Journal of Intelligent & Robotic Systems (2019) 93:385–413
https://doi.org/10.1007/s10846-018-0838-0

Leveraging sUAS for Infrastructure Network Exploration and Failure Isolation

Andrew C. Lee[1] · **Mathieu Dahan**[2] · **Andrew J. Weinert**[3] · **Saurabh Amin**[1]

Received: 15 October 2017 / Accepted: 9 April 2018 / Published online: 26 April 2018

Abstract

Large-scale infrastructures are prone to simultaneous faults when struck by a natural or man-made event. The current operating procedure followed by many utilities needs improvement, both in terms of monitoring performance and time to repair. Motivated by the recent technological progress on small Unmanned Aerial Systems (sUAS), we propose a practical framework to integrate the monitoring capabilities of sUAS into standard utility repair operations. A key aspect of our framework is the use of monitoring locations for sUAS-based inspection of failures within a certain spatial zone (called a localization set). This set is defined based on the alerts from fixed sensors or customer calls. The positioning of monitoring locations is subject to several factors such as sUAS platform, network topology, and airspace restrictions. We formulate the problem of minimizing the maximum time to respond to all failures by routing repair vehicles to various localization sets and exploring these sets using sUAS. The formulation admits a natural decomposition into two sub-problems: the sUAS Network Exploration Problem (SNEP); and the Repair Vehicle Routing Problem (RVRP). Standard solvers can be used to solve the RVRP in a scalable manner; however, solving the SNEP for each localization set can be computationally challenging. To address this limitation, we propose a set cover based heuristic to approximately solve the SNEP. We implement the overall framework on a benchmark network.

Keywords Unmanned aerial systems · Network inspection and repair operations · Localization · Failure identification · Vehicle routing problems

✉ Andrew C. Lee
aclee@mit.edu

Mathieu Dahan
mdahan@mit.edu

Andrew J. Weinert
andrew.weinert@ll.mit.edu

Saurabh Amin
amins@mit.edu

[1] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[2] Center for Computational Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[3] Massachusetts Institute of Technology Lincoln Laboratory, Lexington, MA 02144, USA

1 Introduction

1.1 Motivation

Monitoring and inspection systems for large-scale infrastructures such as water, natural gas, oil, and electric networks typically rely on pre-installed (fixed) sensors to obtain failure alerts. Timely response to these alerts is critical for minimizing the impact of disruptions, which can otherwise lead to significant economic losses and even loss of life. For example, in the US pipeline industry alone, there were 323 fatalities and 1,337 injuries reported in the last 20 years with an estimated total cost of over $7 billion in damages [28]. Natural gas infrastructures are prone to service disruptions due to leaks and pipeline failures (bursts); these events can be respectively detected by customer call

reports and pressure sensors. In some cases, flow sensors and acoustic sensors are also utilized. However, the technological limitations of fixed sensors and budget constraints often limit the distribution utility's ability to monitor every critical network component. Therefore, the utility's situational awareness from fixed sensor alerts cannot be narrowed beyond a certain spatial zone which we call a *localization set*. Consequently, additional manual inspections by survey teams are conducted to isolate and repair the failure(s). The goal of these survey teams is to isolate the leak by using additional sensors or cues like the smell of gas, appearance of exposed pipes, and slope failure. Once isolated, a repair crew addresses the failure and restores the system [19].

Thus, one of the main tasks of the operator (i.e., the utility's command center) is to inspect the network and localize the actual failure events (or the events of interest that led to them) in a timely manner, based on the alert and localization set information. There are two main inefficiencies in the inspection processes that are currently followed by the utilities: (i) significant time delays in localizing the failures with loss of service due to delayed repair operations; and (ii) high operational costs due to sub-optimal allocation of inspection and repair crews [1]. Motivated by these challenges, we focus on the problem of routing repair vehicles from yards (i.e., staging locations) to the localization sets; in addition, we integrate the inspection capabilities of small Unmanned Aerial System (sUAS) for the identification of failure events within the localization sets. Our assertion is that sUAS-based inspection can contribute to reducing both inefficiencies (i) and (ii) by reducing time to failure isolation and improving repair operations.

In recent years, utilities have been employing manned aerial systems such as helicopters to supplement manual surveys for network inspection. While the cost of operating helicopters is estimated to be anywhere from $1,000 to $2,000 per hour, sUAS can provide comparable or improved services at $200 to $300 per hour while also reducing the safety risks to the inspection team [24]. The total difference in person-hours of labor between conventional and sUAS inspections can also be significant [4, 10, 27]. Table 1 shows the comparison of person-hours of labor mentioned in these studies. The ongoing improvements in sensor technology and on-board processing of data are likely to further improve the prospects of using sUAS for infrastructure inspection over manual surveys. For example, recent advances in gas detectors and infrared (IR) sensors [30] can be potentially useful for on-board detection of leak events.

To integrate the sUAS into the current monitoring and inspection processes followed by utilities, we need to address the following questions:

1) How to design an end-to-end operational framework that accounts for the infrastructure inspection requirements and sUAS platform constraints?
2) How to formulate an sUAS-enabled failure localization problem to isolate the failure(s) in a timely manner given the operating environment and the sensing requirements?
3) How should the operator route the repair vehicles to the various localization sets considering the time required for the sUAS to isolate the failure(s)?

This article contributes to the abovementioned questions by building on the existing literature on Vehicle Routing Problems (VRP) and by proposing new models of sUAS-based inspection that can be readily integrated into VRP-type formulations. The failure events considered in our work are typical of emergency situations when the operator is expected to respond to multiple failure alerts. The localization sets determined from these failure alerts typically span a few hundred meters in length, which is consistent with the search zones of conventional ground survey teams. We introduce the operational requirements and specific features of the infrastructure network, fixed sensors, repair vehicles, and sUAS platform in Section 2. To model sUAS capabilities and constraints, we consider a rotary-wing sUAS with a total flight time of less than an hour, which is suitable for proximity and localized facility inspection. Unlike fixed-wing sUAS that require continuous forward flight, rotary-wing sUAS can provide a more stable platform to enable extended observation of a particular area from various angles. For rotary-wing sUAS, we also do not need to consider minimum turning radii (i.e., the Dubins curve) since a constant forward speed is not required. We consider that the sUAS platform is small enough to fit in the repair vehicle, because the current and projected limitations on battery size and line-of-sight will likely require the sUAS to be transported via repair vehicles to the localization sets.

In Section 3, we discuss the factors governing the positioning and routing of sUAS for a given localization set. We introduce the concept of a *monitoring location*, from which the sUAS can observe some of the network components. The spatial positioning of these monitoring locations depends on various factors like the sUAS operating range, airspace restrictions, infrastructure stand-off distance, required image resolution for failure isolation, and infrastructure network topology. Along with the discussion in Section 2, these factors motivate key assumptions in our sUAS-enabled infrastructure monitoring framework.

In Section 4 we introduce Mixed Integer Programming (MIP) optimization models for computing the (i) route plan of repair vehicles (equipped with sUAS operators and repair personnel) to localization sets, and (ii) the inspection strategies of an sUAS fleet to isolate all of the failures within each localization set. We refer to the problems addressing (i) and (ii) as the Repair Vehicle Routing Problem (RVRP)

Table 1 Comparison of conventional and sUAS person-hours of labor

Required	Bridge Inspection [27]		Census Survey Task [4]		Powerline [10]	
	Manned A/C	sUAS	Ground Survey	sUAS	Manual	sUAS
# 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

and the sUAS Network Exploration Problem (SNEP), respectively.

The SNEP can be summarized as follows: Given a localization set and a number of homogeneous sUAS, determine the optimal route for each sUAS to monitor every network component where the objective is to minimize the maximum amount of time to explore the localization set. Each sUAS starts and ends its route at a common base within the localization set, either at the end of its mission, or for battery swaps to continue further exploration. Furthermore, each monitoring location is visited by only one sUAS.

The RVRP considers the following problem: Given a set of yards with a number of repair vehicles and activated localization sets, determine the optimal route for each repair vehicle where the objective is to minimize the maximum amount of time elapsed from time of failure alert to the time to repair, among all localization sets. Each repair vehicle starts and ends its route at its respective yard location, and each localization set is visited by only one repair vehicle. Furthermore, in calculating the total time elapsed, we include the optimal sUAS exploration time for each localization set (obtained from the SNEP) and the corresponding repair times (which are assumed to be fixed and given). Both the SNEP and RVRP are constrained by time and vehicle flow conservation constraints.

The overall approach (and our modeling focus) is illustrated in Fig. 1.

1.2 Related Work

SNEP. The SNEP is at its core a routing problem, and could be modeled after several classical problems found in literature and their variants. The VRP seeks to find the optimal set of routes for a fleet of vehicles to serve a set of customers [9]. For our problem, the vehicles (sUAS) would be serving the monitoring locations. One variant of the VRP is the Vehicle Routing Problem with Time Windows (VRPTW), where each customer can only be visited within a specified time window. In our case, this would be relevant only if we considered an upper bound time to repair each localization set. If we considered the criticality of each monitoring location and a limited time budget, the SNEP could also be modeled after the Team Orienteering Problem (TOP) [15], which seeks to find the optimal set of routes for a fleet of vehicles to maximize the total collected value obtained by visiting a subset of customers within a given time. For example, the TOP can model emergency response scenarios where the goal is to identify as many survivors as possible. For our application, the SNEP is better represented by the VRP than the TOP for two reasons. First, all of the components within the localization set must be monitored by the sUAS since the number and location of failures are not known. Second, although criticality could be an important factor, we require that *all* of the failures are addressed. Observing only a subset of the monitoring locations could result in undetected failure events which could worsen over time.

Of the many VRP variants, our SNEP formulation is closely related to the Multi-Trip Vehicle Routing Problem (MTVRP) [32]. Our formulation shares some features of the Green Vehicle Routing Problem (G-VRP) posed by Erdogan et al. [11]. 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 SNEP, we need to account for sUAS platform and infrastructure monitoring constraints, e.g., climb/descent rates, operating restrictions, and required image resolution.

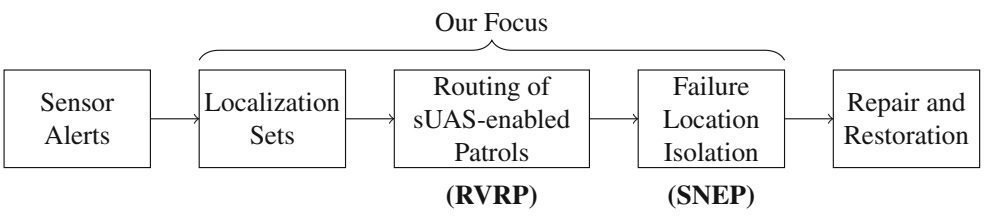


Fig. 1 Process Flow for sUAS-enabled infrastructure monitoring framework

RVRP. The RVRP is closely related to the Multiple Depot Vehicle Routing Problem (MDVRP) and more specifically, the Multiple Depot Multiple Vehicle Routing Problem (MDMVRP) [20]. Given a set of bases with a number of vehicles, the MDMVRP's objective is to minimize the total travel cost so that each location is visited by a vehicle. Our setting is somewhat different because we consider a min-max objective function. Furthermore, the optimal value of the SNEP enters into our RVRP formulation as an input parameter, because the optimal sUAS exploration time for each localization set impacts the longest time from failure alert to repair that the RVRP seeks to minimize.

1.3 Contributions

While the existing literature has considered the planning of flight operations for the use of sUAS in civilian applications, limited emphasis has been placed on detailing how to integrate sUAS into current inspection processes. In this work we detail an operational framework which captures the specific features of sUAS 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). These features and constraints can be included in the MIP formulations of the SNEP and the RVRP, which are presented in Section 4.[1]

Unlike the RVRP which can be solved for realistic problem sizes, solving the SNEP poses a challenge because of the large number of monitoring locations to consider in a typical localization set. Therefore, 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. This heuristic is based on the solution of a weighted minimum set cover problem and is presented in Section 5.

In Section 6, we show that our heuristic can solve a 5 localization set scenario within an acceptable time frame for real world implementation (4.15 seconds). Our heuristic achieves an overall average optimality gap of 0.78% with 10 different localization sets of varying size. We also demonstrate the interrelationship between the SNEP and the RVRP and discuss the practical issues that can impact the sUAS exploration time and repair vehicle travel times.

Finally, we evaluate the performance of our deterministic SNEP in situations when sUAS cruise speeds are stochastic using Monte Carlo simulation. Indeed, 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 localization sets. Following the exploration plan suggested by our SNEP solution can potentially lead to unobserved network components, which can significantly increase the time to localize and respond to failures. Thus, our static (and deterministic) formulation can be viewed only as an initial step toward the research on more general formulations that account for the stochastic and dynamic nature of sUAS-based inspection processes. In Section 7, we provide some comments in this regard and conclude our paper.

