations can lead to significant indirect cost savings in the recovery phase through earlier failure identification that can refine assessment of the "scope, extent, and impact of the event" [30]. Furthermore, timely and accurate intelligence can impact the upstream preparedness phase, by deferring the need for agencies to achieve accuracy in forecasts and risk assessments [70].

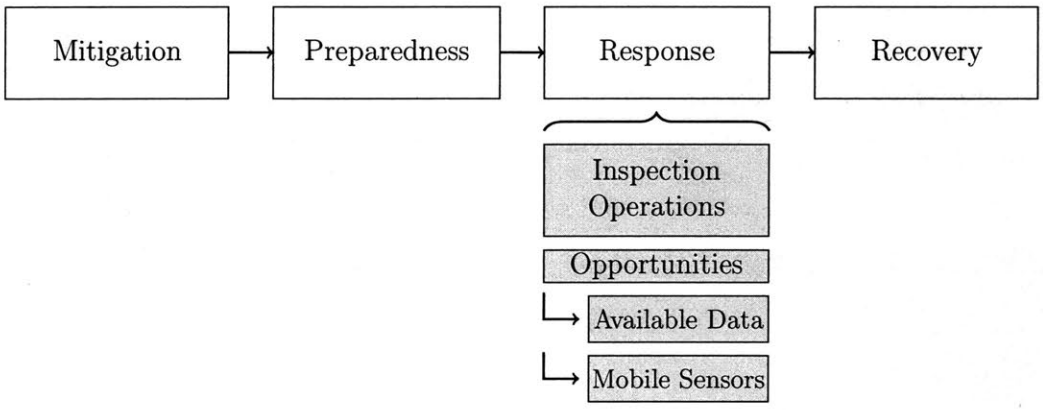


Figure 1-1: Four phases of disaster management and our focus on inspection operations during the response phase

Additionally, network agencies can achieve direct, sustainable cost savings by streamlining the inspection process itself. Current inspection operations lack a systematic process to assign priority levels to inspect various subregions of a large-scale network, given diagnostic uncertainty about the location and type of failures. This is a key omission, given the limited inspection resources and strict time constraints to inspect an extensive infrastructure network. Managers typically rely on experience or a priori fixed inspection schedules to dictate where to send inspection crews. The potential of leveraging available data that reflects the susceptibility of certain regions of the network to failures has not been fully realized. More-

over, gaining access to the infrastructure in a post-storm environment is often challenging due to poor weather, obstructions, and wet terrain that impede travel for inspection crews. This thesis proposes an analytics-driven approach that leverages data from fixed sensors and Unmanned Aerial Systems (UAS) to reduce diagnostic uncertainty, and determines optimal routing strategies for both ground crews and UAS for efficient inspection. We address the accessibility challenge with the integration of aerial sensors such as unmanned aerial sensors (UAS) for timely and comprehensive failure identification.

1.1 Post-Disaster Infrastructure Damage Assessment

To facilitate timely identification of failures, network agencies rely on fixed sensors to assist in localizing them; however, the need for precise isolation and confirmation of actual failure still exists. For example, using a Supervisory Control and Data Acquisition (SCADA), natural gas pipeline networks can detect disruptions due to leaks using fixed sensors such as pressure sensors, flow sensors, or acoustic sensors. Likewise, drainage networks use flood sensors to measure water elevation levels in the event of a storm. Based on the detection of anomalies or pre-determined threshold measures, these sensors can send alerts to the SCADA system. However, technological limitations and budget constraints often limit the network agency's ability to monitor every critical network component with a fixed sensor. In this thesis, we consider that the situational awareness from fixed sensor alerts cannot be narrowed beyond a certain spatial zone, which we call a *subnetwork*.

Typical inspection methods by infrastructure network agencies responding to disaster events are resource-intensive, including both aerial patrols and ground-based inspection crews. Aerial patrols use fixed wing or helicopter assets in order to gain a broad understanding of the damage and determine accessibility. After aerial reconnaissance is completed, inspection crews travel by vehicle from their service station to regions requiring closer inspection. Once at the region, the crew disembarks to conduct inspection, maneuvering through the infrastructure network by foot. Throughout the inspection, the crew gathers data in

the form of photographs, measurements, and geographic coordinates of failures. Once the inspection is complete, the crew can visit another region or return to their service station, depending on the time remaining.

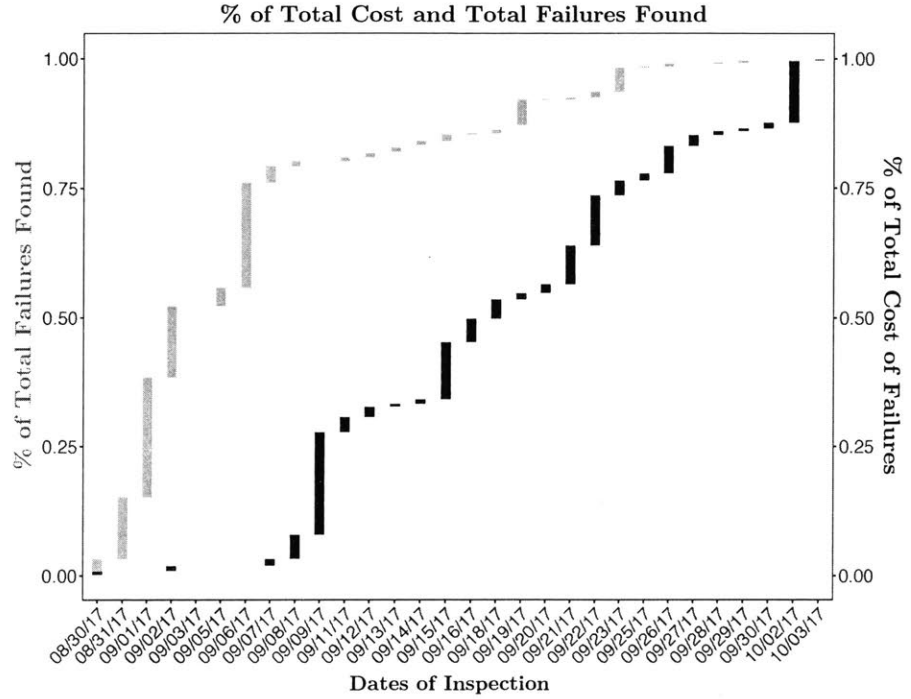


Figure 1-2: Drainage Network Cumulative Cost and Number of Failures Found by Day.

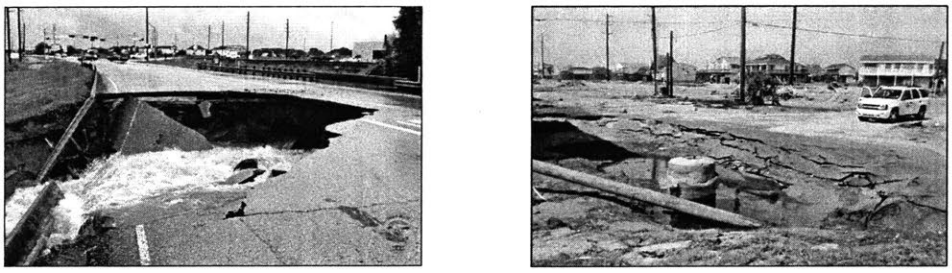
Unfortunately, ground-based inspections to isolate and confirm the failures are inefficient and costly. Early identification of failures with higher repair costs can provide the most cost savings due to the rising costs and risks of deferred maintenance. For example, concrete panel damages in an open-channel drainage network can cost up to $400K and lead to reduction of flow or complete blockage if not addressed immediately. Following a major storm event in 2017, one open-channel drainage network agency spent over $1 million and more than a month to inspect their 2,500-mile infrastructure. The subsequent removal of debris during the recovery phase cost approximately $6 million. In Figure 1-2, we show the cumulative number of failures as well as the cumulative value of failures with repair costs higher than $10K (based on the median repair cost from inspection data) during post-Harvey inspections that lasted from late August 2017 to October 2017. Although the majority of failures were properly identified early in the inspection timeline, the failures with higher repair costs were

not, with most not found until over a week after inspections began.

1.2 Inspection Crew Routing

1.2.1 Key Challenges

There are two main challenges faced by the network agency in current inspection processes. First, access to the infrastructure in a post-disaster environment is challenging. Ground-based crews require adequately favorable weather and terrain conditions to enable access to the network. However, damages inflicted by the disaster event can often impede the ability of inspection crews to isolate failures. Flooded roads, debris obstructions, or wet conditions along the infrastructure can delay or completely obstruct travel for ground-based crews as shown in Figure 1-3. For example, after the 2015 Exxon Mobil oil spill, inspectors sent out to the affected area at 11:30 a.m. could not identify the actual leak until 3 p.m., after some oil had already reached the Pacific [43].



(a) Flooded Roads (b) Damaged Roads

Figure 1-3: Limited Access for Inspection Crews Following Hurricane Harvey and Hurricane Ike in Texas [1]

A second challenge for a network agency is to identify regions more susceptible to failures using available data. Network agencies currently acquire information through pre-installed, fixed sensors deployed across the network. However, using these fixed sensors to assign priority levels for inspection has not been fully explored. Furthermore, in order for network

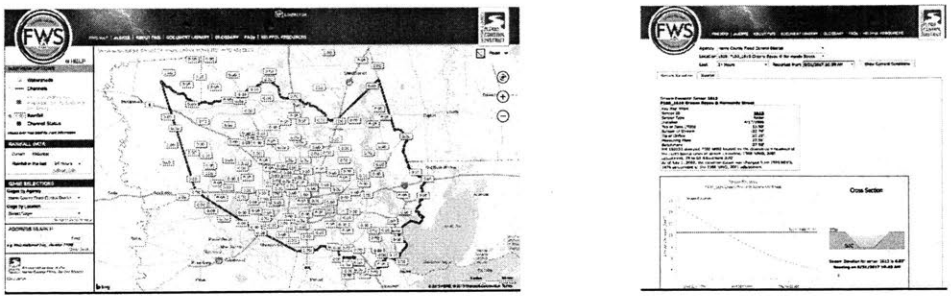
agencies to fully adopt the use of analytics and optimization-driven inspection strategies, they require proof of quantifiable and tangible benefits. Until this proof of concept is presented, current inspection processes will continue to rely largely on experience and limited data. Until recently, the lack of disaster scenario data hampered efforts to provide such a proof of concept. For example, during Tropical Storm Allison in 2001, one drainage network agency captured failure information by hand, and only started maintaining a historical database with Hurricane Ike in 2008. In the past, the lack of standard operating procedures resulted in often subjective reporting and inconsistent data. Although the quality of data has improved in recent years, network agencies still require quantifiable benefits that demonstrate the full potential of integrating available data.

1.2.2 Opportunities

Given the recent emergence of advanced sensors, data-driven approaches have received much attention in transportation research. In this thesis, we focus on integrating data from fixed and aerial sensors into routing problems.

First, we use a fixed sensor network to refine the search area to certain subsets of the network to enable informed, prioritized inspection. For a drainage network, we integrate data from fixed sensors, along with other geomorphological features, to develop a prediction model of failures. Despite the rapid rise in data and the opportunities for improvement through greater efficiency and responsiveness, smart data application for disaster planning and management is still being figured out. More importantly, the potential of leveraging available data to assist in focused inspections in a large-scale infrastructure network has not been fully realized. For example, only recently firefighters leveraged thermal images in wildfires to see through dense smoke and pinpoint hotspots and inform evacuation routes [58]. In the event of a hurricane, emergency responders are starting to use local data analytics to identify the location of vulnerable populations, such as the elderly, in order to inform routing strategies [74]. For instrastructure network inspection, this creates a unique opportunity to explore how to use measurement data from available fixed sensors (e.g., pressure sensors,

flood sensors). For example, Figure 1-4 shows open source flood sensor measurement data, available from a flood control agency website [38].



(a) Flood Sensors (b) Flood Sensor Measurements

Figure 1-4: Open Source Flood Sensor Measurement Data

Second, we integrate UAS to improve accessibility and to identify failures at a fraction of the time required for ground-based inspection crews. For a multi-day inspection, aerial sensors can also provide updates in the form of identified failures; this information is used to revise failure rate beliefs in a subnetwork for subsequent inspection crew routing decisions. Our assertion is that UAS-based inspection can contribute to significant time and cost reductions in post-disaster inspection operations. This is based in part to personal experience as a senior U.S. Army officer leading a team of UAS operators. Originally used for the military, UAS served primarily as Intelligence, Surveillance, and Reconnaissance (ISR) platforms to assess friendly, neutral or adversarial forces. Recent technological advances in UAS have improved its feasibility for many civilian application domains, including remote sensing, humanitarian response, search-and-rescue, product delivery, security and surveillance, and civil infrastructure inspection. The primary benefits of UAS in military applications (i.e., safety, real-time visual access, ability to reach inaccessible regions), are directly transferable to post-disaster infrastructure inspection operations. UAS also permit lower altitude flights to overcome visibility challenges from overhanging vegetation or tree canopies at higher altitudes. The key beneficiaries in this domain are FEMA, US Army Corps of Engineers (USACE), and the infrastructure network agency. While the cost of operating helicopters is estimated to be anywhere from $1,000 to $2,000 per hour, UAS can provide comparable or improved services at $200 to $300 per hour, while also reducing the safety risks to the

inspection crew [53]. UAS can also identify failures in a fraction of the time required for conventional inspection. The total difference in person-hours between conventional inspections (air or ground) compared to UAS inspections implies significant resource cost savings as shown in Table 1.1.

Table 1.1: Comparison of Conventional and UAS Person-Hours of Labor

Required	Bridge Inspection [62]		Census Survey Task [14]		Powerline [21]	
	Aircraft	UAS	Ground	UAS	Ground	UAS
# Workers	7	3	12	2	4	6
Avg Hours	3.5	5.4	4	1.5	40	3
Person-Hrs	24.5	17.3	26	3	160	18

Most importantly, integrating UAS for infrastructure inspection must account for several regulatory and operational factors. The current regulatory environment permits the use of low-cost UAS for disaster response, as evidenced by the recent authorizations granted by the Federal Aviation Administration (FAA) for infrastructure inspections following a recent hurricane [79, 60]. In this thesis, we address several operational considerations such as the design of UAS inspection routes and UAS deployment using ground-based vehicles. We also address regulatory factors such as operating restrictions, infrastructure network stand-off distance, and airspace restrictions.

A point of emphasis is that these two sensors (fixed and aerial) serve two different and distinct roles. Fixed sensors directly provide alerts based on the measured state to refine search regions and if applicable, determine an initial estimate of failures. The role of the UAS is to provide detailed imagery data to identify the location of individual failures.

1.3 Problem Statement and Research Objectives

Motivated by these opportunities, the problem we seek to address is the following: *How to leverage data from sensors, both fixed and mobile, in the design of crew routing strategies to improve the inspection process, subject to sensor, network, and vehicle constraints?*

We are interested in formulating the problem of routing inspection crews to various sub-

networks and exploring these with UAS to isolate failures. We formulate the inspection crew routing problem as two distinct routing problems: the Aerial Sensor Inspection Problem (ASIP) to determine optimal subnetwork inspection routes using aerial sensors, and the Prioritized Inspection Routing Problem (PIRP) to govern the routing of crews to the subnetworks over one or more days. Our overall approach (and our modeling focus) is illustrated in Figure 1-5.

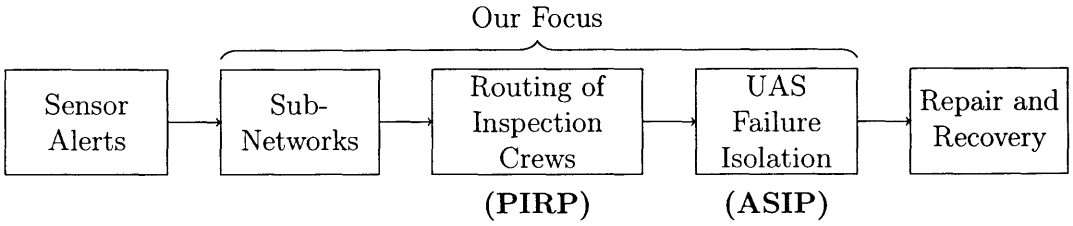


Figure 1-5: Process Flow for UAS-enabled infrastructure inspection

The ASIP can be summarized as follows: Given a subnetwork and a number of homogeneous UAS, determine the optimal routing strategy for each UAS, where the objective is to minimize the maximum amount of time to explore the subnetwork, over all UAS. Each UAS starts and ends its route at a temporary base location within the subnetwork. We consider two use cases that govern the formulation of the ASIP: First, we use the concept of a *monitoring location*, from which the UAS can observe a subset of the network components. Second, we consider a more restricted case where the UAS must traverse every edge of the subnetwork.

The PIRP considers the following: Given a set of service stations and a number of inspection crews and subnetworks, determine the optimal route for each inspection crew, where the objective is to minimize the maximum amount of time elapsed needed to isolate the failures, among all subnetworks. Each crew starts and ends its route at its respective service station. In calculating the total time elapsed, we include the optimal UAS inspection time for each subnetwork, obtained from the ASIP solution. We also consider two use cases that govern the formulation of the PIRP: First, we consider a single day inspection where only a few subnetworks need to be inspected. Second, we consider a multi-day inspection to complete a network-wide inspection, where we use data inputs from fixed and aerial sensors

to prioritize crew routes.

The goals of this thesis are as follows:

1) To design an end-to-end operational approach that integrates UAS and analytics to address the challenges in post-disaster infrastructure network inspections. Our approach accounts for the operating environment as well as vehicle, network, and time constraints.

2) To develop a computationally efficient approach to solve the ASIP. The ASIP formulation accounts for infrastructure inspection requirements, network topology, UAS platform constraints, and sensing requirements.

3) To formulate and solve the PIRP. This formulation combines data-driven methods to enable a prioritized routing strategy to identify failures over one or several days in a timely manner.

4) Determine how diagnostic information from fixed sensors can be used to determine priority regions for inspection, and to predict failures at the beginning of inspection operations.

5) Determine how realized failure data from aerial sensors can be integrated on a daily basis to reduce the diagnostic uncertainty of failure estimates.

1.4 Related Work

Our work is primarily related to two streams of literature: post-disaster response and vehicle routing problems.

In post-disaster response, our work is first related to the extensive literature regarding post-disaster statistical models. While most articles focus on the prediction model itself [37, 57], our work is concerned with the integration of the prediction model into a routing problem. In this manner, our work is relevant to the Learning Enabled Optimization (LEO)

framework suggested by [69], which is a general framework that involves a continuous cycle of data analysis, optimization, and validation. Although a cyclical approach is not possible in our setting due to the lack of longitudinal data, we incorporate a 95% prediction interval in order to forecast possible failure rates in the PIRP in a drainage network. A considerable amount of research effort has been dedicated to obtaining reliable prediction intervals to use for applications such as sales or demand forecasting [32, 42, 6]. We propose using fixed sensor data along with other geomorphological features to narrow the prediction intervals for each subnetwork.

The resilience and restoration of critical infrastructure networks in the aftermath of large-scale disaster events has attracted a high level of interest in the Operations Research and Management Science fields. Of the 4 phases shown in Figure 1-1, there already exists a large body of work on the preparedness phase (e.g., resource allocation), response phase (e.g., search-and-rescue) and recovery phase (e.g., debris clearance) related to Humanitarian Assistance and Disaster Response [4], supply chain networks [71, 26], electrical power systems [81], and transportation infrastructure [27]. In particular, the timely removal of debris to facilitate emergency relief services following large-scale disasters has been a key focus of study in recent years [23, 13, 3]. Table 1.2 summarizes the different domains covered and the approaches used. However, little emphasis has been placed on inspection operations [4, 20] or on quantitative decision-making models that help to bridge the gap between theory and practice [88]. This thesis fills a crucial knowledge gap by showing the results of an analytics-driven computational approach, thereby demonstrating the practical impacts of efficient inspection crew routing strategies. Our work can also be used to provide a higher level of precision for repair times in the final recovery phase, which some papers assume to be stochastic [13].

Survey papers for post-disaster response emphasize the need for stochastic models that explicitly account for information realized over time. Most existing work that does consider stochastic elements employs a two-stage stochastic programming approach [64, 5]. The approach we use to model the realization of failure rates over time is similar to [13] where uncertain debris amounts are realized during clearance activities. By contrast, since our

Table 1.2: Domains of Study in Post-Disaster Response

Domain	Paper	Approach
Power Systems	Guikema et al. [37] Nateghi et al. [57]	Statistical Regression Models
Resource Allocation	Angalakudati et al. [5]	2-Stage Stochastic Programming
Supply Chain	Tomlin, B. [75] Simchi-Levi et al. [70] Qi et al. [67]	Discrete Time Markov Process Linear Programming Continuous Time Markov Process
Debris Removal	Celik et al. [13] Maya Duque et al. [23]	Partially Observable Markov Decision Process Dynamic Programming and Metaheuristic
Transportation	Peeta et al. [64]	2-Stage Stochastic Programming

problem is sequential with updated estimates of failure rates following inspections, we formulate and solve a stochastic dynamic program as opposed to a partially observable Markov decision process model. In addition, while negligible transportation times are relevant for debris clearance problems [7, 13, 12], our inspection approach requires the inclusion of travel times and routing decisions of the inspection crew, along with the inspection times of aerial sensors.

Our formulations for the routing of aerial sensors and inspection crews are variants of classical Vehicle Routing Problem (VRP) formulations. The VRP seeks to find the optimal set of routes for a fleet of vehicles to serve a set of customers [19]. For the use case considering monitoring locations, the ASIP is closely related to the Multi-Trip Vehicle Routing Problem (MTVRP) [72]. Our formulation shares some features of the Green Vehicle Routing Problem (G-VRP) posed by Erdogan et al. [25]. The G-VRP seeks to minimize the total distance traveled by a number of alternative fuel vehicles while visiting a set of important locations that include fueling stations when required. However, in the ASIP, we need to account for the UAS platform and infrastructure monitoring constraints, e.g., climb/descent rates, operating restrictions, and required image resolution. For the use case considering travel along the edges of the subnetwork, the ASIP is most similar to the min max k-Chinese Postman Problem (min max k-CPP) also known as the min max k-Route Inspection Problem described in [24] and [2].

The formulation for the PIRP is closely related to the Multiple Depot Vehicle Routing Problem (MDVRP) [47] and the Team Orienteering Problem (TOP) [35] for a given day of inspection with a limited time budget. What differentiates our work is our offline solution approach to obtain optimal ASIP inspection times, which are used as an input to the PIRP. For the use case where the PIRP is solved over multiple days, accounting for the uncertainty of failure rates, the PIRP is similar to the Dynamic Multi-Period Vehicle Routing Problem (DMPVRP) [85]. The MIP version of the PIRP is considered static in the sense that actual failure rates are revealed over time but routing decisions are made at the beginning of each day. The stochastic dynamic program is related to the Stochastic Vehicle Routing Problem (SVRP) [33, 9] but instead of stochastic demands or travel times, we consider stochasticity in failure rates.

1.5 Thesis Contributions and Outline

The contributions of this thesis include the following:

1) We propose a detailed end-to-end operational approach which captures the specific features of UAS technology (operating range, airspace restrictions, cruise speed) as well as the constraints associated with infrastructure inspections (stand-off distance, image resolution for failure isolation, and network topology constraints).

2) By dèriving a number of theoretical worst-case results, we quantify the time-savings of our approach over ground-based inspections. We find that the magnitude of time-savings depends on the trade-offs that exist between what we call the intra-subnetwork travel time (i.e., within the subnetworks) and the inter-subnetwork travel time (i.e., to and from the subnetworks).

3) We develop a computationally efficient heuristic to solve the ASIP when considering monitoring locations. Solving the ASIP can pose a computational bottleneck if we consider a large number of monitoring locations for a subnetwork. We develop a scalable heuristic

approach that exploits the covering properties of monitoring locations; in particular, the number of components that can be observed from each location. Our heuristic is based on the solution of a weighted minimum set cover problem. We show that our heuristic can solve a 5-subnetwork scenario within an acceptable time frame for real-world implementation (4.15 seconds). Our heuristic achieves an overall average optimality gap of 0.78% when solving the ASIP for 10 different subnetworks of varying size.

4) Using Monte Carlo simulation, we demonstrate the performance of our deterministic ASIP in situations when UAS cruise speeds are stochastic. The stochasticity of cruise speed (and other factors such as airspace restrictions, obstacles, and visibility) can significantly impact the times to explore and inspect the subnetworks. We find that accounting for stochasticity in travel times can significantly increase the overall UAS optimal exploration time and potentially lead to unobserved network components. This points to the importance of choosing the proper UAS platform and conservative route planning to avoid costly setbacks from unidentified failures.

5) We develop quantitative decision-making models that help to bridge the gap between theory and practice. To enable scalable solutions, we present an approach where the ASIP is pre-solved offline for a finite number of possible inspection sequences and used as an input to the PIRP. We formulate two solution approaches for the PIRP. The first is a stochastic dynamic programming approach, which provides an adaptive policy that leverages the failure rate interval of each subnetwork; the second is a certainty-equivalent MIP, which computes a non-adaptive policy that only considers expected failure rates. Both approaches consider updated failure rate estimates at the end of each day based on actual inspection results by the UAS.

6) To determine the value of the integrating data from both types of sensors (fixed and aerial), we show that three quantitative metrics validate the benefits of our approach over the current status quo. Using data from Hurricane Harvey in 2017, we focus on the following metrics: first, the time required to complete inspection of the entire network; second, the cumulative number of failures identified over time; and third, the cumulative value of high-

cost failures identified over time. Since the UAS inspection time accounts for the majority of the time savings with improved speed and accessibility, the first metric serves as a value proposition for the integration of UAS. We show that the integration of UAS leads to a conservative 67% reduction in overall inspection time (from 27 days to 9 days), which can result in cost savings of $360,000 just from personnel expenses alone. The second and third metrics demonstrate the improved rate of failure identification compared to the status quo. For drainage networks, we show that a practical partitioning approach can provide failure identification rates comparable to the results of a stochastic dynamic programming approach.

Our proposed approach is not problem-specific and and can be applied to a wide range of situations that include an array of sensors that detect problems within a region and then use mobile sensors to identify exact locations of failures. One example is earthquake disaster response, where a set of diagnostic seismic sensors can assist in identifying subnetworks for damage inspection for critical infrastructure networks (e.g., gas pipelines). Other applications include treating social media posts or customer calls as sensors for local government agencies to monitor events such as wildfires or accidents. In each of these applications, the procedure in determining the subnetworks using fixed sensors will vary based on the infrastructure and type of data available. Additionally, the formulation of the ASIP and PIRP may differ based on factors such as the network, operating range, airspace restrictions, image resolution required, and network topology [51].

We structure this thesis in the following manner.

In Chapter 2, we introduce the operational requirements and specific features of the infrastructure network, fixed sensors, inspection crews, and UAS platform. We propose the monitoring location concept. The spatial positioning of these monitoring locations depends on various factors such as the UAS operating range, airspace restrictions, infrastructure stand-off distance, required image resolution for failure isolation, and infrastructure network topology. These factors motivate key assumptions in our UAS-enabled infrastructure inspection approach.

In Chapter 3, we first derive theoretical worst-case bounds in order to quantify the

maximum time savings that could be achieved using a UAS-enabled inspection approach. We formulate Mixed Integer Programming (MIP) optimization models to compute the ASIP and PIRP for a single day. To address the computational challenge in solving the ASIP, we develop a scalable heuristic approach based on the weighted minimum set cover. We instantiate our approach on a benchmark pipeline network.

In Chapter 4, we present a priority-based routing approach using predictions of failure rates within subnetworks. This entails incorporating uncertainties in the distribution of failure events into a refined multi-day version of the ASIP and PIRP. We present our results for a stochastic dynamic programming solution and a certainty equivalent MIP formulation, using data from a drainage network inspection in the aftermath of a hurricane. We propose three quantitative metrics to evaluate the benefits of our proposed approach: the total time required to complete inspection, the rate of failures identified over time, and the rate of high-cost failures identified over time.

Chapter 5 presents a summary of our results and provides recommendations for future work.

Chapter 2

Infrastructure Network Inspection

This chapter presents the key requirements and constraints for UAS-enabled infrastructure inspection. In Section 2.1, we start with a description of how to use fixed sensors to partition the network into subnetworks, containing the components that are likely to have experienced failure. In Section 2.2, we describe the generic requirements to route inspection crews, equipped with UAS, to these subnetworks. We discuss characteristics of UAS in 2.3 and address several regulatory and operational factors to consider in Section 2.4. Finally, we present our UAS-enabled infrastructure inspection approach in Section 2.5.

Recall that fixed and aerial sensors serve two different roles. The role of fixed sensors is to provide alerts based on the measured state to refine search regions. The limitation of fixed sensors is that it only offers a coarse idea of the actual failure location, requiring a higher level of detailed inspection. This role is satisfied by the aerial sensor, which can provide detailed imagery data to identify individual failures. We further discuss each of these two types of sensors in the following sections.

2.1 Fixed Sensor-Based Subnetworks

Consider an infrastructure network that is composed of a set of nodes and edges denoted as $(\mathcal{N}, \mathcal{E})$, representing the intersections and the set of components, respectively. The network components are prone to *failure events* that can be random (e.g., pipe bursts) or correlated (e.g., earthquake induced failures).

The infrastructure network is monitored by an agency using a Supervisory Control and Data Acquisition (SCADA) system, which routinely collects data from remote fixed sensors that are pre-installed at certain network locations. When a failure event occurs, a sensor is capable of detecting the resulting fluctuations in its measured state (e.g., local pressure, elevation level, or flow). The sensors are capable of sending alerts to the SCADA system (either directly or using hop-to-hop communication). For cases when the fluctuations are not directly detectable by a fixed sensor, we consider that the agency can be still alerted by some other means, e.g., customer calls or social media data. Thus, for our purpose, any information that helps the agency to identify the area in which one or more failure events are likely to have occurred, counts as a sensor alert. Note that the number of sensors is often limited, thus the sensor alerts cannot be used to perfectly isolate the individual failures in large-scale networks. Furthermore, the sensors' capability to detect fluctuations is constrained by their detection range. Therefore, in most failure situations, the agency can only map each failure event to a certain spatial zone, which we call a *subnetwork* [68]. The subnetworks are dependent on the type of infrastructure and type of fixed sensor. We focus on two instantiations of subnetworks, corresponding to a pipeline and drainage network.

