Joint UAV and Truck Routing Under Uncertain Disruptions: Measuring the Value of Information

Leonardo Lozano
Operations, Business Analytics & Information Systems, University of Cincinnati, Cincinnati, Ohio 45221, leolozano@uc.edu

Juan S. Borrero
School of Industrial Engineering & Management, Oklahoma State University, Stillwater, OK 74078, juan.s.borrero@okstate.edu

We consider a joint UAV and truck routing problem in which the operation of the UAV is subject to uncertain disruptions. The planner initially does not know in which locations a disruption will occur but can refine his/her knowledge by spending additional resources to probe locations for additional information, removing the uncertainty for the probed locations. In order to determine the best planning approach, the planner requires measuring how valuable is the information that is gathered by the probes. To this end, we present a framework that computes the value of information based on a two-stage stochastic optimization problem in which the uncertainty is partially revealed depending on the decisions of the first stage, that is, in the first stage the planner selects the locations to probe and in the second stage the planner selects the route based on the results of the probes. Our approach is able to find optimal probing plans for the case in which the disruptions are statistically independent. We reformulate the two-stage stochastic problem as a bilevel optimization problem with multiple followers, in which the lower-level problem is a challenging joint UAV and truck routing problem. We propose two formulations for the routing problem, one based on subtour elimination constraints, and another based on decision diagrams. We also develop two exact approaches to solve the resulting bilevel problem, one based on a value function reformulation and another based on scenario decomposition. Our extensive computational experiments show that the proposed approaches are able to solve bilevel problems with up to 30 nodes to optimality and that probing, even if it is only on a small subset of locations, can yield significant gains in solution quality compared to not probing any locations.

Key words: Optimization under uncertainty; Bilevel optimization; Value of information; Integer programming; UAV routing

---

1. Introduction

Recent technological advances have increased the application areas where the use of Unmanned Aerial Vehicles (UAVs) can significantly improve operations. UAVs can get to locations that are difficult to reach for ground-operated vehicles; they provide more flexibility than traditional vehicles because they do not require drivers, roads, nor physical infrastructure to reach their destinations; and they can operate in hazardous regions without requiring human physical presence. Given these benefits, UAVs promise to be (or are) particularly useful in last-mile delivery operations (Cornell et al. 2023, Amazon 2022), humanitarian logistics and disaster response (WFP 2018, Unicef 2023), homeland security and defense operations (Harbaugh 2018, Congressional Research Service 2022), among others. On the other hand, UAV operations are highly prone to disruptions and failures due to their connectivity requirements, as well as to the complex environments where these operations take place.

As an illustration consider Intelligence, Surveillance, and Reconnaissance (ISR) operations. In the US, ISR missions involving short distances are normally carried out using small UAVs such as the RQ-11 Raven or the Wasp III, which are controlled by a mobile ground control unit or *truck* that has to be kept in range of the UAV (U. S. Army UAS Center of Excellence 2010, Austin 2010, USAF 2017, 2016). At the same time, counter-unmanned aircraft systems (C-UAS) or jammers have become widely available (Wang et al. 2021) with a growing market size currently valued at around $900 million (Polaris Market Research 2021). Jammers can interfere with RF signals such as remote control or GPS signals (Bhattacharya and Basar 2010, Li et al. 2018, Wang et al. 2021), and they can be made operational relatively easily and cheaply (Nguyen et al. 2014, Farlik et al. 2016, Multerer et al. 2017, Curpen et al. 2018, Mototolea and Stolk 2018, Pärlin et al. 2018). Electronic warfare can be very disruptive for ISR missions. For example, Russia's jammers have been quite successful in recent years against small and large surveillance UAVs (Trevithick 2018, Axe 2022).

Disruptions to the operation of a UAV are not only due to malicious agents but can also come from multiple failure-inducing factors depending on the application area. For instance in a disaster response operation a UAV might not be able to perform its mission due to the state of infrastructure or the environmental conditions after the occurrence of a disaster (Greenwood and Joseph 2020).

As a result, it is important to account for disruptions when developing routing plans involving UAVs. However, planners typically do not know in advance where the disruptions will happen and therefore considering disruptions when planning UAV operations results in a challenging routing problem under uncertainty. From the planner's perspective, an initial way to model this problem is to use a *limited information* (LI) approach, where the uncertainty is estimated using probability distributions or uncertainty sets. These estimates can then be used in a stochastic or robust optimization model to obtain a UAV operation plan. This approach is commonly taken in the literature, see e.g., Evers et al. (2014b), Moskal et al. (2022).

Alternatively, the uncertainty about the locations of the disruptions can be mitigated by investing resources into information collection before planning the operation. For example, in ISR missions, intelligence on the existence of jammers can be gathered by deploying advance static or mobile spectrum analyzers close to the locations of interest (DHS 2019, CRFS 2023). In other application areas, satellites can be used to gather images and identify potential disruptions (Dahm 2020, Development Aid 2020).

We refer to any such activity to gather additional information as *probing*. Thus, in contrast to the LI, the planner can consider a *full-information* (FI) approach, which would completely remove the uncertainty by investing resources into probing all the locations for disruptions. The resulting routing problem would be deterministic, and the planner could use approaches such as the models in e.g., Manyam et al. (2019) or Roberti and Ruthmair (2021) to plan the routes.

There is a trade-off between the LI and FI alternatives. On the one hand, LI plans are cheaper to develop because they do not involve probing costs; however, they risk incurring excessive operational costs due to estimation errors. On the other hand, FI plans do not involve unexpected operational risks; however, the probing costs can make them prohibitively expensive or it could simply be the case that it is impossible to probe all the locations. Deciding which approach to follow, or whether a partial probing approach might be more appropriate, requires to determine if the benefits of having more information justify the higher probing costs. Such analysis, to the best of our knowledge, has not been carried out yet in the optimization literature. To address this need, the overall goal of this paper is to develop an optimization-based framework to evaluate the *value of information* in joint truck and UAV routing operations assuming there is uncertainty regarding the locations in which disruptions occur.

Specifically, we consider that the planner controls a unit consisting of one UAV and one ground vehicle. Successfully serving a location yields a known location-dependent reward. The planner seeks to determine a joint UAV and truck route that maximizes the reward collected. Before determining the route, the planner has limited resources available to probe a fixed number of locations. Probing a location confirms whether a disruption will occur in the location or not. In a first decision stage, the planner selects which locations to probe, whereas in a second decision stage the planner selects the route based on the partial information collected by the probes. The decisions in the second stage are made by maximizing an *estimated reward* associated with the selected route and the partial information discovered. Once the unit is deployed, the disruptions take place (or not) for all the locations visited and the *actual reward* of the route selected is realized.

Formal setup. We consider a directed network $\mathcal{G}=(\mathcal{N},\mathcal{A})$ where the locations are represented by the set of nodes $\mathcal{N}$ and $\mathcal{A}\subseteq\mathcal{N}\times\mathcal{N}$ is the set of arcs. Routes must satisfy a series of constraints regarding the coordination of the ground vehicle and UAV among others (see Section 3.2). We define a *visit plan* as a binary vector that corresponds to all the locations visited by a route, independently of the order in which they are visited. Based on the set of all feasible routes, we define the set of all feasible visit plans as

$$\mathcal{V}=\left\{v\in\{0,1\}^N:\text{Exists a feasible route that visits locations for which } v_i=1,\ \forall i\in\mathcal{N}\right\}. \tag{1}$$

where $N=|\mathcal{N}|$. The occurrence of disruptions at the locations is represented by a random vector $J=(J_i\colon i\in\mathcal{N})\in\{0,1\}^N$ from a probability space $(\Omega,\Psi,\mathbb{P})$. Random variable J_i takes the value 1 if a disruption happens at location i and takes the value 0, otherwise. As it is commonly done in the literature, we assume that there is a finite set of scenarios $\Xi:=J(\Omega)\subseteq\{0,1\}^N$; with $\pi^\xi:=\mathbb{P}(\{\omega\colon J(\omega)=\xi\})$ being the probability that scenario $\xi\in\Xi$ happens, with $\pi^\xi>0$ for all $\xi\in\Xi$ and $\sum_{\xi\in\Xi}\pi^\xi=1$. These probabilities are computed from experts' judgment, situational assessments, among others.