[1] Admittedly, while we do not model the communication aspects of sUAS operation, the framework can be potentially extended to include them.

2 Network, Repair Vehicles, and sUAS

In this section, we describe the key requirements and constraints for sUAS-enabled inspection of infrastructure component failures. We start with a description of how the components that are likely to have experienced failure events can be grouped into geographical regions based on the fixed sensor alerts. We then describe the generic requirements to route repair vehicles (carrying both sUAS operators and repair personnel) to these regions. Finally, we discuss sUAS characteristics to consider for the exploration of localization sets.

2.1 Infrastructure Network with Fixed Sensors

Consider an infrastructure network with the set of components denoted as $\mathcal{E}$. The network components are prone to *failure events* that can be random (e.g., pipe bursts), correlated (e.g., earthquake induced failures), or adversarial in nature (e.g., sabotage). The physical network can either be above ground or underground; nonetheless, in many cases, failure events are detectable from above ground [26]. 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.

The infrastructure network is monitored by an operator (i.e., the utility's command center) through 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

or flow), provided that the event lies in its "detection range". 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 operator 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 operator 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 identify (i.e., perfectly isolate) the individual failures in large-scale networks facing correlated failure events. Furthermore, the sensors' capability to detect fluctuations is constrained by their detection range. Therefore, in most failure situations, the operator can only map each failure event to a certain spatial zone, which we call a *localization set* [31]. Henceforth, we assume that the alerts from fixed sensors correspond to a collection of activated localization sets, where each set is comprised of network components that are likely to have undergone (or are prone to) failure. Isolating these failures requires additional inspection; our focus is on sUAS-based inspection. In our framework, the number and exact location of failures in each localization set is unknown before the inspection. The size of a typical localization set is of the order of a few hundreds of meters, which is consistent with search zones of inspection teams.

Motivated by practical considerations [19], we assume that the operator collects the sensor alerts for a predetermined time interval denoted $[0, t_0]$ and knows the localization sets that are activated during $[0, t_0]$. The operator assigns the available repair vehicles to the group of localization sets at time t_0. This assumption is motivated by the standard operating procedure that the operators follow in assigning survey and repair teams based on the alerts received from multiple areas during the first few hours after a disaster strikes the region. For simplicity, we assume that the interval $[0, t_0]$ also includes the lead time required to prepare vehicles and repair teams. Based on the alerts received during $[0, t_0]$, let $\mathcal{L}_1, \ldots, \mathcal{L}_K \subseteq \mathcal{E}$ denote the K localization sets that need to be further inspected by sUAS. The corresponding sUAS exploration plans are determined by the SNEP. Therefore, at the start of the inspection process, each localization set is assigned to a *yard* (i.e., facility where the repair vehicles and crews are staged). This assignment and corresponding routing plans are determined by the RVRP; see Section 4. Let $\mathcal{Y}$ denote the set of yards. Due to excessive setup costs, we will consider that yard sites are immutable. Figure 2 illustrates a network with a set of 64 fixed sensors, 2 yards, and 5 localization sets.

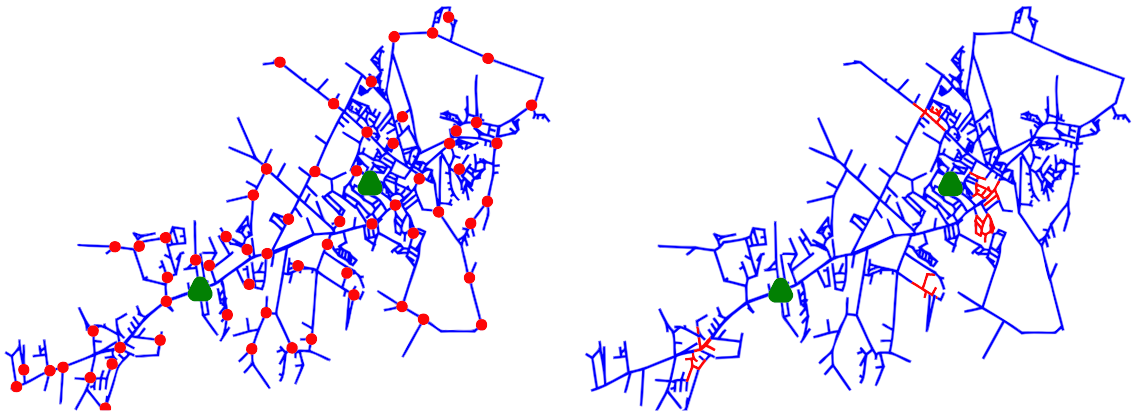
2.2 Repair Vehicles

For every yard $s \in \mathcal{Y}$, we denote n_s the number of repair vehicles that are available at yard s. For simplicity, we assume that the repair vehicles at each yard site have homogeneous capabilities, including speed, range, and crew (repair personnel and sUAS operators). In addition, each sUAS operator controls a single sUAS. We let u denote the number of sUAS in each repair vehicle. Again, for simplicity, we assume that all sUAS have identical sensing capabilities and technical characteristics (i.e., cruise speed, endurance, etc).

Each repair vehicle, starting from a yard, can visit one or more localization sets prior to returning to the same yard. To visit a localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$, a repair vehicle needs to set up a temporary base, which we denote as b_k. In practice, b_k can be chosen as the centroid of $\mathcal{L}_k$. Let $\mathcal{B} := \{b_k,\ k \in [\![1, K]\!]\}$ denote the set of all temporary bases. Setting up a temporary base involves unpacking the sUAS and performing pre-flight checks. We also assume that the repair vehicle remains at its temporary base for the duration of the sUAS flight over the localization set (to inspect and isolate failures) as well as the time to repair these failures. In particular, the repair process starts after sUAS-based inspection is completed, and the repair vehicle can move to another base or yard only after the repair at the current localization set is completed. This assumption is motivated by the fact that, in most practical situations, the repair crews gather the information on the type and location of failures prior to the start of the repair process, since combining repair requirements in close proximity can save costs and reduce the movement of their equipment [19].

For the purpose of route planning, the pairwise time to travel between yards and temporary bases can be assembled into a travel time matrix which we denote Γ. This consists of repair vehicle 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 a repair vehicle 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 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 repair vehicle is negligible.

As mentioned above, once the sUAS complete the inspection of $\mathcal{L}_k$, $k \in [\![1, K]\!]$, the repair process can start. In some practical situations, the repair time can be estimated beforehand; in our framework, we assume this time can be upper bounded by the "worst-case" case repair time, denoted $\tau_{repair,k}$. Historical data on individual repairs



(a) Network with fixed sensors (circles) and 2 yards (triangles)

(b) 5 localization sets arising from fixed sensor alerts

Fig. 2 Illustration of a sensor network and localization sets in an infrastructure network

can be utilized for estimating the expected and worst-case number of failures within a typical localization set; this can be used as a basis to compute $\tau_{repair,k}$. Additionally, data on the average age of the infrastructure, material type, or criticality of network components can be used to build a statistical model to estimate $\tau_{repair,k}$ [36].

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

Next, we discuss the sUAS platform characteristics, based on the standard technical specifications, battery replacement, and communication requirements.

2.3 sUAS Characteristics

A variety of commercially available rotary-wing sUAS platforms can be employed for infrastructure network inspection; 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 sUAS based on their Maximum Gross Take-Off Weight (MGTOW) similar to [35]. These representative sUAS sufficiently represent Commercial Off-The-Shelf (COTS) platforms ranging from the smaller rotary-wing DJI Phantom, to the larger Aeryon Scout. The representative sUAS classes are described in Table 2 using MGTOW (kg), mean cruise airspeed (knots), max airspeed (knots), descent and climb rates (meters per second), and endurance (minutes). In our SNEP formulation, the mean cruise

Fig. 3 Illustration of repair vehicle tours. Starting from a yard, each repair vehicle visits one or more temporary bases (black dots) and returns to its assigned yard (triangle)

Table 2 Representative sUAS

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

airspeed, denoted $\overline{V}_C$, is assumed to be deterministic and defined as the speed at which the sUAS should operate to maintain optimum performance [35]. Max airspeed, denoted V_{max} is the maximum permitted speed. The descent (resp., climb) rate is the vertical speed of the sUAS, 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 an sUAS spends in flight. For rotary-wing sUAS with MGTOW under 9 kgs, the advertised endurance values of different sUAS models exhibit relatively less variability; see Fig. 4. The endurance values shown in Table 2 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}.

When exploring a localization set, we allow the sUAS to return to the temporary base to replace their batteries before exploring other parts of the localization set. Each sUAS requires a deterministic time to replace its battery; we denote this as τ_{batt}. This consists of the time required to swap the battery since many sUAS enable hot-swapping of batteries. All sUAS are fully charged for the initial dispatch, and if a repair vehicle visits more than one localization set, we can reasonably assume that sUAS receive fully charged batteries while enroute to the other locations.

Finally, although communication is an important consideration for safe navigation and connectivity of sUAS, we do not explicitly consider the impacts of unreliable or insecure communication links between the sUAS and the operator. Commonly used communication links for sUAS 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 sUAS for monitoring strategic areas [18]. However, given that our focus in this paper is on establishing a static framework for joint routing of repair vehicles and sUAS-based inspection of physical infrastructures, we do not consider (low-level) communication aspects that are inherently dynamic in nature.

3 sUAS-based Inspection of Localization Sets

In this section we discuss the main factors governing the positioning and routing of sUAS for the purpose of failure isolation. Along with the discussion in Section 2, these

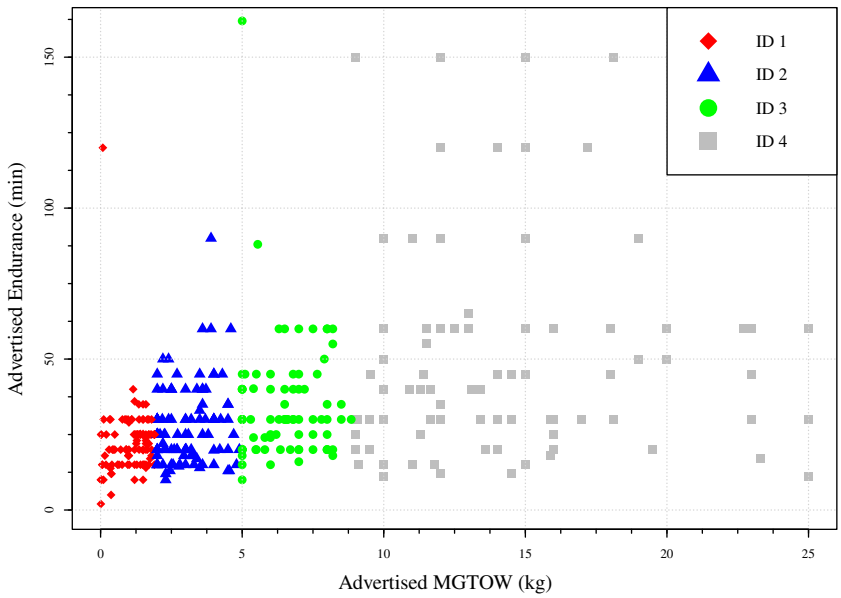


Fig. 4 The advertised MGTOW and endurance for 404 rotary wing sUAS platforms

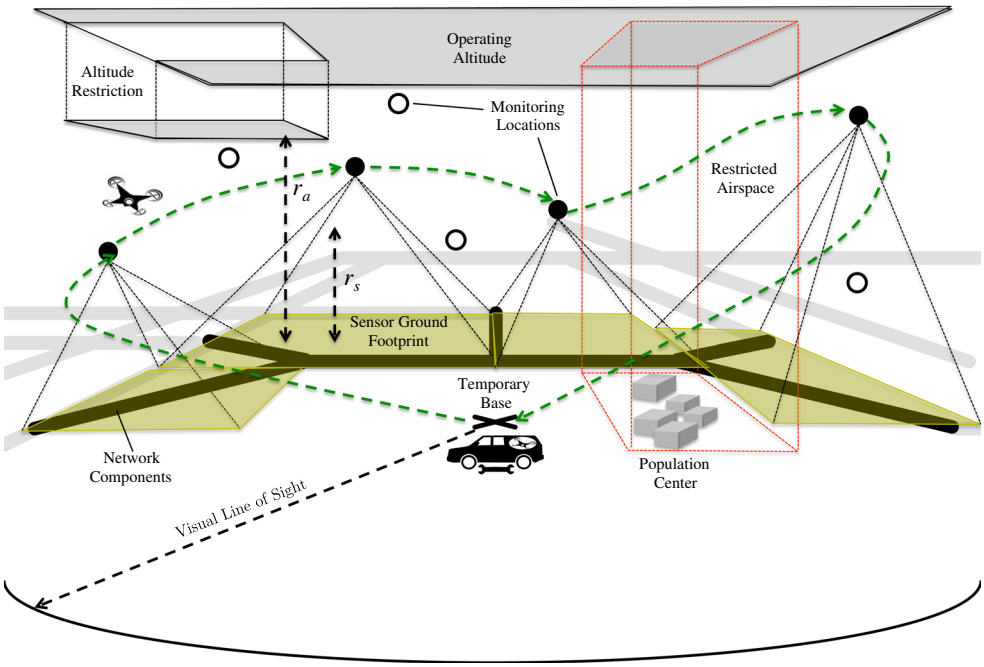


Fig. 5 Illustration of monitoring locations for a localization set. The available monitoring slocations are shown in filled and empty circles and the infrastructure components within the localization set are highlighted with thick lines

factors motivate the key assumptions in our sUAS-enabled infrastructure monitoring framework.