Pipeline Subnetworks. For a pipeline network, when a failure of a network component occurs, a pressure wave spreads through the network and alters the steady state pressure distribution. A failure event can be detected by monitoring the pressure before the signal dissipates, i.e., within the detection range. For example, we can assume that each sensor provides a binary output, i.e., 1 if it detects the failure event, 0 otherwise. If we assume that each network component failure can be detected by at least one fixed sensor, we can deduce the set of network components monitored from each sensor location based on the

shortest distance [31]. This set of components corresponding to a sensor alert make up a high-priority subnetwork. Figure 2-1 illustrates a network with a set of 64 fixed sensors and 5 high-priority subnetworks.

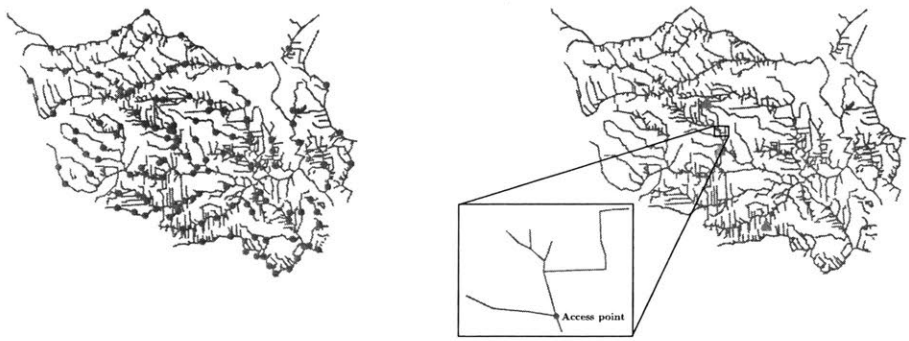
(a) Network with fixed sensors (circles) and 2 service stations (triangles)

(b) 5 high-priority subnetworks arising from fixed sensor alerts

Figure 2-1: Illustration of a fixed sensor network and subnetworks in a pipeline infrastructure network

Drainage Subnetworks. To assist in its role of inspection and maintenance, the drainage network agency installs *flood sensors* at certain locations along the drainage network to monitor water elevation levels (in feet). These sensors routinely transmit elevation data to the service station for daily monitoring by the agency. When the elevation level exceeds a certain pre-determined threshold, designated as the *Top of the Bank (TOB)*, the agency issues flood alerts in the vicinity of the flood sensor. Flood sensor elevation data is publicly available [39]. We associate a set of network components upstream of each flood sensor as a *subnetwork*. We connect one subnetwork per flood sensor for two reasons: first, we want to exploit the value to be gained from each carefully calibrated flood sensor, and second, the flood sensors are generally distributed across the drainage network in a uniform manner as shown in Figure 2-2a. For our purpose, a subnetwork is a connected subtree, where each edge of the subtree lies along the shortest path to the nearest downstream flood sensor. We use the common Depth-First-Search (DFS) algorithm to determine these shortest paths. We denote the total number of flood sensors as K and therefore let $\mathcal{T}_1, \ldots, \mathcal{T}_K \subseteq \mathcal{E}$ denote the K subnetworks associated with each flood sensor. Using this approach, we achieve a perfect partitioning of the network. If a flood alert occurs (i.e., the water elevation exceeds the TOB

during a time period of interest), we classify the corresponding subnetwork as high-priority. Otherwise, the subnetwork is classified as low-priority. We denote a high- and low-priority subnetwork as $\mathcal{T}_k^h$ and $\mathcal{T}_k^l$, respectively. Using the elevations recorded during the period of Hurricane Harvey (August 26, 2017 to September 1, 2017), we partition the network into the high- and low-priority subnetworks as shown in Figure 2-2b.



(a) Flood Sensors (b) Subnetworks

Figure 2-2: Flood sensors are depicted on the left and high-priority subnetworks are shown in red on the right. We highlight one example subnetwork.

For both types of networks, we assume that the alerts from fixed sensors correspond to a collection of high-priority subnetworks, where each subnetwork is comprised of network components that are likely to have undergone (or are prone to) failure. In our approach, the exact number and location of failures in each high-priority subnetwork is unknown, and therefore, isolating these failures requires additional inspection.

Motivated by practical considerations [46], we assume that the agency collects the sensor alerts for a predetermined time interval. Suppose time 0 denotes the time when the disaster event strikes, and time t_0 represents the first day of favorable weather and channel conditions to enable access to the infrastructure network. Based on the alerts received during $[0, t_0]$, let $\mathcal{T}_1, \ldots, \mathcal{T}_K \subseteq \mathcal{E}$ denote the K subnetworks that need to be further inspected by UAS. The agency assigns the available inspection crews to the the K subnetworks at time t_0. Therefore, at the start of the inspection process, each subnetwork is assigned to a service station.

2.2 Inspection Crews

Currently, network agencies do not use the notion of subnetworks, specified by sensor outputs as described previously. Instead, agencies assign general regions to inspect at the start of each day based primarily on experience, geographic proximity, contiguity of the network, and accessibility. These regions are assigned to a set of inspection crews originating from one or more service stations. Furthermore, infrastructure network agencies have only started to experiment using aerial sensors such as UAS in a limited role, such as providing images for construction projects. In our approach, we assume that inspection crews each carry a number of UAS that will allow timely inspection of the subnetworks. We let U denote the total number of UAS with each inspection crew.

Let $\mathcal{Y}$ denote the set of service stations. Due to excessive setup costs, we will consider a small number of immutable service stations. For every service station $s \in \mathcal{Y}$, we denote n_s the number of inspection crews that are available at station s. Each inspection crew, starting from a service station, can visit one or more subnetworks prior to returning to the same station. To visit a subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, a crew needs to first set up a temporary base, which we denote as b_k. In practice, b_k can be chosen as the centroid of $\mathcal{T}_k$, or access points, which are authorized entry points from a road to inspect network components. Let $\mathcal{B} := \{b_k,\ k \in [\![1, K]\!]\}$ denote the set of all temporary bases. Setting up a temporary base involves unpacking the UAS and performing pre-flight checks. We also assume that the inspection crew remains stationary at the temporary base until retrieval of the UAS and can observe the live video feed of the UAS to identify failures. The crew completes a visit to the subnetwork upon retrieval of the UAS.

For the purpose of crew route planning, the pairwise travel time between service stations and temporary bases can be assembled into a travel time matrix which we denote $\mathbf{\Gamma}$. This consists of inspection crew travel times between every pair of locations (k, l) in the set $\mathcal{Y} \cup \mathcal{B}$. We let γ_{kl} denote the time needed by an inspection crew to travel from k to l. One can obtain γ_{kl} by determining the shortest path between k and l in the transportation network and dividing by the empirical average vehicle speed. The temporary base set up time can

also be included in γ_{kl}. For convenience, we assume that the time to refuel a crew vehicle is negligible.

We define an *inspection crew tour* as a sequence of visits to a subset of $\mathcal{B}$, that starts and ends its tour at the same service station $s \in \mathcal{Y}$. Note that we do not allow a inspection crew to return to another service station due to accountability and unbalanced workload issues that might arise. Figure 2-3 illustrates 3 inspection crew tours to 5 different temporary bases corresponding to the subnetworks from Figure 2-1b.

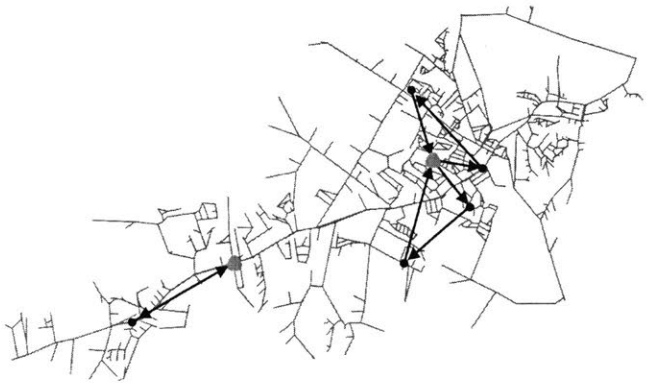


Figure 2-3: Illustration of inspection crew tours. Starting from a service station, each crew visits one or more temporary bases (black dots) and returns to its station (triangles)

2.3 Airborne Sensors

In a military context, unmanned aerial systems (UAS) operate as the eyes of the Commander to "see first, understand first, and act first, decisively" [40]. As UAS technology and its safety record continues to improve, there is widespread interest for its use in commercial applications such as aerial photography, surveillance of land and crops, monitoring of forest fires and environmental conditions, and the recent protection of borders and ports with the Department of Homeland Security. Recent studies have shown that computer-vision based methods to extract features of failures using automated image processing is also possible [83].

For the purpose of infrastructure network inspection, a variety of commercially available

rotary-wing UAS platforms can be employed; however, it would be impractical to individually model each one of them. Using data from the Association for Unmanned Vehicle Systems International (AUVSI) air platform database, we define four representative classes of UAS based on their Maximum Gross Take-Off Weight (MGTOW) similar to [84]. These representative UAS sufficiently represent Commercial Off-The-Shelf (COTS) platforms ranging from the smaller rotary-wing DJI Mavic Pro as shown in Figure 2-4a, to the larger Aeryon Scout. The representative UAS classes are described in Table 2.1 using MGTOW (kg), mean cruise airspeed (knots), max airspeed (knots), descent and climb rates (meters per second), and endurance (minutes).

Table 2.1: Representative UAS

Class ID	1	2	3	4
Notional MGTOW (kg)	[0,2)	[2,5)	[5,9)	[9,25]
Mean Cruise Airspeed (kn)	25	20	30	60
Max Airspeed (kn)	40	30	60	100
Descent Rate (m/s)	-1.5	-2.5	-2.5	-5.0
Climb Rate (m/s)	2.5	3.5	3.5	5.0
Endurance (min)	30	45	45	60

If using a UAS class with limited endurance, we allow the UAS to return to the temporary base to replace their batteries before exploring other parts of the subnetwork. Each UAS requires a deterministic time to replace its battery; we denote this as τ_{batt}. All UAS are fully charged for the initial dispatch, and if an inspection crew visits more than one subnetwork, we can reasonably assume that UAS receive fully charged batteries while enroute to the other locations.

In addition to these UAS classes, we also consider hybrid models (mixture of gas and electric) of UAS with an endurance of up to 5 hours as shown in Figure 2-4b. Coupled with the approval of Beyond Visual Line-of-Sight (BVLOS) operations, this type of UAS can cover greater distances, improving the cost effectiveness and removing the need for multiple trips [8]. We consider this type of hybrid UAS for the ASIP use case requiring traversal of every edge. In this case, the total edge distances of the subnetworks fall well below the maximum range of a hybrid UAS (100 miles).

(a) Example of Shorter Endurance UAS: DJI Mavic Pro

(b) Example of Longer Endurance Hybrid UAS: Skyfront Perimeter

Figure 2-4: Examples of Unmanned Aerial Systems

In our ASIP formulation, the mean cruise airspeed, denoted $\overline{V}_C$, is assumed to be deterministic and defined as the speed at which the UAS should operate to maintain optimum performance [84]. Max airspeed, denoted V_{max} is the maximum permitted speed. The descent (resp., climb) rate is the vertical speed of the UAS, or the rate of negative (resp., positive) altitude change with respect to time. The endurance, denoted as τ_{max}, is defined as the maximum length of *time* that a UAS spends in flight. For rotary-wing UAS with MGTOW under 9 kgs, the advertised endurance values of different UAS models exhibit relatively less variability; see Figure 2-5. The endurance values shown in Table 2.1 are based on the *average* advertised endurance rate for a given class. In practice, the realized endurance depends on a variety of factors, including payload (which is affected by sensor weight), battery age, operating environment, etc. However, we ignore these complications and assume a deterministic τ_{max}.

Finally, although communication is an important consideration for safe navigation and connectivity of UAS, we do not explicitly consider the impacts of unreliable or insecure communication links between the UAS and the operator. Commonly used communication links for UAS operations are: the uplink control, downlink telemetry, and downlink payload communications; these links operate at frequencies dictated by the Federal Communications Commission (FCC). Loss of communication in any of these three links can occur due to loss of line-of-sight or interference from the environment or adversary. Indeed, cyber-security risks have been recently identified as an important barrier to employing UAS for monitoring

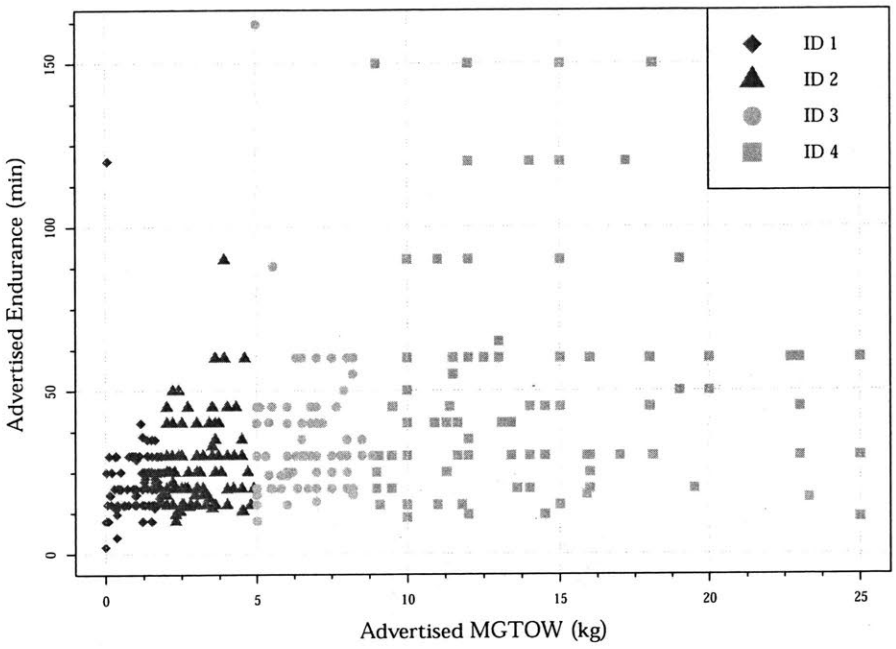


Figure 2-5: The Advertised MGTOW and Endurance for 404 Rotary Wing UAS Platforms

strategic areas [45]. However, given that our focus in this thesis is on establishing a static approach for joint routing of inspection crews and UAS-based inspection of physical infrastructures, we do not consider (low-level) communication aspects that are inherently dynamic in nature.

2.4 Inspection Requirements

In this section we discuss the five main factors governing the positioning and routing of UAS for the purpose of failure isolation. These factors motivate the key assumptions in our UAS-enabled infrastructure monitoring approach.

The physical network can either be above ground or underground depending on the type of infrastructure. For example, an open channel drainage network can be observed from above ground whereas a pipeline network can have portions laid underground. With underground

portions, in many cases, failure events are detectable from above ground [56]. For example, hydrocarbon leaks in an underground gas network can be detected above ground using gas detectors. Other above ground activities that are often main causes of failures include: the presence of unauthorized digging, excavation by third parties, or soil erosion. We include such definitive precursors of actual failures in our definition of failure events.

Once an inspection crew sets up a temporary base, up to U UAS are launched to explore the corresponding subnetwork. For simplicity, we assume that UAS have identical sensing capabilities and technical characteristics (e.g., cruise speed). In our approach, the UAS completes inspection of a subnetwork by either 1) visiting a number of vantage points with a clear and unobstructed view to one or more network components, or 2) traversing each edge, which can also be viewed as visiting a series of several vantage points directly over the network components. For both inspection methods, the spatial positioning of the UAS should account for several factors:

(i) Operating Range

(ii) Airspace Restrictions

(iii) Infrastructure Stand-off Distance

(iv) Required Image Resolution for Failure Isolation

(v) Infrastructure Network Topology

Figure 2-6 illustrates the abovementioned factors influencing the spatial positioning of these vantage points, which we henceforth refer to as *monitoring locations*. We first briefly discuss each of the five factors and provide a formal definition of monitoring locations thereafter.

(i) *Operating Range.* The maximum operating range is the maximum *distance* from a temporary base that the UAS is capable of flying on a round trip mission. We can estimate this operating range with $\frac{\overline{V}_C \cdot \tau_{max}}{2}$, where $\overline{V}_C$ is the mean cruise speed and τ_{max} is the endurance of the UAS. The operating range is also restricted by Visual Line-of-Sight (VLOS)

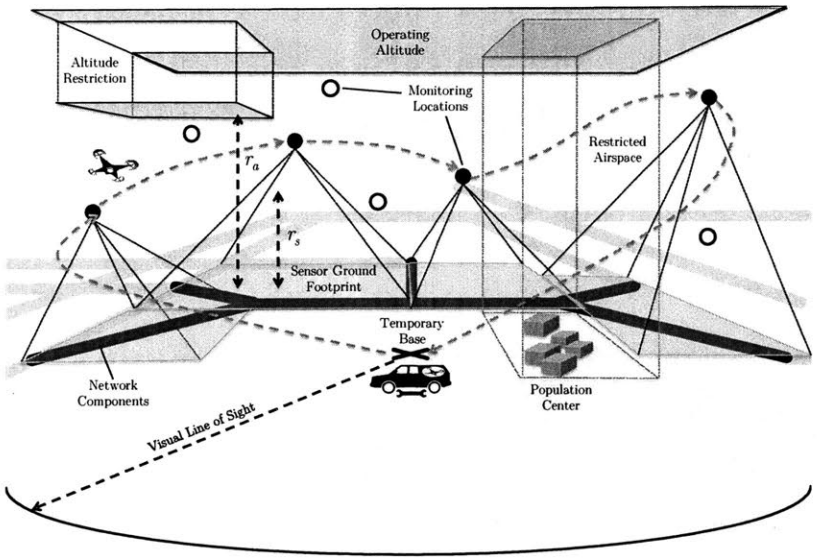


Figure 2-6: Illustration of monitoring locations for a subnetwork. The available monitoring locations are shown in filled and empty circles and the infrastructure components within the subnetwork are highlighted with thick lines

requirements, defined by the FAA as keeping unaided visual contact with the UAS in order to *"maintain safe operational control of the aircraft, know its location, and be able to scan the airspace in which it is operating to see and avoid other air traffic or objects aloft or on the ground"* [28]. To the best of our knowledge, the current literature does not suggest a common quantitative definition of the VLOS range. We came across conservative VLOS ranges of 930 m [28] to a maximum theoretical range of 1050 m [86]. In practice, VLOS can also vary significantly with local weather and other environmental conditions. We argue that for the purpose of infrastructure monitoring, the typical size (maximum radius) of subnetworks is smaller than a conservative VLOS estimate if using UAS with limited endurance. If using a hybrid UAS with BVLOS approval, the positioning of monitoring locations is not constrained by the VLOS requirement [45].

(ii) *Airspace Restrictions.* All monitoring locations are upper bounded at an altitude restriction, denoted r_a, which is typically 122m AGL (Above Ground Level) to align with the current FAA Part 107 UAS regulations [29]. One may also need to consider restricted airspace, which can either be temporary or permanent. Temporary flight restrictions can be

enforced due to hazardous conditions (e.g., a wildfire) or routine events (e.g., stadium event). Examples of permanent flight restrictions include airspace in close proximity to population centers, military operation areas, or airports.

(iii) *Infrastructure Stand-off Distance.* Each monitoring location must also comply with the minimum stand-off distance, denoted r_s, to the infrastructure network components or other ground obstacles (e.g., power lines or buildings). Since misjudgment of distance and speed is a significant flight hazard, stand-off distance provides a safe buffer zone during flight. Given environmental uncertainty, wind gusts, and UAS platform instability, we assume that the UAS will operate outside a 30m stand-off distance. This can be viewed as a conservative estimate based on current best practices as described in [53]. Combined with the restriction imposed by the stand-off distance and altitude restriction, we maintain that any feasible vertical distance between a monitoring location and ground level, denoted R, is constrained as:

$$r_s \leq R \leq r_a \tag{2.1}$$

(iv) *Required Image Resolution for Failure Isolation.* When camera sensors are used for the identification of failures, the spatial positioning of monitoring locations also depends on the required image resolution. To get an idea of the resolution, one can estimate the Ground Sampling Distance (GSD), defined as the distance between two consecutive pixel centers measured on the ground [50]. For example, a GSD of 10 cm can be interpreted as one image pixel representing 10 cm on the ground. Thus, a higher GSD corresponds to lower spatial resolution. The GSD can be estimated using the following equation:

$$GSD = \frac{xR}{f \cos \alpha}, \tag{2.2}$$

where x is the length of the sensor's pixel size (mm), f is the focal length of the camera's lens (mm), R is the vertical distance (m) between the camera (or monitoring location) and ground level, and α is the look angle. Thus, all else equal, a higher altitude R will correspond to a higher GSD value. A sensor's *ground footprint* is defined as the total projection of a

sensor's pixels onto the ground; see Figure 2-7 for an illustration of the GSD and sensor ground footprint.

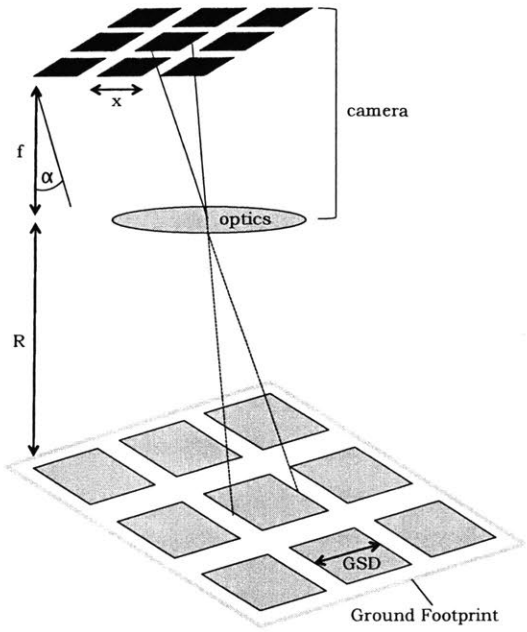


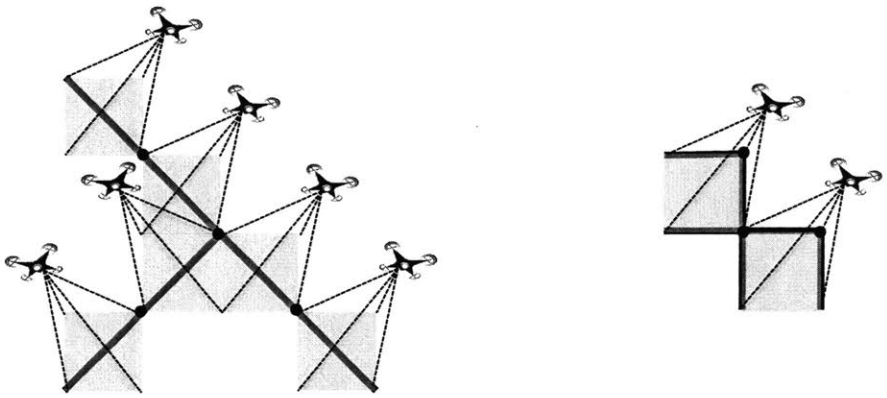
Figure 2-7: Ground Sampling Distance and Sensor Ground Footprint

It is important to note that different failure types may require different GSDs for identification. Based on Eqs. (2.1) and the (2.2), one can check if the on-board camera on the UAS is adequate for the inspection task.[1] In the context of gas pipelines, two types of failure events are of interest based on the type of damage: structural damages and full component disruptions. For structural damage (e.g., leaks), it is critical to achieve high resolution images (i.e., smaller GSD) in order to identify small hairline fractures (a few mm in length). To identify such failures, the monitoring locations need to be positioned at lower altitudes. This would also entail a higher number of monitoring locations to fully explore the subnetwork. On the other hand, for disruptions, such as pipeline bursts, major gas leaks, or fire emergencies, a lower resolution can meet the requirements for failure isolation. In

[1]As an example, for a Sony QX-10 camera with a 1/2.3 inch sensor (6.2 mm by 4.6 mm) that can take pictures of up to 4,896 by 3,672 pixels, the size of each pixel would be 0.0012 mm by 0.0012 mm. With a focal length of 25 mm, determining the altitude to fly the UAS to resolve a 0.5 cm feature on the ground would require a simple rearranging of terms in Eq. (2.2) to solve for R. Assuming a nadir (overhead) aerial view ($\alpha = 0$), the UAS would visit monitoring locations at a height of 100 m which also satisfies Eq. (2.1).

this case, UAS can operate at higher altitudes (with higher GSD), and consequently visit a smaller number of monitoring locations. In the context of a drainage network, lower altitude flights along the water channels are required due to visibility challenges from overhanging vegetation or tree canopies.

(v) *Infrastructure Network Topology.* Finally, the number of monitoring locations also depends on the network topology within the subnetwork. For example, given the same number of network components and ground footprint size, a tree network topology would likely require more monitoring locations to explore the entire subnetwork in comparison to a grid topology; see Figure 2-8.



(a) A network with a tree topology (b) A network with a grid topology

Figure 2-8: For the tree topology (left), 6 network components (black line segments) would require 6 monitoring locations with associated ground footprints (grey). For the grid network (right), only 2 monitoring locations are needed for the same number of network components.

Considering factors (i)-(v), we are now in a position to formally define *monitoring locations* and *monitoring sets.* Each monitoring location provides a vantage point for the UAS to observe some of the network components while considering the requirements for each of the five factors discussed. For every subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, we let $\mathcal{V}_k$ denote the set of monitoring locations that the UAS can visit. Without loss of generality, we assume that $b_k \in \mathcal{V}_k$, that is, the temporary base belongs in this set.

From each monitoring location $i \in \mathcal{V}_k$, we define a *monitoring set*, $\mathcal{C}_i^k \subseteq \mathcal{T}_k$, as the subset

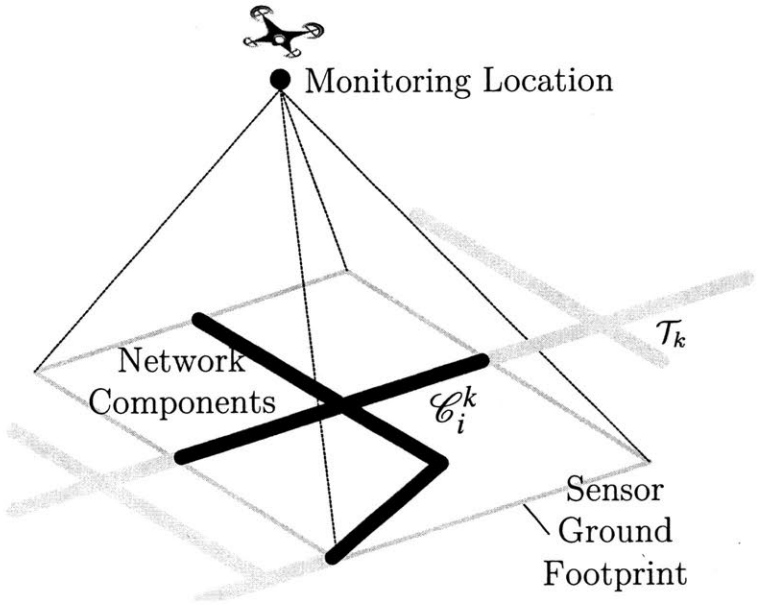


Figure 2-9: The Monitoring Set corresponding to a particular monitoring location is obtained from the sensor ground footprint, which is based on the GSD calculation

of network components that a UAS is capable of monitoring (and isolating). Correspondingly, for every network component $e \in \mathcal{T}_k$, let $\mathcal{V}_k(e)$ denote the subset of monitoring locations from where a UAS can monitor e. We assume that $\mathcal{V}_k$ fully "covers" $\mathcal{T}_k$, i.e., all components in $\mathcal{T}_k$ can be monitored by visiting a subset of $\mathcal{V}_k$. Figure 2-9 illustrates how a monitoring set is obtained from the sensor's ground footprint.[2] We emphasize that our setup provides us with the flexibility to consider different types of monitoring sets for each monitoring location, depending on the five factors.

For each subnetwork $\mathcal{T}_k$, a *UAS tour* is then defined as a sequence of visits to a subset of $\mathcal{V}_k$ that starts and ends at the temporary base b_k. Upon returning to base, the UAS will either replace its battery for additional tours, or complete its mission. Analogous to the travel time matrix for inspection crews, we can define another travel time matrix for each subnetwork $\mathcal{T}_k$; the elements of this matrix are the pairwise travel times between monitoring

[2]To provide an example of a monitoring set, we turn to the ground footprint from our above GSD calculation. The sensor ground footprint in this case would be 24.48 m by 18.36 m, acquired by multiplying the total sensor pixel size (4,896 by 3,672 pixels) by 0.5 cm. Recall that 0.5 cm was the length of the feature to resolve. Therefore, from a monitoring location $i \in \mathcal{V}_k$ at a height of 100 m, we can include all network components within this ground footprint as part of the monitoring set $\mathcal{C}_i^k$, given that there are no obstacles.

locations within $\mathcal{V}_k$. We denote this matrix as $\boldsymbol{T}^k$. Thus, for every ordered pair of locations $(i,j) \in \mathcal{V}_k^2$, let τ_{ij}^k denote the UAS travel time from i to j. One can obtain τ_{ij}^k by determining the shortest path distance from i to j and dividing by the mean UAS cruise speed, which also incorporates the climb or descent rate. We do not necessarily impose $\tau_{ij}^k = \tau_{ji}^k$. The observation time at each monitoring location can also be incorporated into $\boldsymbol{T}^k$.

By visiting a subset of monitoring locations $\mathcal{V}_k$ such that each network component is monitored at least once, we claim that UAS will be capable of decisively isolating the failures in the corresponding subnetwork. This can be achieved if, for example, a trained observer is inspecting the live video feed to provide near real time feedback. Alternatively, state-of-the-art software can be employed to provide rapid automated image processing (i.e., identify failures from the live video feed) with high accuracy. Computer-vision based methods to extract features from images can achieve accuracy levels of 90-95% for certain types of failures [55], and the expanding use of unmanned systems in the future will only increase the amount of training data required to increase accuracy.

Before proceeding further, we summarize the key assumptions that we introduced in Sections 2.1 through 2.4:

A1 Each failure alert obtained by the agency from fixed sensors can be mapped to a subnetwork, which contains the set of network components that need to be inspected in order to isolate failure events. The number and location of failures in each subnetwork is unknown.

A2 Based on fixed sensor alerts (and associated subnetworks) received in the time interval $[0, t_0]$, the agency allocates and dispatches inspection crews at time t_0.

A3 If we do not consider BVLOS operations, the size of the subnetworks for UAS-based inspection is no greater than the VLOS in radius.

A4 The UAS can monitor all network components in the subnetwork $\mathcal{T}_k$ by either visiting a subset of monitoring locations $\mathcal{V}_k$, or by traversing along every edge.

A5 All UAS have identical sensing and technical capabilities. Likewise, all inspection crews have homogeneous capabilities.