Denote by z_i, $i\in\mathcal{N}$, the first-stage binary variable that takes the value 1 if the planner decides to probe location i and 0 otherwise. Let B, $0\le B\le N$, be the available budget for probing in the first stage and let $\hat{R}(z,\xi,v)$ be the estimated reward associated with a probing plan z, scenario ξ, and a route that visits locations given by visit plan v. Function $\hat{R}$ depends on vector ξ; however, it only observes the ξ_i components for which $z_i=1$, i.e., the occurrence or absence of disruptions at the locations probed. Similarly, let $R(\xi,v)$ be the actual reward corresponding to scenario ξ and a visit plan v, and note that the actual reward is independent of the probing plan because once the route is executed, then the disruptions occur (or not) for all the locations visited. We study two-stage stochastic problems of the form

$$\Gamma_B:=\max_z\ \mathbb{E}[\mathcal{F}(z,J)] \tag{2a}$$

$$\text{s.t.} \sum_{i \in \mathcal{N}} z_i \leq B \tag{2b}$$

$$z \in \{0,1\}^N, \tag{2c}$$

where

$$\mathbb{E}[\mathcal{F}(z,J)] = \sum_{\xi \in \Xi} \pi^{\xi} \mathcal{F}(z,\xi), \tag{3}$$

and for each scenario $\xi \in \Xi$, the function $\mathcal{F}(z,\xi)$ is defined as:

$$\mathcal{F}(z,\xi) = \max_{v} \Big\{ R(\xi,v) \colon v \in \arg\max_{v'} \big\{ \hat{R}(z,\xi,v') \colon v' \in \mathcal{V} \big\} \Big\}. \tag{4}$$

The first-stage problem decides the probing plan with the objective of maximizing the expected actual reward (2a), which is computed by solving the second-stage problem. In the second stage, for a given probing plan and a realization of the uncertainty, the planner first selects a route that maximizes the estimated reward and then computes the actual reward after executing the selected route.

A few remarks are in order. First, in contrast to standard two-stage stochastic problems in which the second stage solves a single-level problem for each realization of the uncertainty, our second stage problem is defined by "argmax" constraints resulting in a bilevel structure. Second, our formulation admits different functional forms for the estimated reward function $\hat{R}$ and the actual reward function R, which we define as linear functions in this work (see Section 3.1). Third, the second stage involves solving a challenging discrete joint UAV and truck routing problem. We model this problem as a variation of the Orienteering Problem (OP) (Tsiligirides 1984, Golden et al. 1987, Gendreau et al. 1998a,b) with truck and UAV and include additional operational considerations (see Section 3.2).

Within this framework, Γ_0 corresponds to the expected actual reward associated with the LI approach whereas Γ_N is the expected actual reward when using the FI approach. For any B we refer to $\Gamma_B - \Gamma_0$ as the *value of information* associated with having B probes whereas $\Gamma_N - \Gamma_B$ is referred as the *price of not having full-information* associated with having B probes. The value of information measures the largest possible expected reward gain associated with being able to probe B locations. The price of not having full information measures the largest possible expected reward that is not gathered due to probing only B locations.

Statement of Contribution. The main contributions of this work are as follows. We propose a general two-stage stochastic program for orienteering problems with truck and drone under uncertainty that can quantify the value of knowing with certainty whether disruptions to the UAV occur in a subset of the locations. Our approach is exact for the case in which the disruptions are statistically independent, i.e., it is able to find optimal probing plans. The two-stage problem requires solving a challenging joint UAV and truck routing subproblem, for which we propose two alternative formulations. The first one is based on subtour elimination constraints and the second one is based on decision diagrams. We reformulate the two-stage stochastic problem as a bilevel problem with multiple followers and propose two solution methods, the first one follows a value function reformulation, and the second one is a scenario-based decomposition.

We derive bounds on the value of information and the price of not having full information, as well as valid inequalities. The first of these bounds does not depend on B whereas the second one does. These results suggests that there are instances where the value of information is independent on the number of probes, and that there are instances where optimally placed probes always reveal at least some information about the remaining uncertainty. Some of these bounds are applicable for general problem settings as they are not limited to the UAV and truck routing problem we consider.

In addition, we conduct computational experiments that show that:

1. Having full information often results in a considerable increase to the optimal value.
2. Probing a small subset of the locations can yield large improvements to the optimal value and in many cases get close to the value obtained under full information.
3. Our solution approaches for the bilevel problem with multiple followers are able to solve several problem instances to optimality with a time limit of 4 hours, and the scenario decomposition approach is able to consistently produce solutions within 5% of the optimal value on average.

To the best of our knowledge, our bilevel formulation is the first one to provide optimal probing plans (as opposed to heuristics or approximation algorithms) for these type of information discovery problems over independent uncertain events, as well as the first one to conduct an extensive computational study regarding the value of information in joint truck and UAV operations under uncertainty.

2. Literature review

A first stream of related literature deals with ISR missions. These problems are commonly framed as extensions of the (team) Orienteering Problem (Tsiligirides 1984), which is a challenging generalization of the (NP-hard) Traveling Salesman Problem (TSP) (Golden et al. 1987). A two-stage multi UAV routing problem with uncertainty is considered in Kress and Royset (2008). In the first stage, mobile command control and ground control centers are located; once the uncertainty about the target locations and the availability of UAVs are revealed, an OP-based model prescribes the routing. In Royset and Reber (2009) the authors extend the basic OP model to solve a surveillance problem for improvised explosive devices that involves multiple UAVs and multiple time periods. Adaptations of OP formulations to simultaneously optimize UAVs' sensor selection and their surveillance routes are studied in Mufalli et al. (2012). In Evers et al. (2014b) the authors study a UAV surveillance problem with uncertain resource consumption; they propose an OP-based robust optimization model to solve the problem. A similar setting is studied in Evers et al. (2014c), who consider a two-stage stochastic problem. In the recourse, the decision-maker can re-route the UAV after the uncertain resource consumption is revealed. A two-stage stochastic problem that routes the UAV following a TSP in the first stage and that schedules additional surveillance on the recourse (depending on the amount of uncertain information collected) is presented in Rajan et al. (2022).

Dasdemir et al. (2020) considers a deterministic multi-objective UAV routing problem that optimizes travel times and threat detection. The model assumes a continuous movement space that results in 'multi-arcs' between locations, and assumes that parts of the regions are subjected to radars, whose locations are known in advance. The model is solved heuristically using a multiobjective evolutionary algorithm. Dasdemir et al. (2022) considers a similar tri-objective model where information collection and time windows are also considered. The authors propose an OP-based MIP formulation for the problem, which is solved to scale using a hybrid algorithm. Moskal et al. (2022) consider an OP-based UAV routing problem using waypoints over a grid, where the objective is to maximize the expected information collection. Their model includes constraints for the probability that the flight duration and the radar-detection probability are above certain given thresholds. Assuming specific probability distributions for the uncertainty, they reformulate and solve the problems using MIPs. Other extensions of OP-based models are given in Pietz and

Royset (2013), Evers et al. (2014a), Verbeeck et al. (2016, 2017), Moskal and Batta (2017), Liu et al. (2019) and the references given therein.