Once a repair vehicle sets up a temporary base, up to u sUAS are launched to explore the corresponding localization set. In our framework, the sUAS complete inspection of a localization set by visiting a number of "vantage points" with a clear and unobstructed view to one or more network components. The spatial positioning of these vantage points 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 5 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 sUAS 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 sUAS. We also need to meet Visual Line-of-Sight (VLOS) requirements, which is defined by the FAA as keeping unaided visual contact with the sUAS 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"* [12]. The sUAS operating range is therefore expressed as $\min(\frac{\overline{V}_C \cdot \tau_{max}}{2}$, VLOS range). Unfortunately, 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 [12] to a maximum theoretical range of 1050 m [37]. In practice, VLOS can vary significantly with local weather and other environmental conditions. However, we argue that for the purpose of infrastructure monitoring, the typical size (maximum radius) of localization sets is smaller than a conservative VLOS estimate. Thus, we assume that the positioning of monitoring locations are not constrained by the VLOS requirement. This assumption makes even more sense given the ongoing discussion about relaxing VLOS restrictions and allowing Beyond VLOS (BVLOS) sUAS operations in urban areas to support a range of applications, including infrastructure monitoring [18].

(ii) *Airspace Restrictions*. All monitoring locations are upper bounded at an altitude restriction, denoted r_a, which is typically 122 m AGL (Above Ground Level) to align with the current FAA Part 107 sUAS regulations [13]. 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 inspection. Given environmental uncertainty, wind gusts, and sUAS platform instability, we assume that the sUAS will operate outside a 30 m stand-off distance. This can be viewed as a conservative estimate based on current best practices as described in [24]. Combining the restriction imposed by the stand-off distance and altitude restriction, we maintain that any feasible vertical distance between a monitoring location and the infrastructure (at ground level), denoted R, is constrained as:

$$r_s \leq R \leq r_a \tag{1}$$

(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. Image stability of the video feed can be achieved by utilizing appropriate hardware like a motorized gimbal mount to compensate for turbulence. To get an idea of the resolution, one can estimate the Ground Sampling Distance (GSD), which is the distance between two consecutive pixel centers measured on the ground [23]. 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:

$$GSD = \frac{xR}{f\cos\alpha}, \tag{2}$$

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 the infrastructure at 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 Fig. 6 for an illustration of the GSD and sensor ground footprint. It is important to note that different failure types may require different GSDs for identification. Based on Eq. 1 and the GSD requirement, one can check if the on-board camera on the sUAS is adequate for the inspection task.[2] 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 localization set. 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 this case, sUAS can operate at higher altitudes (with higher GSD), and consequently visit a smaller number of monitoring locations.

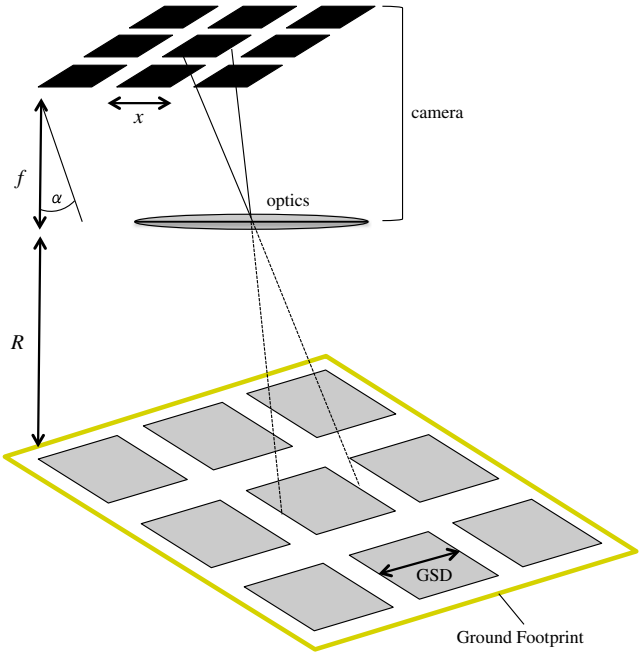
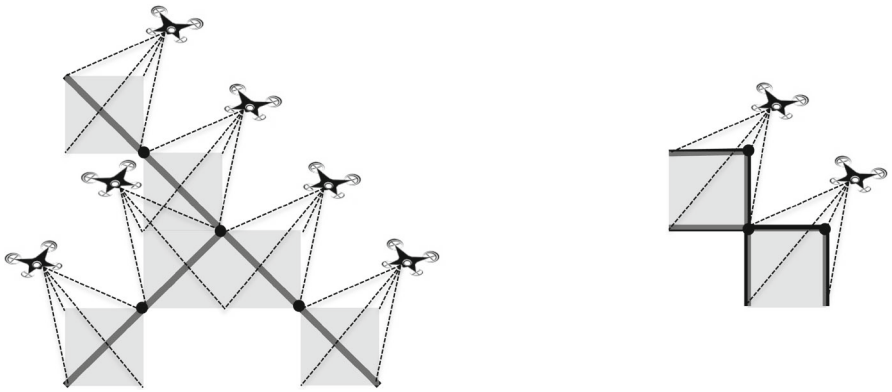


Fig. 6 Ground sampling distance and sensor ground footprint

(v) *Infrastructure Network Topology*. Finally, the number of monitoring locations also depends on the network topology within the localization set. 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

[2] 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 sUAS to resolve a 0.5 cm feature on the ground would require a simple rearranging of terms in Eq. 2 to solve for R. Assuming a nadir (overhead) aerial view ($\alpha = 0$), the sUAS would visit monitoring locations at a height of 100 m which also satisfies Eq. 1.



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

Fig. 7 Possible monitoring locations for 2 different network topologies. For the tree topology (left), 6 network components (black line segments) would require 6 monitoring locations with associated ground footprints (gray). For the grid network (right), only 2 monitoring locations are needed for the same number of network components

localization set in comparison to a grid topology; see Fig. 7.

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 sUAS to observe some of the network components while considering the requirements for each of the five factors discussed. For every localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$, we let $\mathcal{V}_k$ denote the set of monitoring locations that the sUAS can visit. Without loss of generality, we assume that the temporary base is $b_k \in \mathcal{V}_k$ in this set.

From each monitoring location $i \in \mathcal{V}_k$, we define a *monitoring set*, $\mathcal{C}_i^k \subseteq \mathcal{L}_k$, as the subset of network components that an sUAS is capable of monitoring (and isolating). Correspondingly, for every network component $e \in \mathcal{L}_k$, let $\mathcal{V}_k(e)$ denote the subset of monitoring locations from where an sUAS can monitor e. We assume that $\mathcal{V}_k$ fully "covers" $\mathcal{L}_k$, i.e., all components in $\mathcal{L}_k$ can be monitored by visiting a subset of $\mathcal{V}_k$. Figure 8 illustrates how a monitoring set is obtained from the sensor's ground footprint.[3] 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 localization set $\mathcal{L}_k$, an *sUAS tour* is then defined as a sequence of visits to a subset of $\mathcal{V}_k$ that starts and ends at the base b_k. Upon returning to base, the sUAS will either replace its battery for additional tours, or complete its mission. Analogous to the travel time matrix for repair vehicles, we can define another travel time matrix for each localization set $\mathcal{L}_k$; the elements of this matrix are the pairwise travel times between monitoring locations within $\mathcal{V}_k$. We denote this matrix as T^k. Thus, for every ordered pair of locations $(i, j) \in \mathcal{V}_k^2$, let τ_{ij}^k denote the sUAS 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 sUAS 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 T^k. We consider that by visiting a subset of monitoring locations $\mathcal{V}_k$ such that each network component is monitored at least once, the sUAS will be capable of decisively isolating the

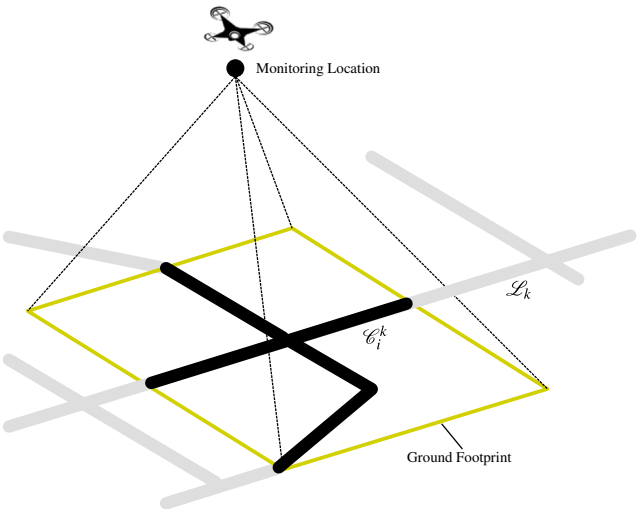
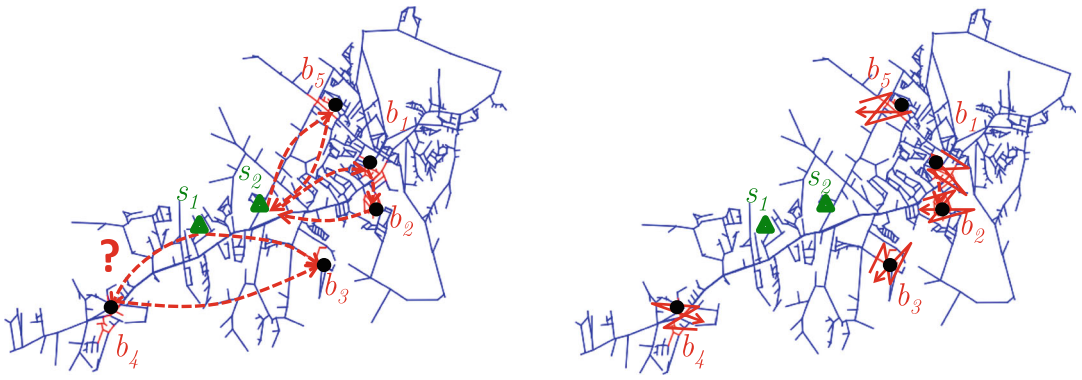


Fig. 8 The Monitoring Set corresponding to a particular monitoring location is obtained from the sensor ground footprint, which in turn is based on the GSD calculation

[3] 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.



(a) RVRP: Each repair vehicle can visit one or more localization sets. Following repair, each repair vehicle returns to its assigned yard.

(b) SNEP: From the temporary base location, the sUAS isolate the failure(s) by visition a subset of monitoring locations.

Fig. 9 Illustration of the RVRP and SNEP

failure(s) in the corresponding localization set. 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 [25], and the expanding use of unmanned systems in the future will only increase the amount of training data required to increase accuracy. Any additional image post-processing times can be incorporated in the repair time. Before proceeding further, we summarize the key assumptions that we introduced in Sections 2 and 3:

- A1 Each failure alert obtained by the network operator from fixed sensors can be mapped to a localization set, 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 localization set is unknown.
- A2 Based on fixed sensor alerts (and activated localization sets) received in the time interval $[0, t_0]$, the operator allocates and dispatches repair vehicles at time t_0.
- A3 The size of the localization sets for sUAS-based inspection is no greater than the VLOS in radius.
- A4 The sUAS can monitor all network components in the localization set $\mathcal{L}_k$ by visiting a subset of monitoring locations $\mathcal{V}_k$.
- A5 To fix the isolated failure(s) in a given localization set, the worst-case repair time, $\tau_{repair,k}$ for each localization set $\mathcal{L}_k$ is known. The repair starts only after the entire localization set has been inspected for possible failures.
- A6 All sUAS have identical sensing and technical capabilities. Likewise, all repair vehicles have homogeneous capabilities.
- A7 The travel times for both the repair vehicles and sUAS are assumed to be deterministic.
- A8 The sUAS endurance time as well as the time to replace the on-board battery are assumed to be deterministic.
- A9 The communication link between the sUAS and operator is secure and reliable, and does not impose any constraint on the route planning of repair vehicles and sUAS.