A6 The travel times for both the inspection crews and UAS are assumed to be deterministic.

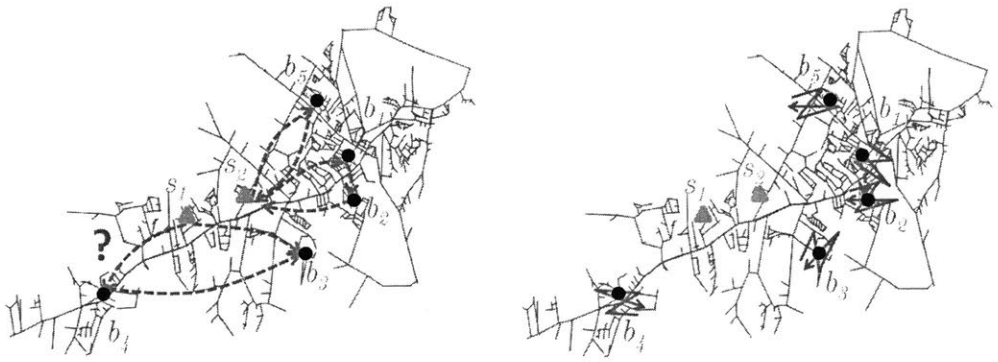
A7 The UAS endurance time as well as the time to replace the on-board battery are assumed to be deterministic.

A8 The communication link between the UAS and operator is secure and reliable, and does not impose any constraint on the route planning of inspection crews and UAS.

2.5 Proposed Approach

Considering the subnetworks that arise from failure alerts, our focus is to study how joint optimization of UAS and inspection crew route plans can create efficiency and timely detection. To do so, we propose an approach that (i) solves the PIRP, which optimally dispatches inspection crews to the subnetworks to minimize the worst-case time to inspect, and (ii) solves the ASIP, which optimally routes the UAS to isolate failure locations within a given subnetwork. These two problems are nested in that the optimal value of the ASIP is taken as an input in solving the PIRP; see the illustration in Figure 2-10.

Recall that we model our problems based on failures that arrive during the time period $[0, t_0]$, i.e., the decision for vehicle dispatches is made in a batch. The failure alert data is processed and mission planning is completed in preparation of inspection crew dispatch from each service station at time t_0. For each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, we denote θ_k the amount of time during which $\mathcal{T}_k$ was alerted prior to the dispatch of the inspection crews. Note that within a given subnetwork, there can be several alerts corresponding to the failure of different components at different times. However, we are only concerned with the largest such time, i.e., $\forall k \in [\![1, K]\!]$, θ_k is determined by the first alert that is received in $\mathcal{T}_k$. Note that high values of θ_k can occur when the interval $[0, t_0]$ is large. This may happen in situations when



(a) PIRP: Each inspection crew can visit one or more subnetworks. At the end of each day, the crew returns to its assigned service station.

(b) ASIP: From the temporary base location, the UAS isolate the failures by visiting a subset of monitoring locations or by traversing each edge.

Figure 2-10: Illustration of the PIRP and ASIP

the inspection crews are not readily available or when inspection crews are engaged in other jobs and are not positioned at the service station [5]. The timeline of various phases of inspection operations is illustrated in Figure 2-11. An inspection crew is dispatched from its service station s at time t_0 and completes the set up of a temporary UAS base at location b_k at time $t_0+\gamma_{sk}$. The optimal UAS inspection time for $\mathcal{T}_k$ is denoted ξ_k^*. For this example, the total time elapsed from failure alert to the completion of inspection for $\mathcal{T}_k$, denoted $t_{total,k}$, is equal to $\theta_k + \gamma_{sk} + \xi_k^*$. We can see how the optimal value of the ASIP is embedded within the PIRP, which seeks to minimize the maximum of $t_{total,k}$ over all K subnetworks.

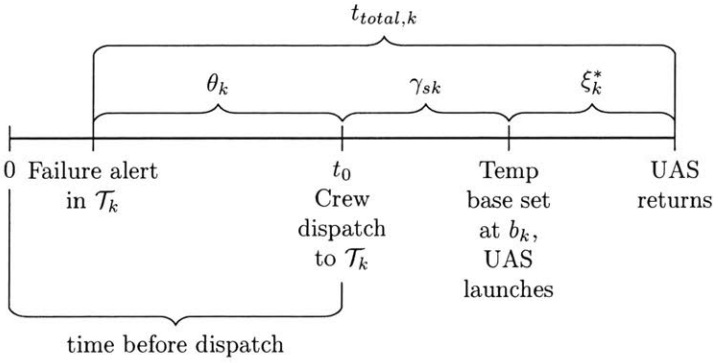


Figure 2-11: Example Timeline from failure alert for a single day in the case of a single inspection crew, single subnetwork, and single UAS

Chapter 3

Network Inspection with Aerial Sensors

In this chapter, our focus is on how to integrate aerial sensors (i.e., UAS) into inspection operations. Specifically, the problem statement we seek to address is the following: *How to optimally route UAS for the localization of failures within alerted regions (subnetworks) with the goal of: minimizing the maximum inspection time, subject to constraints due to the operating environment and sensing requirements?* In Section 3.1 we first provide the results of a worst-case analysis using comparisons between classical vehicle routing problems and adapting their solutions to our setting. In Section 3.2 we provide a Mixed Integer Programming (MIP) formulation that provides optimal multi-trip UAS routes for inspecting a subnetwork. To address the computational challenges of implementing this MIP formulation on a realistic network size, we propose a set-cover based heuristic. Finally, we show the results of a computational study using our overall proposed approach on a benchmark pipeline network in Section 3.4.

3.1 Maximum Value of Aerial Sensor-based Inspection

In order to study the advantages of using UAS over ground-based inspection, we derive a number of theoretical worst-case results. Specifically, we are interested in quantifying the

time savings that could be achieved using our UAS-enabled inspection approach, compared to the current ground-based approach. We find that the results depend on the trade-offs that exist between what we call the intra-subnetwork travel time (i.e., within the subnetworks) and the inter-subnetwork travel time (i.e., to and from the subnetworks).

In [80], the authors derive worst-case results for a problem called the Vehicle Routing Problem with Drones (VRPD), where a set of customers demanding parcels are served by either a truck with limited cargo capacity or a drone with the capacity to carry one parcel at a time. Our problem differs in that the PIRP is an inspection problem with no vehicle capacity constraints, the UAS can inspect several network components, and we do not allow crews to inspect network components in parallel with the UAS.

For this section, we focus on the use case considering monitoring locations and limited endurance. We refer to a ground-based inspection crew as a vehicle and a temporary base location within a subnetwork as a base. For simplicity, we make the following assumptions:

A1. For any given subnetwork, we assume that the UAS endurance is sufficient to reach all of the monitoring locations with one or more trips.

A2. The speed of the UAS is α times the speed of the vehicle (which is set to 1). We assume that $\alpha \geq 1$.

A3. We do not consider altitude for monitoring locations in this study (i.e., ground-based crews can also visit monitoring locations to complete inspections).

A4. For ground-based inspections, the vehicle travels at the same speed for both intra and inter-subnetwork travel.

The last assumption implies conservative results since intra-subnetwork travel for ground-based inspections is typically slower than inter-subnetwork travel. Recall that according to our UAS-enabled inspection approach, the UAS begins subnetwork inspection only after the vehicle sets up at a base. The vehicle remains idle at the base until UAS inspection is complete. We use the following notation for this section:

- $Z_k(TSP)$: The optimal value of the Traveling Salesman Problem for subnetwork k, which finds the shortest possible route for a vehicle that starts at a base and visits each monitoring location, prior to returning to the base. We denote the value of a feasible TSP solution as $Z_k^f(TSP)$.

- $Z(VRP_n)$: The optimal value of the Vehicle Routing Problem with n ground vehicles, which minimizes the total travel time required to visit a set of temporary bases, prior to returning to the service station. For simplicity, we only assume one service station for this study.

- $Z_k(ASIP_{u,\alpha})$: The optimal value of the ASIP for subnetwork k where u is the number of UAS available for each vehicle and the speed of the UAS is α times the speed of the vehicle. We denote the value of a feasible ASIP solution as $Z_k^f(ASIP_{u,\alpha})$.

- $Z(PIRP_{n,u,K})$: The optimal value of the PIRP with n vehicles, u UAS and K subnetworks.

- $Z(SQ_{n,K})$: The optimal value of the status quo inspection approach with n vehicles and K subnetworks.

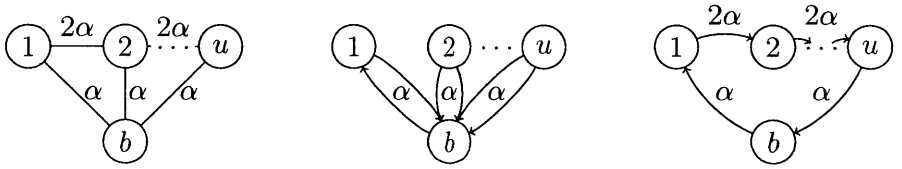
Since $PIRP_{n,u,K}$ employs UAS with speeds that are α times the speed of the vehicle, where $\alpha \geq 1$, we expect that $Z(PIRP_{n,u,K}) \leq Z(SQ_{n,K})$. Our goal is to seek to find a tight upper bound on the ratio $\frac{Z(SQ_{n,K})}{Z(PIRP_{n,u,K})}$. We gradually build up to this ratio by first presenting our results from the viewpoint of a single subnetwork. We then define the worst case ratio for a simple case, with one vehicle, u UAS, and one subnetwork. Finally, we use both of these results to generalize for K subnetworks.

We first present the results for a single subnetwork. For the sake of brevity, we leave out the subnetwork index, using $Z(ASIP_{u,\alpha})$ instead of $Z_k(ASIP_{u,\alpha})$, and $Z(TSP)$ instead of $Z_k(TSP)$.

Lemma 1. $\dfrac{Z(TSP)}{Z(ASIP_{u,\alpha})} \leq u\alpha$

Proof. We first start with the optimal $ASIP_{u,\alpha}$ solution. For simplicity, we assume that all of the monitoring locations must be visited to fully inspect the subnetwork. An optimal $ASIP_{u,\alpha}$ solution will consist of u UAS routes (consisting of trips from the base to each monitoring location and back). Next, we construct a feasible TSP solution (vehicle inspection route) by combining the u UAS routes into one large route. Finally, we multiply by α since we the UAS speed is α times the speed of the vehicle. Thus we have $\frac{Z(TSP)}{Z(ASIP_{u,\alpha})} \leq \frac{Z^f(TSP)}{Z(ASIP_{u,\alpha})} \leq u\alpha$.

We show that this bound is tight by considering an example shown in Figure 3-1 with u monitoring locations (not including the base). All of the edge distances are set to α with the exception of adjacent edges between the monitoring locations, which are set to 2α.



(a) Example problem (b) Feasible $ASIP_{u,\alpha}$ (c) Optimal TSP

Figure 3-1: Example for $ASIP_{u,\alpha}$

Recall that the vehicle travels at a speed of 1 while the UAS travels at speed α. By inspection, the optimal TSP solution will visit the monitoring locations in sequence for a total time of $2u\alpha$. A feasible $ASIP_{u,\alpha}$ solution has the UAS traveling individually to each monitoring location and back. Since the ASIP is a min-max problem, and each UAS takes a total time of 2, $Z(ASIP_{u,\alpha}) = 2$. Therefore, we have:

$$\frac{Z(TSP)}{Z(ASIP_{u,\alpha})} \geq \frac{Z(TSP)}{Z^f(ASIP_{u,\alpha})} = \frac{2u\alpha}{2} = u\alpha \tag{3.1}$$

□

It is not too difficult to show that at best, $Z(TSP) \leq \alpha\, Z(ASIP_{u,\alpha})$. This is the case where the routes for the UAS and vehicle remain the same, and thus the difference is simply

the speed of the UAS.

We now use our result from Lemma 1 to derive the worst case ratio $\frac{Z(SQ_{n,K})}{Z(PIRP_{n,u,K})}$, where we have one vehicle, u UAS, and one subnetwork.

Proposition 2. $\frac{Z(SQ_{1,1})}{Z(PIRP_{1,u,1})} \leq 1 + \frac{u\alpha - 1}{\beta + 1}, \quad \beta = \frac{Z(VRP_1)}{Z(ASIP_{u,\alpha})}$

Proof. We first decompose both $Z(SQ_{1,1})$ and $Z(PIRP_{1,u,1})$ into two components. The first component is the vehicle route to the base and back, which we denote as $Z(VRP_1)$. The result for $Z(VRP_1)$ is the same for both $Z(SQ_{1,1})$ and $Z(PIRP_{1,u,1})$. The second component is the inspection route, which we denote as $Z(TSP)$ and $Z(ASIP_{u,\alpha})$ for the vehicle inspection and UAS inspection, respectively. Using this decomposition, we have:

$$\frac{Z(SQ_{1,1})}{Z(PIRP_{1,u,1})} = \frac{Z(VRP_1) + Z(TSP)}{Z(VRP_1) + Z(ASIP_{u,\alpha})} \tag{3.2}$$

$$= 1 + \frac{Z(TSP) - Z(ASIP_{u,\alpha})}{Z(VRP_1) + Z(ASIP_{u,\alpha})} \tag{3.3}$$

$$= 1 + \frac{Z(ASIP_{u,\alpha})}{Z(VRP_1) + Z(ASIP_{u,\alpha})} \left(\frac{Z(TSP)}{Z(ASIP_{u,\alpha})} - 1 \right) \tag{3.4}$$

$$\leq 1 + \frac{Z(ASIP_{u,\alpha})}{Z(VRP_1) + Z(ASIP_{u,\alpha})} (u\alpha - 1) \tag{3.5}$$

$$= 1 + \frac{u\alpha - 1}{\beta + 1} \tag{3.6}$$

Equation (3.2) is the decomposition of $Z(SQ_{1,1})$ and $Z(PIRP_{1,u,1})$ into their two components. Equations (3.3) and (3.4) rearrange the terms to form the ratio $\frac{Z(TSP)}{Z(ASIP_{u,\alpha})}$ defined earlier in Lemma 1. Inequality (3.5) is valid since we replace $\frac{Z(TSP)}{Z(ASIP_{u,\alpha})}$ with $u\alpha$. Here we recognize that the ratio $\frac{Z(ASIP_{u,\alpha})}{Z(VRP_n)+Z(ASIP_{u,\alpha})}$ is very meaningful as the ratio between the ASIP and PIRP. To be precise, we introduce a term β to be the ratio between the $Z(VRP_n)$ and $Z(ASIP_{u,\alpha})$; this is a measure of the total inter-subnetwork travel time compared to the total intra-subnetwork travel time. Substituting for this β term, we get equation (3.6).

Based on our example shown in Figure 3-1, we show that this bound is tight. Since the result for $Z(VRP_1)$ is the same for both $Z(SQ_{1,1})$ and $Z(PIRP_{1,u,1})$, we have:

$$\frac{Z(SQ_{1,1})}{Z(PIRP_{1,u,1})} \geq \frac{Z(VRP_1) + Z(TSP)}{Z(VRP_1) + Z^f(ASIP_{u,\alpha})} = \frac{Z(VRP_1) + 2u\alpha}{Z(VRP_1) + 2} \tag{3.7}$$

For simplicity, we assume that the base is located at a distance of d from the service station. $Z(VRP_1)$ then is equal to $2d$ (to the base and back). Therefore, we have:

$$\frac{Z(VRP_1) + 2u\alpha}{Z(VRP_1) + 2} = \frac{2d + 2u\alpha}{2d + 2} = \frac{d + u\alpha}{d + 1} \tag{3.8}$$

For this example we can confirm that $\beta = d$, and therefore, $\frac{Z(SQ_{1,1})}{Z(PIRP_{1,u,1})} \geq \frac{d+u\alpha}{d+1} = 1 + \frac{u\alpha - 1}{\beta + 1}$.

□

Next we generalize the above result for K subnetworks.

Proposition 3. $\frac{Z(SQ_{1,K})}{Z(PIRP_{1,u,K})} \leq 1 + \frac{u\alpha - 1}{\beta + 1}, \quad \beta = \frac{Z(VRP_1)}{\sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})}$

Proof. We again decompose both $Z(SQ_{1,K})$ and $Z(PIRP_{1,u,K})$ into two components as before, but in this case we account for K subnetworks. Using a similar decomposition procedure, we have:

$$\frac{Z(SQ_{1,K})}{Z(PIRP_{1,u,K})} = \frac{Z(VRP_1) + \sum_{k=1}^{K} Z_k(TSP)}{Z(VRP_1) + \sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})} \tag{3.9}$$

$$= 1 + \frac{\sum_{k=1}^{K} \left(Z_k(TSP) - Z_k(ASIP_{u,\alpha})\right)}{Z(VRP_1) + \sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})} \tag{3.10}$$

$$= 1 + \frac{\sum_{k=1}^{K} \left(Z_k(ASIP_{u,\alpha})\left(\frac{Z_k(TSP)}{Z_k(ASIP_{u,\alpha})} - 1\right)\right)}{Z(VRP_1) + \sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})} \tag{3.11}$$

$$\leq 1 + \frac{(u\alpha - 1)\sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})}{Z(VRP_1) + \sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})} \tag{3.12}$$

$$= 1 + \frac{u\alpha - 1}{\beta + 1} \tag{3.13}$$

Equation (3.9) is the decomposition of $Z(SQ_{1,K})$ and $Z(PIRP_{1,u,K})$ into their respective components. Equations (3.10) and (3.11) rearrange the terms to form the ratio $\frac{Z_k(TSP)}{Z_k(ASIP_{u,\alpha})}$ defined in Lemma 1. Inequality (3.12) is valid since we replace $\frac{Z_k(TSP)}{Z_k(ASIP_{u,\alpha})}$ with $u\alpha$ as before. The β term in this case is the ratio between $Z(VRP_1)$ and $\sum_{k=1}^{K} Z_k(ASIP_{u,\alpha})$. Substituting for this term we get equation (3.13).

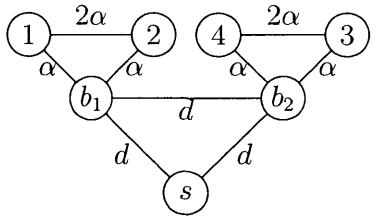


Figure 3-2: Example problem with $K = 2$ subnetworks

Next, we show that this bound is also tight with another example. For simplicity, we assume that all subnetworks have $u+1$ monitoring locations that must be visited to complete a full inspection. We use an arbitrary distance of d for the edges connecting the subnetworks to service station s as well as adjacent subnetworks. Figure 3-2 shows an example with $K = 2$ subnetworks, where each subnetwork consists of 3 monitoring locations (including the base).

The result for $Z(VRP_1)$ is the same for both $Z(SQ_{1,K})$ and $Z(PIRP_{1,u,K})$, and therefore, $Z(VRP_1) = d(K+1)$ by inspection. Thus we have:

$$\frac{Z(SQ_{1,K})}{Z(PIRP_{1,u,K})} \geq \frac{Z(VRP_1) + \sum_{k=1}^{K} Z_k(TSP)}{Z(VRP_1) + \sum_{k=1}^{K} Z_k^f(ASIP_{u,\alpha})} = \frac{d(K+1) + 2u\alpha K}{d(K+1) + 2K} \tag{3.14}$$

For this example $\beta = \frac{d(K+1)}{2K}$, and we confirm that the bound is tight using substitution:

$$\frac{Z(SQ_{1,K})}{Z(PIRP_{1,u,K})} \geq \frac{d(K+1) + 2u\alpha K}{d(K+1) + 2K} = 1 + \frac{u\alpha - 1}{\beta + 1} \tag{3.15}$$

□

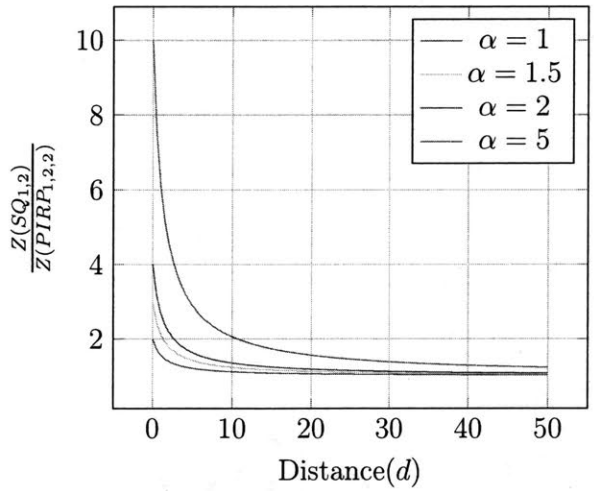


Figure 3-3: Relationship Between Distance d and α

We summarize our results by examining the worst-case ratio in different settings. Using the last example with 1 vehicle, 2 UAS, and $K = 2$ subnetworks, we focus on the tradeoff between distance d and α when determining the ratio $\frac{Z(SQ_{1,2})}{Z(PIRP_{1,2,2})}$. Figure 3-3 shows that the worst case bounds starts at $u\alpha$ when $d = 0$, and approaches 1 as d becomes large. Moreover, the value of α influences the magnitude of the worst case ratio. To generalize,

what we confirm is that more time savings can be achieved if subnetworks are larger and vehicle distances to the subnetworks are shorter. Furthermore, as the distance to the bases get larger, the time savings to be gained from the UAS-enabled inspection is diminished. Using these insights, we can confirm that the advantages of UAS-enabled inspection can be significant, especially when considering that our results are conservative. We next formulate and solve the ASIP and PIRP on a benchmark pipeline network.

3.2 Aerial Sensor Inspection Problem

Consider a subnetwork $\mathcal{T}_k$ that is alerted during the time interval $[0, t_0]$. Recall that the exact number of failures, and their exact locations in $\mathcal{T}_k$ are unknown. Thus, the ASIP considers optimally dispatching UAS to isolate every network component in $\mathcal{T}_k$. The u UAS leave the base b_k, visit a subset of monitoring locations in $\mathcal{V}_k$ from where they can inspect the network components, and return to the base either to complete the mission, or to replace their batteries for further inspection. The objective is to minimize the time to explore $\mathcal{T}_k$, which we formulate as the maximum amount of time, among the u UAS, to return to the base for mission completion. We choose this min-max objective function for two main reasons highlighed in [5]. In that work, a min-max objective was widely accepted by the key stakeholders of a natural gas utility over an alternative objective of minimizing the overall cost, which was primarily viewed as a symptom of the root problem. Second, the min-max objective achieves a level of fairness, which complies with union regulations for equal distribution of labor.

3.2.1 Mixed Integer Programming Formulation

For the sake of brevity in presenting our formulation of the ASIP, we use the notation b to denote the base b_k, τ_{ij} as the travel time τ_{ij}^k, and ξ as the longest UAS inspection time ξ_k. For each pair of monitoring locations $i \neq j \in \mathcal{V}_k$ we define a binary variable x_{ij} equal to 1 if an UAS goes from i to j, and 0 otherwise. We also define two real variables z_{ij} and

t_{ij}. If z_{ij} is nonzero, then it represents the cumulative travel time taken by the UAS that visits node j (coming from node i). Note that this quantity is reset every time an UAS replaces its battery. If t_{ij} is nonzero, then it represents the time traveled so far by the UAS that is currently visiting node j (which comes from i). This quantity is NOT reset when an UAS replaces its battery. For every pair of monitoring locations different from the base, $i \neq j \in \mathcal{V}_k \backslash \{b\}$, let $x_{ij}{}'$ be another binary variable equal to 1 if an UAS goes from i to j after replacing its battery at b, and is equal to 0 otherwise. This is similar to the concept of the replenishment arc as discussed by Boland et al. in [10]. Our ASIP can be formulated with constraints (3.16)-(3.28).

Constraint (3.16) ensures that no more than u UAS are used for inspection. Constraint (3.17) is the flow conservation constraint, taking replenishment arcs into account. Constraint (3.18) ensures that each monitoring location is visited at most once. Constraint (3.19) ensures that each network component is monitored at least once. Constraint (3.20) initializes and resets the time traveled by an UAS after replacing its battery. Constraint (3.21) enforces z_{ij} to be 0 when there is no UAS that goes from i to j and between 0 and τ_{max} otherwise. Constraint (3.22) enforces z_{ib} to be 0 when there is no UAS that goes from i to b or that goes from i to any other node j after replacing its battery at the base b. Constraint (3.23) updates the time traveled so far by each UAS since the last battery replacement. Constraint (3.24) initializes the cumulative time traveled so far by the UAS. Constraints (3.25) and (3.26) make sure that $t_{ij} = 0$ when there is no UAS that goes from i to j (whether directly or by a replenishment arc). The right hand sides of these constraints constitute a "Big-M", which makes the corresponding inequalities non-restricting when an UAS goes from i to j. We illustrate the Big-M upper bound for constraint (3.26) in Figure 3-4. Constraint (3.27) updates the time traveled so far by the UAS and takes into account the time to replace the batteries, if required. Since we want to minimize the maximum travel time of the UAS which is given by $\min \max_{i \in \mathcal{V}_k \backslash \{b\}} t_{ib}$, we can reformulate it by using the variable ξ, along with the constraint (3.28).

The ASIP solution provides optimal UAS routes that can be described as *simple* or *multi-trip* routes. Consider a subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$. We let p represent an UAS *simple*

$$\underset{x,x',z,t,\xi}{\text{minimize}} \quad \xi$$

Subject to

$$\sum_{i \in \mathcal{V}_k \backslash \{b\}} x_{bi} \leq u \tag{3.16}$$

$$\sum_{i \in \mathcal{V}_k \backslash \{b,j\}} (x_{ij} + x_{ij}{}') + x_{bj} = \sum_{i \in \mathcal{V}_k \backslash \{b,j\}} (x_{ji} + x_{ji}{}') + x_{jb}, \qquad \forall j \in \mathcal{V}_k \backslash \{b\} \tag{3.17}$$

$$\sum_{i \in \mathcal{V}_k \backslash \{b,j\}} (x_{ij} + x_{ij}{}') + x_{bj} \leq 1, \qquad \forall j \in \mathcal{V}_k \backslash \{b\} \tag{3.18}$$

$$\sum_{i \in \mathcal{V}_k(e)} \left(\sum_{j \in \mathcal{V}_k \backslash \{b,i\}} (x_{ij} + x_{ij}{}') + x_{ib} \right) \geq 1, \qquad \forall e \in \mathcal{E} \backslash \mathcal{C}_b^k \tag{3.19}$$

$$z_{bj} = \tau_{bj} \left(x_{bj} + \sum_{i \in \mathcal{V}_k \backslash \{b,j\}} x_{ij}{}' \right), \qquad \forall j \in \mathcal{V}_k \backslash \{b\} \tag{3.20}$$

$$0 \leq z_{ij} \leq \tau_{max} x_{ij}, \qquad \forall (i,j) \in (\mathcal{V}_k \backslash \{b\})^2 \mid i \neq j \tag{3.21}$$

$$0 \leq z_{ib} \leq \tau_{max} \left(x_{ib} + \sum_{j \in \mathcal{V}_k \backslash \{b,i\}} x_{ij}{}' \right), \qquad \forall i \in \mathcal{V}_k \backslash \{b\} \tag{3.22}$$

$$\sum_{j \in \mathcal{V}_k \backslash \{i\}} z_{ij} = \sum_{j \in \mathcal{V}_k \backslash \{i\}} z_{ji} + \sum_{j \in \mathcal{V}_k \backslash \{i\}} \tau_{ij} x_{ij} + \tau_{ib} \sum_{j \in \mathcal{V}_k \backslash \{b,i\}} x_{ij}{}', \qquad \forall i \in \mathcal{V}_k \backslash \{b\} \tag{3.23}$$

$$t_{bj} = \tau_{bj} x_{bj}, \qquad \forall j \in \mathcal{V}_k \backslash \{b\} \tag{3.24}$$

$$0 \leq t_{ij} \leq ((|\mathcal{V}_k| - 1)\tau_{max} + (|\mathcal{V}_k| - 2)\tau_{batt})(x_{ij} + x_{ij}{}'), \qquad \forall (i,j) \in (\mathcal{V}_k \backslash \{b\})^2 \mid i \neq j \tag{3.25}$$

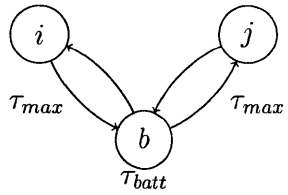
$$0 \leq t_{ib} \leq ((|\mathcal{V}_k| - 1)\tau_{max} + (|\mathcal{V}_k| - 2)\tau_{batt}) x_{ib}, \qquad \forall i \in \mathcal{V}_k \backslash \{b\} \tag{3.26}$$

$$\sum_{j \in \mathcal{V}_k \backslash \{i\}} t_{ij} = \sum_{j \in \mathcal{V}_k \backslash \{i\}} t_{ji} + \sum_{j \in \mathcal{V}_k \backslash \{i\}} \tau_{ij} x_{ij} + \sum_{j \in \mathcal{V}_k \backslash \{b,i\}} (\tau_{ib} + \tau_{batt} + \tau_{bj}) x_{ij}{}', \qquad \forall i \in \mathcal{V}_k \backslash \{b\} \tag{3.27}$$

$$\xi \geq t_{ib}, \qquad \forall i \in \mathcal{V}_k \backslash \{b\} \tag{3.28}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{V}_k^2 \mid i \neq j$$

$$x_{ij}{}' \in \{0,1\}, \qquad \forall (i,j) \in (\mathcal{V}_k \backslash \{b\})^2 \mid i \neq j$$



$$0 \leq t_{ib} \leq ((|\mathcal{V}_k| - 1)\tau_{max} + (|\mathcal{V}_k| - 2)\tau_{batt})x_{ib}$$

Figure 3-4: Explanation of Big-M Upper Bound. Here $|\mathcal{V}_k| = 3$. If $x_{ib} = 0$ then $t_{ib} = 0$. If $x_{ib} = 1$, then t_{ib} is at most $2 \times \tau_{max}$ and $1 \times \tau_{batt}$ after having visited node j first

route, defined as a sequence of n monitoring locations $(i_1, i_2, ..., i_n)$ where $i_1 = i_n = b_k$, each monitoring location in p (not including the base) is visited only once with no interim base visit, and the feasibility requirement is met, i.e., the cumulative travel time $\sum_{m=1}^{n-1} \tau_{i_m i_{m+1}}$ does not exceed τ_{max}. We let $\mathcal{P}$ denote the set of simple routes. We define a *multi-trip route* as a route that contains one or more interim base visits (for replacing the battery), e.g., $(b_k, 1, 2, b_k, 3, b_k)$.