Regarding computational results, the exact approaches considered in the aforementioned works are able to solve small to medium-sized instances ranging from 14 to 30 nodes (with the exception of specific instances, see Kress and Royset (2008)). For example, as shown in Dasdemir et al. (2022), solving a deterministic multiobjective problem using a state-of-the-art MIP solver can take days for instances with 30 nodes; likewise, Moskal et al. (2022) solved the MIP formulations within minutes to hours in instances of less than 20 nodes. Most adopted techniques that solve larger instances rely on hybrid and heuristic methods.

The aforementioned studies base their approach to routing in the OP, as we do in this paper; however, there are two important differences between our setting and these studies. First, these models do not simultaneously route UAVs and ground units (only Kress and Royset (2008) considers this setting, but does it sequentially rather than simultaneously). Second, these UAV routing problems are not concerned with the value of information and do not consider the probing decision stage, which is our main subject of study.

Another stream of related literature deals with jointly scheduling UAVs and trucks to deliver packages across a set of customers. Particularly, it has been shown that using UAVs in parallel with trucks can result in substantial performance improvements (Wang et al. 2017, Poikonen et al. 2017, Carlsson and Song 2018, Reed et al. 2022). Joint UAV-truck routing problems are framed as variants of the TSP and most of them are solved heuristically, see Mathew et al. (2015), Murray and Chu (2015), Agatz et al. (2018), Ha et al. (2018), Boysen et al. (2018), Otto et al. (2018), Poikonen et al. (2019) and the references therein.

Exact methods for the aforementioned problems are relatively scarce and mostly focused on the TSP-D, which is a single-vehicle TSP extension where a UAV can detach at customer locations to make a delivery in other node, after which it must reattach to the truck (Agatz et al. 2018). Such methods include solving MIP formulations using either state-of-the-art solvers or specialized branch-and-price methods (Roberti and Ruthmair 2021, Yang et al. 2023); Benders' decomposition (Vásquez et al. 2021); dynamic programming (Bouman et al. 2018); branch-and-price-and-cut (Li and Wang 2022); and other enumerative methods (Yurek and Ozmutlu 2018). In general, these approaches can solve instances with up to 30 nodes within an hour; instances with more nodes, however, are not typically solved to optimality in an hour. We remark that TSP-D models do not account for longer UAV-only subtours (in the TSP-D the UAV can only visit one node before returning to the truck), and uncertainty in the success of visiting a node.

In this regard, an exact approach for a problem similar to the TSP-D is presented by Manyam et al. (2019). The main differences with respect to the TSP-D is that the UAV can visit more than one node per sub-tour, and that the truck has to wait in the same location while the UAV is detached. The authors provide a branch-and-cut method based on a MIP formulation that is capable of solving 50-nodes instances within six hours. A further generalization allowing uncertainties and arbitrary rendezvous points is solved using a chance-constrained MIP by Du et al. (2020). We model the routing problem in a similar way as in these models, the key difference is that we do not impose that all nodes have to be visited.

Regarding using probing to estimate the value of information, Gupta et al. (2016) consider a probing setting with unlimited budget for combinatorial problems. In their model, only 'items' probed in the first stage can be included in the objective function of the second stage. The authors propose an approximation algorithm and focus on bounding an 'adaptivity gap' between optimal online and offline policies. Similar models are studied by Gupta and Nagarajan (2013) and Adamczyk et al. (2016), who also propose approximation algorithms. Guha and Munagala (2012), and particularly, Goel et al. (2010), consider a probing setting that is closer to what we do in this paper. The main differences is that their focus is on developing approximation algorithms. They show that for specific combinatorial problems there is a constant-factor approximation algorithm (based on solving the 'outlier problem') for both the online and offline versions of the problem. We note, however, that the complexity of the algorithm depends on the computational complexity of the outlier problem. This complexity, in turn, is problem-dependent, thus constant-factor approximations might not be available to all problems. Finally, more general probing problems with a similar approach to Goel et al. (2010) have also been considered in the two-stage stochastic programming literature. Their focus, is however, on the statistical and mathematical properties of the model, see Artstein and Wets (1993), Artstein (1994, 1999).

3. Definitions and problem description

Next, we continue the description given in the Formal Setup of Section 1. We introduce additional notation and definitions, describe our assumptions, and provide a bilevel reformulation for the two-stage stochastic problem (2).

3.1. Defining the reward functions

The optimization framework described in the Formal Setup admits different functions to represent the estimated reward $\hat{R}(z,\xi,v)$ (associated with a probing plan z, scenario ξ, and visit plan v) and the actual reward $R(\xi,v)$ (corresponding to scenario ξ and a visit plan v). A standard approach, following the literature on orienteering problems (Tsiligirides 1984, Vansteenwegen et al. 2011), is to define $R(\xi,v)$ as a linear function. The estimate $\hat{R}(z,\xi,v)$, on the other hand, can be based on any estimation method, from closed-form functions as we do in this paper, to deep learning-based methods that might be more accurate and can include additional features of the instance (Bengio et al. 2013). However, it is important to note that these functions directly impact the difficulty of the problem because they are to be maximized both in the inner and outer optimization problems given by (4). We propose linear functions (with respect to visit decisions v) based on standard expectations over the scenarios to model both the estimated and actual reward. In this way we are able to exploit existing MIP solvers that excel at optimizing linear functions while maintaining reasonable modeling capabilities.

Recall that the planner does not know where disruptions occur, but keeps a set of possible scenarios $\Xi \subseteq \{0,1\}^N$, where scenario ξ happens with probability π^ξ . In scenario $\xi \in \Xi$ a disruption occurs at location i if $\xi_i = 1$, otherwise $\xi_i = 0$. Let r_i be the reward corresponding to successfully serving the i-th location using the UAV, $i \in \mathcal{N}$. If the planner probes $i \in \mathcal{N}$ and no disruption occurs, then the actual reward at i is given by r_i. Else, if the planner probes $i \in \mathcal{N}$ and a disruption happens, then the actual reward at i is 0. Finally, if the planner does not probe location $i \in \mathcal{N}$, then the planner *estimates* the reward to be $0 \cdot p_i + r_i(1-p_i)$, where p_i is the probability that a disruption occurs at i. This probability is computed from the scenario probabilities by

$$p_i := P[J_i = 1] = \sum_{\xi \in \Xi:\, \xi_i = 1} \pi^\xi \qquad i \in \mathcal{N}. \tag{5}$$

The estimated reward if the planner selects to probe the locations in z, given scenario ξ, and given the visit plan $v \in \mathcal{V}$ is defined by

$$\hat{R}(z,\xi,v) = \sum_{i \in \mathcal{N}:\, z_i = 1} r_i(1-\xi_i)v_i + \sum_{i \in \mathcal{N}:\, z_i = 0} r_i(1 - P_i(z,\xi))v_i \tag{6a}$$

$$= \sum_{i \in \mathcal{N}} r_i\big((1-\xi_i)z_i + (1 - P_i(z,\xi))(1-z_i)\big)v_i, \tag{6b}$$

recalling that $v_i = 1$ if and only if location $i \in \mathcal{N}$ is visited, zero otherwise. In equation (6), $P_i(z, \xi)$ is the updated probability that there is a disruption at i, given the information discovered by the probing decisions in z and given that the scenario is ξ. Specifically,

$$P_i(z, \xi) := P[J_i = 1 | J_z = \xi_z], \tag{7}$$

where $J_z := (J_k \colon z_k = 1)$ and $\xi_z := (\xi_k \colon z_k = 1)$. These probabilities can be computed via highly non-linear expressions in terms of z (see Appendix A). On the other hand, if the J_is are independent, then we have that $P_i(z, \xi) = p_i$, for any z and ξ. Thus, by assuming independence the estimated reward simplifies to

$$\hat{R}(z, \xi, v) = \sum_{i \in \mathcal{N}} r_i \big((1 - \xi_i) z_i + (1 - p_i)(1 - z_i) \big) v_i. \tag{8}$$

When a visit plan $v \in \mathcal{V}$ is executed, then disruptions occur (or not) at all the locations visited and the planner receives the actual reward $R(\xi, v)$, which is given by

$$R(\xi, v) = \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) v_i. \tag{9}$$

The estimated and actual reward are the same for nodes that are probed. Additionally, note that the actual reward does not directly depend on the probing plan. On the other hand, an optimal visit plan will depend on the probing plan since it maximizes the estimated reward, see Equation (4).