4 Modeling Approach and Formulation

In this section we present the overall infrastructure monitoring framework and provide the MIP formulations for both the SNEP and RVRP based on assumptions A1–A9.

4.1 Infrastructure Monitoring Framework

Given the localization sets that arise from failure alerts, our focus is to study how joint optimization of sUAS inspection plans and repair vehicle route plans can create efficiency and timely detection. To do so, we propose an approach that (i) solves the RVRP, which consists in optimally dispatching repair vehicles to the localization sets to minimize the worst-case time to inspect (and repair), and (ii) solves the SNEP which optimally routes the sUAS to isolate failure locations

within a given localization set. These two problems are nested in that the optimal value of the SNEP is taken as an input in solving the RVRP; see the illustration in Fig. 9.

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 repair vehicle dispatch from each yard at time t_0. For each localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$, we denote θ_k the amount of time during which $\mathcal{L}_k$ was activated prior to the dispatch of the repair vehicles. Note that within a given localization set, 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 $\mathcal{L}_k$. Note that high values of θ_k can occur when the interval $[0, t_0]$ is large. This may happen in situations when the repair vehicles are not readily available or when repair crews are engaged in other jobs and are not positioned at the yard [1]. The timeline of various phases of inspection and repair operations is illustrated in Fig. 10. A repair vehicle is dispatched from its yard s at time t_0 and completes the set up of a temporary sUAS base at location b_k at time $t_0 + \gamma_{sk}$. The optimal sUAS exploration time for $\mathcal{L}_k$ is denoted ξ_k^*. For this example, the total time elapsed from failure alert to time of repair for $\mathcal{L}_k$, denoted $t_{total,k}$, is equal to $\theta_k + \gamma_{sk} + \xi_k^* + \tau_{repair,k}$. We can see how the optimal value of the SNEP is embedded within the RVRP, which seeks to minimize the maximum of $t_{total,k}$ over all K localization sets.

4.2 sUAS Network Exploration Problem

Consider a localization set $\mathcal{L}_k$ that is activated during the time interval $[0, t_0]$. Recall that the exact number of failures, and their exact locations in $\mathcal{L}_k$ are unknown. Thus, the SNEP considers optimally dispatching sUAS to monitor and isolate every network component in $\mathcal{L}_k$. The u sUAS leave the base b_k, visit a subset of monitoring locations in $\mathcal{V}_k$ from where they can monitor the network components, and return to the base either to complete the mission, or to replace their batteries for further exploration. The objective is to minimize the time to explore $\mathcal{L}_k$, which we formulate as the maximum amount of time, among the u sUAS, to return to the base for mission completion. We choose this min-max objective function for two main reasons highlighted in [1]. 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.

For the sake of brevity in presenting our formulation of the SNEP, 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 sUAS exploration 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 sUAS 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 sUAS that visits node j (coming from node i). Note that this quantity is reset every time an sUAS replaces its battery. If t_{ij} is nonzero, then it represents the time traveled so far by the sUAS that is currently visiting node j (which comes from i). This quantity is NOT reset when an sUAS 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 sUAS 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 [2]. Our SNEP can be formulated with constraints (3)–(15).

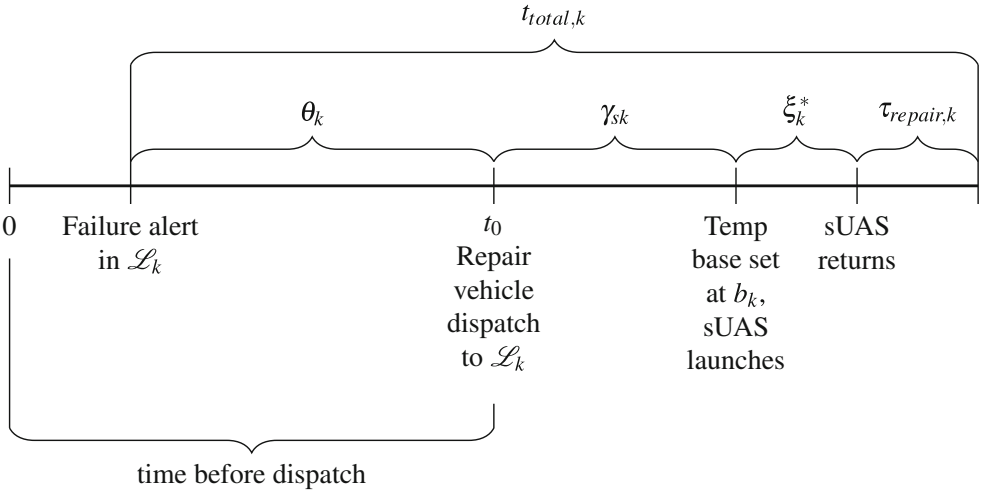


Fig. 10 Timeline from failure alert to repair for the trivial case of a single repair vehicle, single localization set, and single sUAS

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

Subject to

$$\sum_{i\in\mathcal{V}_k\backslash\{b\}} x_{bi} \leq u \tag{3}$$

$$\sum_{i\in\mathcal{V}_k\backslash\{b,j\}} (x_{ij}+x'_{ij}) + x_{bj} = \sum_{i\in\mathcal{V}_k\backslash\{b,j\}} (x_{ji}+x'_{ji}) + x_{jb}, \qquad \forall j \in \mathcal{V}_k\backslash\{b\} \tag{4}$$

$$\sum_{i\in\mathcal{V}_k\backslash\{b,j\}} (x_{ij}+x'_{ij}) + x_{bj} \leq 1, \qquad \forall j \in \mathcal{V}_k\backslash\{b\} \tag{5}$$

$$\sum_{i\in\mathcal{V}_k(e)} \left(\sum_{j\in\mathcal{V}_k\backslash\{b,i\}} (x_{ij}+x'_{ij}) + x_{ib} \right) \geq 1, \qquad \forall e \in \mathcal{E}\backslash\mathcal{C}_b^k \tag{6}$$

$$z_{bj} = \tau_{bj} \left(x_{bj} + \sum_{i\in\mathcal{V}_k\backslash\{b,j\}} x'_{ij} \right), \qquad \forall j \in \mathcal{V}_k\backslash\{b\} \tag{7}$$

$$0 \leq z_{ij} \leq \tau_{max} x_{ij}, \qquad \forall (i,j) \in (\mathcal{V}_k\backslash\{b\})^2 \mid i \neq j \tag{8}$$

$$0 \leq z_{ib} \leq \tau_{max} \left(x_{ib} + \sum_{j\in\mathcal{V}_k\backslash\{b,i\}} x'_{ij} \right), \qquad \forall i \in \mathcal{V}_k\backslash\{b\} \tag{9}$$

$$\sum_{j\in\mathcal{V}_k\backslash\{i\}} z_{ij} = \sum_{j\in\mathcal{V}_k\backslash\{i\}} z_{ji} + \sum_{j\in\mathcal{V}_k\backslash\{i\}} \tau_{ij} x_{ij} + \tau_{ib} \sum_{j\in\mathcal{V}_k\backslash\{b,i\}} x'_{ij}, \qquad \forall i \in \mathcal{V}_k\backslash\{b\} \tag{10}$$

$$t_{bj} = \tau_{bj} x_{bj}, \qquad \forall j \in \mathcal{V}_k\backslash\{b\} \tag{11}$$

$$0 \leq t_{ij} \leq ((|\mathcal{V}_k|-1)\tau_{max} + (|\mathcal{V}_k|-2)\tau_{batt})(x_{ij}+x'_{ij}), \qquad \forall (i,j) \in (\mathcal{V}_k\backslash\{b\})^2 \mid i \neq j \tag{12}$$

$$0 \leq t_{ib} \leq ((|\mathcal{V}_k|-1)\tau_{max} + (|\mathcal{V}_k|-2)\tau_{batt})x_{ib}, \qquad \forall i \in \mathcal{V}_k\backslash\{b\} \tag{13}$$

$$\sum_{j\in\mathcal{V}_k\backslash\{i\}} t_{ij} = \sum_{j\in\mathcal{V}_k\backslash\{i\}} t_{ji} + \sum_{j\in\mathcal{V}_k\backslash\{i\}} \tau_{ij} x_{ij} + \sum_{j\in\mathcal{V}_k\backslash\{b,i\}} (\tau_{ib}+\tau_{batt}+\tau_{bj}) x'_{ij}, \qquad \forall i \in \mathcal{V}_k\backslash\{b\} \tag{14}$$

$$\xi \geq t_{ib}, \qquad \forall i \in \mathcal{V}_k\backslash\{b\} \tag{15}$$

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

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

Constraint (3) ensures that no more than u sUAS are sent for the exploration of the localization set. Constraint (4) is the flow conservation constraint, taking replenishment arcs into account. Constraint (5) ensures that each monitoring location is visited at most once. Constraint (6) ensures that each network component is monitored at least once. Constraint (7) initializes and resets the time traveled by an sUAS after replacing its battery. Constraint (8) enforces z_{ij} to be 0 when there is no sUAS that goes from i to j and between 0 and τ_{max} otherwise. Constraint (9) enforces z_{ib} to be 0 when there is no sUAS 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 (10) updates the time traveled so far by each sUAS since the last battery replacement. Constraint (11) initializes the cumulative time traveled so far by the sUAS. Constraints (12) and (13) make sure that $t_{ij} = 0$ when there is no sUAS 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 sUAS goes from i to j. We illustrate the Big-M upper bound for constraint (13) in Appendix C. Constraint (14) updates the time traveled so far by the sUAS and takes into account the time to replace the batteries, if required. Since we want to minimize the maximum travel time of the sUAS 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 (15).

The SNEP solution provides optimal sUAS routes that can be described as *simple* or *multi-trip* routes. Consider a localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$. We let p represent an sUAS *simple* 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 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 in several ways. First, instead of multiple refueling stations available, there is only one refuel location (temporary base) for each localization set. 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 (5) and (6). 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.

4.3 Repair Vehicle Routing Problem

Given the set of yards $\mathcal{Y}$ and the set of temporary base locations $\mathcal{B}$, the RVRP seeks to find the optimal route for each repair vehicle starting and ending at its corresponding yard such that (i) each base corresponding to a localization set is visited, and (ii) the maximum amount of time elapsed from time of failure to time of repair among all localization sets, is minimized. In calculating (ii) we include the optimal sUAS exploration time, ξ_k^*, required for each localization set $\mathcal{L}_k, k \in [\![1, K]\!]$.

For each yard $s \in \mathcal{Y}$, 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 repair vehicle that originates from yard 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 yard different from s; this ensures that a repair vehicle will return to the yard 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 repair vehicle arrives at location l (coming from location k). Note that this quantity takes into account the time to travel between yards and localization sets, the time to explore the localization sets with the sUAS, and the repair time. 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 RVRP is given in constraints (16)–(23).

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

Subject to

$$\sum_{l \in \mathcal{B}} y_{sl}^s \leq n_s, \forall s \in \mathcal{Y} \tag{16}$$

$$\sum_{l \in \mathcal{B} \cup \{s\} \setminus \{k\}} y_{kl}^s = \sum_{l \in \mathcal{B} \cup \{s\} \setminus \{k\}} y_{lk}^s, \qquad \forall (k, s) \in \mathcal{B} \times \mathcal{Y} \tag{17}$$

$$\sum_{s \in \mathcal{Y}} \sum_{l \in \mathcal{B} \cup \{s\} \setminus \{k\}} y_{lk}^s = 1, \qquad \forall k \in \mathcal{B} \tag{18}$$

$$w_{sl} = \gamma_{sl} y_{sl}^s, \qquad \forall (l, s) \in \mathcal{B} \times \mathcal{Y} \tag{19}$$

$$0 \leq w_{kl} \leq M \sum_{s \in \mathcal{Y}} y_{kl}^s, \qquad \forall (k, l) \in \mathcal{B}^2 \mid k \neq l \tag{20}$$

$$0 \leq w_{ks} \leq M y_{ks}^s, \qquad \forall (k, s) \in \mathcal{B} \times \mathcal{Y} \tag{21}$$

$$\sum_{l \in \mathcal{B} \cup \mathcal{Y} \setminus \{k\}} w_{kl} = \sum_{l \in \mathcal{B} \cup \mathcal{Y} \setminus \{k\}} w_{lk} + \sum_{l \in \mathcal{B} \setminus \{k\}} \gamma_{kl} \sum_{s \in \mathcal{Y}} y_{kl}^s + \sum_{s \in \mathcal{Y}} \gamma_{ks} y_{ks}^s + \xi_k^* + \tau_{repair,k}, \qquad \forall k \in \mathcal{B} \tag{22}$$

$$t_{worst} \geq \sum_{l \in \mathcal{B} \cup \mathcal{Y} \setminus \{k\}} w_{lk} + \xi_k^* + \tau_{repair,k} + \theta_k, \qquad \forall k \in \mathcal{B} \tag{23}$$

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

Constraints (16)–(18) define the classic network flow constraints, while constraints (19)–(22) keep track of the arrival times for each repair vehicle. Specifically, constraint (16) ensures that no more than n_s repair vehicles leave yard s. Constraint (17) ensures that if a vehicle from yard s enters a localization set, it also leaves. Constraint (18) ensures that each localization set is visited by exactly one vehicle. Constraint (19) initializes the time traveled by the vehicle if it departs from yard s. Constraints (20)–(21) make sure that the arrival time is 0 when no vehicle travels from

one localization set to another localization set or yard. Otherwise, we use a large constant, M, to ensure that there is no restriction when a location is visited. Constraint (22) updates the arrival time by taking into account the vehicle travel time, the repair time, $\tau_{repair,k}$, as well as ξ_k^*. Finally, since we want to minimize the maximum amount of time elapsed from time of failure to time of repair among all localization sets, we add the variable t_{worst}, along with constraint (23).