Our formulation differs from the G-VRP introduced by [25] in several ways. First, instead of multiple refueling stations available, there is only one refuel location (temporary base) for each subnetwork. Second, our formulation involves the notion of a monitoring set. Therefore, unlike the traditional VRP formulation, there is no need to visit every monitoring location because of constraints (3.18) and (3.19). Third, instead of using dummy vertices, we use the concept of replenishment arcs which eliminates the need to set a condition on the number of refueling visits. Finally, whereas the G-VRP aims to minimize the total distance traveled by the vehicles, our objective is to minimize the maximum time to observe all network components.

3.2.2 Set-Cover-Based Heuristic

In order to improve the scalability of the ASIP, we propose a heuristic approach that takes advantage of the monitoring set constraint (3.19). This heuristic can be described in five main steps: Solving a weighted set cover problem, initial route construction, improvement

procedures, route combination, and relocation with base insertion. The improvement procedures consist of the relocation, exchange, and 2-opt procedures, which are three well known local search algorithms to solve the Traveling Salesman Problem (TSP) and other related VRPs [17]. Our primary contributions in developing this heuristic are the following: the initial route construction step using a weighted set cover, the route combination step, and the relocation with base insert step.

The computation time to solve the ASIP can pose a computational bottleneck because of the large number of monitoring locations to consider for a typical subnetwork. For example, one ASIP solution for a subnetwork consisting of 33 edges, took almost 3 hours to solve to optimality. Utilities that require efficient dispatch of resources in a timely manner cannot afford to wait this long, and so we next propose a heuristic approach that can promptly reach optimal or near optimal solutions.

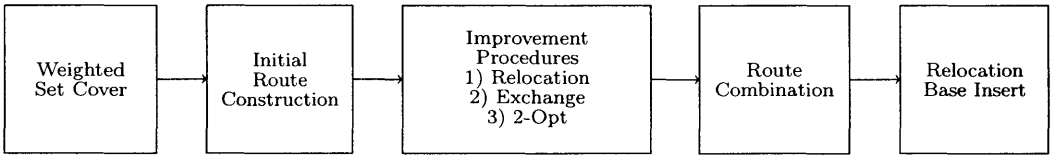


Figure 3-5: Five Steps Used in the Heuristic to Solve the ASIP

Step 1. Weighted Set Cover

Recall that in each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, the UAS need to visit a subset of monitoring locations in order to isolate every network component in $\mathcal{T}_k$. This implies that, although each monitoring location does not need to be visited, the UAS need to visit a subset that forms a *set cover*. In our context, a set cover is a set of monitoring locations $\mathcal{S} \subseteq \mathcal{V}_k$ such that each network component in $\mathcal{T}_k$ is isolated if each monitoring location in $\mathcal{S}$ is visited by the UAS. A minimum set cover (MSC) is a set cover of minimum cardinality that can observe every network component in the subnetwork $\mathcal{T}_k$. The MSC problem is known to be NP-hard but many commercial solvers can solve this problem efficiently using exact or approximation algorithms.

For the ASIP heuristic, we consider a weighted variant of the MSC problem as discussed in [15] since distance from the base must also be considered due to limited endurance. For

each monitoring location $i \in \mathcal{V}_k$, we define x_i to be a binary variable equal to 1 if i is chosen as part of the set cover and 0 otherwise. Furthermore, we consider the shortest distance from the base to monitoring location i, or $\tau_{b_k,i}$ as the weights in the objective function. A set cover $\mathcal{S}$ is a weighted MSC if it is an optimal solution of the following problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in \mathcal{V}_k} \tau_{b_k,i} x_i & & \\
\text{subject to} \quad & \sum_{i \in \mathcal{V}_k(e)} x_i \geq 1, & \forall e \in \mathcal{E} & \quad (3.29) \\
& x_i \in \{0,1\}, & \forall i \in \mathcal{V}_k & \quad (3.30)
\end{aligned}$$

Constraint (3.29) ensures that for each network component, at least one of the monitoring locations from $\mathcal{V}_k(e)$, is part of the set cover. Note that because this is a minimization problem and the weights are defined as distances from the base, we can always consider the base to be part of the set cover (since $\tau_{b_k,b_k} = 0$). For the remainder of this section, we will refer to the monitoring locations simply as nodes and b_k as b.

The intuition behind the use of the weighted variant of the MSC as opposed to the unweighted case can be explained using a "pathological" case shown in Figure 3-6, where three nodes are positioned at the specified unit distances from a base. In general we would prefer to visit a higher number of closer nodes as opposed to a few nodes farther away. Recall that we consider the base as a monitoring location and the edges in this graph represent network components. If we assume that the UAS can effectively isolate adjacent components incident to the nodes, we can verify that the unweighted MSC solution is $\{b, 3\}$ but we would prefer to use the weighted MSC solution $\{b, 1, 2\}$ in order to minimize the maximum travel time. If only one UAS is available, the unweighted MSC solution would equate to a total distance of 20, i.e., to node 3 and back, whereas the weighted MSC solution only requires a total travel distance of 4, i.e., to nodes 1 (resp. 2) and back.

To show that the weighted MSC can lead to a feasible ASIP solution, we highlight one

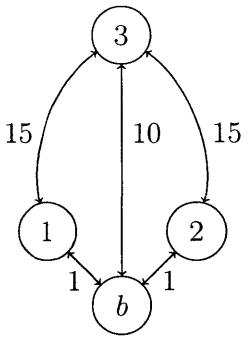


Figure 3-6: Example network for justifying the use of weighted MSC in the ASIP heuristic

example from Table 3.1. Consider the multi-trip routes from the ASIP solution for 2 UAS exploring $\mathcal{T}_2$ from base node 16: UAS 1 $\rightarrow$ **(16, 10, 1, 16)**, **(16**, 5, **6, 16)**, UAS 2$\rightarrow$ **(16**, 8, 9, **7**, **4, 16)**, **(16, 12, 15, 16)**. Each UAS requires one interim base visit. The nodes in bold make up the optimal weighted MSC. Figure 3-7 shows the optimal weighted MSC along with the topology of $\mathcal{T}_2$. We observe that the weighted MSC will always provide a subset of nodes from which to generate feasible routes for the ASIP. We use this insight to construct the initial set of routes.

Step 2. Initial Route Construction

The initial route construction step takes an optimal set cover $\mathcal{S}$ from the weighted MSC problem, the endurance τ_{max}, and the travel times τ_{ij} between each pair of nodes $(i, j) \in \mathcal{V}_k^2$ as its input and provides an initial set of simple routes as its output. To accomplish this, we apply the well known Clarke & Wright Savings Algorithm. Two versions of the savings algorithm exist; a sequential version, where only one route is expanded at a time, and a parallel version, where more than one route may be considered simultaneously [16]. We choose to construct the routes in parallel since it generally provides better results as described in [48]. For each pair of nodes $(i, j) \in \mathcal{V}_k^2$, let S_{ij} be defined as the time "savings" gained by visiting nodes i and j in succession from the base node b and back, i.e., (b, i, j, b) as opposed to one at a time, i.e., $(b, i, b), (b, j, b)$. Using notation from Section 3.2.1, we let t'_{jb} denote the total travel time for the route (b, i, j, b) and t_{jb} denote the total travel time for the route (b, i, b, j, b). Thus the total savings is given by $S_{ij} = t_{jb} - t'_{jb}$. For example, suppose that the two nodes i and j were originally visited using two separate routes as seen in Figure 3-8a.

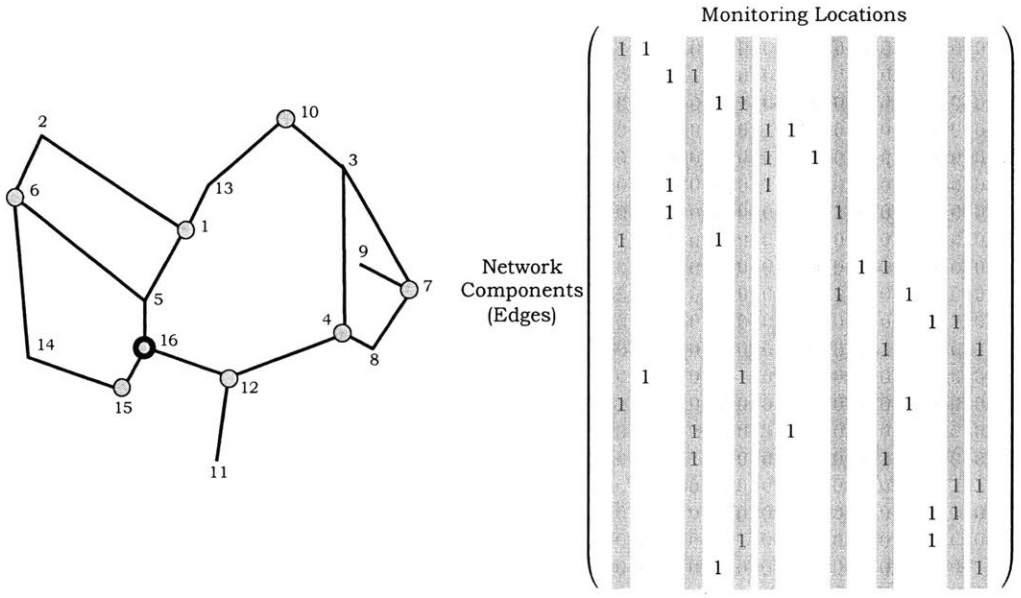
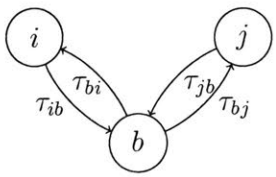


Figure 3-7: Weighted Minimum Set Cover embedded within the ASIP Solution for subnetwork 2. The topology is shown on the left with weighted MSC nodes in grey. The matrix on the right shows the associated monitoring sets with the weighted MSC nodes highlighted

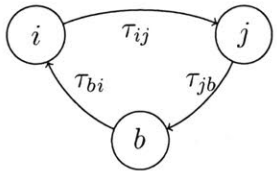
The total travel time in this case is given by $t_{jb} = (\tau_{bi} + \tau_{ib} + \tau_{bj} + \tau_{jb})$. Alternatively, if the two nodes are visited successively in the same route as shown in Figure 3-8b, the total travel time is $t'_{jb} = (\tau_{bi} + \tau_{ij} + \tau_{jb})$. The total travel time savings is then given by:

$$S_{ij} = t_{jb} - t'_{jb} = (\tau_{bi} + \tau_{ib} + \tau_{bj} + \tau_{jb}) - (\tau_{bi} + \tau_{ij} + \tau_{jb}) = \tau_{ib} + \tau_{bj} - \tau_{ij} \tag{3.31}$$



$t_{jb} = (\tau_{bi} + \tau_{ib} + \tau_{bj} + \tau_{jb})$

(a) Nodes visited one at a time



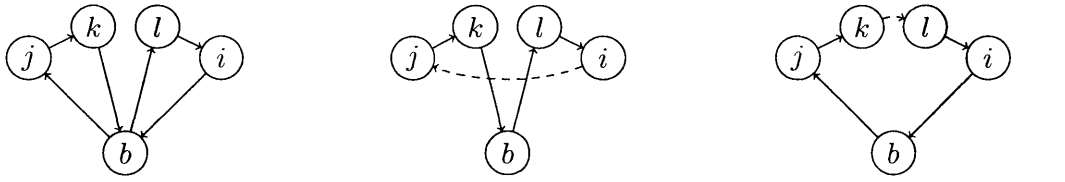
$t'_{jb} = (\tau_{bi} + \tau_{ij} + \tau_{jb})$

(b) Nodes visited in succession

Figure 3-8: Illustration of Travel Time Savings

The intuition behind this savings approach is that pairs of nodes with larger time savings

should be prioritized when constructing the initial routes. It is also important to note that this approach works with asymmetric travel times ($\tau_{ij} \neq \tau_{ji}$) as discussed in [78]. In the asymmetric case, the routes can be considered to be oriented and so we only calculate the savings for S_{ij} if i is the last node visited in a route and j is the first of the other. Suppose we have two oriented routes shown in Figure 3-9a. Since i is the last node visited in one route and j is the first node of the other, $S_{ij} = \tau_{bj} + \tau_{ib} - \tau_{ij}$, which is the difference in the travel times shown below Figs. 3-9a and 3-9b. Respectively, if k is the last node visited in a route and l is the first of the other, $S_{kl} = \tau_{kb} + \tau_{bl} - \tau_{kl}$, using the difference in travel times below Figs. 3-9a and 3-9c. Since S_{kl} provides the larger savings, we only select this as a savings pair and do not consider S_{ij}.



(a) $(\tau_{bj}+\tau_{jk}+\tau_{kb}+\tau_{bl}+\tau_{li}+\tau_{ib})$ (b) $(\tau_{bj} + \tau_{jk} + \tau_{kl} + \tau_{li} + \tau_{ib})$ (c) $(\tau_{bl} + \tau_{li} + \tau_{ij} + \tau_{jk} + \tau_{kb})$

Figure 3-9: Illustration of Savings Calculation for Asymmetric Travel Times

We calculate the savings S_{ij} for every pair of nodes in $\mathcal{S}\backslash\{b\}$ and sort them in descending order of magnitude to create a savings list. Starting from the highest savings pair in the savings list, we construct one or more simple routes based on the following cases for each pair (i, j), keeping inventory of the nodes that have not been assigned to a route [49]. For each case, we check for feasibility, to ensure that total travel time for each route is less than or equal to τ_{max}.

Case 1: If both nodes in the pair do not already belong to a simple route, create a new simple route that consists of the pair bookended by the base, i.e., (b, i, j, b), given that the feasibility requirement is met.

Case 2: If exactly one of the two nodes in the pair (suppose i) belongs to an existing simple route, then we insert j in that same route only if i is an edge node and the feasibility requirement is met. If i is preceded by the base, then j is inserted before i, otherwise,

j is inserted after i. We follow this guideline in order to preserve the integrity of savings pairs.

Case 3: If both i and j already belong to a simple route, then we skip to the next pair in the savings list.

Once the savings list is exhausted, if there are any nodes that were not included in a route, we create a new simple route for each omitted node bookended by the base. The initial route construction is completed once all of the nodes in $\mathcal{S}$ are included within the set of simple routes. Recall that $\mathcal{P}$ denotes the set of simple routes. Note that the Clarke & Wright Savings Algorithm does not allow for the control of $|\mathcal{P}|$. If $|\mathcal{P}| < u$, we ensure that more simple routes consisting of only two basenodes, i.e., (b, b) are created until $|\mathcal{P}| = u$. In the case of more than one UAS, this guarantees an available route to insert a node into for the upcoming relocation procedure. Otherwise, we risk inequity in UAS workload.

Step 3. Improvement Procedures

The next three procedures attempt to improve $\mathcal{P}$ through a sequence of moves, which we define as a modification of nodes either within a route (intra-route) or between routes (inter-route) to obtain a neighborhood solution out of an existing one. We only consider feasible moves based on τ_{max}. We use the relocation, exchange, and 2-Opt procedures in this order based on computational results on routing problems described in [66] as well as our own computational tests on the 5 subnetworks in Table 3.1. After each of these procedures, we do not remove any simple routes; even if a route is or becomes empty, consisting of only two base nodes, (b, b), we carry them over until Step 4, thus guaranteeing that $|\mathcal{P}| \geq u$.

1. **Relocation**. This inter-route procedure takes the initial simple routes obtained from Step 2 as an input and for all possible pairs of routes, completes the following two stages:

   a) For each pair of simple routes, choose the route with the maximum travel time as the "donor" route; the max travel time is set as the incumbent best time to improve upon.

b) For each node in the donor route, not including b, remove and insert it into the other "recipient" route in all of the possible positions between the base nodes. For each move, compute the maximum travel time between the two routes. Choose the move that results in a smallest maximum travel time compared to the incumbent best.

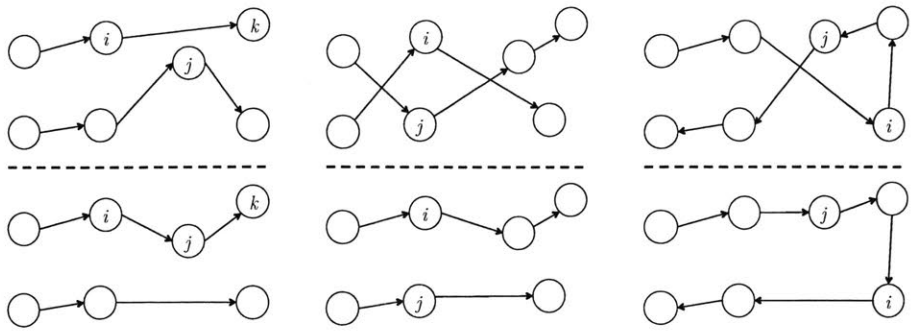
This procedure is shown in Figure 3-10a and is similar to the *relocation* procedure described in [78]. We extend it to consider both τ_{max} and the min-max objective function of the ASIP. We apply the relocation procedure first since it often produces the best results of the three improvement procedures as discussed in [66].

2. **Exchange.** The exchange inter-route improvement procedure as described in [78] considers every possible pair of routes and attempts to exchange two nodes between the two routes as shown in Figure 3-10b. For each pair of routes, we exchange all pairwise combinations of nodes (not including the base). Like the relocation procedure, we compute the maximum travel time between the two routes after each move. We choose the move that results in the smallest maximum travel time compared to the incumbent best.

3. **2-Opt.** As a final improvement procedure, we apply the 2-Opt local search algorithm proposed by [17] for the traveling salesman problem. This is an intra-route improvement procedure that replaces two edges with new ones so that a single route is maintained. The example shown in Figure 3-10c illustrates a valid 2-opt move. The edges $(i-1, i)$ and $(j, j+1)$ are replaced by edges $(i-1, j)$ and $(i, j+1)$, which then reverses the direction of nodes between i and j [11]. We systematically apply the 2-opt procedure on all pairwise combinations of edges in each route. For each swap, we accept the new route if it results in a shorter travel time.

Step 4. Route Combination

This step ensures that the number of generated UAS routes accounts for the number of UAS available and inserts an interim base visit if required. This problem is a variant



(a) Relocation (b) Exchange (c) 2-Opt

Figure 3-10: Improvement Procedures. For each procedure, the top figure shows the route(s) prior to implementation and the bottom figure shows the resulting route(s) [66]

of the multi-processor scheduling or load balancing problem [36]. In the multi-processor scheduling problem, n jobs $j_1, j_2, ..., j_n$ are assigned to m machines, each job j_i has a non-negative processing time, and the goal is to minimize the maximum load over all machines. In our problem, the jobs are the set of simple routes that we obtain after Step 3, each with varying durations less than τ_{max}, and the machines are the u homogeneous UAS. There are two additional considerations for our problem: first, if an UAS is assigned n simple routes, we incur additional time equal to $(n-1)\tau_{batt}$ to account for the battery replacement times; and second, we also need to consider the possibility of merging two simple routes into one simple route (with no interim base visit) if the total travel time is within τ_{max}.

If $|\mathcal{P}| > u$, our greedy approach is to successively combine the shortest two simple routes into a larger route until $|\mathcal{P}| = u$. Recall that we will not see a case where $|\mathcal{P}| < u$ based on the previous steps. We define an *edge node* as a node that is adjacent to the base in a route. It follows that for a simple route there are two edge nodes. We also define an *edge node pair* as a combination of two edge nodes where each node in the pair belongs to a different route. For example, given two simple routes $(b_k, 1, 2, 3, b_k)$ and $(b_k, 4, 5, 6, b_k)$, the edge node pairs would be $(1, 4), (1, 6), (3, 4)$, and $(3, 6)$. It follows that given $|\mathcal{P}|$ routes, with $2|\mathcal{P}|$ edge nodes, the number of edge node pairs is given by:

$$\frac{(2|\mathcal{P}|)!}{2!(2|\mathcal{P}|-2)!} - |\mathcal{P}| \tag{3.32}$$

We subtract $|\mathcal{P}|$ from the number of possible combinations of edge nodes since we do not include edge node pairs where both nodes belong in the same route. Using the savings pairs from the initial route construction procedure, we attempt to merge the two shortest simple routes (with the lowest cumulative travel times) by the highest savings *edge node pair*, reversing the order of one of the routes, if necessary, to create a merged simple route. Note that this approach can also work for the asymmetric travel time case ($\tau_{ij} \neq \tau_{ji}$) but we would need to account for the possibility of longer travel times from the reversal of routes. We choose the merged route that is both feasible and results in the shortest cumulative travel time. Otherwise, an interim base visit is placed where we would have merged the two simple routes, resulting in a multi-trip route.

Step 5. Relocation with Base Insert

This final improvement step executes another iteration of the relocation procedure for all possible pairs of routes (simple or multi-trip) but with the additional consideration of a base insertion to accompany the inserted node. This could be required in order to validate what could otherwise be an infeasible route. We follow the same two stages from the relocation procedure but with routes that can be either simple or multi-trip. Suppose a node k is removed from a donor route and inserted into a recipient route. If the recipient route is a multi-trip route, we need to decompose it into one or more simple routes in order to determine feasibility. If we determine that a simple route is infeasible due to the insertion of node i, we insert an additional interim base visit according to a greedy approach. We represent the simple route that received node k, as a sequence of nodes $(i_1, \ldots, i_n)$. We determine the cumulative travel time to a given node i_q and back to the base using the equation $\tau_{i_q,b+} \sum_{m=1}^{q-1} \tau_{i_m i_{m+1}}$, where $q = 2, ..., n-1$. If the cumulative travel time exceeds τ_{max}, an interim base visit will be inserted before node i_q. Indeed, this will result in an increase in the cumulative travel time for the recipient route (due to the additional travel time to and from the new interim base and/or the τ_{batt}), but it can result in an overall decrease in the maximum travel time *over all routes.*

By taking advantage of the special structure of the ASIP, specifically, the monitoring set

constraint (3.19), our heuristic reduces the number of monitoring locations to consider by solving the weighted set cover problem. We refine the initial simple routes created using the Clarke & Wright Savings Algorithm with additional improvement steps that consider the min-max objective function of the ASIP. We also ensure that the final simple or multi-trip routes account for the number of available UAS. To clarify the steps of our heuristic, we show how it is applied for one example subnetwork in Appendix A.2.

3.3 Inspection Crew Routing Problem

Given a set of service stations $\mathcal{Y}$ and a set of temporary base locations $\mathcal{B}$, the PIRP seeks to find the optimal route for each inspection crew vehicle (referred hereafter as vehicle) starting and ending at its corresponding station such that (i) each base corresponding to a subnetwork is visited, and (ii) the maximum amount of time elapsed from time of failure to inspection completion among all subnetworks, is minimized. In calculating (ii) we include the optimal UAS inspection time, ξ_k^*, required for each subnetwork $\mathcal{T}_k, k \in [\![1, K]\!]$.

For each service station $s \in \mathcal{Y}$, and for every pair of locations $k \neq l \in \mathcal{B} \cup \{s\}$, we define a binary variable y_{kl}^s which is equal to 1 if a vehicle that originates from station s goes from location k to location l, and 0 otherwise. There is no binary variable y_{kl}^s where k or l is a station different from s; this ensures that a vehicle will return to the service station it originated from. For every pair of locations $k \neq l \in \mathcal{B} \cup \mathcal{Y} \mid k \notin \mathcal{Y}$ or $l \notin \mathcal{Y}$, we define a real variable w_{kl} which represents the time at which a vehicle arrives at location l (coming from location k). Note that this quantity takes into account the time to travel between service stations and subnetworks and the time to explore the subnetworks with the UAS. With a slight abuse of notation, for every base $b_k \in \mathcal{B}$, we denote $\xi_{b_k}^* := \xi_k^*$ and $\theta_{b_k} := \theta_k$. A MIP formulation of the PIRP is given in constraints (3.33)-(3.40).

Constraints (3.33)-(3.35) define the classic network flow constraints, while constraints (3.36)-(3.39) keep track of the arrival times for each vehicle. Specifically, constraint (3.33) ensures that no more than n_s vehicles leave service station s. Constraint (3.34) ensures

$$\underset{t_{worst},y,w}{\text{minimize}} \quad t_{worst}$$

Subject to

$$\sum_{l\in\mathcal{B}} y^s_{sl} \le n_s, \qquad \forall s \in \mathcal{Y} \tag{3.33}$$

$$\sum_{l\in\mathcal{B}\cup\{s\}\setminus\{k\}} y^s_{kl} = \sum_{l\in\mathcal{B}\cup\{s\}\setminus\{k\}} y^s_{lk}, \qquad \forall (k,s) \in \mathcal{B}\times\mathcal{Y} \tag{3.34}$$

$$\sum_{s\in\mathcal{Y}}\sum_{l\in\mathcal{B}\cup\{s\}\setminus\{k\}} y^s_{lk} = 1, \qquad \forall k \in \mathcal{B} \tag{3.35}$$

$$w_{sl} = \gamma_{sl} y^s_{sl}, \qquad \forall (l,s) \in \mathcal{B}\times\mathcal{Y} \tag{3.36}$$

$$0 \le w_{kl} \le M \sum_{s\in\mathcal{Y}} y^s_{kl}, \qquad \forall (k,l) \in \mathcal{B}^2 \mid k \ne l \tag{3.37}$$

$$0 \le w_{ks} \le M y^s_{ks}, \qquad \forall (k,s) \in \mathcal{B}\times\mathcal{Y} \tag{3.38}$$

$$\sum_{l\in\mathcal{B}\cup\mathcal{Y}\setminus\{k\}} w_{kl} = \sum_{l\in\mathcal{B}\cup\mathcal{Y}\setminus\{k\}} w_{lk} + \sum_{l\in\mathcal{B}\setminus\{k\}} \gamma_{kl} \sum_{s\in\mathcal{Y}} y^s_{kl} + \sum_{s\in\mathcal{Y}} \gamma_{ks} y^s_{ks} + \xi^*_k, \qquad \forall k \in \mathcal{B} \tag{3.39}$$

$$t_{worst} \ge \sum_{l\in\mathcal{B}\cup\mathcal{Y}\setminus\{k\}} w_{lk} + \xi^*_k + \theta_k, \qquad \forall k \in \mathcal{B} \tag{3.40}$$

$$y^s_{kl} \in \{0,1\}, \qquad \forall s \in \mathcal{Y},\ \forall (k,l) \in (\mathcal{B}\cup\{s\})^2 \mid k \ne l$$

that if a vehicle from service station s enters a subnetwork, it also leaves. Constraint (3.35) ensures that each subnetwork is visited by exactly one vehicle. Constraint (3.36) initializes the time traveled by the vehicle if it departs from service station s. Constraints (3.37)-(3.38) make sure that the arrival time is 0 when no vehicle travels from one subnetwork to another subnetwork or station. Otherwise, we use a large constant, M, to ensure that there is no restriction when a location is visited. Constraint (3.39) updates the arrival time by taking into account the vehicle travel time as well as ξ^*_k. Finally, since we want to minimize the maximum amount of time elapsed from time of failure to inspection completion among all subnetworks, we add the variable t_{worst}, along with constraint (3.40).

3.4 Computational Results for Pipeline Network Inspection

In this section, we first compare our heuristic solutions against the exact solutions of the ASIP. Next, we solve the PIRP, considering different assignments of inspection crews to service stations, and different elapsed times from failure alert to inspection crew dispatch.

We implement the ASIP and PIRP formulations on a case study based on a Kentucky-based urban water network. We assume that a set of fixed sensors are placed at some nodes and monitor the edges in the pipeline network, which form the set of vulnerable components $\mathcal{E}$ [44]. Assuming that a failure in a network component can be detected by a sensor if it is within a given distance [68], the Floyd Warshall algorithm is applied in order to calculate the shortest distance between each pair of nodes and deduce the set of components monitored from each sensor location [31]. The subnetworks are then provided by partitioning the set of components depending on the sensors' outputs.

3.4.1 ASIP Results

For the ASIP, we first consider the activation of five subnetworks with sizes ranging from 16 to 33 edges. We label these subnetworks $\mathcal{T}_1, \ldots, \mathcal{T}_5$.

For each subnetwork $\mathcal{T}_k$, $k \in [\![1, 5]\!]$, we assume that the UAS monitoring locations, $\mathcal{V}_k$, are positioned directly above the nodes of the subnetwork induced by $\mathcal{T}_k$, i.e., the set of end nodes of the edges in $\mathcal{T}_k$. We consider the scenario where an UAS, positioned directly above a given monitoring location $i \in \mathcal{V}_k$, can inspect the adjacent edges of i (i.e., the adjacent pipelines). Without loss of generality, we restrict the UAS to travel only along the edges of the given pipeline network and we assume symmetry with respect to travel times τ_{ij}^k. We place the temporary base within each subnetwork by finding the node that minimizes the total distance from that node to all other nodes. As a conservative estimate, we assume a maximum endurance, τ_{max}, of 1 hour, and a battery replacement time, τ_{batt} of 5 minutes. By solving the ASIP instances, we find that the optimal time required to isolate all components for each subnetwork with 2 UAS is $\boldsymbol{\xi}^* = (0.88, 1.46, 0.86, 0.87, 0.63)$ (in hours). The final routes for the 2 UAS are shown in Table 3.1. Note that the inspection of subnetwork $\mathcal{T}_2$ takes longer because it requires a multi-trip route for each UAS. Also note that due to the min-max objective function of the ASIP, the final solution can result in extraneous node

visits for route(s) with shorter duration. For example, for $\mathcal{T}_2$, the visit to node 9 for the second UAS is one such case.

Table 3.1: ASIP Solutions

$\mathcal{T}_k$	Nodes	Edges	ξ_k^*	b_k	UAS Routes
$\mathcal{T}_1$	31	33	0.88	4	$(4,18,24,1,29,13,12,11,26,5,4)$, $(4,3,7,6,8,9,22,19,28,4)$
$\mathcal{T}_2$	16	20	1.46	16	$(16,10,1,16)$, $(16,5,6,16)$, $(16,8,9,7,4,16)$, $(16,12,15,16)$
$\mathcal{T}_3$	15	18	0.86	6	$(6,8,3,15,14,1,6)$, $(6,5,10,11,6)$
$\mathcal{T}_4$	18	17	0.87	8	$(8,10,15,12,14,8)$, $(8,4,1,5,6,3,8)$
$\mathcal{T}_5$	16	16	0.63	11	$(11,16,5,4,7,10,11)$, $(11,8,12,14,1,11)$

Using the MIP formulation, it took over 5 hours to achieve the exact solutions for the 5-subnetwork scenario referred to in Table 3.1. Comparatively, our heuristic was able to obtain the same optimal solutions in only 4.15 seconds, an acceptable time frame for real world implementation. Using the same Kentucky based pipeline network, we altogether tested our heuristic on 10 different subnetworks consisting of up to 37 edges (i.e., components). The largest of these represents the biggest reasonable size that can be assigned to an inspection crew given the scale of the network. Computational results show that high quality solutions can be obtained using the ASIP heuristic for 2 and 3 UAS. Table 3.2 shows our results using 2 UAS. The results for 3 UAS are included in Appendix A.1. The subnetworks are listed in ascending order based on the number of edges with the 5 subnetworks used in our scenario highlighted in grey. The overall average optimality gap was 0.78%. All of our problem instances were solved on a computer with a 2 GHz Intel Core i7 processor and 8 GB of RAM.