In our optimization models and hereafter, we assume that the disruptions are independent and thus we use the estimated reward defined by (8). Under this assumption, our models are exact, i.e., they find optimal probing plans that maximize the expected actual reward as defined in (9). If the random variables J_i, $i \in \mathcal{N}$ are not independent, then the estimated reward $\hat{R}$ given in equation (8) can be considered as a *partial update* as it only uses the information discovered in probed nodes to update the probabilities. In this case our models can still be used as heuristics but they cannot guarantee optimal probing plans as it may be the case that for some of the locations not probed, $P_i(z, \xi) \neq p_i$.

3.2. Assumptions on the joint routing of the UAV and truck

Our routing problem considers truck and UAV operational requirements, which leads us to the following assumptions:

A1. The UAV can visit several locations each time it is airborne; its maximum airborne time is $T_U > 0$, which is a known quantity.

A2. The UAV has to be within $U > 0$ units of distance to the truck at all times. U is known and fixed.

A3. The locations $\mathcal{N}_V \subseteq \mathcal{N}$ can only be visited by the UAV.

A4. The truck takes a time t_{ij} to traverse arc $(i,j) \in \mathcal{A}$; the UAV travel time is equal to $\alpha_{ij} t_{ij}$, where $\alpha_{ij} < 1$.

A5. The complete operation can take at most T time units, T is known and fixed.

A6. The UAV can only be deployed at a location and it has to be recovered at the same location.

A7. The truck waits in the same location where the UAV is deployed while the UAV is airborne and can move to other locations only after recovering the UAV.

These assumptions are standard and are motivated by operational requirements in joint truck and UAV routing see Austin (2010), Manyam et al. (2019).

We now provide an illustration of the flow of information, the estimated reward, the actual reward, and the joint UAV and truck routing setting.

Example 1. *Consider the network depicted in Figure 1. There are six locations that can be visited jointly by the truck and the UAV and four locations that can be visited only by the UAV. The particular scenario depicted is* $\xi_i = 0$ *for* $i \neq 9, 10$ *and* $\xi_9 = \xi_{10} = 1$*. Initially, the planner is only able to observe the network, the probabilities* p_i *and the rewards* r_i*, see Figure 2a. Suppose that the planner probes nodes 8 and 10. Then he/she discovers that a disruption will occur at node 10 and that node 8 has no risk of a disruption happening, see Figure 2b. Using this information, the planner decides the joint route in Figure 2c. The truck visits nodes 1, 2, 4, and 6 and the UAV is deployed from the truck at node 2. While the truck waits there, the UAV visits nodes 8, 9, and 7, and then returns to node 2. The estimated reward of this route is* $\hat{R} = \sum_{i \in \{1,2,4,6,7,9\}} r_i p_i + r_8$*. The actual reward once the routing takes place and the scenario* ξ *is revealed is* $R = \sum_{i \in \{1,2,4,6,7,8\}} r_i$.

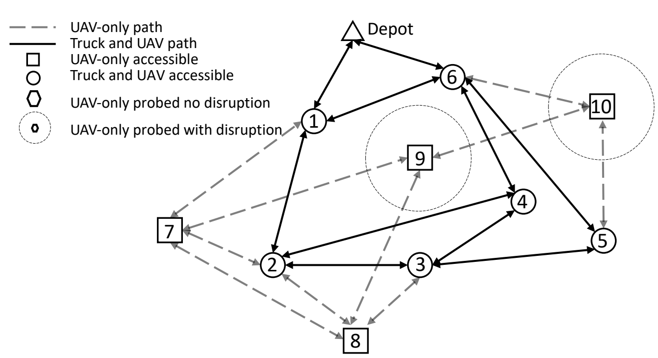
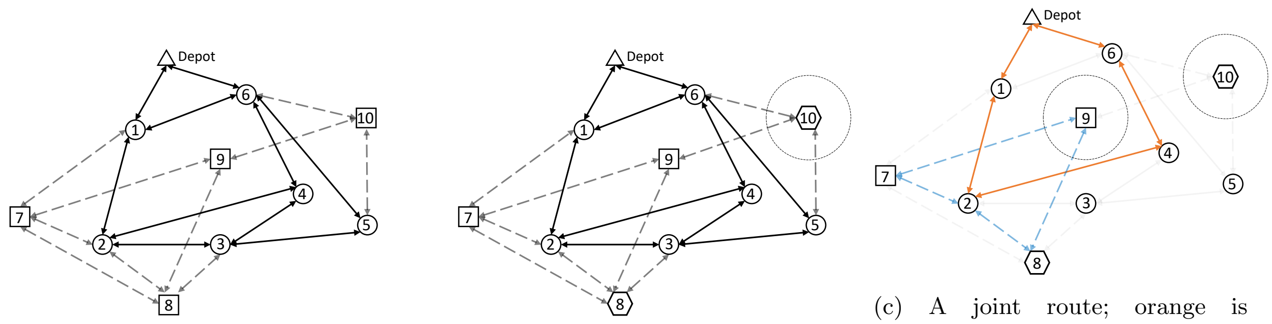


Figure 1 An illustrative problem instance for a given scenario



(a) What the planner observes (b) The planner probes 8 and 10 (c) A joint route; orange is truck/UAV and blue is UAV

Figure 2 Decision-making process for the planner

3.3. A bilevel reformulation of Problem (2)

A standard approach to solve two-stage stochastic programs is to use a deterministic equivalent monolithic formulation, in which the expectation is represented by a weighted sum over all possible scenarios and copies of the second-stage variables are introduced for each scenario. We generate such monolithic formulation for problem (2) as follows. For scenario $\xi \in \Xi$, let $v^\xi \in \mathcal{V}$ be the visit plan under scenario ξ and let all visit plans be represented by $v^\Xi = \{v^\xi\}_{\xi\in\Xi}$, that is, v^ξ is an optimal solution to the inner problem in (4). Then

$$\Gamma_B = \max_{z, v^\Xi} \sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} r_i(1-\xi_i) v_i^\xi \tag{10a}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} z_i \le B \tag{10b}$$

$$v^\xi \in \arg\max_v \Big\{ \sum_{i\in\mathcal{N}} r_i\big((1-\xi_i)z_i + (1-p_i)(1-z_i)\big)v_i \ : \ v \in \mathcal{V} \Big\} \qquad \forall \xi \in \Xi \tag{10c}$$

$$z \in \{0,1\}^N. \tag{10d}$$

Formulation (10) describes a bilevel problem with multiple followers (one per scenario) under the so-called *optimistic assumption* (Dempe 2002), that is, it breaks possible ties in the arg max by selecting the routes that maximize the objective function in (10a). The probing stage corresponds to the *leader's* problem, while the joint UAV and truck routing problem corresponds to the *follower's* problem. The objective is to maximize the expected actual reward while ensuring that the visit plans maximize the estimated reward for each scenario. As we will show next, solving the follower problem and enforcing constraints $v^\xi \in \mathcal{V}$ for all $\xi \in \Xi$ involves formulating a challenging joint UAV and truck routing problem.