4.4 Implementation on a Real Pipeline Network

We implement the SNEP and RVRP 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 $\mathcal{E}$ [17]. Assuming that a failure in a network component can be detected by a sensor if it is within a given distance [31], 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 [14]. The localization sets are then provided by partitioning the set of components depending on the sensors' outputs. For this example, we considered the activation of five localization sets with sizes ranging from 16 to 33 components. We label these localization sets $\mathcal{L}_1, \ldots, \mathcal{L}_5$.

For each localization set $\mathcal{L}_k$, $k \in [\![1, 5]\!]$, we assume that the sUAS monitoring locations, $\mathcal{V}_k$, are positioned directly above the nodes of the subnetwork induced by $\mathcal{L}_k$, i.e., the set of end nodes of the edges in $\mathcal{L}_k$. We consider the scenario where an sUAS, positioned directly above a given monitoring location $i \in \mathcal{V}_k$, can monitor the adjacent edges of i (i.e., the adjacent pipelines). Without loss of generality, we restrict the sUAS 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 localization set 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 SNEP instances, we find that the optimal time required to monitor and isolate all components for each localization set with 2 sUAS is $\boldsymbol{\xi}^* = (0.88, 1.46, 0.86, 0.87, 0.63)$ (in hours). The final routes for the 2 sUAS are shown in Table 3. Note that the exploration of localization set $\mathcal{L}_2$ takes longer because it requires a multi-trip route for each sUAS. Also note that due to the min-max objective function of the SNEP, the final solution can result in extraneous node visits for route(s) with shorter duration. For example, for $\mathcal{L}_2$, the visit to node 9 for the second sUAS is one such case. Next, we consider that two yards, s_1 and s_2, are located in the network with yard s_1 containing one repair vehicle and yard s_2 containing two. Each repair vehicle carries two sUAS. For simplicity, we assume a repair time, $\tau_{repair,k}$, of 10 minutes for each localization set. We derived the travel time between yards and localization set bases b_k by computing the Euclidean distances between each pair of locations. 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).

Table 3 SNEP solutions

$\mathcal{L}_k$	Nodes	Edges	ξ_k^*	b_k	sUAS Routes
$\mathcal{L}_1$	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)
$\mathcal{L}_2$	16	20	1.46	16	(16, 10, 1, 16), (16, 5, 6, 16), (16, 8, 9, 7, 4, 16), (16, 12, 15, 16)
$\mathcal{L}_3$	15	18	0.86	6	(6, 8, 3, 15, 14, 1, 6), (6, 5, 10, 11, 6)
$\mathcal{L}_4$	18	17	0.87	8	(8, 10, 15, 12, 14, 8), (8, 4, 1, 5, 6, 3, 8)
$\mathcal{L}_5$	16	16	0.63	11	(11, 16, 5, 4, 7, 10, 11), (11, 8, 12, 14, 1, 11)

Using the formulation for the RVRP, as described in Section 4.3, we obtain the following optimal solution: The single repair vehicle from yard s_1 travels along the route (s_1, b_4, s_1), and the routes for the two repair vehicles from yard s_2 are (s_2, b_5, b_1, s_2) and (s_2, b_2, b_3, s_2). Figure 11 illustrates the optimal solution. The longest time elapsed from time of failure to time of repair is 5.87 hours.

From this initial computational study, we can make the following observations: First, we verified that the overall solutions are sensitive to $\boldsymbol{\theta}$. In general, a localization set $\mathcal{L}_k$ with a larger θ_k will be visited first. Second, we can check that the RVRP solution is affected by the magnitude of the sUAS optimal exploration times, ξ_k^*. For example, a repair vehicle can take a longer route if sUAS exploration times for the localization sets in that route are relatively small. We realize that the computation time to solve the SNEP can be large for practical instances. Unlike the RVRP which can be solved efficiently given the low number of localization sets, the SNEP poses a computational bottleneck because of the large number of monitoring locations to consider for a typical localization set. For example, one SNEP solution for a localization set 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 propose a heuristic approach that can promptly reach optimal or near optimal solutions.[4]

[4] As of this writing, we are also exploring path based formulations which can further reduce the computational times, especially if used in combination with our heuristic approach.

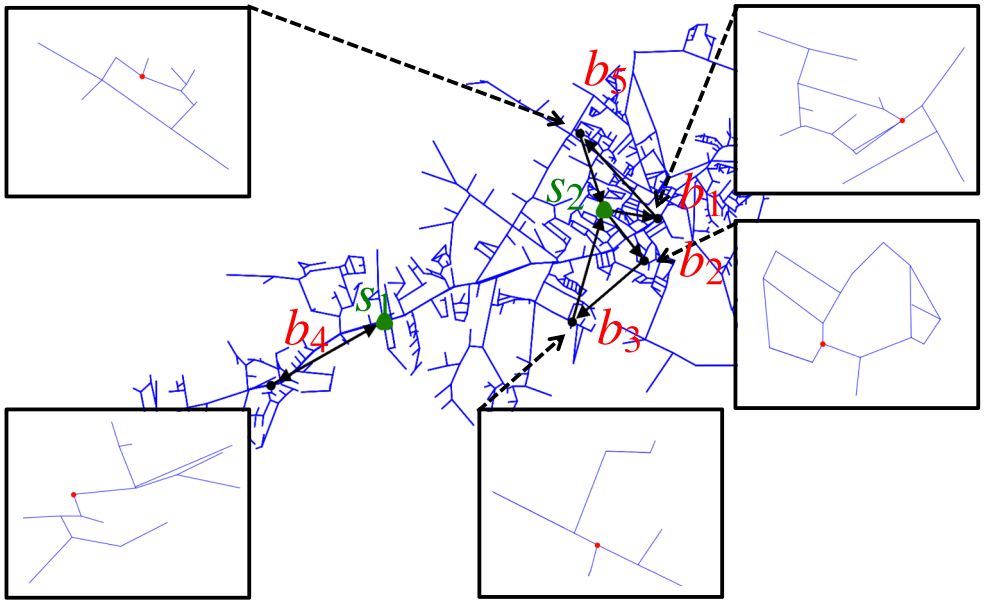


Fig. 11 Optimal RVRP solution where all failures alerts occur one hour prior to t_0. The arrows depict the routes for the repair vehicles, which originate from the two yards (triangles) and visit the temporary bases within each localization set

5 Heuristic Approach

In order to improve the scalability of the SNEP, we propose a heuristic approach that takes advantage of the monitoring set constraint (6). 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 [7]. Our primary contributions in developing this heuristic are the weighted set cover based initial route construction, and the final route combination and relocation with base insert steps as shown in Fig. 12.

5.1 Heuristic Steps

Step 1 Weighted Set Cover Recall that in each localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$, the sUAS need to visit a subset of monitoring locations in order to isolate every network component in $\mathcal{L}_k$. This implies that, although each monitoring location does not need to be visited, the sUAS 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 $\mathcal{L}_k$ is isolated if each monitoring location in $\mathcal{S}$ is visited by the sUAS. A minimum set cover (MSC) is a set cover of minimum cardinality that can observe every network component in the localization set $\mathcal{L}_k$. 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 SNEP heuristic, we consider a weighted variant of the MSC problem as discussed in [5] 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:

$$\begin{aligned} &\text{minimize} && \sum_{i \in \mathcal{V}_k} \tau_{b_k,i} x_i && && \\ &\text{subject to} && \sum_{i \in \mathcal{V}_k(e)} x_i \geq 1, && \forall e \in \mathcal{E} && (24) \\ & x_i \in \{0, 1\}, && && \forall i \in \mathcal{V}_k && (25) \end{aligned}$$

Constraint (24) 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

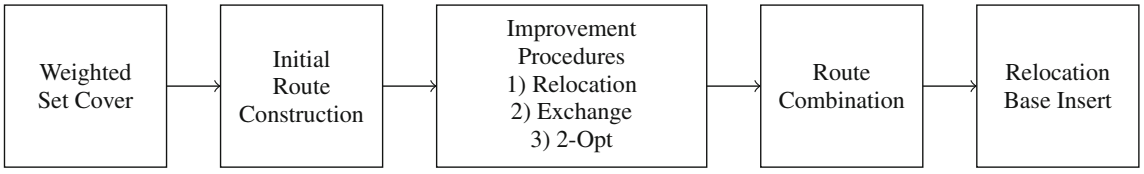


Fig. 12 Five steps used in the heuristic to solve the SNEP

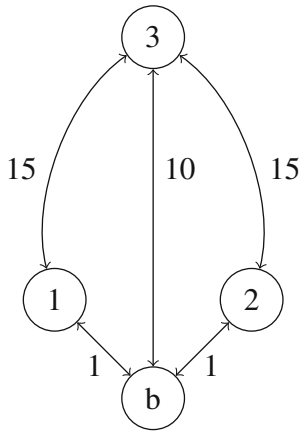


Fig. 13 Example network for justifying the use of weighted MSC in the SNEP heuristic

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 Fig. 13, 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 the sUAS 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 sUAS 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 SNEP solution, we highlight one example from Table 3. Consider the multi-trip routes from the SNEP solution for 2 sUAS exploring $\mathcal{L}_2$ from base node 16: sUAS 1 $\rightarrow$ (**16**, **10**, **1**, **16**), (**16**, 5, **6**, **16**), sUAS 2$\rightarrow$ (**16**, 8, 9, **7**, **4**, **16**), (**16**, **12**, **15**, **16**). Each sUAS requires one interim base visit. The nodes in bold make up the optimal weighted MSC. Figure 14 shows the optimal weighted MSC along with the topology of $\mathcal{L}_2$. We observe that the weighted MSC will always provide a subset of nodes from which to generate feasible routes for the SNEP. We use this insight to construct the initial routes, as described next.

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

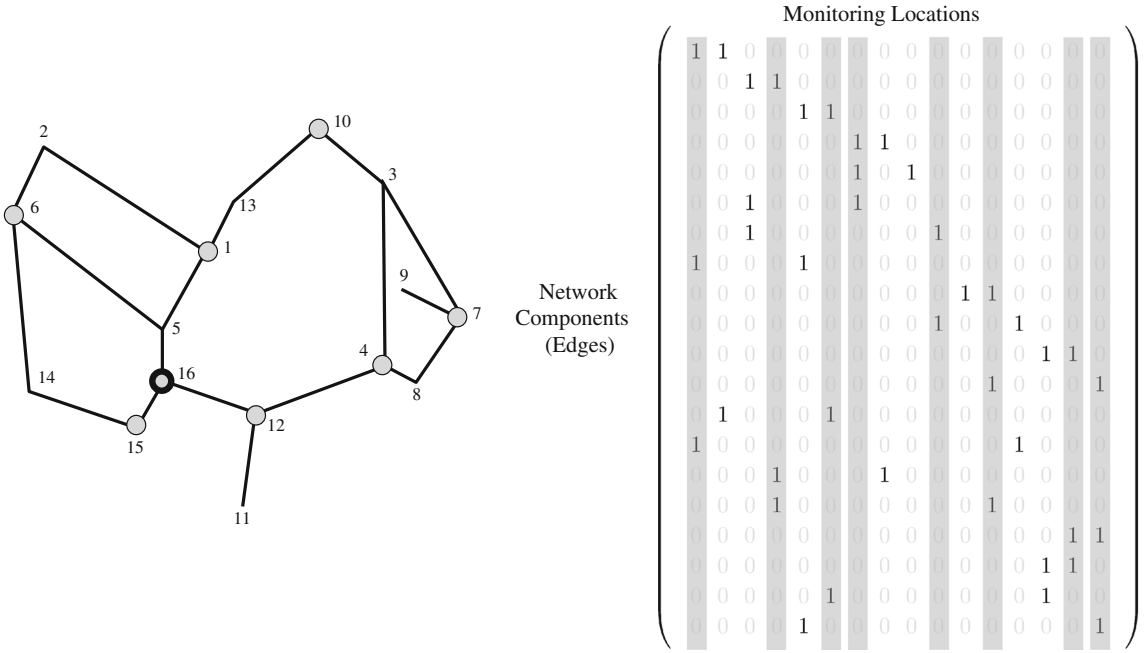
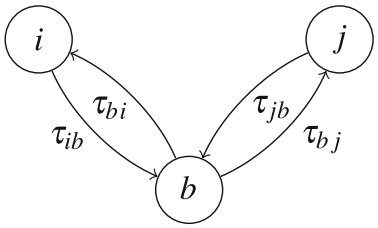
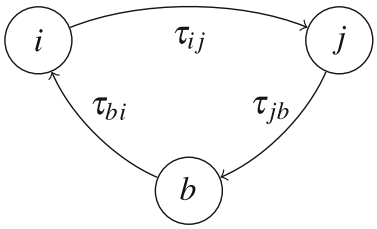