Our computational study for the ASIP shows that our heuristic can provide high quality solutions within a time frame that meets operational requirements.

Table 3.2: Comparison of ASIP Exact Solutions with Heuristic for 2 UAS

Subnetwork		MIP		Heuristic		Optimality
Nodes	Edges	Obj	Time (sec)	Obj	Time (sec)	Gap
5	6	0.209	0.02	0.209	0.013	0%
11	10	0.413	0.41	0.413	0.042	0%
16	16	0.635	33	0.635	0.093	0%
18	17	0.869	10	0.869	1.115	0%
15	18	0.857	94	0.857	0.077	0%
16	20	1.464	7742	1.464	1.208	0%
17	22	0.745*	100000	0.745	0.157	0%*
22	29	1.582*	100000	1.611	0.271	2%*
31	33	0.882	10354	0.882	0.608	0%
36	37	1.291*	100000	1.369	0.715	6%*

* Figures based on the best incumbent MIP objective value found in 100000 sec limit.

3.4.2 PIRP Results

Using the ASIP solutions from Table 3.1, we consider two service stations, s_1 and s_2, that are located in the network with station s_1 containing one inspection crew, and station s_2 containing two. Each inspection crew travels in a vehicle that carries two UAS. We derive the travel time by dividing euclidean distances between each pair of locations (k, l) in the set $\mathcal{Y} \cup \mathcal{B}$ by the average vehicle speed. We assume that the maximum distance can be covered in 6 hours, and that $\boldsymbol{\theta} = (1, 1, 1, 1, 1)$ for simplicity (i.e., all failure alerts occur one hour prior to t_0).

Using the formulation for the PIRP, as described in Section 3.3, we obtain the following optimal solution: The single inspection crew from service station s_1 travels along the route (s_1, b_4, s_1), and the routes for the two inspection crews from service station s_2 are (s_2, b_5, b_1, s_2) and (s_2, b_2, b_3, s_2). Figure 3-11 illustrates the optimal solution. The longest time elapsed from time of failure to inspection completion is 5.87 hours.

From this initial computational study, we can make the following observations: First, we verify that the overall solutions are sensitive to θ. In general, a subnetwork $\mathcal{T}_k$ with a larger θ_k will be visited first. Second, we observe that the PIRP solution tends to compensate for the magnitude of the UAS optimal inspection times, ξ_k^*. For instance, an inspection crew can

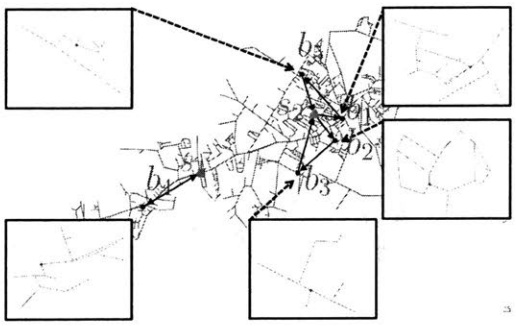


Figure 3-11: Optimal PIRP solution where all failures alerts occur one hour prior to t_0. The arrows depict the routes for the inspection crews, which originate from the two service stations (triangles) and visit the temporary bases within each subnetwork

take a longer route if UAS inspection times for the subnetworks in that route are relatively small, and vice versa.

Next, we solve for the PIRP using different values for θ and n_s. Note that we can use the ASIP solutions as inputs for the PIRP only because we assumed the same number of homogeneous UAS for each inspection crew. We also consider an alternative scenario where $\boldsymbol{\theta} = (1, 5, 1, 1, 3)$ meaning that subnetwork $\mathcal{T}_2$ (resp. $\mathcal{T}_5$) was alerted 5 hours (resp. 3 hours) prior to t_0. With all else equal, this results in an increase in priority to visit these subnetworks earlier. Larger values of θ may arise due to the unavailability of inspection crews as discussed in Chapter 2, Section 2.5. For each scenario, we also consider the availability of 1 to 2 inspection crews at each service station to assess the impacts of initial crew placement on the PIRP solution. Table 3.3 shows the resulting optimal values for t_{worst} and inspection crew routes for each scenario.

Figure 3-12 shows the optimal solution for the PIRP considering the scenario $\theta = (1, 5, 1, 1, 3)$ with 1 inspection crew at service station 1, and 2 vehicles at service station 2. As anticipated, b_2 and b_5 are visited first by the crew originating from service station s_2. Surprisingly, the optimal solution sends one of the crews along the longer route (s_2, b_5, b_3, s_2). This is due to the longer UAS inspection time required for $\mathcal{T}_2$. Since the UAS take more time to explore $\mathcal{T}_2$, the crew which visits $\mathcal{T}_2$ compensates by traveling to the closer subnetwork $\mathcal{T}_1$, thus resulting in the shorter crew route (s_2, b_2, b_1, s_2). This implies that the other crew is

Table 3.3: Comparison of PIRP solutions

Time Since Alert	Crew Vehicles		Obj	PIRP
$\boldsymbol{\theta}$	Station 1	Station 2	t_{worst}	Solution
$(1,1,1,1,1)$	1	2	5.87	(s_1, b_4, s_1), (s_2, b_5, b_1, s_2), (s_2, b_2, b_3, s_2)
$(1,1,1,1,1)$	2	2	5.05	(s_1, b_4, s_1),(s_1, b_3, s_1) (s_2, b_5, s_2),(s_2, b_1, b_2, s_2)
$(1,1,1,1,1)$	2	1	7.40	(s_1, b_5, s_1),(s_1, b_4, s_1) $(s_2, b_1, b_2, b_3, s_2)$
$(1,5,1,1,3)$	1	2	7.52	(s_1, b_4, s_1), (s_2, b_5, b_3, s_2),(s_2, b_2, b_1, s_2)
$(1,5,1,1,3)$	2	2	7.52	(s_1, b_4, s_1), (s_1, b_3, s_1) (s_2, b_2, s_2),(s_2, b_1, b_5, s_2)
$(1,5,1,1,3)$	2	1	8.03	(s_1, b_5, s_1),(s_1, b_4, s_1) $(s_2, b_2, b_1, b_3, s_2)$

left to travel along a longer route. The t_{worst} for this problem, is 7.52 hours. When allocated an additional crew at service station 1 (2 crews at both service stations), t_{worst} remains at 7.52 hours. This is because for both cases, $\mathcal{T}_2$ overwhelmingly takes the longest time from time of failure alert to inspection completion. Indeed, 7.52 hours in this case represents a tight lower bound for t_{worst} regardless of crew allocation since $\mathcal{T}_2$ will always be visited first and take the longest time.

Figure 3-12: Optimal PIRP solution for the case where the failure alert for subnetwork 2 (resp. 5) occurs 5 (resp. 3) hours prior to inspection crew dispatch

Finally, we investigate the impact of inspection crew spatial positions prior to dispatch.

Of the 5 subnetworks, $\mathcal{T}_4$ is located near s_1, and is farther from the rest of the subnetworks, which are clustered around s_2. Therefore, the case where 2 inspection crews are at service station 1 and only 1 is at service station 2 represents an "unbalanced" situation. In this case, the additional distance traveled by the inspection crews increases t_{worst} to 8.03 hours compared to 7.53 hours. Although in general, the availability of inspection crews are subject to numerous factors (e.g., utilization rates), this indicates the importance of proper spatial allocation, which we consider as part of our future work.

In summary, our results provide insights in the coupling between the ASIP and the PIRP, the interconnection between UAS inspection time and inspection crew travel times, and the overall solution's sensitivity to inspection crew spatial positioning and θ.

3.4.3 Stochastic Travel Times

In this section, we evaluate the solution of the ASIP for the case when travel times are stochastic by employing Monte Carlo simulation on representative network topologies.

Although we solved the ASIP using deterministic values for the travel times, the complexity of real time navigation for UAS warrants a stochastic model to account for environmental factors (e.g., headwind versus tailwind), the need to circumvent restricted airspace (e.g., temporary flight restrictions), and obstacle or terrain avoidance (e.g., manned aircraft or powerlines). Ignoring the stochasticity of travel times could otherwise lead to over-optimistic or infeasible solutions. This is especially pertinent for smaller UAS that cannot fully compensate for atmospheric disturbances given their slower speeds and lower propulsion capacity [18]. Obstacle avoidance as well as atmospheric turbulence prevalent at low altitudes could also cause the UAS to deviate from the expected travel time between any two locations [82].

Based on the available a priori information with regards to the environment, we can assume that the cruise speed, denoted as V_C, is a random variable with a known probability distribution. Specifically, we assume that V_C is an independent and normally distributed random variable. This assumption is relevant for low altitude flights, where unpredictable

winds prevent the use of even the simplest wind models [82]. Due to the lack of available UAS flight data, we use mean cruise airspeed ($\overline{V}_C$), and the maximum allowable operating speed (V_{max}) from Table 2.1, resulting in the following model for V_C, as described in [84].

$$V_C \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}, \quad \mu = \overline{V}_C, \quad \sigma = \frac{V_{max} - \overline{V}_C}{3} \tag{3.41}$$

Likewise, we also assume that the climb (resp. descent) rates are independent, normally distributed variables, where the mean value is equal to the climb (resp. descent) rate from Table 2.1, and the maximum climb (resp. descent) rate is 5 m/s (resp. -3 m/s), respectively, based on empirical data from the first class of representative UAS.

For each simulation, we determine τ_{ij}, between any two locations $(i, j) \in \mathcal{V}_k^2$ by dividing the shortest path distance from i to j with V_C. Recall that we do not necessarily impose $\tau_{ij} = \tau_{ji}$. We also incorporate the climb and descent rates into the stochastic travel time based on the altitude change from i to j.

We now investigate the impacts of these stochastic travel times with notional topologies; this allows us to assess the impact of not only the random travel times but also the network topology in the overall ASIP solution. We focus on the tree and extended star topologies for our representative subnetworks. We select a binary tree with a height of 3, an extended star consisting of 4 internal nodes with 3 degrees each, an extended star with 10 internal nodes with 3 degrees each, and an extended star with with 17 internal nodes with 4 degrees each. Figure 3-13 shows the four representative subnetworks, in the order of decreasing network density for each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$. We calculate the network density by taking the number of edges and dividing by the number of potential edges, $|\mathcal{V}_k|(|\mathcal{V}_k| - 1)/2$.

For each chosen topology, we use 2 of the first classes of UAS shown in Table 2.1 with an endurance time of 30 minutes. We assume each edge has a distance of 1 km for simplicity, and we normalize the distances so that the longest distance can be reached given the mean

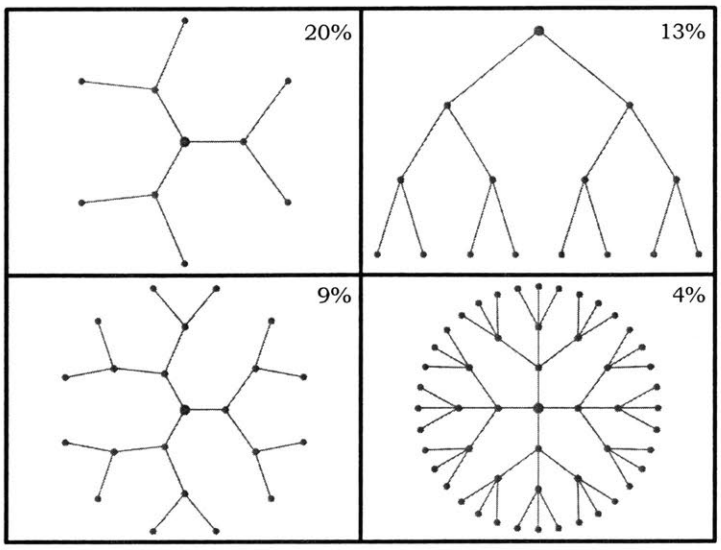


Figure 3-13: Representative topologies. Clockwise from top left: extended star (4 internal nodes with 3 degrees each), binary tree (height of 3), extended star (10 internal nodes with 3 degrees each), and extended star (17 internal nodes with 4 degrees each). The network densities for each are shown on the top right. The lower right has the lowest density measure

cruise airspeed and endurance of the UAS. Also, we restrict the UAS to travel along the edges of the graph. For simplicity, we set the operating altitude to 120 m, equivalent to the maximum allowable altitude for each monitoring location and 80 m for the base location, since we assume that $b_k \in \mathcal{V}_k$. Finally, we assume that the UAS can inspect adjacent components (i.e., edges) incident to a monitoring location (i.e., node).

We employ Monte Carlo simulation to investigate the impacts of the proposed stochastic travel times on the UAS optimal inspection time, ξ_k^*. For each selected topology, we first solve the ASIP using our heuristic, assuming deterministic values for τ_{ij}. Using the optimal routes for each UAS as a guide, we attempt to follow each prescribed route using the simulated random travel times. This results in some incomplete routes, that is, the UAS will not be able to reach some of the monitoring locations towards the end of the route if the cumulative travel times are longer than the endurance. Therefore, there is a likelihood of missing some components in each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, depending on the composition of the monitoring set $\mathcal{C}_i^k$ for each missed monitoring location $i \in \mathcal{V}_k$. We are concerned with two measures of performance: first, the distribution of the UAS optimal inspection times, or ξ_k,

for those UAS flights that successfully observed all components, and second, the number of missed components for a given UAS endurance. Figure 3-14 shows the histograms showing the distribution of inspection times based on 1000 simulations.

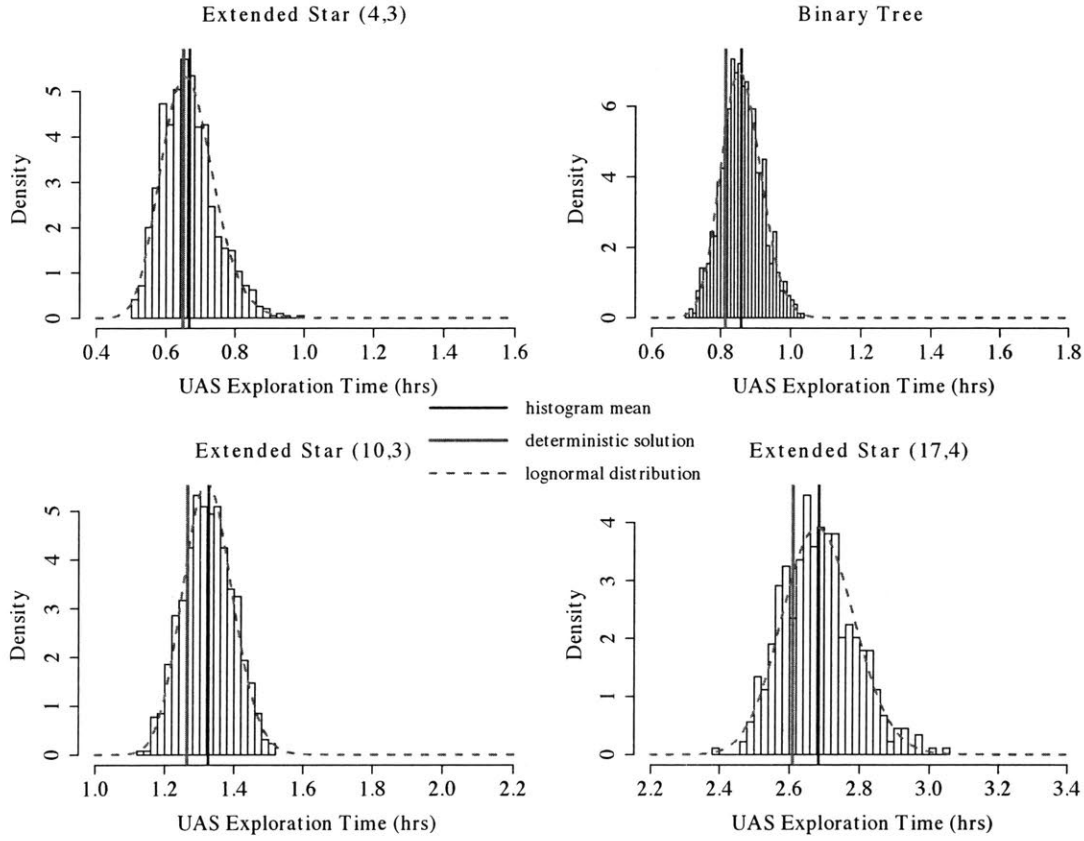


Figure 3-14: Histograms showing the distribution of 1000 simulated UAS inspection times on select topologies

We can now make a few key observations. First, we find that each histogram is best approximated with a log-normal distribution, based on the Akaike Information Criterion (AIC) compared to other distributions like the Gamma and Weibull. We also note that the lognormal distribution's shape parameter decreases as the network density decreases. Furthermore, the histogram mean (respectively the mean of the associated log-normal distribution) is larger compared to the deterministic ASIP optimal value. The difference between these two values grows as the network density decreases (with differences of 0.046, 0.018, 0.062, and 0.075). In general, given the same number of nodes (and therefore the same number of potential

connections), a lower network density will lead to longer inspection times due to increased travel distances.

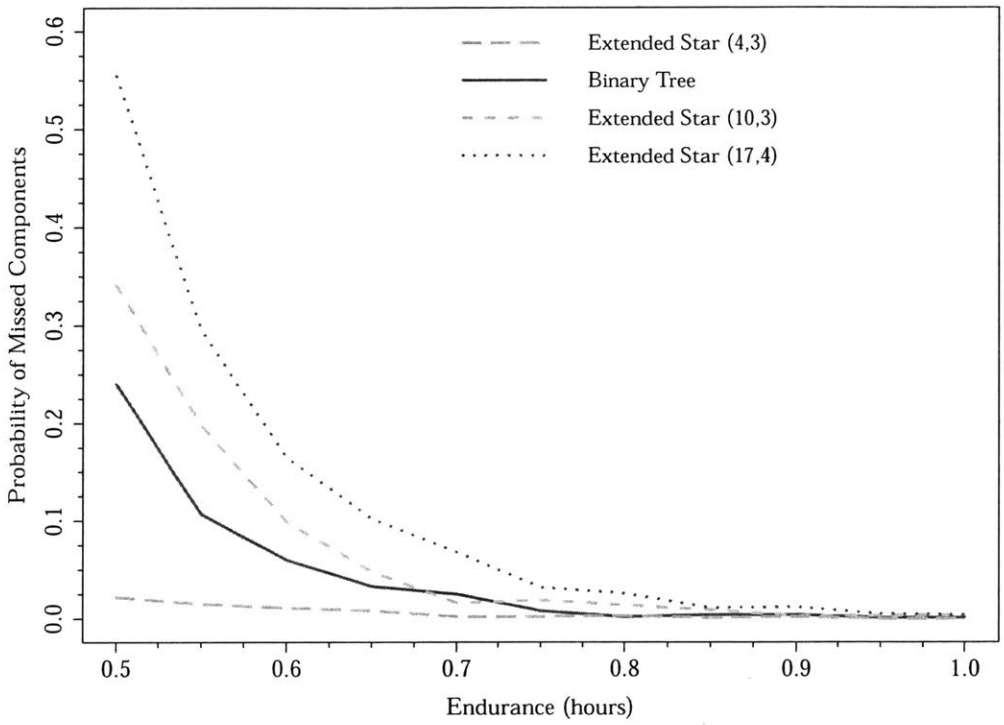


Figure 3-15: Probability of missed network components from 1000 simulations given the endurance

Next, we investigate the impact of UAS endurance on the number of missed components as a result of the simulated travel times. A network component e is missed if the UAS do not visit any of the monitoring locations in $\mathcal{V}_k(e)$. We used a benchmark endurance of 30 minutes to normalize the distances based on the first category of UAS in Table 2.1. We set the maximum distance to a node from the base to half of the endurance. Figure 3-15 shows the probability of missed components from 1000 simulations with different endurance values ranging from 30 minutes to 1 hour. By using these values, we can represent different UAS platforms to some extent. For each of the endurance values, we observe the number of times a network component is missed. We note that the extended star with 4 internal nodes resulted in very few missed components. This is a direct result of its topology, which requires only half of the endurance to travel to the 3 internal nodes adjacent to the base and back; by visiting the internal nodes, the UAS can successfully observe all of the components. As expected, with

higher endurance values we observe a lower probability of missed components. In general, we observe that topologies with lower network density measures result in a higher probability of missed components. We also observe that endurance values higher than 45 minutes result in virtually no missed components for each of the network topologies shown. This suggests that class 2 and 3 UAS from Table 2.1 with capable sensors (larger monitoring sets) could satisfy mission requirements despite the uncertainty from the external environment.

In summary, accounting for stochasticity in travel times suggests that the overall PIRP solution will likely incur delays in practice, due to longer UAS inspection times. Moreover, in some situations, following the prescribed route from the ASIP output can lead to missed network components depending on the endurance of the UAS platform. Thus, in real-world environments, proper choice of UAS platform and conservative route planning is needed to avoid costly setbacks from unidentified failures. Still, UAS-based inspection can lead to significant cost and time savings in comparison to conventional, purely ground-based, operations.

Chapter 4

Value of Predictive Failure Analytics for Priority-Based Routing

This chapter presents an adaptation of our infrastructure inspection approach presented in Chapter 2 to a drainage network. Specifically, the problem we seek to address in this chapter is the following: *Given an extreme storm event, how to leverage data from flood sensors and UAS to: 1) reduce diagnostic uncertainty of failures in various subnetworks, and 2) design optimal inspection crew routing strategies for complete network inspection, subject to time, vehicle, and network constraints?* In Section 4.1 we partition the network into subnetworks based on the range of fixed sensors. Sensor alerts show strong correlation to the extent of damage in subnetworks; these alerts help determine inspection priorities and failure rate predictions. In Section 4.2 we first obtain UAS inspection routes for each subnetwork by solving auxiliary MIPs. We integrate UAS inspection times and stochastic failure rates for network inspection, which is solved in two ways: a stochastic dynamic program that considers prediction intervals of failure rates; and a certainty-equivalent MIP that only accounts for mean failure rates. These solutions allow us to evaluate the utility of integrating sensor data into inspection operations. In Section 4.3, we demonstrate the benefits of our approach using data on drainage network failures and inspections following 2017's Hurricane Harvey.

The work presented in this chapter is motivated by a data project with a particular drainage network agency, whose total maintenance budget is approximately $30 million a year. Delays in the identification of failures such as debris, structural damages, and blockages heighten the catastrophic effects of prolonged storm-induced flooding (e.g., levee structural failures, reduction of flow, limited access to clean drinking water, and loss of basic sanitation).

Following a storm event, the main challenge for a drainage agency is to identify regions more susceptible to failures with higher repair costs while completing a network-wide inspection. Drainage network agencies currently acquire information through two primary forms of sensors. First, pre-installed flood sensors deployed across the network can provide water elevation measures to identify regions of high flood risk. However, using flood sensors to assign priority levels for failure identification has not been fully explored. Second, ground-based inspection crews can be viewed as mobile sensors, who complete detailed network inspection to identify failures. However, narrow Right-of-Ways (ROW) require crews to walk along the water channels, resulting in significant delays. Furthermore, damages inflicted by the storm along the channels can impede access, leading to additional delays or worse, uninspected portions of the network.

4.1 Failure Prediction Model

For drainage networks, we also use data collected from existing fixed sensors to inform inspection crew routing strategies, through priority recommendations. In this section we discuss our approach to utilize data from flood sensors, located on a drainage network, to partition the network into high- and low-priority subnetworks. We use flood sensor measures, along with other geomorphological data, to develop a statistical model for estimating failure rates in each subnetwork. To achieve this, we rely on available data and results from inspection operations following Hurricane Harvey in Houston, TX.

4.1.1 Network and Data

We consider a drainage network consisting of over 2,500 miles of open water channels that are either natural (i.e., creeks, streams) or man-made (i.e., canals, ditches). During a storm event, surface runoff, defined as excess storm water that flows across the land instead of absorbing into the ground, can lead to flooding. The risk of flooding is especially magnified in urban settings, where impervious concrete and pavement surfaces resist the natural absorption of water. The drainage network's primary role is to ensure proper conveyance of surface runoff, so it can ultimately reach a river, lake, or other body of water. We refer to the direction of gravity-induced flow as downstream. The network naturally resembles a tree-like, or dendritic structure.

The main entity responsible for inspection and maintenance of the drainage network is the Flood Control District (FCD). In many states, FCDs are special purpose districts created in response to flood threats, authorized by a number of Flood Control Acts passed by US Congress [87]. The FCD operates out of a small set of service stations. Inspection crews travel to a set of *access points*, which represent authorized entry points off of a road to access the drainage network. The dispatch of inspection crews from the service stations to the access points is enabled by a road network. We consider the set of roads to be accessible (i.e., not flooded or damaged). Otherwise, data collected from sensors or customer calls can be incorporated as a way to estimate accessibility and travel times of certain road segments. To determine the travel time along the roads, we use the primary and secondary road lengths, along with empirical average travel speeds available from the Texas Department of Transportation [73].

For the estimation of failure rates, we also gather additional data that describes the physical attributes of each edge of a subnetwork. Each edge is distinguished by the channel type (man-made or natural), and each channel type is further classified as a water body type. For example, natural channels consist of the following water types: bayou, creek, gulch, river, or swamp; man-made channels consist of either canal/ditch or concrete channel. Using publicly available information from the Soil Survey Geographic Database and the

Houston-Galveston Area Council, we also collect geomorphological data such as soil type (i.e., sand, silt, or clay) and various land cover categories (e.g., evergreen forest) within a 500 ft buffer zone around each edge, based on consultation from the network agency [76, 41].

Following a storm event, wind damage can cause extensive damage to vegetation and buildings, introducing debris to a drainage network such as solid waste, roof material, and downed tree limbs. Furthermore, rain-induced flooding can introduce additional debris, which lead to significant reduction in flow or complete blockages. We focus on debris failures, which include organic or inorganic debris, blockages resulting from slope failure or sediment build-up, and failed concrete. The timely removal of debris following large-scale disasters has been a key focus of study in recent years [23, 13, 3]. Therefore, we collect data on the number of confirmed debris failures found along each edge during inspection operations following Hurricane Harvey. For each failure, we also account for repair cost estimates determined by qualified engineers following inspection operations.

The spatial reach for the aforementioned attributes and failure data to consider for each subnetwork is governed by a distance parameter. We find that 10,000m is ideal to ensure sufficient coverage of the network (90%). For each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$, we count up the number of failures along the edges within this distance (along the shortest paths) from the corresponding flood sensor. We divide the total number of failures by the total edge distance in $\mathcal{E}_k$, to obtain a failure rate per 1000m for each subnetwork. We denote this actual failure rate from the data for each subnetwork as x_k, $k = 1, \ldots, K$.

4.1.2 Failure Rates and Flood Sensor Data

In this section, we propose an approach to utilize flood sensor data to estimate failure rates in each subnetwork. A useful hydrologic measure to determine rate of flow is the *discharge*, defined as the total volume of water that flows past a certain cross sectional area of the channel. The discharge, typically measured in cubic feet per second, is estimated from an empirical elevation-discharge curve, calibrated for each sensor location [77]. For our purpose,

the discharge corresponding to elevations over the TOB is of interest. Using the publicly available data from a drainage network agency's website [39], we collect elevation measurements for 133 flood sensors, over the dates coinciding with Hurricane Harvey. Using the elevation-discharge curves for each flood sensor, we first convert the elevation measurements to discharge. We then obtain the total volume of water over the TOB (cubic feet) by integrating the resulting discharge curve over the time period between the first TOB overflow to the final time the elevation receded below the TOB. We denote this total volume as v_k, $k \in [\![1, K]\!]$. Additional details on total volume calculation can be found in Appendix B.1.

We first examine the relationship between the failure rate and the total volume over the TOB. To do this, we let P_k represent the number of failures and D_k represent the rate of failures (per 1000m) for all $k = 1, \ldots, K$. If we suppose P_k follows a Poisson distribution, the probability density function is given by:

$$Pr(P_k = p_k) = \frac{\lambda_k^{p_k} e^{-\lambda_k}}{p_k!}, \quad p_k = 0, 1, 2, \ldots \tag{4.1}$$

with parameter $\lambda_k = \mathbb{E}(P_k) = Var(P_k)$. The Poisson regression model is derived from this distribution by parameterizing the relation between λ_k and the explanatory variable v_k. This is given by $\lambda_k = \mathbb{E}(p_k \mid v_k) = e^{\beta_0 + \beta_1 v_k}$, where β_0 and β_1 are numeric coefficients. If we let $L(\mathcal{E}_k)$ denote the distance of the edges in subnetwork k, we can model the rate of failures, using a Poisson regression model of the form:

$$\log\left(\mathbb{E}\left(\frac{P_k}{L(\mathcal{E}_k)}\right)\right) = \beta_0 + \beta_1 v_k, \tag{4.2}$$

Using log properties, this is equivalent to the following model, which we use to model the failure rate per 1000m. Note that $L(R_k)$ is treated as another explanatory variable with a fixed coefficient of 1.

$$D_k = \log(\mathbb{E}(P_k)) = \beta_0 + \beta_1 v_k + \log(L(\mathcal{E}_k)), \tag{4.3}$$

Given data from only one storm event, we train our univariate model using the entire drainage network with the exception of a region called the White Oak Bayou region, which we reserve for testing based on consultation with the agency. This region represents a good mix of several physical attributes, such as channel type and water type. For our purpose, the exact total volume over the TOB is not required for failure rate prediction; in fact, the noise in the data can lead to overprediction errors. Therefore, we group the total volume over the TOB into 4 natural bins to create categorical variables representing very low, low, medium, and high volumes. We observe a statistically significant relationship between P_k and these total volume categories (p-values of < 0.001 for each category). In general we observe that higher total volumes over the TOB correspond to higher debris failures, suggesting that overflow and discharge both impact the number of debris failures.