4. Mathematical models for the single-level routing problem

A key challenge for solving problem (10) is solving the follower's problem for each scenario $\xi \in \Xi$ to determine the visit plan $v^{\xi} \in \mathcal{V}$; that is, we need to solve a joint UAV and truck routing problem that maximizes the estimated reward for a given scenario and probing plan. In this section we examine this problem in more detail and derive two mixed-integer programming formulations for the set of feasible visit plans $\mathcal{V}$ based on assumptions **A1**–**A7** (see Section 3.2). The first formulation is build upon standard subtour elimination constraints. The second formulation is based on Decision Diagrams (DDs). We close the section by analyzing two limiting cases, the first one is the case without probing ($B = 0$) and the second one is the case with full information ($B = N$).

4.1. Subtour elimination formulation

Subtour elimination constraints, when used within a cutting generation framework, have been proven effective to solve routing problems such as the TSP. For the joint routing problem under consideration, we have three types of subtour elimination constraints, one for the truck and two for the UAV. The truck-based constraints do not allow subtours unless the subtour starts at the depot. The UAV-based constraints, on the other hand, coordinate the UAV-only subtours with the truck.

Let x_{ij} be a binary variable that takes a value of 1 if arc $(i,j) \in \mathcal{A}$ is traversed by the truck in the route and let y_{ij} be a binary variable that takes a value of 1 if arc $(i,j) \in \mathcal{A}$ is used by the UAV in a UAV-only route. For our formulation we allow self loops (i.e., $y_{ii} = 1$), which correspond to a location being visited jointly by the UAV and truck. Let w_i be a binary variable that takes a value of 1 if node $i \in \mathcal{N}$ is visited by the truck and let v_i be a binary variable that takes a value of 1 if node $i \in \mathcal{N}$ is visited by the UAV (even if visited jointly with the truck).

For any $\mathcal{S} \subseteq \mathcal{N}$ let $\mathcal{A}(\mathcal{S})$ be all the arcs with both endpoints in $\mathcal{S}$. For any $\mathcal{C} \subseteq \mathcal{A}$ let $x(\mathcal{C})$ and $y(\mathcal{C})$ be $\sum_{(i,j)\in\mathcal{C}} x_{ij}$ and $\sum_{(i,j)\in\mathcal{C}} y_{ij}$, respectively, and let $\mathcal{N}(\mathcal{C})$ be the set of all nodes incident to arcs in $\mathcal{C}$. Let Δ be the set of all candidate closed loops starting at any given node in $\mathcal{N}$. A closed loop is defined as a set of arcs $\mathcal{C} = \{(i_1, j_1), \ldots, (i_q, j_q)\}$ such that $i_k = j_{k-1}$ for all $k = 2, \ldots, q$ and $i_1 = j_q$, and such that $j_k \neq j_{k'}$, $k, k' = 1, \ldots, q$, $k \neq k'$. Candidate Loops in Δ have a total duration of at most T_U, that is, $\sum_{(i,j)\in\mathcal{C}} \alpha_{ij} t_{ij} \le T_U$

for all $\mathcal{C} \in \Delta$. For a given probing plan z and a given scenario ξ, the integer programming formulation of the joint UAV and truck routing problem is:

$$\max \sum_{i\in\mathcal{N}} r_i\big((1-\xi_i)z_i + (1-p_i)(1-z_i)\big)v_i \tag{11a}$$

$$\text{s.t.} \sum_{j\in\mathcal{N}\setminus\{0\}} x_{0j} = \sum_{j\in\mathcal{N}\setminus\{0\}} x_{j0} = 1 \tag{11b}$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij} = \sum_{(j,i)\in\mathcal{A}} x_{ji} = w_i \qquad \forall i \in \mathcal{N}\setminus\{0\} \tag{11c}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij} = \sum_{(j,i)\in\mathcal{A}} y_{ji} = v_i \qquad \forall i \in \mathcal{N}\setminus\{0\} \tag{11d}$$

$$y_{ii} \le w_i \qquad \forall i \in \mathcal{N}\setminus\{0\} \tag{11e}$$

$$2y_{ij} \le v_i + v_j \qquad \forall (i,j) \in \mathcal{A} \tag{11f}$$

$$\sum_{(0,j)\in\mathcal{A}} y_{0j} = \sum_{(j,0)\in\mathcal{A}} y_{j0} = 0 \tag{11g}$$

$$\sum_{(i,j)\in\mathcal{A}} t_{ij}x_{ij} + \sum_{(i,j)\in\mathcal{A}} \alpha_{ij}t_{ij}y_{ij} \le T \tag{11h}$$

$$w_i = 0 \qquad \forall i \in \mathcal{N}_V \tag{11i}$$

$$x(\mathcal{A}(\mathcal{S})) \le |\mathcal{S}| - 1 \qquad \forall \mathcal{S} \subset \mathcal{N} \mid 0 \notin \mathcal{S} \tag{11j}$$

$$y(\mathcal{C}) \le |\mathcal{C}| - 1 + \sum_{i\in\mathcal{N}(\mathcal{C}) \ \mid \ t_{ij}\le U \ \ \forall j\in\mathcal{N}(\mathcal{C})} w_i \qquad \forall \mathcal{C} \in \Delta \tag{11k}$$

$$y(\mathcal{C}) \le |\mathcal{C}| + 1 - \sum_{i\in\mathcal{N}(\mathcal{C})} w_i \qquad \forall \mathcal{C} \in \Delta \tag{11l}$$

$$x \in \{0,1\}^{|\mathcal{A}|}, w \in \{0,1\}^N \tag{11m}$$

$$y \in \{0,1\}^{|\mathcal{A}|}, v \in \{0,1\}^N. \tag{11n}$$

In Formulation (11), the objective maximizes the estimated reward $\hat{R}$. Constraints (11b)-(11d) are the flow preservation constraints for the truck and the UAV. Constraints (11e) implies that there can be a self-loop at $i \in \mathcal{N}$ only if i is visited by the truck. Constraints (11f) ensure that the UAV can traverse arc (i,j) only if it visits both nodes i and j. Constraint (11g) state that the UAV cannot depart nor land in the depot. Constraint (11h) assures that the total duration of the operation is at most T units and constraints (11i) state that UAV-only nodes cannot be visited by the truck. Constraints (11j) are the subtour elimination constraints for the truck, which assure that all nodes but the depot are visited at most once by the truck; Constraints (11k) are the subtour constraints for the UAV, which assure that in each UAV cycle each node is visited at most once, and that the starting point of any UAV-cycle should be at a distance of at most U of all the nodes in the cycle (assuming, for convenience, that time and distance units are equivalent). Finally, Constraints (11l) imply that exactly one node of any UAV-cycle should be visited by the truck. Based on this formulation, we can define set $\mathcal{V}$ as $\mathcal{V} = \{v \in \{0,1\}^N:$ Exists $x \in \{0,1\}^{|\mathcal{A}|}$, $w \in \{0,1\}^N$, $y \in \{0,1\}^{|\mathcal{A}|}$ such that Eqs. (11b)–(11n) hold$\}$.