Fig. 14 Weighted Minimum Set Cover embedded within the SNEP Solution for localization set 2. The topology is shown on the left with weighted MSC nodes in gray. The matrix on the right shows the associated monitoring sets with the weighted MSC nodes highlighted



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

(a) Nodes visited one at a time



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

(b) Nodes visited in succession

Fig. 15 Illustration of travel time savings

simultaneously [6]. We choose to construct the routes in parallel since it generally provides better results as described in [21]. 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 4, 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 Fig. 15a. 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 Fig. 15b, the total travel time is $t'_{jb} = (\tau_{bi} + \tau_{ij} + \tau_{jb})$. The total travel time savings is then given by:

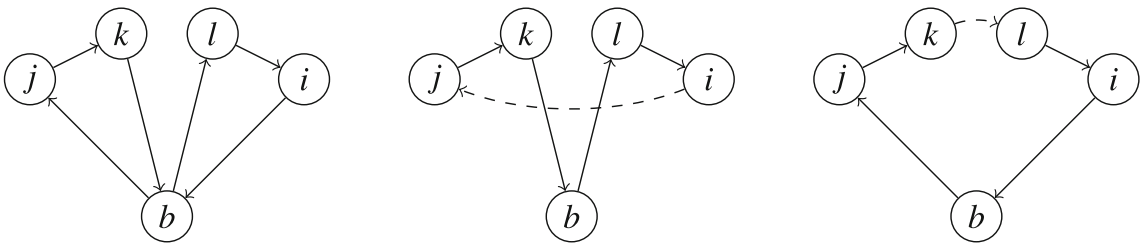
$$\begin{aligned} S_{ij} = t_{jb} - t'_{jb} &= (\tau_{bi} + \tau_{ib} + \tau_{bj} + \tau_{jb}) \\ &-(\tau_{bi} + \tau_{ij} + \tau_{jb}) = \tau_{ib} + \tau_{bj} - \tau_{ij} \end{aligned} \tag{26}$$

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 [33]. 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 Fig. 16a. 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. 16a and b. 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 Fig. 16a and c. Since S_{kl} provides the larger savings, we only select this as a savings pair and do not consider S_{ij}.

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) until the savings list is exhausted [22]. For each case, we check for feasibility, i.e., total travel time for each route is less than or equal to τ_{max}. We also keep inventory of the nodes that have not been assigned to a route.

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.



(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) $(\tau_{bl} + \tau_{li} + \tau_{ij} + \tau_{jk} + \tau_{kb})$

Fig. 16 Illustration of savings calculation for asymmetric travel times

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. Let $\mathcal{P}$ denote the set of simple routes resulting from this step. 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 sUAS, this guarantees an available route to insert a node into for the upcoming relocation procedure. Otherwise, we risk inequity in sUAS 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 [29] as well as our own computational tests on the 5 localization sets in Table 3. 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.

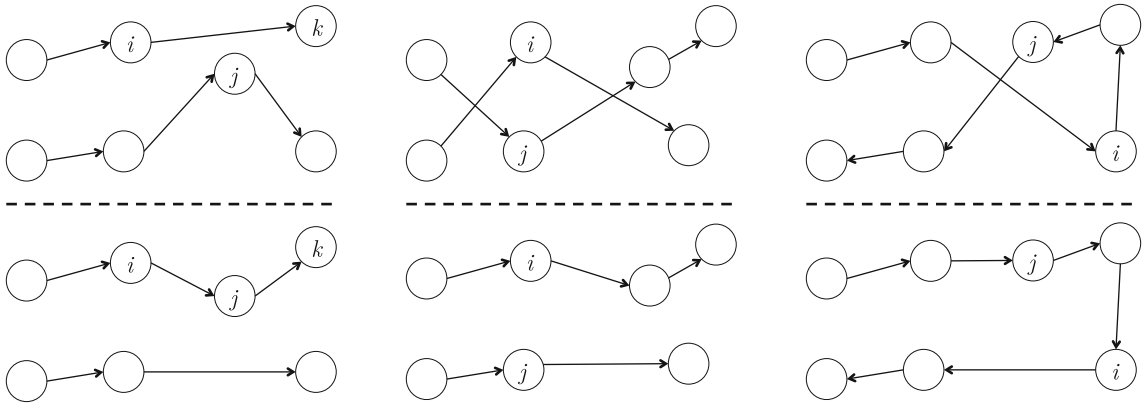
   This procedure is shown in Fig. 17a and is similar to the *relocation* procedure described in [33]. We extend it to consider both τ_{max} and the min-max objective function of the SNEP. We apply the relocation procedure first since it often produces the best results of the three improvement procedures as discussed in [29].

2. **Exchange.** The exchange inter-route improvement procedure as described in [33] considers every possible pair of routes and attempts to exchange two nodes between the two routes as shown in Fig. 17b. 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 [7] 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 Fig. 17c 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 [3]. 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 final sUAS routes created accounts for the number of sUAS available and inserts an interim base visit if required. This problem is a variant of the multi-processor scheduling or load balancing problem [16]. 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 sUAS. There are two additional considerations for our problem: first, if an sUAS 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 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 $|\mathcal{P}| < u$. 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. If there is more than one simple route created from the above steps, note that we will have node-disjoint simple routes except with the terminal base visits. 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



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

Fig. 17 Improvement Procedures. For each procedure, the top figure shows the route(s) prior to implementation and the bottom figure shows the resulting route(s) [29]

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:

$$\frac{(2|\mathcal{P}|)!}{2!(2|\mathcal{P}|-2)!} - |\mathcal{P}| \tag{27}$$

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 results in the shortest cumulative travel times and that is feasible. 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 SNEP, specifically, the monitoring set constraint (6), 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 SNEP and

Table 4 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

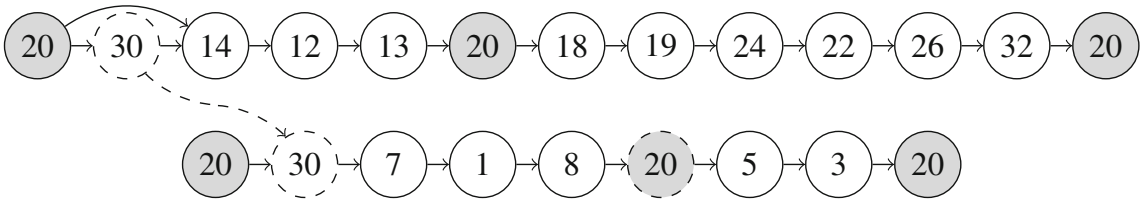


Fig. 18 Illustration of the Relocation with Base Insert Step. With this pair of routes, the top route is the donor route with the larger cumulative travel time (which includes an interim base visit). Node 30 is selected to be moved from the donor route into the recipient route prior to node 7. Since the recipient route's total travel time was longer than the sUAS endurance, a base visit is inserted according to a greedy approach

ensures that the final simple or multi-trip routes account for the number of available sUAS. To clarify the steps of our heuristic, we show how it is applied for one example localization set. We set τ_{max} to 1 hour and u to 2 sUAS.

5.2 Example Problem

Step 1 Weighted Set Cover. In a localization set from the same pipeline network which consists of 36 nodes (monitoring locations) and 37 edges (components), an 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 localization set is shown in Table 4. 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.

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.

Table 5 Comparison of SNEP exact solutions with heuristic solutions for 2 sUAS

Localization Set		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

Table 6 Comparison of RVRP solutions

Time Since Alert	Repair Vehicles		Obj	RVRP
θ	Yard 1	Yard 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) $(s_2, b_2, b_1, b_3, s_2)$

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: sUAS 1 $\rightarrow$ (20, 7, 1, 8, 5, 3, 20), and sUAS 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.

Step 5 Relocation with Base Insert. In Fig. 18, 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 sUAS 1: (20, 30, 7, 1, 8, 20), (20, 5, 3, 20), sUAS 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.

6 Computational Study

In this section, we first compare our heuristic solutions against the exact solutions of the SNEP. Next, we solve the RVRP, considering different assignments of repair vehicles to yards, and different elapsed times from failure alert to repair vehicle dispatch. Finally, we evaluate our solution for the case when travel times are stochastic by employing Monte Carlo simulation on representative network topologies.

6.1 SNEP Results

The MIP formulation for the SNEP described in Section 4 took over 5 hours to achieve the exact solutions for the 5 localization set scenario referred to in Table 3. Comparatively, our heuristic was able to obtain the same optimal solutions in only 4.15 seconds. Thus, for this 5 localization set scenario, our heuristic provided the optimal solution within an acceptable time frame for real world implementation. Using the same Kentucky based pipeline network, we altogether tested our heuristic on 10 different localization sets consisting of up to

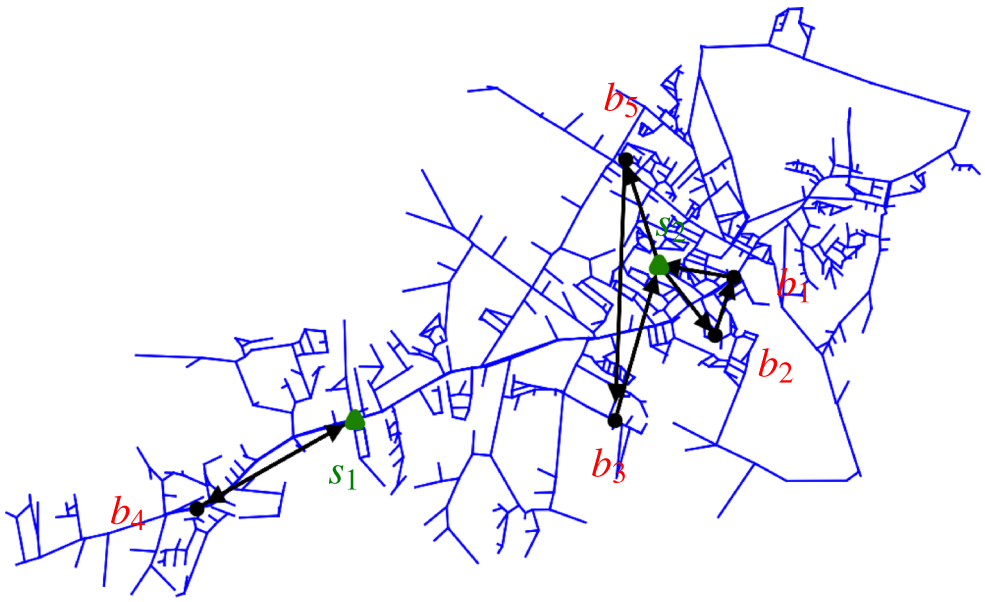


Fig. 19 Optimal RVRP solution for the case where the failure alert for localization set 2 (resp. 5) occurs 5 (resp. 3) hours prior to repair vehicle dispatch

37 edges (i.e., components). The largest of these represents the biggest reasonable size that can be assigned to a repair vehicle crew given the scale of the network. Computational results show that high quality solutions can be obtained using the SNEP heuristic for 2 and 3 sUAS. Table 5 shows our results using 2 sUAS. The results for 3 sUAS are included in Appendix B. The localization sets are listed in ascending order based on the number of edges with the 5 localization sets used in our scenario highlighted in gray. 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.

6.2 RVRP Results

Using the SNEP solutions for our 5 localization set scenario, we also show the results of the RVRP, using different values for θ and n_s. Note that we can use the SNEP solutions as inputs for the RVRP only because we assumed the same number of homogeneous sUAS for each repair vehicle. For each problem instance, we consider two yards, s_1 and s_2, two sUAS for each repair vehicle, and a repair time, $\tau_{repair,k}$, of 10 minutes for $k = 1, \ldots, 5$. For simplicity, we assume the temporary base set up time is negligible. For the repair vehicle travel time, Γ, we divide Euclidean distances between each pair of locations (k, l) in the set $\mathcal{Y} \cup \mathcal{B}$ by the average vehicle speed.