Uncertainty with respect to actual debris failure rates is a critical aspect in determining the order of precedence with which to approach subnetwork inspection. In our proposed approach, the accuracy (or lack thereof) of failure rates can significantly affect the timeliness of failure identification. For example, given inaccurate failure rates, crews might spend a longer time in the earlier days of inspection visiting regions with a disproportionately smaller number of failures. On the other hand, accurate failure rates can assist in focusing earlier inspection efforts on subnetworks with a higher number of failures. To specify the uncertainty of failure rates, we use bootstrapped samples from the training data to acquire a prediction interval based on the Poisson model. The prediction interval defines where a future observation will fall with a certain probability based on our model. For example, with a 95% prediction interval of [0,3], we can be 95% confident that a future observation will fall in this range. A prediction interval provides an alternative to point forecasts and has been shown to be very useful when decisions are affected by uncertainty [32, 6].

Using our univariate Poisson model, we find the prediction interval to be too large for our purpose. For example, an upper prediction interval of 6 equates to a rate of 6 per 1000m,

which would equate to 58 failures in one example subnetwork despite having only 8 actual failures. To reduce the size of the prediction interval (and hence improve on the accuracy), we build upon this univariate model by introducing additional explanatory variables.

We develop a multivariate additive Poisson regression model to estimate the failure rate per 1000m for each subnetwork, using the total volume over the TOB and additional edge attribute data discussed in Section 4.1.1. Specifically, we choose three additional explanatory variables that were shown to be statistically significant for predicting debris failure rates: percent natural channel, percent canal or ditch, and percent evergreen forest. The soil type and other water types did not turn out to be significant. We let $\mathbf{c}_k$ to be a $(1+j)$ vector, where the first element is unity for all k subnetworks and j is the number of explanatory variables.

Therefore, the multivariate Poisson model takes the form:

$$\log(\mathbb{E}(P_k)) = \boldsymbol{\beta}^{'}\boldsymbol{c_k} + \log(L(\mathcal{E}_k)), \tag{4.4}$$

where $\boldsymbol{\beta}^{'}$ represents the vector of numeric coefficients.

Given a $(1+j)\times K$ matrix of explanatory variables and a $1\times K$ vector of actual failure rates from the available data, we obtain coefficient estimates by maximizing the likelihood. We show the results of a training and test split in Figure 4-1 along with the upper bound of the prediction interval (lower bound is 0). Of the list of explanatory variables, the total volume over the TOB stands out as one of the more significant variables in determining the rates of debris failures. As discussed earlier, this volume is calculated by integrating the discharge curve, corresponding to the elevation measurements over the TOB. Additional details on the results of our Poisson models can be found in Section B.2 of the online appendix.

We compare the performance of our univariate and multivariate Poisson models on the White Oak Bayou test set. To estimate the average generalization error, we perform a 10-fold Cross Validation (CV) to obtain an overall mean squared error calculated as $\frac{1}{n}\sum_{i=1}^{n}(P_i - \hat{D}_i)^2$

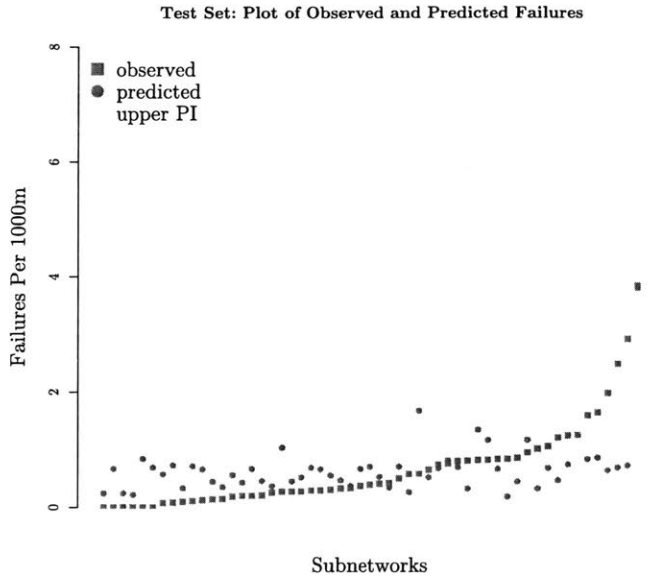


Figure 4-1: Multivariate Poisson Regression Results on Test Set

where $\hat{D}_i$ represents the predicted debris failure rate from the model and n denotes the number of observations. The multivariate model shows an improvement based on the Cross Validation results (3.14 MSE for the Multivariate model compared to a 3.91 MSE for the Univariate model). In Table 4.1, we show the total edge distance, the volume over the TOB, the actual failure rate, the expected failure rate per 1000m based on the models, and the upper bound for the prediction intervals for the 11 subnetworks in the region (6 high-priority and 5 low-priority). We note that there is an improvement in the prediction intervals by incorporating additional variables.

Henceforth, we assume an uninformative prior, where the range of the failure rate is based on the 95% prediction interval. Specifically, we consider a uniform discrete distribution given by $P(D_k = d_k) = \frac{1}{(h_k+1)}$, where h_k is the upper bound from the Poisson model's prediction interval for subnetwork k. This serves as an input to the Inspection Routing Problem, which we discuss next in Section 4.2.

Table 4.1: Model Results for the White Oak Bayou Region

Sub-Network	Distance $L(\mathcal{E}_k)$	Volume $\log(v_k)$	Actual Rate	Predicted Rate		Pred Interval h_k	
				Univar	Multivar	Univar	Multivar
$\mathcal{T}_1^h$	9689	19.91	3.00	0.98	0.90	3	3
$\mathcal{T}_2^h$	17526	23.55	1.76	1.39	1.08	4	3
$\mathcal{T}_3^h$	522	24.51	0.00	2.14	1.50	6	4
$\mathcal{T}_4^h$	24343	24.54	1.03	2.14	1.83	5	3
$\mathcal{T}_5^h$	17729	26.43	1.52	1.73	1.31	5	4
$\mathcal{T}_6^h$	21682	26.22	1.38	1.73	1.39	4	4
$\mathcal{T}_7^l$	18740	26.28	1.17	1.73	1.52	5	4
$\mathcal{T}_8^l$	11924	24.49	1.26	2.15	1.32	5	4
$\mathcal{T}_9^l$	9895	24.68	0.81	2.15	1.56	6	4
$\mathcal{T}_{10}^l$	2016	25.41	1.98	2.15	1.45	6	4
$\mathcal{T}_{11}^l$	9474	25.54	2.74	2.15	2.09	5	4

4.2 Prioritized Inspection Crew Routing

In this section, we describe our proposed approach to solve the Prioritized Inspection Routing Problem (PIRP) for a drainage network, using the subnetworks and failure rate probability distributions described in Section 4.1.

The overall goal in a drainage network is to rapidly identify not only all failures, but more importantly the high-cost failures, which we define as having estimated repair costs of over \$10K (based on the median repair cost from inspection data). However, this depends on knowing the estimated repair costs, which is realized only after post-evaluation of the inspection data by qualified engineers. Therefore, this motivates our choice for the objective. We focus on a proxy measure to find all failures, since high-cost failures cannot be immediately confirmed. In the PIRP, our objective is to maximize the number of debris failures over the inspection operations timeframe, which is discretized into N days. Our goal is to determine if this objective, combined with the failure rate probability distributions, can lead to a routing schedule where the net effect is the timely identification of high-cost failures. We first partition the drainage network into subnetworks, and for each subnetwork, we determine the a-priori failure rate probability distribution $P(D_k)$, as described in the previous section. Suppose time 0 denotes the time when the storm event strikes and time t_0

represents the first day of favorable weather and channel conditions to enable access to the drainage network. Based on the flood sensor alerts collected during the time period $[0, t_0]$, we consider a collection of subnetworks $\mathcal{T}_k$, $k \in [\![1, K]\!]$, each represented with an access point. Recall from Section 2.1 that we can refer to these subnetworks as high-priority, $\mathcal{T}_k^h$, or low-priority, $\mathcal{T}_k^l$, based on alerts.

We consider a set of service stations denoted as $\mathcal{Y}$. We let n_s denote the number of inspection crews that are available for each day from service station $s \in \mathcal{Y}$. For simplicity, we assume that each crew is equipped with U homogeneous UAS to assist with inspections. UAS can identify failures across the entire subnetwork at a fraction of the time required for ground-based inspection crews. As a result of this increased rate of inspection, crews can complete piecemeal or fractional inspections of subnetworks. This approach would otherwise be difficult using ground-based crews due to challenges in accessibility and work hour considerations.

For each day $m \in [\![1, N]\!]$, an inspection crew must depart and return to the same service station within a time budget which we denote as T_m. In practice, T_m can be different for each day depending on various factors such as weather, accessibility, and worker availability. For the purpose of network inspection, we designate a set of nodes, $\mathcal{B} := \{b_k,\ k \in [\![1, K]\!]\}$, as access points. In practice, there can be more than one access point for each subnetwork. We let the set $\mathcal{J}$ contain the edges (k, l) for each ordered pair of locations in the set $\mathcal{Y} \cup \mathcal{B}$, assuming a complete graph. The travel time needed by a crew to travel along $(k, l) \in \mathcal{J}$ is denoted by γ_{kl}, which is determined by the shortest paths along the road network using the empirical average travel speed data. We do not necessarily impose $\gamma_{kl} = \gamma_{lk}$. Once the inspection crew arrives at an access point, UAS are used to complete inspection. The inspection crew remains stationary at the access point while UAS travel along the Right-Of-Way (ROW) to complete inspection; this policy is especially practical when ground conditions prevent movement by foot to the drainage network. Upon retrieval of the aerial sensor, the inspection crew can choose to either continue inspection of the same subnetwork or visit another subnetwork within the remaining time budget.

The routing of the inspection crew over a transportation network is driven by two decisions: first, the sequence in which to visit the access points (i.e., subnetworks), and second, for those visited subnetworks, what fraction of the subnetwork to inspect with the aerial sensor. For each day $m \in [\![1, N]\!]$, let $R_k^m \subseteq \mathcal{E}_k$ denote the set of edges of $\mathcal{T}_k$ that have not been inspected after days $1, \ldots, m-1$, and let $L(R_k^m)$ denote the total length of the edges in R_k^m.

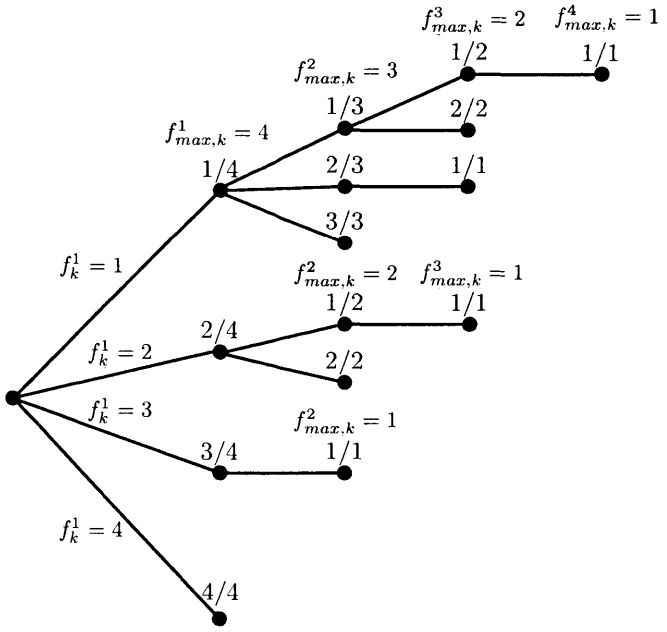


Figure 4-2: Fractional Inspection Sequence Example for a Subnetwork

For each day m, an inspection team may decide to inspect a fraction of a subnetwork, which we model as a choice index denoted as $f_k^m \in \mathbb{N}$ that satisfies $f_k^m \leq f_{max,k}^m$, where $f_{max,k}^m$ represents the largest choice index. The largest index choice satisfies: $\forall m \in [\![1, N-1]\!]$, $f_{max,k}^{m+1} = f_{max,k}^m - f_k^m$. This ensures that the entirety of the subnetwork will be inspected with at most $f_{max,k}^1$ visits. For simplicity, we assume that $f_{max,k}^1$ is the same for all subnetworks. Figure 4-2 shows an example of the possible sequence of fractional inspections for a subnetwork with $f_{max,k}^1 = 4$. A choice index of $f_k^1 = 1$ would represent an inspection of 1/4 of the edges in R_k^1 for the first day. The largest choice index for the following day is given by $f_{max,k}^2 = f_{max,k}^1 - f_k^1 = 3$. For smaller subnetworks where the total length $L(R_k^m)$, does not exceed the length of the longest edge in the network, we assume the only available

fraction choice to be $f^1_{max,k}$; in other words, the only choice is to inspect the entire subnetwork. Based on the objective to maximize the number of expected failures, the routing decisions are based on the failure rate distribution for each subnetwork, which reflects the current belief. The initial distribution is assumed to be uniform, and with each subsequent fractional inspection of a subnetwork, we update the respective distribution using a Bayesian update, which we discuss in Section 4.2.2. Figure 4-3 illustrates a general timeline using our notation for inspection operations requiring a total of N days after t_0. Note that in the case of a drainage network, we set $\theta_k = 0,\ k \in [\![1, K]\!]$ since the information about the delay from the time of sensor alert to t_0 is embedded in the Total Volume over TOB measure.

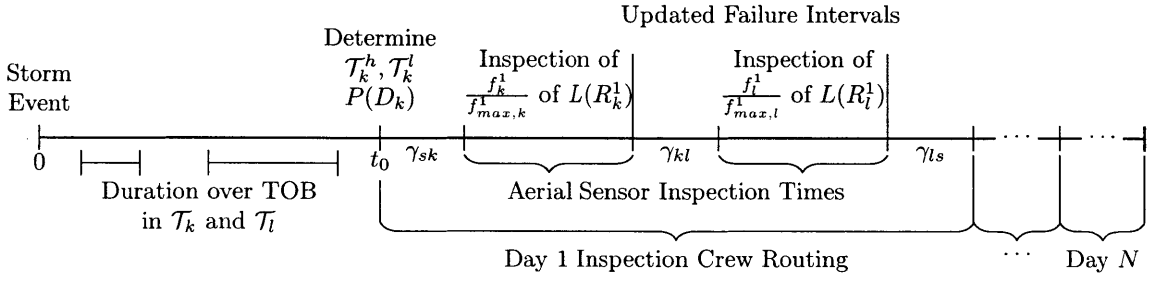


Figure 4-3: Timeline of Inspection Operations for our Proposed Approach

Since the PIRP solution is partially determined by UAS inspection time, we solve a second routing problem to determine the optimal time required to inspect a chosen fraction of a subnetwork using available UAS. We call this problem the Aerial Sensor Inspection Problem (ASIP), and we solve it offline for a finite set of fraction choices. In the next section, we describe how to solve the ASIP.

4.2.1 Refined Aerial Sensor Inspection Problem

We also integrate UAS as described in Chapter 3 to improve accessibility and to identify failures at a fraction of the time required for ground-based inspection crews. However, instead of using monitoring locations, we consider a more restricted case for drainage networks, where the UAS must traverse every edge of the subnetwork.

Consider a subnetwork $\mathcal{T}_k$ with $k \in [\![1, K]\!]$. Recall that we let $\mathcal{N}_k$ (resp. $\mathcal{E}_k$) denote the

set of nodes (resp. edges) in $\mathcal{T}_k$. For every pair of nodes (i,j) in $\mathcal{N}_k$, we let τ_{ij} represent the distance along the edges from i to j. For each day $m \in [\![1, N]\!]$, let $E_k^m \subseteq \mathcal{E}_k$ denote the edges chosen for inspection on day m. The set of remaining edges still requiring inspection satisfies the following: $R_k^1 = \mathcal{E}_k$, and $R_k^{m+1} = R_k^m \backslash E_k^m \ \forall m \in [\![1, N-1]\!]$. Based on a chosen fraction from the PIRP, we denote the corresponding total length to inspect as $\frac{f_k^m}{f_{max,k}^m} L(R_k^m)$.

The goal of our Aerial Sensor Inspection Problem (ASIP) is to determine the optimal routing of the UAS in $\mathcal{T}_k$ in order to inspect the total edge length requirement $\frac{f_k^m}{f_{max,k}^m} L(R_k^m)$. The objective is to minimize the longest time to complete this inspection requirement, over all UAS. The routing solution of the ASIP provides a subset of edges E_k^m to inspect using UAS. We denote the total time taken to inspect E_k^m using UAS as ξ^*_{k,f_k^m}.

Due to the tree topology of the network, we allow *bypassing* behavior. Bypassing can occur in two cases. First, if an edge was already inspected on a former visit to the subnetwork, the UAS can travel faster or 'bypass' that edge. Second, if one aerial sensor already inspected an edge during a current visit, another aerial sensor can bypass that same edge.

For the sake of brevity in presenting our formulation of the ASIP MIP for a given subnetwork and fraction choice, we leave out the subnetwork subscript and use $(\mathcal{N}, \mathcal{E})$ to denote the set of nodes and edges in the subnetwork being inspected, R to denote the remaining edges to be inspected, b to denote the base b_k, f to represent the chosen fraction index f_k^m and ξ as the total exploration time ξ_{k,f_k^m}. Once an inspection crew arrives at an access point, a total of U UAS are deployed and travel along the ROW to identify failures. For each edge $e = (i,j) \in \mathcal{E}$, we define $2U$ binary variables x_{ij}^u, x_{ji}^u that are equal to 1 if edge e is traversed by aerial sensor u traveling from i to j or from j to i, respectively. We also define $2U$ binary variables y_{ij}^u, y_{ji}^u that take a value of 1 if an aerial sensor bypasses or travels faster across the edge from i to j or from j to i, respectively. For each edge e we also define $2U$ continuous variables t_{ij}^u, t_{ji}^u which represent the cumulative distance at which an aerial sensor arrives at location j (coming from location i), or arrives at i (coming from j) respectively. Finally, we introduce a binary variable z_e, which takes the value of 1 if e is inspected by an aerial sensor. For each node $i \in \mathcal{N}$, we denote the adjacent nodes as $\delta(i)$. Likewise for each edge

$e \in \mathcal{E}$, we denote the adjacent edges as $\delta(e)$. For a given fraction choice of a subnetwork to inspect, we formulate the ASIP with equations (4.5) to (4.12).

$$\underset{x,y,t,\xi}{\text{minimize}} \quad \xi$$

Subject to

$$\sum_{e} \tau_e z_e \geq \frac{f}{f_{max}} L(R) \tag{4.5}$$

$$(x_{ij}^u + x_{ji}^u - y_{ij}^u - y_{ji}^u) \geq 2z_e \qquad \forall e = (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U \tag{4.6}$$

$$\sum_{j \in \delta(i)} x_{ij}^u = \sum_{j \in \delta(i)} x_{ji}^u \qquad \forall i \in \mathcal{N} \backslash \{b\}, \forall u = 1,\ldots,U, \tag{4.7}$$

$$t_{bj}^u = \tau_{bj} x_{bj}^u \qquad \forall j \in \delta(b), \forall u = 1,\ldots,U \tag{4.8}$$

$$\sum_{j \in \delta(i)} t_{ij}^u = \sum_{j \in \delta(i)} t_{ji}^u + \sum_{j \in \delta(i)} \tau_{ij} x_{ij}^u - \eta \sum_{j \in \delta(i)} \tau_{ij} y_{ij}^u \qquad \forall i \in \mathcal{N} \backslash \{b\}, \forall u = 1,\ldots,U \tag{4.9}$$

$$x_{ij}^u \geq y_{ij}^u \qquad \forall (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U \tag{4.10}$$

$$0 \leq t_{ij}^u \leq M x_{ij}^u, \qquad \forall (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U \tag{4.11}$$

$$\xi \geq t_{ib}^u, \qquad \forall i \in \delta(b), \forall u = 1,\ldots,U \tag{4.12}$$

$$x_{ij}^u \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U$$

$$y_{ij}^u \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U$$

$$z_e \in \{0,1\}, \qquad \forall e = (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U$$

$$t_{ij}^u \geq 0, \qquad \forall (i,j) \in \mathcal{E}, \forall u = 1,\ldots,U$$

Constraint (4.5) ensures that the UAS inspect at least the minimum distance required (based on the fraction chosen from the PIRP). Note that the actual length of inspected edges can be larger than the required length based on the fraction choice. Constraint (4.6) ensures that an aerial sensor can only choose to bypass an edge if that edge is not being inspected. Constraint (4.7) maintains flow conservation. Constraint (4.8) initializes the cumulative time traveled so far by the aerial sensor. Constraint (4.9) updates the cumulative distance

traveled so far by each aerial sensor. We include the parameter η, which is determined by the ratio between the cruise speed and bypass speed. Constraint (4.10) ensures that y_{ij}^u can only be 1 if an aerial sensor travels from i to j. Constraint (4.11) enforces t_{ij}^u to be 0 when there is no aerial sensor that goes from i to j and between 0 and less than a sufficiently large number M otherwise. Finally, since we want to minimize the maximum travel time over the available aerial sensor, we use the variable ξ along with constraint (4.12). The ASIP can be viewed as a modified min-max k-Chinese Postman Problem [24, 2] that allows for bypassing and takes into account a minimum distance to inspect.

We ensure that the upper bound of Big-M in constraint (4.11) is tight by setting it to twice the total length of the edges, using the property of tree networks. Using Gurobi optimization software in Julia, we find our formulation can efficiently reach optimal solutions for even the largest subnetwork (consisting of 101 nodes and 100 edges), foregoing the need for heuristics.

We refer to the sequence of non-zero fractional inspections of a subnetwork as an inspection sequence. Each inspection sequence consists of up to $f_{max,k}^1$ rational numbers that represent the chosen fractions. Therefore we denote the inspection sequence as $\left\{\frac{f_k^m}{f_{max,k}^m}\right\}_{m=1}^N$. We illustrate a 3-element inspection sequence $\{\frac{1}{4}, \frac{1}{3}, \frac{2}{2}\}$ for one subnetwork in Figure 4-4. Table 4.2 shows the associated fraction choices made and the total required inspection times with and without bypassing. In this example, we see improvements of 16% to 41% for inspection times by introducing bypassing behavior. Due to the tree topology, we note that the time required to inspect the final remaining portion is the longest despite having the smallest remaining distance. Therefore, there is an added delay in subsequent inspections if a portion of a subnetwork is not inspected that day.

Table 4.2: Example Subnetwork Sequential Inspection Times

Day m	Distance (m) R^m	Fraction f^m/f_{max}^m	No Bypass ξ^*	Bypass ξ^*
1	11924	1/4	22.6	14.7
2	8582	2/3	58.7	48.9
3	2763	1/1	85.4	50.3

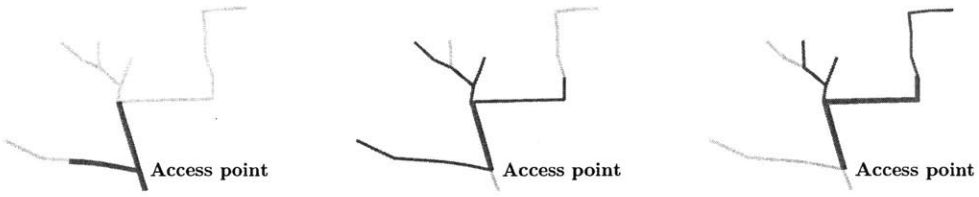


Figure 4-4: Example of ASIP Sequential Inspection for a Subnetwork using 2 UAS

We solve the ASIP offline prior to time t_0 for all possible inspection sequences. Hereafter, we assume that $f^m_{max} = 4$ for all subnetworks. For each day m, we store the ASIP solution times in a $K \times 5$ matrix denoted as Ξ_m with the subnetworks along the rows and the fraction choices indices along the columns. If a fraction choice index is not available during a particular sequence, the corresponding ASIP solution time is set as ∞. By doing so, we can retrieve the value of f^m_{max} for each subnetwork by finding for the largest column index with a real number. Therefore, Ξ_m serves as another input to the PIRP along with the initial uniform failure rate probability distribution.

4.2.2 Failure Distribution Updates

In the case of a fractional subnetwork inspection, we update the failure rate distribution based on the realized number of failures.

Consider a subnetwork k with $k \in [\![1, K]\!]$. On a given day $m \in [\![1, N]\!]$, the failure rate distribution for this subnetwork, $P(D_k = d_k)$, is discrete with finite support on the interval $[\![0, h^m_k]\!]$. Recall from Section 4.1.2 that the a-priori failure distribution for each subnetwork is given by a uniform distribution, $P(D_k = d_k) = \frac{1}{(h^1_k+1)}$, where h^1_k is the upper bound from the Poisson model's prediction interval. Suppose UAS complete inspection of E^m_k corresponding to a fraction $\frac{f^m_k}{f^m_{max,k}}$ of the total remaining edge length in R^m_k. For ease of explanation, we interpret d_k and h^m_k as *numbers* of failures as opposed to the failure rate for the remainder of this subsection.

Upon completing the inspection, we use a Bayesian update procedure to revise the failure

rate distribution. To accomplish this, we account for the number of realized failures from the fractional inspection, which we denote as $x_k \in \mathbb{N}$, where $x_k \leq h_k^m$. We update the probability distribution using the following equation:

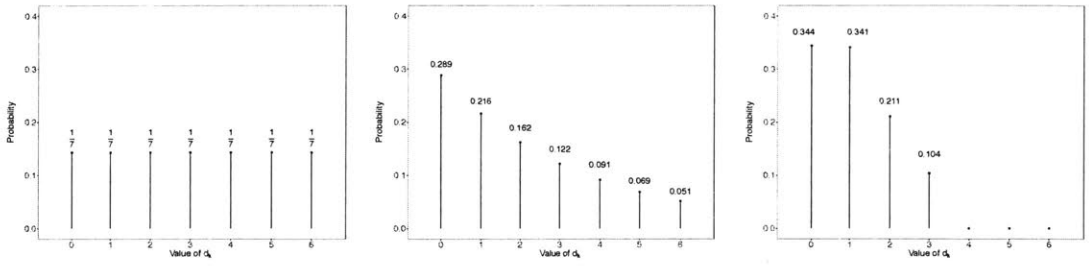
$$P_{f_k^m}(D_k = d_k \mid x_k) = \frac{P_{f_k^m}(x_k \mid d_k)P(d_k)}{P_{f_k^m}(x_k)} \tag{4.13}$$

The prior, $P(d_k)$, is the probability of d_k failures within R_k^m given the prior failure distribution. The likelihood, $P_{f_k^m}(x_k \mid d_k)$, is the probability of finding x_k failures after inspecting fraction $\frac{f_k^m}{f_{max,k}^m}$ of R_k^m, given that the actual number of failures is d_k. The likelihood is computed using a binomial distribution:

$$P_{f_k^m}(x_k \mid d_k) = \binom{d_k}{x_k}\left(\frac{f_k^m}{f_{max,k}^m}\right)^{x_k}\left(1 - \left(\frac{f_k^m}{f_{max,k}^m}\right)\right)^{d_k - x_k} \tag{4.14}$$

The binomial distribution gives the probability with which x_k failures belong in the inspected fraction, given that d_k failures exist in R_k^m. Whether or not a realized failure belongs in the inspected fraction of a subnetwork is a Bernoulli random variable with a success probability equal to $\frac{f_k^m}{f_{max,k}^m}$. Finally, the denominator, $P_{f_k^m}(x_k)$, is computed using the marginal probabilities for x_k given all of the possibilities for d_k. Specifically, this is computed by $\sum_{d_k=0}^{h_k^m} P_{f_k^m}(x_k \mid d_k)P(d_k)$. Note that if $x_k > d_k$, the likelihood function is 0, and therefore $P_{f_k^m}(d_k \mid x_k) = 0$, since the realized number of failures exceeds that of the total assumed failures in the R_k^m. Once we determine the updated failure distribution for R_k^{m+1}, we shift the distribution so that the new finite support accounts for the realized number of failures with possible values in $[\![0, h_k^m - x_k]\!]$. Therefore, as subnetworks are partially inspected, we expect the failure distribution in the remaining uninspected fractions of the subnetwork to narrow over time. Figure 4-5 shows an example of the probability distribution updates for one subnetwork. The initial failure rate probability distribution, which we assume to be uniform, is updated based on the fraction index chosen and realized failures.

Using this failure distribution update procedure, we next describe two solution approaches to solve the PIRP.



(a) Initial Failure Rate Probability Distribution

(b) After inspection of 1/4 and 0 failures found

(c) Inspection of remaining 2/3 and 3 failures found

Figure 4-5: Updating of Failure Rate Probability Distribution for Subnetwork

4.2.3 Solution Approaches

In this section, we describe two solution approaches to solve the PIRP: a stochastic dynamic program and certainty equivalent MIP. Both approaches use the update procedure for failure rate distributions as described in the previous section.

Stochastic Dynamic Program Approach. We first present a stochastic dynamic program to solve the PIRP, incorporating uncertainty in the failure rate of all subnetworks. Let $(\mathcal{Y} \cup \mathcal{B}, \mathcal{J})$ denote the set of nodes and edges representing the service stations/access points and roads, respectively. The goal of the stochastic dynamic program is to find an optimal routing policy to inspect K subnetworks over N days that maximizes the number of expected failures for each day. Each day $m \in [\![1, N]\!]$ has a time budget T_m and contains n^m stages, where each stage is denoted as $t = 1, \ldots, n^m$. For brevity, we consider only one day and leave out the day index for the remainder of this section. Each stage consists of three components: the states, the decision, and the optimal value.

States. The state, denoted as s_t, keeps accountability of the ASIP inspection times, the failure rate distributions, the current location of the inspection crew, and the time remaining (out of the time budget T), at the beginning of each stage. We denote the collection of ASIP inspection times as a matrix Ξ_t. This contains the ASIP solutions for the next available fraction choices for each subnetwork $\mathcal{T}_k$, $k \in [\![1, K]\!]$. Additionally, the matrix provides the value of $f_{max,k}$ as described in Section 4.2.1. The collection of failure rate distributions for

each subnetwork at the beginning of stage t is denoted as C_t. The location of the inspection crew at the beginning of stage t is either at a service station $s \in \mathcal{Y}$ or access point $b_k \in \mathcal{B}$, and is denoted as l_t. Finally, the time remaining at the beginning of each stage is denoted as r_t. Therefore the stage-t state variable is highly structured as (Ξ_t, C_t, l_t, r_t). This represents a finite state space, since we only consider a finite set of fraction choices for each subnetwork,

Decision. The second component for each stage is a decision variable denoted y_t that governs which node to visit for stage $t+1$, and if it is a subnetwork, what fraction $\frac{f_k}{f_{max,k}}$ to inspect. From a given state s_t, we consider a decision leading to state s_{t+1} as feasible if the inspection crew can visit the selected subnetwork, complete the chosen fractional inspection, and return to the service station from l_{t+1} within the time remaining, such that $r_{t+1} > 0$. We denote the set of feasible decisions from state s_t as $Y_t(s_t)$. Note that this set of feasible decisions also includes the decision to stay at a subnetwork for a successive inspection.