We propose valid inequalities that state that if the UAV visits node i, then the truck should visit a node j that is within a distance of U units from node i, given by

$$v_i \leq \sum_{j \in \mathcal{N} | t_{ij} \leq U} w_j \quad \forall i \in \mathcal{N}. \tag{12}$$

Observe that there is an exponential number of constraints (11j)- (11l), which we generate via a cut-generation algorithm, within a branch-and-cut framework. Identifying a violated cut is done in $\mathcal{O}(N)$ time by simply verifying that truck cycles include the depot and do not visit any location more than once and that UAV-cycles visit every node at most once and that there is a truck stop in exactly one of the nodes visited by the UAV-cycle.

4.2. Decision diagrams-based formulation

We propose an alternative formulation for the joint UAV and truck routing problem that exploits the network structure provided by DDs to represent feasible UAV-cycles. We create a DD for each node that could potentially be the start of a UAV-cycle, i.e., nodes not in set $\mathcal{N}_V$ having at least one neighbor within U units of distance. We define this set of candidate *hub* locations as

$$\mathcal{K} = \left\{ i \in \mathcal{N} \setminus \mathcal{N}_V : \ \exists j \in \mathcal{N} \text{ such that } t_{ij} \leq U \right\}. \tag{13}$$

Let each DD be a layered-acyclic digraph $\mathcal{D}^k = (\mathcal{N}^k, \mathcal{A}^k)$, where $\mathcal{N}^k$ and $\mathcal{A}^k$ are the set of nodes and arcs for the DD corresponding to hub location $k \in \mathcal{K}$. Paths in $\mathcal{D}^k$ represent feasible UAV-cycles starting and ending at location k. We construct the DDs based on a dynamic programming model, ensuring that each $\mathcal{D}^k$ encodes the set of all feasible UAV-cycles for location k (see Appendix B). Each arc $a \in \mathcal{A}^k$ represents the UAV flying from a tail node $s(a)$ to a head node $h(a)$ in the original network, where functions s and h map arcs in the DDs to nodes in the original network following

$$s : \bigcup_{k \in \mathcal{K}} \mathcal{A}^k \to \mathcal{N} \text{ and } h : \bigcup_{k \in \mathcal{K}} \mathcal{A}^k \to \mathcal{N}. \tag{14}$$

To avoid confusion in the notation, we use $(i,j)\in\mathcal{A}$ to refer to arcs in the original network and $a\in\mathcal{A}^k$ to refer to arcs in the DDs. Similarly, we use indexes i, j, and k in $\mathcal{N}$ to refer to nodes/locations in the original network and index u to refer to nodes in the DDs. Let $\gamma^+(u)/\gamma^-(u)$ be the arcs directed out of/into node u, respectively. Let 0^k denote the root node for $\mathcal{D}^k$, and define the estimated reward for a location visited, given probing plan z and a given scenario ξ as

$$\hat{r}(z,\xi,i)=r_i\big((1-\xi_i)z_i+(1-p_i)(1-z_i)\big). \tag{15}$$

Let x_{ij} be a binary variable that takes a value of 1 if arc $(i,j)\in\mathcal{A}$ is used by the truck in the optimal route. Let y_{ak} be binary variable that takes a value of 1 if (DD) arc in $a\in\mathcal{A}^k$, $k\in\mathcal{K}$ is selected, which corresponds to the UAV traveling from $s(a)$ to $h(a)$ in the original network. Let w_i be a binary variable that takes a value of 1 if node $i\in N$ is visited by the truck in the optimal route. For a given probing plan z and a given scenario ξ, our proposed alternative formulation is:

$$\max\sum_{i\in\mathcal{N}}\hat{r}(z,\xi,i)w_i+\sum_{k\in\mathcal{K}}\sum_{a\in A^k\ \mid\ h(a)\neq k}\hat{r}(z,\xi,h(a))y_{ak} \tag{16a}$$

$$\text{s.t.}\quad \sum_{j\in\mathcal{N}\setminus\{0\}}x_{0j}=\sum_{j\in\mathcal{N}\setminus\{0\}}x_{j0}=1 \tag{16b}$$

$$\sum_{(i,j)\in\mathcal{A}}x_{ij}=\sum_{(j,i)\in\mathcal{A}}x_{ji}=w_i \qquad \forall i\in\mathcal{N}\setminus\{0\} \tag{16c}$$

$$\sum_{\gamma^+(0^k)}y_{ak}\le w_k \qquad \forall k\in\mathcal{K} \tag{16d}$$

$$\sum_{a\in\gamma^+(u)}y_{ak}=\sum_{a\in\gamma^i(u)}y_{ak} \qquad \forall k\in\mathcal{K},\ u\in\mathcal{N}^k\setminus\{0^k\} \tag{16e}$$

$$\sum_{k\in\mathcal{K}}\sum_{a\in A^k\ \mid\ i\neq k,\ h(a)=i}y_{ak}\le 1-w_i \qquad \forall i\in\mathcal{N} \tag{16f}$$

$$\sum_{(i,j)\in\mathcal{A}}t_{ij}x_{ij}+\sum_{k\in\mathcal{K}}\sum_{a\in A^k}\alpha_{ij}t_{s(a)h(a)}y_{ak}\le T \tag{16g}$$

$$w_i=0 \qquad \forall i\in\mathcal{N}_V \tag{16h}$$

$$x(\mathcal{A}(\mathcal{S}))\le|\mathcal{S}|-1 \qquad \forall\mathcal{S}\subset\mathcal{N}\ \mid\ 0\notin\mathcal{S} \tag{16i}$$

$$x\in\{0,1\}^{|\mathcal{A}|},\ w\in\{0,1\}^N \tag{16j}$$

$$y_k\in\{0,1\}^{|\mathcal{A}^k|}\quad \forall k\in\mathcal{K}. \tag{16k}$$

The objective function maximizes the estimated reward. Constraints (16d) ensure that a UAV-cycle can only start from a location visited by the truck by enforcing that the total flow out of the root node of $\mathcal{D}^k$ is bounded by w_k. Constraints (16e) enforce flow conservation for the DDs, which implicitly enforce feasibility conditions for the UAV-cycles. Constraints (16f) ensure that every node is visited at most once either by the truck or the UAV. Constraints (16g) ensure that the total duration of the operation is less than or equal to T. As before, there is an exponential number of truck subtour elimination constraints (16i), which we generate via a cut-generation algorithm and we include lifted valid inequalities (12) (replacing variables v for the corresponding flow variables in the DDs) to strengthen the formulation.

4.3. Limiting cases

We now turn our attention to two limiting cases of problem (10) that can be readily solved as single-stage problems using the proposed formulations. For the first limiting case, suppose that there is no probing budget, that is, $B = 0$, which implies that $z_i = 0$ for all $i \in \mathcal{N}$. In this case, the estimated reward function $\hat{R}$ simplifies to

$$\hat{R}(z,\xi,v) = \sum_{i\in\mathcal{N}} r_i(1-p_i)v_i \tag{17}$$

and evaluates to the same value under all scenarios for a given visit plan. As a result, optimal routing decisions for all scenarios coincide and all constraints in (10c) can be replaced by

$$v \in \arg\max\Big\{\sum_{i\in\mathcal{N}} r_i(1-p_i)v'_i : v' \in \mathcal{V}\Big\}. \tag{18}$$

On the other hand, since we can select a single visit plan v for all the scenarios, the objective function of (10) simplifies to

$$\sum_{\xi\in\Xi}\pi^\xi \sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i = \sum_{i\in\mathcal{N}} r_i v_i \sum_{\xi\in\Xi}\pi^\xi(1-\xi_i) = \sum_{i\in\mathcal{N}} r_i v_i \sum_{\xi\in\Xi:\, \xi_i=0}\pi^\xi = \sum_{i\in N} r_i(1-p_i)v_i, \tag{19}$$

that is, the leader's objective (10a) of maximizing the expected actual reward coincides with the follower's objective (18) of maximizing the estimated reward. As a result, if there is no probing ($B = 0$), then the two-stage problem (10) simplifies to

$$\Gamma_0 = \max\Big\{\sum_{i\in\mathcal{N}} r_i(1-p_i)v_i : v \in \mathcal{V}\Big\}, \tag{20}$$

which is a single-stage joint UAV and truck routing problem of the form (11) or (16) where an optimal route is selected entirely based on the estimated probabilities p_i, $i \in \mathcal{N}$. An alternative interpretation of (20) is that in the absence of probing, both the actual reward and estimated reward functions simplify to a naive expected reward function based on the probabilities p.