Recall that earlier in Section 4 we considered a simple scenario where we set $\boldsymbol{\theta} = (1, 1, 1, 1, 1)$ for $\mathcal{L}_1, \ldots, \mathcal{L}_5$. We now consider an alternative scenario where $\boldsymbol{\theta} = (1, 5, 1, 1, 3)$ meaning that localization set $\mathcal{L}_2$ (resp. $\mathcal{L}_5$) was activated 5 hours (resp. 3 hours) prior to t_0. With all else equal, this results in an increase in priority to visit these localization sets earlier. Larger values of θ may arise due to the unavailability of repair vehicles or crews as discussed in Section 4. For each scenario, we also consider the availability of 1 to 2 repair vehicles at each yard to assess the impacts of initial repair vehicle placement (i.e., spatial positioning) on the RVRP solution. Table 6 shows the resulting optimal values for t_{worst} and repair vehicle routes for each scenario.

Figure 19 shows the optimal solution for the RVRP considering the scenario $\theta = (1, 5, 1, 1, 3)$ with 1 repair vehicle at Yard 1, and 2 vehicles at Yard 2. As anticipated, b_2 and b_5 are visited first by the repair vehicles originating from yard s_2. Surprisingly, the optimal solution sends one of these repair vehicles along the longer route (s_2, b_5, b_3, s_2). This is due to the longer sUAS exploration time required for $\mathcal{L}_2$. Since the sUAS take more time to explore $\mathcal{L}_2$, the repair vehicle which visits $\mathcal{L}_2$ compensates by traveling to the closer localization set $\mathcal{L}_1$, thus resulting in the shorter repair vehicle route (s_2, b_2, b_1, s_2). This implies that the other repair vehicle is left to travel along a longer route. The t_{worst} for this problem, is 7.52 hours. When allocated an additional repair vehicle at Yard 1 (2 repair vehicles at both yards), t_{worst} remains at 7.52 hours. This is because for both cases, $\mathcal{L}_2$ overwhelmingly takes the longest time from time of failure alert to repair. Indeed, 7.52 hours in this case represents a tight lower bound for t_{worst} regardless of repair vehicle allocation since $\mathcal{L}_2$ will always be visited first and take the longest time.

Finally, we investigate the impact of repair vehicle spatial positions prior to dispatch. Of the 5 localization sets, $\mathcal{L}_4$

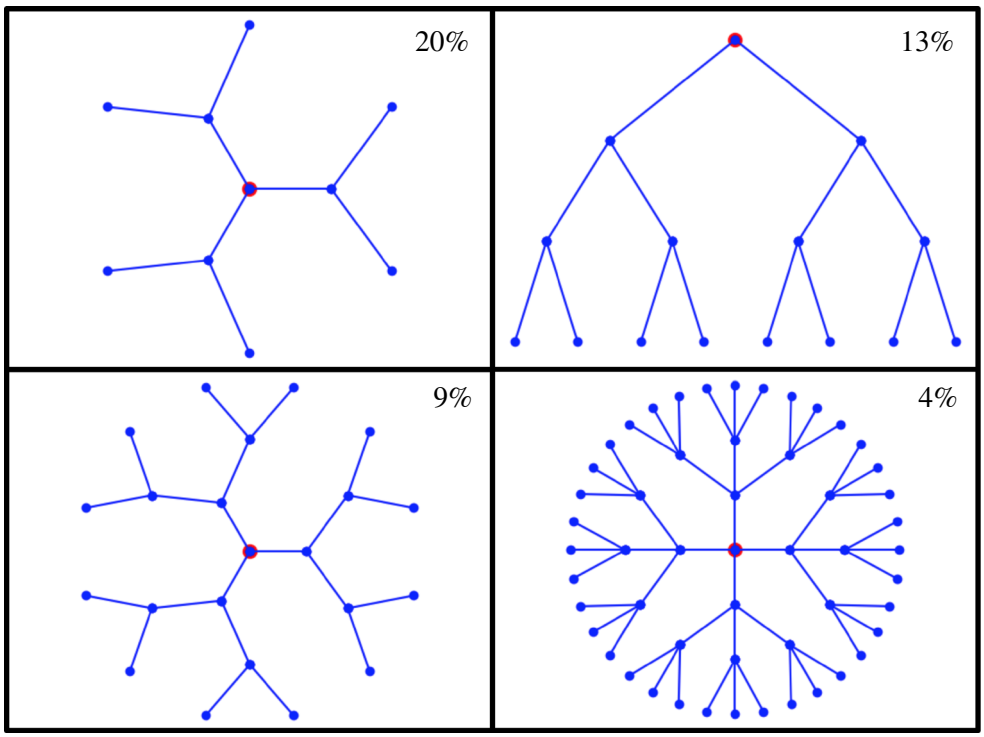


Fig. 20 Representative topologies. Clockwise from top left: extended star (4 internal nodes with 3 degrees each), binary tree (height of 3), extended star (10 internal nodes with 3 degrees each), and extended star (17 internal nodes with 4 degrees each). The network densities for each are shown on the top right. The lower right has the lowest density measure

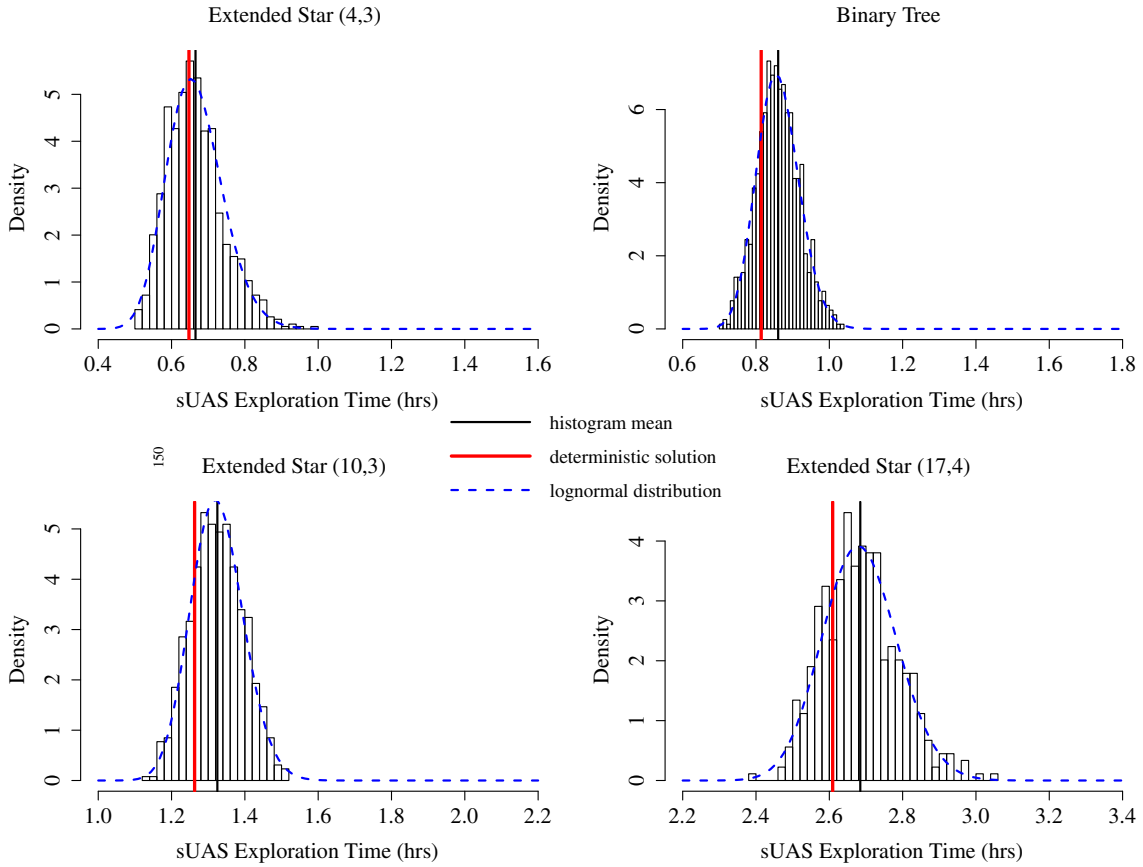


Fig. 21 Histograms showing the distribution of 1000 simulated sUAS exploration times on select topologies

is located near s_1, and is farther from the rest of the localization sets, which are clustered around s_2. Therefore, the case where 2 repair vehicles are at Yard 1 and only 1 is at Yard 2 represents an "unbalanced" situation. In this case, the additional distance traveled by the repair vehicles increases t_{worst} to 8.03 hours compared to 7.53 hours. Although in general, the availability of repair vehicles 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 SNEP and the RVRP, the interconnection between sUAS exploration time

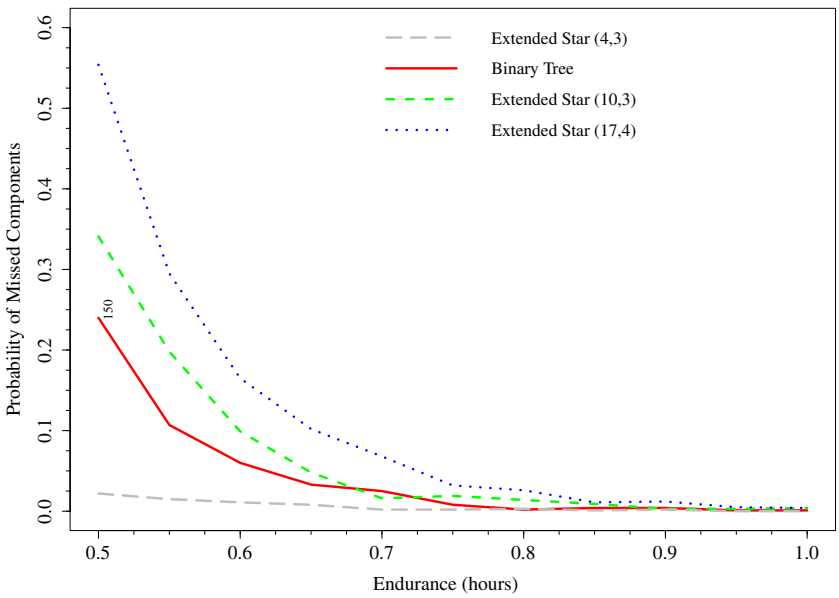


Fig. 22 Probability of missed network components from 1000 simulations given the endurance

and repair vehicle travel times, and the overall solution's sensitivity to repair vehicle spatial positioning and $\boldsymbol{\theta}$.

6.3 Stochastic Travel Times

Although we solved the SNEP using deterministic values for the travel times, the complexity of real time navigation for sUAS 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 [8]. Obstacle avoidance as well as atmospheric turbulence prevalent at low altitudes could also cause the sUAS to deviate from the expected travel time between any two locations [34].

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 [34]. Due to the lack of available sUAS flight data, we use mean cruise airspeed ($\overline{V}_C$), and the maximum allowable operating speed (V_{max}) from Table 2, resulting in the following model for V_C, as described in [35].

$$V_C \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}, \quad \mu = \overline{V}_C, \quad \sigma = \frac{V_{max} - \overline{V}_C}{3} \tag{28}$$

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, 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 sUAS.

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 SNEP solution. We focus on the tree and extended star topologies for our representative localization sets. 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. Fig. 20 shows the four representative localization sets, in the order of decreasing network density for each localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$. We calculate the network density as follows:

$$\frac{|\mathcal{L}_k|}{|\mathcal{V}_k|(|\mathcal{V}_k| - 1)/2} \tag{29}$$

For each chosen topology, we employ 2 of the first class of sUAS shown in Table 2 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 cruise airspeed and endurance of the sUAS. Also, the sUAS 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 sUAS can monitor 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 sUAS optimal exploration time, ξ_k^*. For each selected topology, we first solve the SNEP using our heuristic, assuming deterministic values for τ_{ij}. Using the optimal routes for each sUAS 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 sUAS will not be able to reach some of the monitoring locations toward 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 localization set $\mathcal{L}_k$, $k \in [\![1, K]\!]$, 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 sUAS optimal exploration times, or ξ_k, for those sUAS flights that successfully observed all components, and second, the number of missed components for a given sUAS endurance. Figure 21 shows the histograms showing the distribution of exploration times based on 1000 simulations.

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 SNEP 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 higher exploration time values due to increased travel distances.