Optimal Value. The final component of the stage is the optimal value of expected failures at s_t, which we denote as $g_t(s_t)$. We solve the stochastic dynamic program on a daily basis by maximizing the total expected failures over stages $t, t+1, \ldots, n^m$. The objective is given by the Bellman-style recursion equation of the form:

$$g_t(s_t) = \max_{y_t \in Y_t(s_t)} \left\{ F(s_t, y_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, y_t) g_{t+1}(s_{t+1}) \right\} \tag{4.15}$$

where function $F(s_t, y_t)$ represents the expected failures per stage given s_t and y_t, and $P(s_{t+1}|s_t, y_t)$ represents the conditional probability of a state s_{t+1} given a decision y_t in state s_t. Recall from Section 4.2.2 that the decision to inspect a fraction $\frac{f_k}{f_{max,k}}$ of a subnetwork associated with b_k, $k \in [\![1, K]\!]$ will lead to an updated distribution based on the possible values of d_k and x_k. Therefore, $P(s_{t+1}|s_t, y_t)$ is equivalent to $P_{f_k^t}(D_k = d_k \mid x_k)$ from Equation (4.13), where we use f_k^t for the stage, instead of f_k^m. Specifically, the decision y_t for the subnetwork to visit and fraction to inspect, prescribes the values of d_k and x_k to consider based on the failure rate distribution of subnetwork k at stage t.

For each day, we solve the stochastic dynamic program using a recursive algorithm that computes $g_t(s_t)$ by expansion of a decision tree. To calculate $g_t(\cdot)$, we rely on recursive calls for $g_{t+1}(\cdot), g_{t+2}(\cdot), \ldots g_n(\cdot)$. Given an initial state s_t, which begins at the service station, and a set of feasible decisions $Y_t(s_t)$, our optimal policy is to visit the fraction of the subnetwork that maximizes the expected number of failures. From this decision, we use the actual realized failures x_k to determine the updated failure distribution for s_{t+1}. We continue this policy at each stage as long as there are feasible decisions available. At the end of each day, we carry forward the state (Ξ_n, C_n, l_n, r_n) to be part of the first stage of the following day, and reset r_t to the time budget. We continue this procedure until inspection of all of the subnetworks is complete.

$$\underset{y,w,z}{\text{maximize}} \quad \sum_{k\in\mathcal{B}} \mathbb{E}(D_k)L(R_k) \times \frac{1}{f_{max,k}} \times \sum_{f_k=0}^{f_{max,k}} f_k z_k^{f_k}$$

Subject to

$$\sum_{j\in\delta(s)} y_{sj}^s \le n_s, \qquad \forall s \in \mathcal{Y} \quad (4.16)$$

$$\sum_{j\in\delta(i)} y_{ij}^s = \sum_{j\in\delta(i)} y_{ji}^s, \qquad \forall s \in \mathcal{Y}, i \in \mathcal{B} \quad (4.17)$$

$$\sum_{s\in\mathcal{Y}} \sum_{i\in\delta(k)} y_{ik}^s \le 1, \qquad \forall k \in \mathcal{B} \quad (4.18)$$

$$w_{sj} = \gamma_{sj} y_{sj}^s, \qquad \forall s \in \mathcal{Y}, j \in \delta(s) \quad (4.19)$$

$$0 \le w_{ij} \le T \sum_{s\in\mathcal{Y}} y_{ij}^s, \qquad \forall (i,j) \in \mathcal{J} \mid i \notin \mathcal{Y}, j \notin \mathcal{Y}, s \in \mathcal{Y} \quad (4.20)$$

$$0 \le w_{js} \le T y_{js}^s, \qquad \forall s \in \mathcal{Y}, \forall j \in \delta(s) \quad (4.21)$$

$$\sum_{j\in\delta(k)} w_{kj} = \sum_{j\in\delta(k)} w_{jk} + \sum_{j\in\delta(k)} \gamma_{kj} \sum_{s\in\mathcal{Y}} y_{kj}^s + \sum_{s\in\mathcal{Y}} \gamma_{ks} y_{ks}^s + \sum_{f_k=0}^{f_{max,k}} \xi_{k,f_k}^* z_k^{f_k}, \qquad \forall k \in \mathcal{B} \quad (4.22)$$

$$1 - z_k^0 \le \sum_{s\in\mathcal{Y}} \sum_{j\in\delta(k)} y_{jk}^s, \qquad \forall k \in \mathcal{B} \quad (4.23)$$

$$\sum_{f_k=0}^{f_{max,k}} z_k^{f_k} = 1, \qquad \forall k \in \mathcal{B} \quad (4.24)$$

$$y_{ij}^s \in \{0,1\}, \qquad \forall s \in \mathcal{Y},\ \forall i \ne j \in \{s\} \times \mathcal{B} \cup \mathcal{B}^2 \cup \{s\}$$

$$z_k^{f_k} \in \{0,1\}, \qquad \forall k \in \mathcal{B},\ f_k = 0, \ldots, f_{max,k}$$

$$w_{ij} \ge 0, \qquad \forall (i,j) \in \mathcal{J}$$

Mixed Integer Program Approach. Although the stochastic dynamic program can provide high quality solutions, its direct application is limited due to its complexity. In particular, each decision leads to a different failure rate probability distribution and consequently,

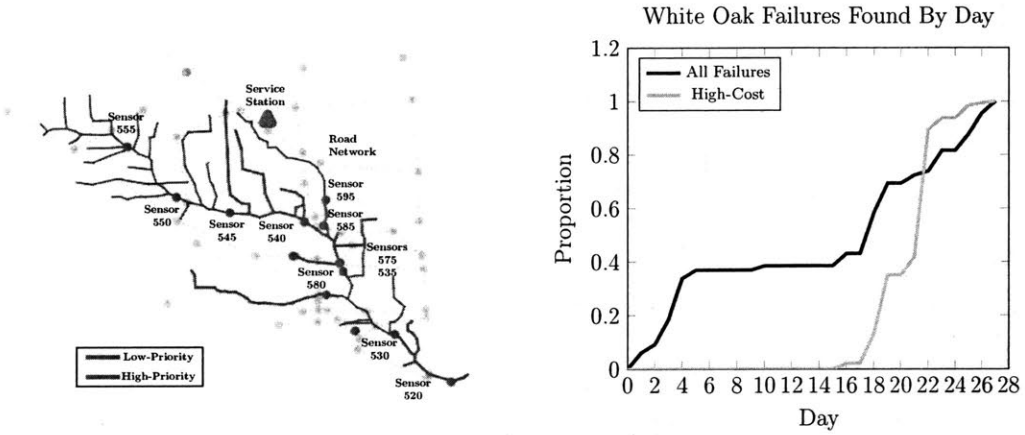
the number of possible states that must be considered grows exponentially. Therefore, we present an alternate certainty equivalent solution approach using an MIP formulation to solve at the beginning of each day. There are two main differences between this MIP approach and the stochastic dynamic program. First, we consider the $\mathbb{E}(D_k)$ as opposed to the entire range of values in the failure rate interval from $P(D_k)$. Secondly, the MIP approach is closer to a myopic policy for each day since each subnetwork can only be inspected once and the routing decisions are made at the beginning of each day.

For each service station $s \in \mathcal{Y}$ and for every pair of locations (i, j) such that $i \neq j \in \mathcal{B} \cup \{s\}$, we define a binary decision variable y_{ij}^s that is equal to 1 if an inspection crew from service station s travels along the road network connecting i to j, and 0 otherwise. We define w_{ij} to represent the time an inspection crew arrives at location j (coming from location i). This time includes both the travel time between service stations and access points, along with the ASIP inspection time for the subnetwork associated with access point i. We define another binary variable $z_k^{f_k}$ that is equal to 1 if a fraction choice index f_k is chosen for subnetwork $k \in \mathcal{B}$. For all subnetworks k on a given day, we consider $f_{max,k}$, R_k, Ξ and $P(D_k)$ as inputs to the MIP formulation, which is given in constraints (4.16)-(4.24).

Constraints (4.16)-(4.18) define the classic network flow constraints, while constraints (4.19)-(4.22) keep track of the arrival times for each inspection crew. Specifically, constraint (4.16) ensures that no more than n_s inspection crews leave service station s. Constraint (4.17) ensures that if an inspection crew from a service station enters a subnetwork, it also leaves. Constraint (4.18) ensures that only one inspection crew can visit each subnetwork. Constraint (4.19) initializes the time traveled by the inspection crew if it departs from a service station. Constraints (4.20)-(4.21) make sure that the arrival time is less than the time budget if travel takes place and 0 otherwise. Constraint (4.22) updates the arrival time by taking into account the inspection crew travel time as well as ξ^*_{k,f_k}. The term $\sum_l \xi^*_{k,f_k} z_k^{f_k}$ for subnetwork $\mathcal{T}_k$ gives the time to inspect the chosen fraction index. Constraint (4.23) forces z_k^0 to be 1 if subnetwork k is never visited. Finally, constraint (4.24) ensures that only one fraction index is chosen for each subnetwork.

4.3 Computational Results for Drainage Network

In this section, we solve the ASIP and PIRP introduced in Section 4.2 and implement the routing solutions in a computational study involving the White Oak Bayou region. The goals of this study are threefold: to (i) analyze the performance of our proposed UAS-enabled inspection approach in comparison to the current ground-based approach, (ii) evaluate the relationship between the amount of data integration and the quality of the PIRP solution, and (iii) extract practical insights from our results to improve inspection operations.



(a) Service Station, Flood Sensors, Subnetworks, and Road Network

(b) Failures and High-Cost Failures Found by Day

Figure 4-6: White Oak Bayou Region.

The White Oak Bayou region, located near downtown Houston, consists of 146 miles of both natural and man-made (concrete-lined) waterways monitored by 11 flood sensors. We consider one service station, with an inspection crew equipped with two UAS. We choose a conservative parameter of 3 hours for the crew's daily time budget, which we assume is a continuous block of time. We use average crew vehicle speeds along the primary and secondary roads (35 mph and 25 mph, respectively). During the dates coinciding with Hurricane Harvey, from 26 to 29 August 2017, we use flood sensor readings to generate 6 high-priority and 5 low-priority subnetworks using our approach described in Section 4.1.2. The inspection operations commence a day after, on 30 August 2017 due to adverse weather conditions that precluded ground movement prior to that time. Figure 4-6a depicts our

study region along with the flood sensors, priority subnetworks, service station, and road network. The actual inspection results for the White Oak Bayou study region are shown in Figure 4-6b. Note that the results of this study region are consistent with the results of the entire FCD region shown in Figure 1-2. In particular, while debris failures in general were found early in the inspection timeline, the failures with high repair costs were not. We use three quantitative metrics to evaluate inspection performance, driven by Figure 4-6b: the overall duration required to complete inspection, the rate of cumulative failures found over time, and the cumulative value of high-cost failures found over time.

To solve the ASIP described in Section 4.2.1, we consider a hybrid UAS (gas and electric) and allow Beyond Visual Line of Sight (BVLOS) operations. By allowing this, the entirety of a subnetwork can be inspected without the need to refuel or recharge, improving the cost effectiveness [8]. The total edge distances of the subnetworks fall well below the 100 mile maximum range of a hybrid UAS [54]. To address privacy issues, flight operations are restricted to a narrow width within the ROW. Finally, to ensure high resolution imagery, we assume an operating altitude of 400 ft Above Ground Level (AGL) and a conservative 5 mph for the inspection cruise speed based on comparable UAS inspections [34]. We assume a conservative speed of 15 mph when bypassing, and set η in constraint (4.9) to 0.7, based on the bypass speed.

To determine the minimum appropriate level of data required for early identification of failures and high-cost failures, we consider different levels of data integration in the PIRP. In contrast to the non-adaptive MIP solution approach that considers expected failure rates, and the adaptive stochastic dynamic program that considers the full interval of possible failure rates as described in Section 4.2.2, we also consider two simpler solution approaches. First, we consider an *Unadvised* approach, which assumes no data input from either the flood sensor or predictive model at t_0. Specifically, we set initial failure rates to be the same normalized value for each subnetwork. Second, we consider a *Hi-Low Priority* approach, which only accounts for data in the form of flood sensor alerts at t_0. Specifically, these alerts act as a binary classifier to label each subnetwork as high- or low-priority. We use two classes of normalized failure rates based on this classification. We summarize our four approaches in

Table 4.3. Note that the approaches are ordered, where the Unadvised approach represents the least data integration and the stochastic dynamic program represents the most. Hereafter we refer to the stochastic dynamic program as the *Stochastic* approach.

We solve the first three approaches with the MIP presented in Section 4.2.3. For the Unadvised and Hi-Low Priority approaches, failure rates are updated and appropriately scaled based on the cumulative number of failures and inspected lengths. For the MIP and Stochastic approaches, failure rates are updated according to the Bayesian update procedure described in Section 4.2.2. In the remainder of this section, we describe how to compute each metric, and discuss the results using the four approaches in Table 4.3.

Table 4.3: Summary of Approaches Reflecting Level of Data Integration

Name of Approach	Flood Sensor Alert Info	Failure Rate Distribution
Unadvised	×	×
Hi-Low Priority	✓	×
Non-Adaptive MIP	✓	✓(Expected Value)
Adaptive Stochastic	✓	✓(Full Distribution)

4.3.1 Inspection Duration

First, we compare the inspection duration using our proposed approach to the actual timeline. To compute this metric, we solve the PIRP using the four approaches for the level of data integration, and record the number of days required for complete inspection of each subnetwork. Recall that the total duration is a conservative estimate based on selected parameters for the time budget, aerial sensor speed, and inspection crew vehicle speed. Figure 4-7 compares the timeline of our approach to the actual recorded ground-based inspection dates following Hurricane Harvey. Under the alert period, we show the duration used to calculate the total volume over the TOB for each flood sensor. For example, Flood Sensor 520 reported a TOB overflow on 26 August at 1130 hours and the water elevation receded below the TOB on 29 August at 1200 hours. We also include unalerted flood sensors since their associated subnetworks also require inspection.

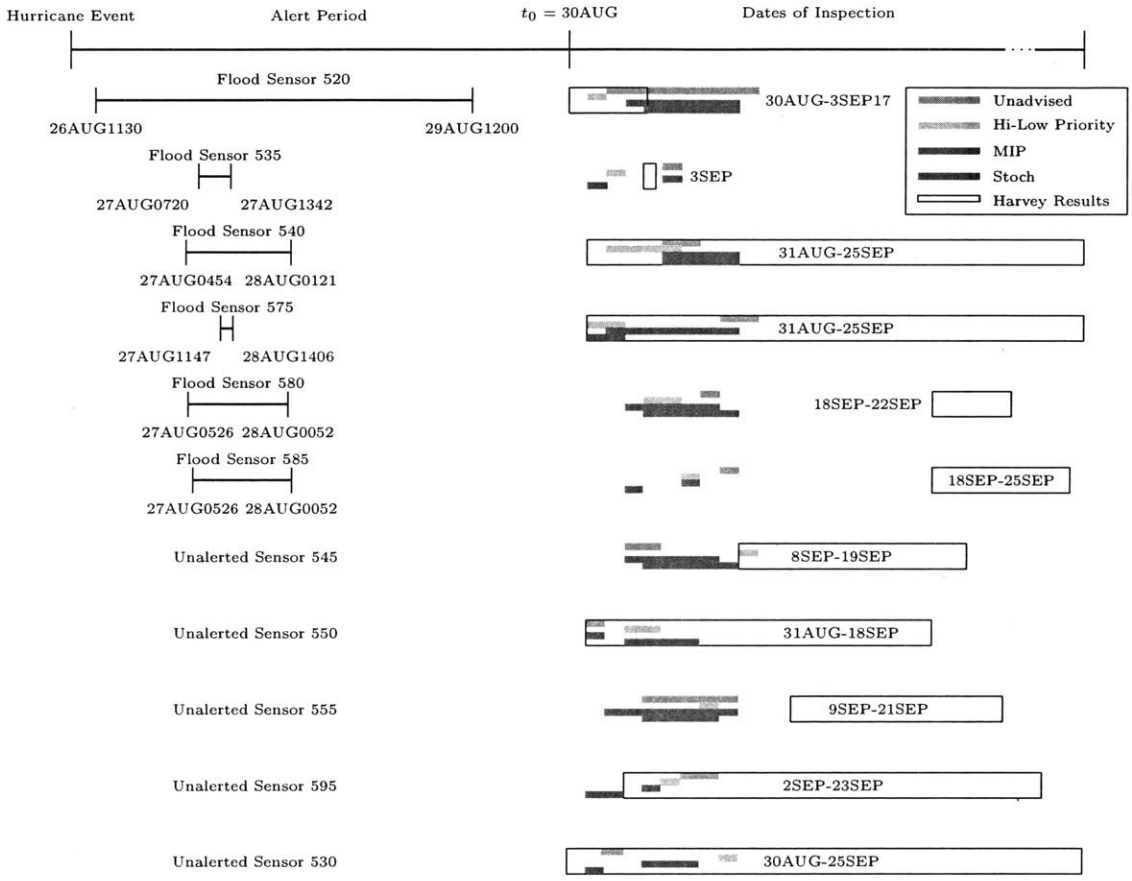


Figure 4-7: Timeline of PIRP inspection results compared to ground-based inspection times. Since subnetworks were not considered in the original inspection, the actual inspection dates we depict align with when the edges in the subnetwork were inspected. For each subnetwork corresponding to a flood sensor, the horizontal span of each solution approach represents the days elapsed from the first inspection to the last.

Based on our results, we achieve a conservative 67% decrease in inspection time (9 days compared to 27) for the first two approaches (Unadvised and Hi-Low Priority) and a 70% decrease (8 days compared to 27) for the last two approaches (MIP and Stochastic). Ground-based inspections cost an estimated $20K per day based mostly on the cost of crews, so the reduction of 19 days using our approach could lead to a conservative savings of $380K from personnel costs alone. Since the UAS inspection time accounts for the majority of the time savings, this duration metric serves as a value proposition for the integration of UAS.

Next, we observe that each solution approach exhibits a different sequence of subnetwork inspections in Figure 4-7. Naturally, the Hi-Low Priority approach attempts to visit all of the high-priority subnetworks as soon as possible. It is important to note that the high- and low-priorities assigned to the subnetworks in this case do not entail a mandatory precedence; that is, all high-priority subnetworks do not need to be inspected before low-priority subnetworks. This is important since we must also account for crew travel time and UAS inspection time. We find that the Stochastic approach exhibits the most exploratory behavior, which can be characterized by the number of fractional inspections completed over all subnetworks. The resulting 18 fractional visits follow the expected behavior in balancing travel time and inspection time. The MIP approach has the next highest fractional inspections at 15. On the other hand, the Hi-Low Priority approach exhibits the lowest exploratory behavior with only 8 fractional visits.

4.3.2 Rate of Failures

Next, we analyze the rate of cumulative debris failures found over time. Using the PIRP solution, this metric accounts for the daily number of actual failures found over the edges inspected by UAS.

We again use the four approaches to solve the PIRP and compare with the actual inspection results. In the actual post-Harvey inspection timeline shown in Figure 4-8, we note a distinct plateau between days 5 and 15, when only one failure was identified, reflecting

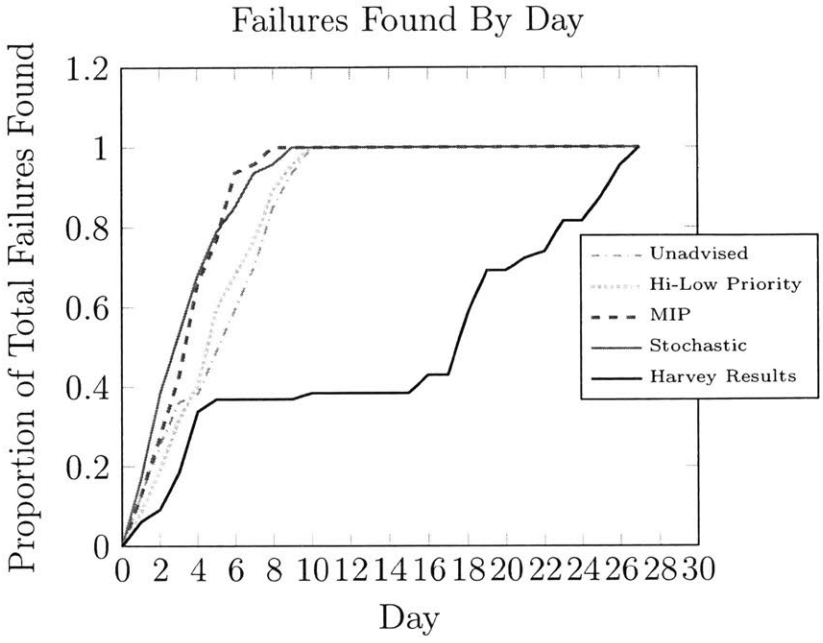


Figure 4-8: Total Cumulative Failures Found By Day Compared to Actual Hurricane Harvey Results

misemployment of limited inspection resources. As expected, we find that the Stochastic approach performs the best in terms of total area under the curve, followed closely by the MIP approach, with a difference of 3%. This is a direct result of the objective function, which is to maximize the expected number of failures over each day. The Unadvised approach performs the worst when compared with the others, but still offers a significant improvement over ground-based inspection due to the compressed inspection timeline.

We note that the variability between approaches is a result of choosing different subnetworks and fractions to inspect for each day. The Unadvised approach tends to visit larger subnetworks earlier, which generally contain a higher number of failures in this study region. This is a result of treating each subnetwork the same with respect to the failure rate, leaving the total distance of a subnetwork to be the main differentiator. The Hi-Low Priority approach on the other hand, places an emphasis on visiting the high-priority subnetworks before the larger subnetworks, resulting in a nearly inverse result. The Stochastic and MIP approaches perform well because of a more balanced approach to inspect both high- and

low-priority subnetworks using updated failure rate distributions.

4.3.3 Rate of Failure Costs

It is important to reiterate that our choice of maximizing the number of expected failures for the objective is a proxy measure for the more important objective, which is the timely identification of high-cost failures. Recall that high-cost failure identification can be completed only as a post-evaluation, after qualified engineers evaluate imagery data from inspections.

To determine the rate of high-cost failures found over time, we focus on identifying failures with repair costs of higher than \$10K (50th percentile of all failure costs). Using the PIRP solution, we account for these costs for failures found along the edges inspected by UAS.

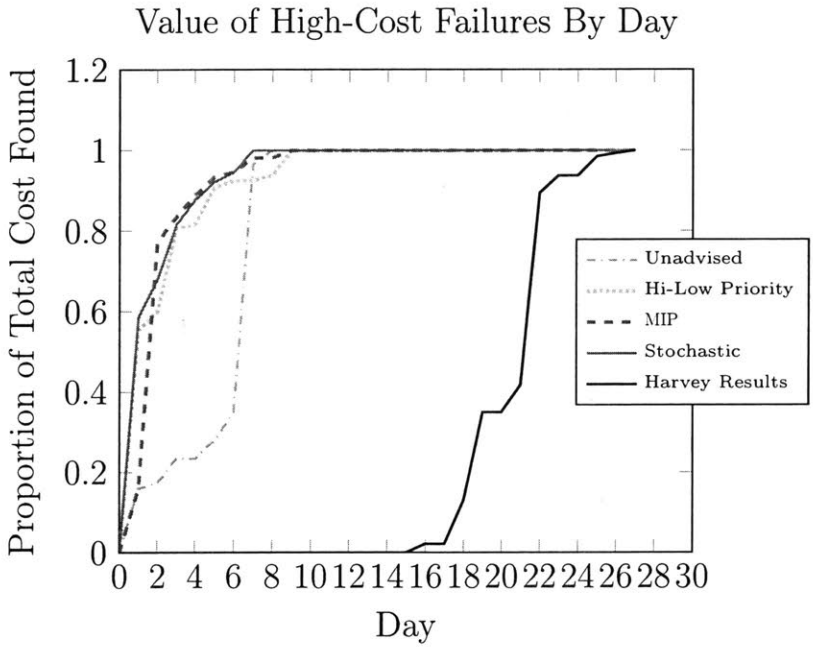


Figure 4-9: Cumulative Value of High-Cost Failures By Day Compared to Actual Hurricane Harvey Results

Inspecting the PIRP solutions using the four approaches, we make the following observations based on the results shown in Figure 4-9. First, we note that actual Harvey inspection results did not find the first high-cost failure until halfway through the inspection timeline

on Day 16, implying a misallocation of inspection efforts. The Unadvised curve is almost parallel to these actual inspection results, but still offers an improvement attributed mostly to faster UAS speeds. Using the area under the curve, the Stochastic approach offers an improvement of $4.3M over the Unadvised approach, or a 41% improvement. We also note that the benefits of integrating flood sensor alert information (Hi-Low Priority, MIP, and Stochastic approaches) are mostly captured earlier in the inspection timeline. This is a direct result of data integration, which ensures that the high-priority subnetworks are visited in the beginning. The total repair costs in these subnetworks amount to $622,520, mostly for concrete lining failure expenses associated with project sites affected by Harvey.

Most surprisingly, we find that using a partition of the network based solely on flood sensor alerts (i.e., the Hi-Low Priority approach) can assist in identifying these high-cost failures at a rate close to the Stochastic approach. The value of the Hi-Low Priority over the Unadvised approach is $3.8M, representing a 39% improvement in value. Moreover, the Hi-Low Priority approach only differs by 4.2% and 0.34% from the stochastic and MIP approaches, respectively. Therefore, although capturing the most high-cost failures is not directly built into our model, the net effect is the capture of high-cost failures in post-evaluation based on assigned priorities. This result lends credence to our approach to partition the network into high- and low-priority subnetworks, suggesting that high-priority subnetworks contain a disproportionate amount of high-cost failures.

Based on our computational results, we summarize our key observations for an FCD to consider in post-storm inspection operations:

Observation 1. (Data Collection) The failure rate in each subnetwork and its relationship with the total volume over the TOB is statistically significant.

A key finding in our research is the relationship between the failure rate of a subnetwork and the total volume over the TOB at the flood sensor location. Our statistical model coefficient for this feature indicates a multiplicative effect on failure rates. This suggests that the total volume over the TOB, calculated from the elevation-discharge curve, can be used as a key variable to achieve a more accurate forecast of failure rates. We show that

this can provide the added benefit of timely failure identification. In Figure 4-8 both of our approaches that integrate information from the failure rate distribution (i.e., the MIP and Stochastic approaches) outperform the simpler approaches that do not integrate this information.

Observation 2. (Method) The Hi-Low Priority approach offers a practical procedure to use for the timely identification of high-cost failures.

Another key finding in our research is that the alerted subnetworks disproportionately contain more high-cost failures than non-alerted, or low-priority subnetworks. To determine whether or not our results were specific to the White Oak Bayou, we also inspect the number of high-cost failures in two other regions (Greens and Brays Bayou). Whereas the Greens Bayou is 308 miles in length and consists of more natural channels, the Brays Bayou is 121 miles and consists of more man-made channels. Using our subnetwork approach described in Section 4.1, we first use flood sensor readings to partition these network into high- and low-priority subnetworks. Figure 4-10 shows the count of high-cost failures within these subnetworks.

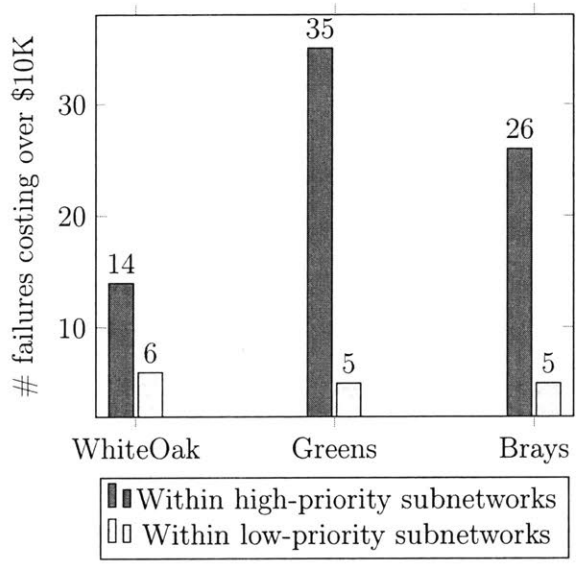


Figure 4-10: High-Cost Failures Study With Different Regions

We find that our claim is supported very well; using a paired t-test, high-priority sub-

networks contain a significantly higher frequency of high-cost failures (p-value of 0.045). Furthermore, we find that this result is independent of total edge distance in the subnetworks. That is, our claim is supported even for regions where the total edge distance in low-priority subnetworks is substantially larger than the total edge distance of high-priority subnetworks.

Observation 3. (Approach) Total inspection time can be reduced substantially with the integration of UAS.

By comparing our UAS-enabled inspection timeline with the current ground-based inspection timeline, we provide a measurable value proposition with respect to efficiency gained from the UAS. Our proposed approach suggests a conservative 67-70% decrease in total inspection time compared to ground-based inspection as shown in Figure 4-7. We demonstrate that our approach can lead to a cost-savings of $380,000 just from personnel expenses alone. It is important to note that this cost saving estimate does not include the potential reductions in the severe societal and economic losses that the proposed approach could enable. In other words, this cost saving only quantifies reduction in the inspection costs, which would be likely dwarfed by the reductions in the costs to local economy, public health, and recovery efforts. By compressing the inspection timeline, our approach would enable earlier repair schedules and shorter delays, to avoid the hidden costs of deferred maintenance.

In summary, there are three practical implications of our study for drainage network agencies. First, we find that partitioning of the network into high- and low-priority subnetworks at the beginning of the inspection timeline can offer the best value for timely identification of high-cost failures with a low computational requirement. In fact, our results in solving the PIRP show that the additional effort required to integrate a full statistical model may not be worth the marginal benefits. This is of particular interest to drainage networks, since the computational burden of using a statistical approach or stochastic dynamic programming model can be prohibitive in post-disaster environments due to time, budget, or resource considerations. Second, our validation of the proposed total volume over the TOB in our computational study provides valuable insight into how discharge and overflow can assist

in forecasting the number of failures. FCDs can easily adopt this metric to be captured by hydrologists and sent to inspection crews for future storm events. Finally, as drainage network agencies continue to look towards adopting UAS as a cost-effective tool for collecting maintenance-related data and imagery, our study provides conservative benchmarks using realistic UAS route planning algorithms that can be used to justify investment into this technology.

Chapter 5

Conclusion

5.1 Summary of Results

This thesis studied the problem of integrating data from two types of sensors (fixed and aerial) into post-disaster infrastructure damage assessments in order to reduce time and cost of inspection. We introduced important features of the UAS platform and the operating environment in the development of MIP formulations for the ASIP and PIRP, where the overall objective is to minimize the maximum time elapsed from time of failure alert to time of failure identification, over all subnetworks.