For the second limiting case consider that all nodes can be probed, i.e., $B = N$. We have that $z_i = 1$ for all $i \in N$ and the estimated reward becomes

$$\hat{R}(z, \xi, v) = \sum_{i \in \mathcal{N}} r_i (1 - \xi) v_i. \tag{21}$$

In other words, the estimated reward for each scenario is equal to the actual reward since there is full information and it is readily seen that problem (10) becomes the single-stage problem

$$\Gamma_N = \max\Big\{ \sum_{\xi \in \Xi} \pi^{\xi} \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) v_i^{\xi} \colon v^{\xi} \in \mathcal{V} \quad \forall \xi \in \Xi \Big\}. \tag{22}$$

Moreover, as in (22) there are no coupling requirements between the visit plans v^{ξ}, we have that

$$\Gamma_N = \sum_{\xi \in \Xi} \pi^{\xi} \max\Big\{ \sum_{i \in \mathcal{N}} r_i (1 - \xi_i) v_i^{\xi} \colon v^{\xi} \in \mathcal{V} \Big\}. \tag{23}$$

As a result, Γ_N can be computed by decomposing problem (22) and solving $|\Xi|$ single-stage routing problems of the form (11) or (16) independently (and potentially in parallel).

5. Bounds on the value of information and on the price of not having full information

In this section we derive bounds on $\Gamma_B - \Gamma_0$ and $\Gamma_N - \Gamma_B$. We first provide results for the effect of additional probing in the optimal estimated reward for a given scenario and then construct bounds for the value of probing in terms of the expected actual reward.

For any given $\mathcal{P} \subseteq \mathcal{N}$ and any scenario $\xi \in \Xi$, let $\phi_{\mathcal{P}}^{\xi}$ be the optimal estimated reward for scenario ξ when the locations in $\mathcal{P}$ are probed. That is,

$$\phi_{\mathcal{P}}^{\xi} = \max\Big\{ \sum_{i \in \mathcal{P}} r_i (1 - \xi_i) v_i + \sum_{i \in \mathcal{N} \setminus \mathcal{P}} r_i (1 - p_i) v_i \colon v \in \mathcal{V} \Big\}. \tag{24}$$

Let $v^{\xi, \mathcal{P}}$ be an optimal visit plan associated with $\phi_{\mathcal{P}}^{\xi}$. Lemma 1 presents upper and lower bounds on the difference in optimal estimated reward for two nested probing plans, that is, two plans such that one probes a subset of the locations probed by the other plan. The proof is in Appendix D.

Lemma 1. *Let $\xi \in \Xi$ and $\mathcal{Q} \subseteq \mathcal{P} \subseteq \mathcal{N}$ be given. Then*

$$\sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{Q}} \le \phi_{\mathcal{P}}^{\xi} - \phi_{\mathcal{Q}}^{\xi} \le \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}}. \tag{25}$$

Lemma 1 has interesting implications. Let $\xi \in \Xi$ be given and assume that the locations in $\mathcal{Q}$ have been probed. If in scenario ξ all locations in $\mathcal{P}\setminus Q$ will suffer disruptions, then probing $\mathcal{P}\setminus Q$ cannot provide any additional estimated reward. Conversely, if in ξ there are no disruptions in $\mathcal{P}\setminus\mathcal{Q}$, then the additional estimated reward for probing $\mathcal{P}\setminus Q$ cannot be negative.

Now we turn our attention to the effects of probing on the expected actual reward. For a given $\mathcal{P}\subset\mathcal{N}$, let $\gamma(\mathcal{P})$ be the expected actual reward corresponding to nodes in $\mathcal{P}$ being probed, i.e., $\gamma(\mathcal{P})$ is the objective function in (10) for the probing plan $z_i=1$ for any $i\in\mathcal{P}$ and $z_i=0$ $i\notin\mathcal{P}$. From the definition of $v^{\xi,\mathcal{P}}$ and constraint (10c) it follows that,

$$\gamma(\mathcal{P}) = \sum_{\xi\in\Xi}\pi^{\xi}\sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}}. \tag{26}$$

Note that the relationship between $\gamma(\mathcal{P})$ and Γ_B is given by

$$\Gamma_B = \max\{\gamma(\mathcal{P})\colon |\mathcal{P}|=B, \mathcal{P}\subseteq\mathcal{N}\}. \tag{27}$$

Theorem 1 presents upper and lower bounds for the change in optimal expected actual reward corresponding to two nested probing plans.

Theorem 1. *Let $\mathcal{Q}\subseteq\mathcal{P}\subseteq\mathcal{N}$ be given. Then,*

$$\sum_{\xi\in\Xi}\pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(p_i-\xi_i)(v_i^{\xi,\mathcal{P}}-v_i^{\xi,\mathcal{Q}})\Big] \le \gamma(\mathcal{P})-\gamma(\mathcal{Q}) \le \sum_{\xi\in\Xi}\pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)(v_i^{\xi,\mathcal{P}}-v_i^{\xi,\mathcal{Q}})\Big]. \tag{28}$$

Theorem 1 can be used to provide non-trivial bounds on $\Gamma_{B'}-\Gamma_B$ for any $B'>B\ge 0$. These bounds, however, require to know in advance optimal UAV and truck routes. Corollary 1, shown next, removes some of these limitation and provides upper-bounds that only depend on the surveillance rewards and on the locations visited by the 'inner-most' optimal plan.

Corollary 1. *Let $\mathcal{Q}\subseteq\mathcal{P}\subseteq\mathcal{N}$ be given. Then,*

$$\gamma(\mathcal{P})-\gamma(\mathcal{Q}) \le \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_ip_i(1-p_i) \le \frac{1}{4}\sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i. \tag{29}$$

Interestingly, the bound in Corollary 1 does not depend on $\mathcal{P}$ (as long as $\mathcal{Q} \subseteq \mathcal{P}$). Let $\mathcal{B} \subseteq \mathcal{N}$ denote the locations probed in an optimal probing plan associated with Γ_B. Corollary 1 implies that the value of information, $\Gamma_B - \Gamma_0$, is upper-bounded by

$$\Gamma_B - \Gamma_0 \leq \sum_{i \in \mathcal{N}} r_i p_i (1 - p_i) \leq \frac{1}{4} \sum_{i \in \mathcal{N}} r_i, \tag{30}$$

for any $1 \leq B \leq N$. Similarly, for the price of not having full information, $\Gamma_N - \Gamma_B$, we have

$$\Gamma_N - \Gamma_B \leq \sum_{i \in \mathcal{N} \setminus \mathcal{B}} r_i p_i (1 - p_i) \leq \frac{1}{4} \sum_{i \in \mathcal{N} \setminus \mathcal{B}} r_i. \tag{31}$$

Observe that there is a remarkable asymmetry in these bounds. On the one hand, the bound on the value of information does not depend on the number of locations that are probed, whereas the bound on the price of not having full information does. The bound on the value of information might be explained from the observation that, in general, one might gather all the relevant uncertain information of the system by just probing one location (see for example, Remark 1). The bound for the price of not having full information might be interpreted as implying that there exist instances where optimally placed probes always reveal at least some information about the remaining uncertainty. Next, we provide an example where these bounds are tight.