Next, we investigate the impact of sUAS endurance on the number of missed components as a result of the simulated travel times. A network component e is missed if the sUAS 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 sUAS in Table 2. We set the maximum distance to a node from the base to half of the endurance. Figure 22 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 sUAS 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 sUAS 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 sUAS from Table 2 with capable sensors (larger monitoring sets) could satisfy mission requirements despite the uncertainty from the external environment.

In summary, our computational study shows that our heuristic can provide high quality solutions for the SNEP within a time frame that meets operational requirements. Accounting for stochasticity in travel times can significantly increase the overall sUAS optimal exploration time. This suggests that the overall RVRP solution will likely incur even further delays in practice. Moreover, in some situations, following the prescribed route from the SNEP output can lead to missed network components depending on the endurance of the sUAS platform. Thus, in real-world environments, proper choice of sUAS platform and conservative route planning is needed to avoid costly setbacks from unidentified failures. Still, sUAS-based inspection can lead to significant cost and time savings in comparison to conventional, purely ground-based, operations.

7 Summary and Future Work

In this paper we provided an end-to-end operational framework to model an sUAS-based inspection process in large-scale infrastructure networks. We introduced important features of the sUAS platform and the operating environment in the development of MIP formulations for the SNEP and RVRP, where the overall objective is to minimize the maximum time elapsed from time of failure alert to time of repair, over all localization sets. Given the significant solution time required for larger instances of the SNEP, 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 sUAS within seconds. We also studied the effect of stochastic travel times on the SNEP solution based on performance metrics such as distribution of sUAS exploration times and probability of missed network components. Further study is required to assess how we might reformulate our heuristic to provide robust SNEP solutions and to analyze the effects on the RVRP solution.

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

Another extension to this work could incorporate randomness in repair times for the RVRP. As mentioned in Section 2, one can use data on the average age of the infrastructure, or material type of network components in a localization set to build a statistical model to estimate $\tau_{repair,k}$ for each $k \in \mathcal{B}$. One can also consider using a criticality factor for each localization set to determine the necessary repair time. This factor could be based on the proximity to population centers or environmentally sensitive areas that would indicate higher priority over others for faster repair. Finally, since we assume that repair time includes the repair vehicle travel time to the exact failure location(s), one can also consider randomness in this travel time (e.g., due to road obstructions).

Acknowledgements A shorter version of this paper was presented at the 2017 International Conference on Unmanned Aircraft Systems (ICUAS). This article elaborates on the overall framework, computational study, and heuristic approach. The work of M. Dahan, S. Amin, and A. Weinert was supported by ICAST: Intelligent Constrained Autonomous Strategic Tasking, which received financial support from MTSI Inc. and MIT MIT Lincoln Laboratory. The project benefited from useful advice by Mike Munizzi. The support of National Science Foundation through grants CNS-1239054 and CNS-1453126 is also greatly acknowledged.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

Appendix A: Table of Notations

Table 7 Table of notations

Notation	Definition
$\mathcal{E}$	Set of vulnerable network components, $e \in \mathcal{E}$
θ_k	Time since failure alert in $\mathcal{L}_k$ to t_0
t_0	Time of dispatch
$\mathcal{Y}$	Set of all yards, $s \in \mathcal{Y}$
$\mathcal{B}$	Set of temporary bases
$\mathcal{L}_k$	Localization set
$\mathcal{V}_k$	Set of monitoring locations sUAS can visit where $k \in \mathcal{B}$
$\mathcal{C}_i^k$	Subset of network components an sUAS can isolate from location i in $\mathcal{V}_k$
b_k	Temporary base location at $\mathcal{L}_k$
ξ_k	sUAS exploration time of $\mathcal{L}_k$
t_{worst}	RVRP objective value, the time elapsed from failure alert to time of repair
$\tau_{repair,k}$	Time to repair the failure(s) in $\mathcal{L}_k$
τ_{batt}	Time to replace the sUAS battery
R	Altitude
u	Number of sUAS in each repair vehicle
n_s	Number of repair vehicles at yard s
τ_{max}	sUAS endurance
τ_{ij}^k	sUAS travel times for every pair of locations $(i, j) \in \mathcal{V}_k^2$
T^k	sUAS Travel Time Matrix for $\mathcal{L}_k$
Γ	Repair Vehicle Travel Time Matrix
γ_{kl}	The repair vehicle travel time from k to l in the set $\mathcal{Y} \cup \mathcal{B}$.
$\mathcal{P}$	Set of simple routes
p	Simple Route

Appendix B: Results for 7 Localization Sets Using 3 sUAS

Table 8 Results of SNEP MIP exact solutions for 3 sUAS compared to heuristic

3 sUAS Localization Set			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%

The last row indicates figures based on the best incumbent MIP objective value found in the given time

Appendix C: Explanation of Big-M Upper Bound for Constraint 13

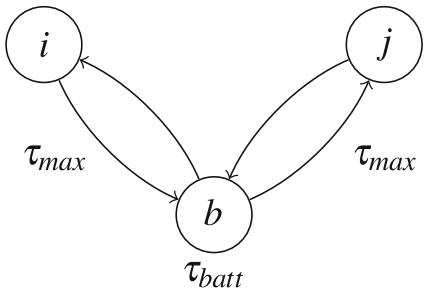


Fig. 23 Here $|\mathcal{V}_k| = 3$. If $x_{ib} = 0$ then $t_{ib} = 0$. If $x_{ib} = 1$, then t_{ib} is at most $2 \times \tau_{max}$ and $1 \times \tau_{batt}$ after having visited node j first

References

1. Angalakudati, M., Balwani, S., Calzada, J., Chatterjee, B., Perakis, G., Raad, N., Uichanco, J.: Business analytics for flexible resource allocation under random emergencies. Manag. Sci. **60**(6), 1552–1573 (2014)
2. Boland, N., Clarke, L., Nemhauser, G., et al.: The asymmetric traveling salesman problem with replenishment arcs. Euro. J. Oper. Res. **123**(2), 408–427 (2000)
3. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part i: route construction and local search algorithms. Transp. Sci. **39**(1), 104–118 (2005)
4. Chabot, D., Craik, S.R., Bird, D.M.: Population census of a large common tern colony with a small unmanned aircraft. PloS one **10**(4), e0122,588 (2015)
5. Chvatal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**(3), 233–235 (1979)
6. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
7. Croes, G.A.: A method for solving traveling-salesman problems. Oper. Res. **6**(6), 791–812 (1958)
8. Dadkhah, N., Mettler, B.: Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance. J. Intell. Robot. Syst. **65**(1-4), 233–246 (2012)
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manag. Sci. **6**(1), 80–91 (1959)
10. Deng, C., Wang, S., Huang, Z., Tan, Z., Liu, J.: Unmanned aerial vehicles for power line inspection: a cooperative way in platforms and communications. J. Commun. **9**(9), 687–692 (2014)
11. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transport Res E-Log **48**(1), 100–114 (2012)
12. Federal Aviation Administration: Unmanned Aircraft Systems (UAS) Operational Approval (N 8900.227). https://www.faa.gov/documentLibrary/media/Notice/N_8900.227.pdf. Accessed: Sept 12, 2017 (2013)
13. Federal Aviation Administration: Authorizations Granted via Section 333 Exemptions. https://www.faa.gov/uas/beyond_the_basics/section_333/333_authorizations/ (2016)
14. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**(6), 345 (1962)
15. Golden, B.L., Levy, L., Vohra, R.: The orienteering problem. Nav. Res. Logist. **34**(3), 307–318 (1987)
16. Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell Labs Tech. J. **45**(9), 1563–1581 (1966)
17. Kentucky Water Resources Research Institute: Water Distribution System Research Database. http://www.uky.edu/WDST/database.html. [Online; accessed February 22, 2017] (2016)
18. Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., Robinson, J.: Unmanned Aircraft System Traffic Management (Utm) Concept of Operations. In: AIAA Aviation Forum (2016)
19. Krefta, L.: Asset knowledge and integrity management earthquake playbook, gas pipeline integrity management program pacific gas electric company (2015)
20. Laporte, G., Nobert, Y., Arpin, D.: Optimal Solutions to Capacitated Multidepot Vehicle Routing Problems. Université de Montréal, Centre de recherche sur les transports (1984)
21. Laporte, G., Semet, F.: Classical Heuristics for the Capacitated Vrp. In: The Vehicle Routing Problem, pp. 109–128. SIAM (2002)
22. Larson, R.C., Odoni, A.R.: Urban operations research. Monograph Dynamic Ideas (1981)
23. Leachtenauer, J.C., Driggers, R.G.: Surveillance and reconnaissance imaging systems: modeling and performance prediction. Artech House (2001)
24. Lusk, R.M., Monday, W.H.: 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 (2017)
25. Mohan, A., Poobal, S.: Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal (2017)
26. Murvay, P.S., Silea, I.: A survey on gas leak detection and localization techniques. J. Loss Prev. Process Ind. **25**(6), 966–973 (2012)
27. Otero, L.: Proof of concept for using unmanned aerial vehicles for high mast pole and bridge inspections (fdot report no. bdv 28 977–02) (2015)
28. Pipeline and Hazardous Materials Safety Administration: National Pipeline Performance Measures. https://phmsa.dot.gov/pipeline/library/data-stats/performance-measures
29. Prosser, P., Shaw, P.: Study of greedy search with multiple improvement heuristics for vehicle routing problems (1996)
30. Rangel, J., Garzón, J., Sofrony, J., Kroll, A.: Gas leak inspection using thermal, visual and depth images and a depth-enhanced gas detection strategy. Revista de Ingeniería **1**(42), 8–15 (2015)
31. Sela Perelman, L., Abbas, W., Koutsoukos, X., Amin, S.: Sensor placement for fault location identification in water networks. Automatica **72**(C), 166–176 (2016). https://doi.org/10.1016/j.automatica.2016.06.005
32. Taillard, É.D., Laporte, G., Gendreau, M.: Vehicle routeing with multiple use of vehicles. J. Oper. Res. Soc. **47**(8), 1065–1070 (1996)
33. Vigo, D.: A heuristic algorithm for the asymmetric capacitated vehicle routing problem. Eur. J. Oper. Res. **89**(1), 108–126 (1996)
34. Ware, J., Roy, N.: An analysis of wind field estimation and exploitation for Quadrotor Flight in the urban canopy layer. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1507–1514. IEEE (2016)
35. Weinert, A., Campbell, S., Vela, A., Schuldt, D., Kurucar, J.: A Well Clear Recommendation for Small Unmanned Aircraft Systems Based on Unmitigated Collision Risk. Tech. rep., MIT Lincoln Laboratory (2017)
36. Whipple, S.D.: Predictive Storm Damage Modeling and Optimizing Crew Response to Improve Storm Response Operations. Ph.D. thesis, Massachusetts Institute of Technology (2014)
37. Williams, K.W., Gildea, K.M.: A review of research related to unmanned aircraft system visual observers. Federal Aviation Administration Final Report (2014)

Andrew C. Lee is currently an Active Duty U.S. Army Officer and Ph.D. candidate in the Interdepartmental Doctoral Program in Transportation at the Massachusetts Institute of Technology (MIT), and focuses on applications of optimization methods, Unmanned Aerial Systems, and machine learning in the context of infrastructure network monitoring. He received his B.S. degree from the United States Military Academy in 2003 and his M.S. degree in Transportation from MIT in 2012.

Mathieu Dahan received his B.S. degree in Mathematics from the University of Paris XI-Sud, France, in 2014, his B.Eng. and M.Eng. degrees from the Ecole Centrale Paris, France, in 2013 and 2016, and his M.S. degree in Computation for Design and Optimization from the Massachusetts Institute of Technology (MIT), USA, in 2016. He is currently a Ph.D. candidate at MIT, and focuses on the applications of optimization methods and game-theoretic tools for assessing and improving the survivability of infrastructure networks to disruptions.

Andrew J. Weinert is a member of the Humanitarian Assistance and Disaster Relief Systems Group in the Homeland Protection and Air Traffic Control Division at MIT Lincoln Laboratory. His master's thesis focused on optimization of aircraft avoidance systems using information theoretics and parallel processing techniques. Joining the Laboratory in 2009, he has support programs in Public Safety systems and manned and unmanned aircraft avoidance. He received a MS in Electrical and Computer Engineering at Boston University and a BS in Security and Risk Analysis with minors in Information Science Technology for Aerospace Engineering and Natural Science from the Pennsylvania State University.

Saurabh Amin is Robert N. Noyce Career Development Associate Professor in the Department of Civil and Environmental Engineering at MIT. His research focuses on the design and implementation of high confidence network control algorithms for infrastructure systems. He works on robust diagnostics and control problems that involve using networked systems to facilitate the monitoring and control of critical infrastructures. He also studies the effect of security attacks and random faults on the survivability of networked systems, and designs incentive-compatible control mechanisms to reduce network risks. Dr. Amin received his Ph.D. from the University of California, Berkeley in 2011.