For gas pipeline networks, we introduced UAS monitoring locations used to observe network components. Given the significant computation time required for larger instances of the ASIP, we developed a scalable heuristic based on the weighted set cover problem to limit the number of monitoring locations to consider. We showed that our heuristic can achieve quality solutions for 2 and 3 UAS within seconds. From a theoretical standpoint, we uncovered an important ratio between intra-subnetwork travel time (i.e., within the subnetworks) and the inter-subnetwork travel time. This shows that more time savings can be achieved from UAS integration if subnetworks are larger and vehicle distances to the subnetworks are shorter. We also studied the effect of stochastic travel times on the ASIP

solution based on performance metrics such as distribution of UAS exploration times and probability of missed network components. We show that accounting for stochasticity can significantly increase the overall UAS optimal inspection time and the possibility of missed network components. This points to the importance of choosing the right UAS platform as well as conservative route planning to avoid costly setbacks from unidentified failures. Taken as a whole, we demonstrate that UAS-based inspection can lead to significant cost and time savings in comparison to conventional, purely ground-based, operations.

For drainage networks, we used geomorphological and fixed sensor data to develop a statistical model, determining the order of subnetworks to inspect. UAS provided a learning mechanism in the form of identified failures to update beliefs about failure rates in a subnetwork for subsequent inspection crew routing decisions. To quantify the benefits of our approach, we developed three key performance metrics: the total inspection time, the cumulative number of failures found over time, and the cumulative value of high-cost failures found over time. We showed that our analytics-driven inspection approach can substantially reduce overall inspection times by a conservative 67-70%, and most importantly, assist in the timely inspection of high-cost failures. Surprisingly, we found that the relatively simple approach of partitioning the network into high or low-priority subnetworks based on flood sensor alerts alone resulted in good performance. This practical approach provided results similar to our stochastic dynamic program approach in finding high-cost failures. This result was primarily driven by a variable that we called the total volume over TOB. From an operational perspective, this result is promising for implementation, since our results suggest that the marginal benefits to be gained from a stochastic model may not be worth the computational burden on the network agency.

Our drainage network study has both direct and indirect societal and economic impacts as informed by our direct collaboration with a certain Flood Control Agency. We prescribe an innovative and practical analytics-driven inspection approach that integrates flexible diagnostic information from fixed and aerial sensors to minimize the loss of property and adverse health effects. Long-term indirect benefits of interest to agencies like the Federal Emergency and Management Agency (FEMA) include reduced taxpayer costs for recovery operations

and lessening the psychological impact on flood victims.

5.2 Future Work

For the ASIP, one can also consider an adaptive model that can adjust to a dynamic operating environment. Unexpected changes in monitoring requirements, airspace restrictions, communication strength, and obstacles in the operating environment can impact the UAS inspection times and performance. One approach could be to formulate the ASIP as a Markov Decision Process to address dynamic travel times arising from these changes. This could provide valuable insights to inform and support UAS policy decisions to reduce UAS exploration time.

For the PIRP, an extension to this work could incorporate uncertainty in travel times for the inspection crews. For example, one could leverage data from social media, traffic cameras, flood sensors, and elevation data to build a statistical model that could estimate travel time, which could be considered infinite along an edge, if flooded or blocked by debris obstacles. One can also consider using a criticality factor for each subnetwork in the objective function based on the proximity to population centers or environmentally sensitive areas that would indicate higher priority over others for earlier inspections.

For our work on drainage networks, an important research direction is to determine how our approach can be applied to other FCDs from other states, since they utilize similar flood sensors from the United States Geological Survey (USGS) to monitor their drainage networks. It would also be of interest to see if our approach can be used for smaller scale storm events.

As part of future work, we suggest incorporating optimal resource allocation with our work, considering the high operational costs stemming from sub-optimal allocation of inspection crews [5]. For example, prior to the realization of uncertainty in a two-stage stochastic optimization, this approach can help determine the optimal number and location of inspec-

tion crews and UAS at various service stations.

Appendix A

Appendices of Chapter 3

A.1 Results for 7 Subnetworks Using 3 UAS

Along with testing subnetworks of various sizes for 2 UAS, we also test our heuristic to see how it would perform using 3 UAS. The results show a 0% optimality gap for subnetworks with nodes ranging from 5 to 17.

Table A.1: Results of ASIP MIP exact solutions for 3 UAS compared to the heuristic.

Subnetwork			MIP		Heuristic		Optimality
No.	Nodes	Edges	Obj	Time (sec)	Obj	Time (sec)	Gap
1	5	6	0.209	0.02	0.209	0.014	0%
2	11	10	0.413	2.04	0.413	0.022	0%
3	16	16	0.586	770	0.586	4.122	0%
4	18	17	0.706	34	0.706	0.187	0%
5	15	18	0.857	14851	0.857	0.099	0%
6	16	20	0.825	54	0.825	0.86	0%
7	17	22	0.745	36545	0.745	0.134	0%

A.2 Example Heuristic Solution

To illustrate the 5 steps in our heuristic, we solve an example problem. In this example, we solve the ASIP using our heuristic for a subnetwork consisting of 36 nodes (monitoring locations) and 37 edges (components). We set τ_{max} to 1 hour and u to 2 UAS.

Step 1. Weighted Set Cover. The optimal solution of the weighted set cover is $\mathcal{S} = \{1, 3, 5, 7, 8, 12, 13, 14, 18, 19, 20, 22, 24, 26, 30, 32\}$, of which 20 is the base node. This step immediately reduces the overall problem size by limiting the number of monitoring locations to consider from the original 36 to 16.

Step 2. Initial Route Construction. A portion of the ordered savings list generated from the Clark & Wright Savings heuristic for this subnetwork is shown in Table A.2. For the purpose of illustration, only the first 9 rows are shown. Each row shows the time in hours that could be saved by visiting the node pair in succession as opposed to one at a time.

Table A.2: Savings List from the Clark and Wright Parallel Savings Algorithm

Pair	Savings
(1,8)	0.61
(12,13)	0.55
(1,7)	0.40
(7,8)	0.40
(24,26)	0.35
(22,24)	0.33
(22,26)	0.32
(26,32)	0.31
(19,24)	0.29

The following set $\mathcal{P}$ of initial simple routes is created following the three cases described earlier: $(20, 7, 1, 8, 5, 3, 20), (20, 14, 12, 13, 20), (20, 22, 24, 19, 18, 26, 30, 32, 20)$. Each simple route meets feasibility requirements (the duration of the routes are 0.84, 0.61, and 0.95 hours respectively). Additionally, note that $|\mathcal{P}| > u$.

Step 3. Improvement Procedures. All three improvement procedures provide incremental improvements to one or more simple routes. In the relocation procedure, node 19 is deleted from the third simple route and inserted into the second. The maximum travel

time is reduced from the 0.95 hours associated with the initial route pair $(20, 14, 12, 13, 20)$, $(20, 22, 24, 19, 18, 26, 30, 32, 20)$ to 0.91 hours associated with the new route pair $(20, 19, 14, 12, 13, 20)$, $(20, 22, 24, 18, 26, 30, 32, 20)$. The final routes after this procedure are: $(20, 7, 1, 8, 5, 3, 20)$, $(20, 19, 14, 12, 13, 20)$, $(20, 22, 24, 18, 26, 30, 32, 20)$.

In the exchange procedure, three exchanges take place between the second and third simple routes. First, nodes 18 and 19 are exchanged, followed by 18 and 26, and then 26 and 30. The final routes after this procedure are: $(20, 7, 1, 8, 5, 3, 20)$, $(20, 30, 14, 12, 13, 20)$, $(20, 22, 24, 19, 18, 26, 32, 20)$. The duration of the routes are now 0.84, 0.67, and 0.83 hours respectively.

In the 2-Opt procedure, one valid 2-Opt move for the third simple route results in a reduction in travel time from 0.83 to 0.67. The final routes after this procedure are: $(20, 7, 1, 8, 5, 3, 20)$, $(20, 30, 14, 12, 13, 20)$, and $(20, 18, 19, 24, 22, 26, 32, 20)$. The duration of the routes are now 0.84, 0.67, and 0.67 hours respectively.

Step 4. Route Combination. This step entails trying to combine the shorter duration routes $(20, 30, 14, 12, 13, 20)$ and $(20, 18, 19, 24, 22, 26, 32, 20)$ with the edge node pairs: $(30, 18)$, $(30, 32)$, $(13, 18)$, or $(13, 32)$. Out of the 4 edge node pairs, $(13, 18)$ offers the highest savings, so we attempt to create a new route $(20, 30, 14, 12, 13, 18, 19, 24, 22, 26, 32, 20)$. In this case, this combined route exceeds τ_{max} and therefore we end up with a multi-trip route. The final routes after this procedure are: UAS 1 $\rightarrow$ $(20, 7, 1, 8, 5, 3, 20)$, and UAS 2 $\rightarrow$ $(20, 30, 14, 12, 13, 20)$, $(20, 18, 19, 24, 22, 26, 32, 20)$. The duration of the routes are now 0.84, 1.43 hours respectively.

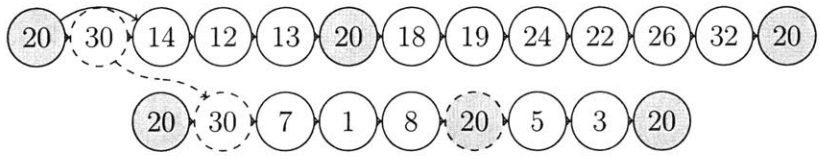


Figure A-1: Illustration of the Relocation with Base Insert Step. With this pair of routes, the top route is the donor route with the larger cumulative travel time (which includes an interim base visit). Node 30 is selected to be moved from the donor route into the recipient route prior to node 7. Since the recipient route's total travel time was longer than the UAS endurance, a base visit is inserted according to a greedy approach.

Step 5. Relocation with Base Insert. In Fig. A-1, we show the result of the last step in our heuristic. In this step, node 30 is relocated. A base visit is inserted due to infeasibility from endurance limitations. While this results in an increase in the cumulative travel time for the recipient route from 0.84 to 1.34, it results in an overall decrease in the maximum travel time *over all routes*, from 1.43 to 1.37 hours. The final routes are UAS 1: $(20, 30, 7, 1, 8, 20)$, $(20, 5, 3, 20)$, UAS 2: $(20, 14, 12, 13, 20)$,$(20, 18, 19, 24, 22, 26, 32, 20)$. This completes the heuristic; the final maximum duration route is 1.37 hours.

Appendix B

Appendices of Chapter 4

B.1 Elevation and Discharge

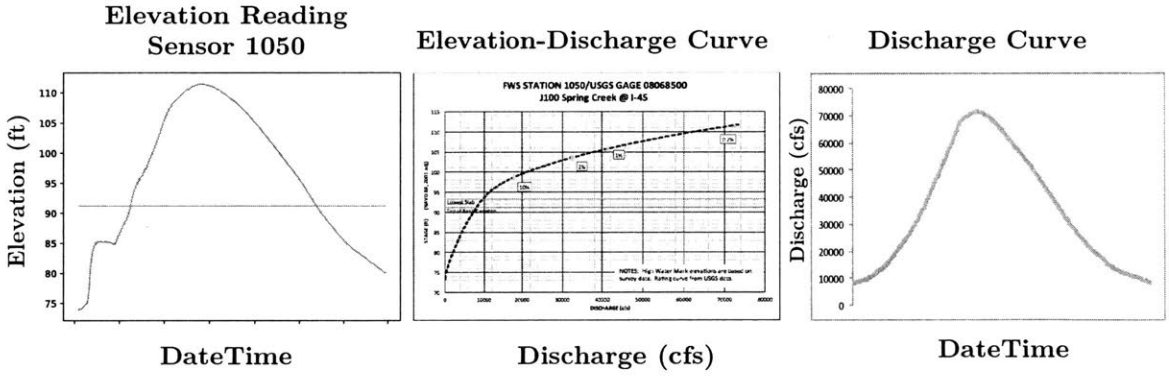


Figure B-1: Example Elevation and Discharge Curve

To calculate the total volume over the Top of the Bank (TOB), we first get the elevation reading for a sensor available from [38]. For each elevation above the TOB, we determine the discharge using the elevation-discharge curve. This generates a new curve called the discharge curve that shows the discharge (cfs) for the times when the flood sensor reported elevations over the TOB. Using the time stamps from the elevation reading data, we integrate

to get the area under the curve, which is the total volume over the TOB. During Hurricane Harvey, a few flood sensors ended up getting damaged. For these sensors, we use extrapolated elevations from the last known reported measurement and follow our procedure.

B.2 Failure Model

We leverage R statistical software to develop and evaluate our statistical models.

Table B.1: Summary of Count Regression Models for Debris Failures

Poisson Model	Multivar	Univar
(Intercept)	-8.263	-6.832
	(0.495***)	(0.098***)
Low Vol over TOB	0.912	0.461
	(0.481*)	(0.104***)
Med Vol over TOB	0.866	0.994
	(0.452*)	(0.108**)
High Vol over TOB	1.056	0.750
	(0.461**)	(0.113***)
Percentage Natural	-0.290	
	(0.144*)	
Percentage Canal/Ditch	-0.811	
	(0.163**)	
Evergreen Forest	3.280	
	(0.554***)	
Degrees of freedom	7	4
Log-Likelihood	-361.45	-721.59
AIC	738.91	1451.20
10-fold CV MSE	3.14	3.91

***$p < 0.01$
**$0.01 < p < 0.05$
*$0.05 < p < 0.10$

We compare the univariate and multivariate Poisson regression models using two common penalized-likelihood information criteria: the Log-Likelihood and the Akaike Information Criterion (AIC), defined as $AIC = -\ln L + P$ where $\ln L$ is the log-likelihood of the model and P is the number of estimated parameters. We also performed a 10-fold Cross Validation (CV), using different training and test splits to attain an overall mean squared error. Using these metrics, we find that the multivariate model provides a better model. For our purpose,

the multivariate serves to provide narrower prediction intervals.

With both of our models, we applied the log link and so e^{β_j} represents a multiplicative effect of the jth explanatory variable on the response. The signs of the coefficients are intuitive. For example, channels classified as manmade or canal/ditch typically have lower rates of failure. Similarly, the positive coefficients associated with the volume over the TOB variables indicate higher failures rates.

Bibliography

[1] Sidney Marshall Adams and Carol J. Friedland. A Survey of Unmanned Aerial Vehicle (UAV) Usage for Imagery. 2011.

[2] Dino Ahr and Gerhard Reinelt. A Tabu Search Algorithm for the Min-Max k-Chinese Postman Problem. *Computers & OR*, 33:3403–3422, 2006.

[3] Dilek Tuzun Aksu and Linet Ozdamar. A Mathematical Model for Post-Disaster Road Restoration: Enabling Accessibility and Evacuation. *Transportation Research Part E: Logistics and Transportation Review*, 61:56 – 67, 2014.

[4] Nezih Altay and Walter G Green III. OR/MS Research in Disaster Operations Management. *European Journal of Operational Research*, 175(1):475–493, 2006.

[5] Mallik Angalakudati, Siddharth Balwani, Jorge Calzada, Bikram Chatterjee, Georgia Perakis, Nicolas Raad, and Joline Uichanco. Business Analytics for Flexible Resource Allocation Under Random Emergencies. *Management Science*, 60(6):1552–1573, 2014.

[6] J.S. Armstrong. *Principles of Forecasting.* Kluwer Academic Publishers, Boston, 2001.

[7] Igor Averbakh. Emergency Path Restoration Problems. *Discrete Optimization*, 9(1):58–64, 2012.

[8] Avionics International. GE-Owned Avitas Granted First Large Commercial BVLOS Waiver. `https://www.aviationtoday.com/2018/10/19/ge-owned-avitas-granted-first-large-commercial-bvlos-waiver/`, Oct 2018. Accessed: October 23, 2018.

[9] Dimitris J Bertsimas. A Vehicle Routing Problem with Stochastic Demand. *Operations Research*, 40(3):574–585, 1992.

[10] Natashia Boland, LW Clarke, GL Nemhauser, et al. The Asymmetric Traveling Salesman Problem with Replenishment Arcs. *European Journal of Operations Research*, 2000.

[11] Olli Bräysy and Michel Gendreau. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, 39(1):104–118, 2005.

[12] Burak Cavdaroglu, Erik Hammel, John E. Mitchell, Thomas C. Sharkey, and William A. Wallace. Integrating Restoration and Scheduling Decisions for Disrupted Interdependent Infrastructure Systems. *Annals of Operations Research*, 203(1):279–294, Mar 2013.

[13] Melih Celik, Özlem Ergun, and Pinar Keskinocak. The Post-Disaster Debris Clearance Problem Under Incomplete Information. *Operations Research*, 63:65–85, 2015.

[14] Dominique Chabot, Shawn R Craik, and David M Bird. Population Census of a Large Common Tern Colony with a Small Unmanned Aircraft. *PloS one*, 10(4):e0122588, 2015.

[15] Vasek Chvatal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[16] Geoff Clarke and John W Wright. Scheduling of Vehicles From a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581, 1964.

[17] Georges A Croes. A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6):791–812, 1958.

[18] Navid Dadkhah and Bérénice Mettler. Survey of Motion Planning Literature in the Presence of Uncertainty: Considerations for UAV Guidance. *Journal of Intelligent & Robotic Systems*, 65(1-4):233–246, 2012.

[19] George B Dantzig and John H Ramser. The Truck Dispatching Problem. *Management science*, 6(1):80–91, 1959.

[20] Luis E. de la Torre, Irina S. Dolinskaya, and Karen R. Smilowitz. Disaster Relief Routing: Integrating Research and Practice. *Socio-Economic Planning Sciences*, 46(1):88 – 97, 2012. Special Issue: Disaster Planning and Logistics: Part 1.

[21] Chuang Deng, Shengwei Wang, Zhi Huang, Zhongfu Tan, and Junyong Liu. Unmanned Aerial Vehicles for Power Line Inspection: A Cooperative Way in Platforms and Communications. *J. Commun*, 9(9):687–692, 2014.

[22] Department of Energy. Quadrennial Energy Review: Energy Transmission, Storage, and Distribution Infrastructure. `https://www.energy.gov`, April 2015. Accessed Feb 25, 2017.

[23] Pablo A. Maya Duque, Irina S. Dolinskaya, and Kenneth Sörensen. Network Repair Crew Scheduling and Routing for Emergency Relief Distribution Problem. *European Journal of Operational Research*, 248(1):272 – 285, 2016.

[24] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. Arc Routing Problems, Part I: The Chinese Postman Problem. *Operations Research*, 43(2):231–242, 1995.

[25] Sevgi Erdoğan and Elise Miller-Hooks. A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

[26] Ozlem Ergun, Gonca Karakus, Pinar Keskinocak, Julie Swann, and Monica Villarreal. *Operations Research to Improve Disaster Supply Chain Management.* American Cancer Society, 2011.

[27] Reza Faturechi and Elise Miller-Hooks. Measuring the Performance of Transportation Infrastructure Systems in Disasters: A Comprehensive Review. *Sensors Journal, IEEE*, 14(12):4134–4142, Dec 2014.

[28] Federal Aviation Administration. Unmanned Aircraft Systems (UAS) Operational Approval (N 8900.227). `www.faa.gov/documentLibrary/media/Notice`, July 2013. Accessed: Sept 12, 2017.

[29] Federal Aviation Administration. Authorizations Granted via Section 333 Exemptions. `www.faa.gov/uas/beyond_the_basics/section_333/333_authorizations/`, 2016.

[30] FEMA. Remote Sensing in Federal Disaster Operations, 1999.

[31] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[32] Everette Gardner. A Simple Method of Computing Prediction Intervals for Time Series Forecasts. *Management Science*, 34:541–546, 04 1988.

[33] Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic Vehicle Routing. *European Journal of Operational Research*, 88(1):3 – 12, 1996.

[34] Daniel Gillins, Christopher Parrish, Matthew Gillins, and Chase H. Simpson. Eyes in the Sky: Bridge Inspections with Unmanned Aerial Vehicles Final Report SPR 787, 02 2018.

[35] Levy L. Golden, B. L. and R Vohra. The orienteering problem. *Naval Research Logistics*, 34:307–318, 1987.

[36] Ronald L Graham. Bounds for Certain Multiprocessing Anomalies. *Bell Labs Technical Journal*, 45(9):1563–1581, 1966.

[37] Seth D Guikema, Rachel A Davidson, and Haibin Liu. Statistical Models of the Effects of Tree Trimming on Power System Outages. *IEEE Transactions on Power Delivery*, 21(3):1549–1557, 2006.

[38] Harris County Flood Control District. Flood Warning System. `https://www.harriscountyfws.org`, Sept 2017. Accessed: November 1, 2017.

[39] HCFCD. Harris County Flood Control District. `https://www.harriscountyfws.org/About`, Sept 2017. Accessed: November 1, 2017.

[40] Department of the Army Headquarters. *Field Manual 6-0 Mission Command.* Washington, D.C., 2011.

[41] Houston-Galveston Area Council. Land Use and Land Cover Data. http://www.h-gac.com, February 2018. Accessed: February 10, 2018.

[42] Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. Prediction Intervals for Exponential Smoothing Using Two New Classes of State Space Models. *Journal of Forecasting*, 24(1):17–37, 2005.

[43] Wall Street Journal. The Trouble With Inspection Tools for Oil Pipelines. http://www.wsj.com/articles/pipeline-inspection-tools-are-far-from-perfect-1435875737, Jul 2015. Accessed: January 3rd, 2017.

[44] Kentucky Water Resources Research Institute. Water Distribution System Research Database. http://www.uky.edu/WDST/database.html, 2016. Accessed: February 22, 2017.

[45] Parimal Kopardekar, Joseph Rios, Thomas Prevot, Marcus Johnson, Jaewoo Jung, and J Robinson. Unmanned Aircraft System Traffic Management (UTM) Concept of Operations. In *AIAA Aviation Forum*, 2016.

[46] Lydia Krefta. Asset Knowledge and Integrity Management Earthquake Playbook Gas Pipeline Integrity Management Program. 2015.

[47] Gilbert Laporte, Yves Nobert, and Danielle Arpin. *Optimal Solutions to Capacitated Multidepot Vehicle Routing Problems.* Université de Montréal, Centre de recherche sur les transports, 1984.

[48] Gilbert Laporte and Frédéric Semet. *Classical Heuristics for the Capacitated VRP*, pages 109–128.

[49] Richard C Larson and Amedeo R Odoni. *Urban Operations Research.* Number Monograph. Dynamic Ideas, 1981.

[50] Jon C Leachtenauer and Ronald G Driggers. *Surveillance and Reconnaissance Imaging Systems: Modeling and Performance Prediction.* Artech House, 2001.

[51] Andrew Lee, Mathieu Dahan, Andrew Weinert, and Saurabh Amin. Leveraging sUAS for Infrastructure Network Exploration and Failure Isolation. *Journal of Intelligent and Robotic Systems*, 93, 04 2018.

[52] Los Angeles Times. PG&E Admits Its Equipment Likely Sparked California's Most Destructive Wildfire. https://www.latimes.com/local/lanow/la-me-pge-paradise-fire-20190228-story.html, Feb 2019. Accessed: March 1, 2019.

[53] Richard M. Lusk and William H. Monday. An Early Survey of Best Practices for the Use of Small Unmanned Aerial Systems by the Electric Utility Industry. Manual ORNL/TM-2017/93, Oak Ridge National Laboratory, February 2017.

[54] Matheson, R. Hybrid Drones Carry Heavier Payloads for Greater Distances. http://news.mit.edu/2017/hybrid-drones-carry-heavier-payloads-greater-distances-0804, Aug 2017. Accessed: Dec 1, 2018.

[55] Arun Mohan and Sumathi Poobal. Crack Detection Using Image Processing: A Critical Review and Analysis. *Alexandria Engineering Journal*, 2017.

[56] Pal-Stefan Murvay and Ioan Silea. A Survey on Gas Leak Detection and Localization Techniques. *Journal of Loss Prevention in the Process Industries*, 25(6):966–973, 2012.

[57] Roshanak Nateghi, Seth D Guikema, and Steven M Quiring. Comparison and Validation of Statistical Methods for Predicting Power Outage Durations in the Event of Hurricanes. *Risk Analysis: An International Journal*, 31(12):1897–1906, 2011.

[58] New York Times. F.A.A.'s Concerns Hold Up Use of Wildfire Drones. `https://www.nytimes.com/2013/05/22/us/faas-concerns-hold-up-use-of-wildfire-drones.html`, May 2013. Accessed: March 1, 2019.

[59] New York Times. A Cyberattack Hobbles Atlanta and Security Experts Shudder. `https://www.nytimes.com/2018/03/27/us/cyberattack-atlanta-ransomware.html`, March 2018. Accessed: March 1, 2019.

[60] Federal Aviation Administration Office of the Secretary of Transportation. Fact Sheet – Small Unmanned Aircraft Regulations (14 CFR Part 107). `www.faa.gov/news`, 2016.

[61] Office for Coastal Management, National Oceanic and Atmospheric Administration. Hurricane Costs. `https://coast.noaa.gov/states/fast-facts`, November 2018. Accessed: November 29, 2018.

[62] L Otero. Proof of Concept for using Unmanned Aerial Vehicles for High Mast Pole and Bridge Inspections, 2015.

[63] Paul W. Parfomak. DOT's Federal Pipeline Safety Program: Background and Key Issues for Congress (R44201). `https://fas.org/sgp/crs/misc/R44201.pdf`, May 20 2016. Accessed: January 3rd, 2017.

[64] Srinivas Peeta, F. Sibel Salman, Dilek Gunnec, and Kannan Viswanath. Pre-Disaster Investment Decisions for Strengthening a Highway Network. *Computers & OR*, 37:1708–1719, 2010.

[65] PEW. What We Don't Know About State Spending on Natural Disasters Could Cost Us. `www.pewtrusts.org/en/research-and-analysis/reports/2018/06/19`, 2018. Accessed: April 1, 2019.

[66] Patrick Prosser and Paul Shaw. Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems, 1996.

[67] Lian Qi, Zuo-Jun Max Shen, and Lawrence V. Snyder. A Continuous-Review Inventory Model with Disruptions at Both Supplier and Retailer. *Production and Operations Management*, 18(5):516–532.

[68] Lina Sela Perelman, Waseem Abbas, Xenofon Koutsoukos, and Saurabh Amin. Sensor Placement for Fault Location Identification in Water Networks. *Automatica*, 72(C):166–176, October 2016.

[69] Suvrajeet Sen and Yunxiao Deng. *Learning Enabled Optimization: Towards a Fusion of Statistical Learning and Stochastic Optimization.* July 2017.

[70] David Simchi-Levi, William Schmidt, Yehua Wei, Peter Yun Zhang, Keith Combs, Yao Ge, Oleg Gusikhin, Michael Sanders, and Don Zhang. Identifying Risks and Mitigating Disruptions in the Automotive Supply Chain. *Interfaces*, 45(5):375–390, 2015.

[71] Lawrence V Snyder, Zümbül Atan, Peng Peng, Ying Rong, Amanda J Schmitt, and Burcu Sinsoysal. OR/MS Models for Supply Chain Disruptions: A Review. *IIE Transactions*, 48(2):89–109, 2016.

[72] Éric D Taillard, Gilbert Laporte, and Michel Gendreau. Vehicle Routing with Multiple Use of Vehicles. *Journal of the Operational research society*, 47(8):1065–1070, 1996.

[73] Texas Department of Transportation. Roadway Inventory Annual Data. `https://www.txdot.gov/inside-txdot/division/transportation-planning/roadway-inventory.html`, 2016. Accessed: May 20, 2018.

[74] The Economist. Improving Disaster Response Efforts Through Data. `https://expectexceptional.economist.com/improving-disaster-response-efforts-through-data.html`, Mar 2019. Accessed: March 1, 2019.

[75] Brian Tomlin. On the Value of Mitigation and Contingency Strategies for Managing Supply Chain Disruption Risks. *Management Science*, 52(5):639–657, 2006.

[76] U.S. Department of Agriculture. Natural Resources Conservation Service Soils. `https://www.nrcs.usda.gov`, January 2018. Accessed: January 15, 2018.

[77] U.S. Geological Survey. Stream Gaging and Flood Forecasting. `https://pubs.usgs.gov/fs/1995/0209/report.pdf`, June 1995. Accessed: November 1, 2017.

[78] Daniele Vigo. A Heuristic Algorithm for the Asymmetric Capacitated Vehicle Routing Problem. *European Journal of Operational Research*, 89(1):108–126, 1996.

[79] Wall Street Journal. Drones Play Increasing Role in Harvey Recovery Efforts. `https://www.wsj.com/articles/drones-play-increasing-role-in-harvey-disaster-recovery-efforts-1504474194`, Sept 2017. Accessed: December 27 2017.

[80] Xingyin Wang, Stefan Poikonen, and Bruce Golden. The Vehicle Routing Problem With Drones: Several Worst-case Results. *Optimization Letters*, 11(4):679–697, 2017.

[81] Yezhou Wang, Chen Chen, Jianhui Wang, and Ross Baldick. Research on Resilience of Power Systems under Natural Disasters - A Review. *IEEE Transactions on Power Systems*, 31(2):1604–1613, 3 2016.

[82] John Ware and Nicholas Roy. An Analysis of Wind Field Estimation and Exploitation for Quadrotor Flight in the Urban Canopy Layer. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1507–1514. IEEE, 2016.

[83] Andrew Weinert. Civil Air Patrol Dataset for Response and Research (CAPFOR). Technical report, 2019. Working paper.

[84] Weinert, A., Campbell, S., Vela, A., Schuldt, D., Kurucar, J. A Well Clear Recommendation for Small Unmanned Aircraft Systems based on Unmitigated Collision Risk. *AIAA Journal of Air Transportation*, 2018. Accepted for Publication.

[85] Min Wen, Jean-François Cordeau, Gilbert Laporte, and Jesper Larsen. The Dynamic Multi-period Vehicle Routing Problem. *Computers & OR*, 37:1615–1623, 2010.

[86] Kevin W Williams and Kevin M Gildea. A Review of Research Related to Unmanned Aircraft System Visual Observers. *Federal Aviation Administration Final Report*, 2014.

[87] James M Wright. *The Nation's Response to Flood Disasters: A Historical Account.* Association of State Floodplain Managers, 2000.

[88] Melih Çelik. Network Restoration and Recovery in Humanitarian Operations: Framework, Literature review, and Research Directions. *Surveys in Operations Research and Management Science*, 21, 01 2017.