Remark 1. *Consider a network with two nodes and a depot. From the depot there is an arc to each node; from each node there is an arc back to the depot. The time to traverse each arc is $T/2$; thus, only one node can be visited. The reward of successfully serving node 1 is $r_1 > 0$ and the reward for node 2 is $r_2 = r_1 + \epsilon$, where $0 < \epsilon < r_1$ is arbitrary. Assume that there are two scenarios: in ξ^1 a disruption occurs at node 1 but not in node 2, whereas in ξ^2 a disruption occurs at node 2 but not at node 1. Let $\pi^{\xi^1} = \pi^{\xi^2} = 1/2$, which implies from the definition of the scenarios, that $p_1 = p_2 = 1/2$.*

For this instance we have that

$$\gamma(\emptyset) = \frac{1}{2} r_1 + \frac{\epsilon}{2}, \;\; \gamma(\{1\}) = \gamma(\{2\}) = \gamma(\{1,2\}) = r_1 + \frac{\epsilon}{2}. \tag{32}$$

In particular, the optimal visit plan for $\mathcal{P} = \emptyset$ is to visit node 2, whereas the optimal visit plans for all the other cases is to visit node 2 under scenario ξ^1 and to visit node 1 under scenario ξ^2. From (32) we obtain that

$$\Gamma_0 = \frac{1}{2} r_1 + \frac{\epsilon}{2}, \;\; \Gamma_1 = \Gamma_2 = r_1 + \frac{\epsilon}{2} \tag{33}$$

and thus $\Gamma_B - \Gamma_0 = \frac{1}{2}r_1$ *for* $B = 1, 2$. *On the other hand,* $\frac{1}{4}\sum_{i\in\mathcal{N}} r_i = \frac{1}{4}(2r_1 + \epsilon) = \frac{1}{2}r_1 + \frac{\epsilon}{4}$. *Therefore, the bound in Equation* (30) *can be as tight as desired by choosing a sufficiently small* ϵ*; note also that this holds true for* $B = 1$ *and* $B = 2$. *Moreover, there are instances where the bounds in Equation* (31) *and Corollary 1 can be made as tight as desired, e.g., setting* $B = 0$ *in the example above.*

We close this section by noting (i) that tighter bounds than those in Corollary 1 may be available if one uses the information given by the visit plans and the scenario probabilities as shown in Theorem 1; and (ii) that the results of this section are independent of the problem. That is, the bounds in Lemma 1, Theorem 1, and Corollary 1 do not require that the probed problem is a UAV and truck joint routing problem, and are valid as long as $\mathcal{V}$ is a set of binary vectors.

6. Solving the bilevel problem reformulation

We consider exact approaches for solving the two-stage probing and routing problem, which we formulate as a bilevel problem with multiple followers in formulation (10). One of the major challenges in solving discrete bilevel problems is the construction of valid relaxations. A common relaxation from the bilevel literature is known as the *High-point relaxation*, which is obtained by dropping the requirement of optimality in the lower-level problem enforced by constraints (10c). After removing these constraints in problem (10), the probing variables become irrelevant because they do not appear in the objective function, and the high-point relaxation reduces to

$$\Gamma^H = \sum_{\xi\in\Xi} \pi^\xi \max\Big\{\sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i \colon v\in\mathcal{V}\Big\} = \Gamma_N, \tag{34}$$

i.e., the high-point relaxation is precisely the full information problem Γ_N (see Equation (23)), which generally yields weak upper bounds on the optimal value.

We study two additional relaxations and propose exact approaches based on each one. The first relaxation comes from a value function reformulation while the second relaxation comes from a scenario-based decomposition.

6.1. Value function approach

Value function approaches have been successfully used in the literature for bilevel problems with one follower (Mitsos 2010, Lozano and Smith 2017). For our multi-follower problem observe that formulation (10) can be equivalently posed as

$$\Gamma_B = \max \sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i^\xi \tag{35a}$$

$$\text{s.t.}\ \sum_{i\in\mathcal{N}} z_i \le B \tag{35b}$$

$$\sum_{i\in\mathcal{N}} r_i((1-\xi_i)z_i + (1-p_i)(1-z_i))v_i^{\xi} \ge$$

$$\sum_{i\in\mathcal{N}} r_i((1-\xi_i)z_i + (1-p_i)(1-z_i))v_i \quad \forall v\in\mathcal{V}, \forall \xi \in \Xi \tag{35c}$$

$$z \in \{0,1\}^N;\ v^{\xi} \in \mathcal{V} \qquad \forall \xi \in \Xi, \tag{35d}$$

which is a single-level non-linear binary optimization problem with potentially exponentially many constraints. Observe that the nonlinearities are products between binary variables ($z_i v_i^{\xi}$) and thus readily linearized.

We define a relaxed value function problem (RVF) by considering a subset of visit plans $\hat{\mathcal{V}} \subseteq \mathcal{V}$. Formally, RFV($\hat{\mathcal{V}}$) is defined as (35), except that $\mathcal{V}$ is replaced by $\hat{\mathcal{V}}$ in (35c). Let $\Gamma_B(\hat{\mathcal{V}})$ be the optimal objective function value of RFV($\hat{\mathcal{V}}$) and note that for any $\hat{\mathcal{V}} \subseteq \mathcal{V}$, it holds that $\Gamma_B(\hat{\mathcal{V}}) \ge \Gamma_B$.

We propose a cutting-plane algorithm that iteratively explores visit plans and adds them to $\hat{\mathcal{V}}$. Solving RFV($\hat{\mathcal{V}}$) for each $\hat{\mathcal{V}}$ provides a sequence of non-increasing upper bounds on Γ_B. Lower bounds are obtained by solving the lower-level problem for fixed probing plans stemming from the relaxed problems. Algorithm 1 formalizes our proposed cutting-plane approach. Line 1 initializes the lower and upper bounds, sets $\hat{\mathcal{V}} = \emptyset$, and creates a trivial probing plan $\bar{z} = 0$. Line 2 computes an upper bound by solving RVF for the visit plans obtained thus far in set $\hat{\mathcal{V}}$. Line 3 solves the joint UAV and truck routing problem to maximize the estimated reward for the probing plan $\hat{z}$ found in Line 2. Line 4 computes the expected actual reward and updates the lower bound if necessary. Line 5 stops the execution of the algorithm if the lower bound is equal to the upper bound. Otherwise, it updates the set of visit plans by adding all the plans discovered in Line 3 and goes back to Line 2 to continue with the cut-generation algorithm.

Algorithm 1 terminates finitely with an optimal solution because the set of all possible visit plans $\mathcal{V}$ is finite. Note that all the problems solved are feasible (setting all the variables to zero gives a trivial feasible solution) and bounded (all the variables are binary) and as a result there is no need to check for unboundedness or feasibility. On the other hand, because of the optimistic assumption that the follower breaks ties among alternative optimal solutions by selecting the one that maximizes the leader's objective,

Algorithm 1: Cutting-plane Algorithm

1: Set $LB = -\infty$, $UB = \infty$, $\hat{\mathcal{V}} = \emptyset$, and incumbent solution $\bar{z} = 0$.

2: Solve RFV($\hat{\mathcal{V}}$). Set $UB = \Gamma_B(\hat{\mathcal{V}})$ and record the optimal probing plan found $\hat{z}$.

3: For each scenario $\xi \in \Xi$, solve lower-level problem $\max\left\{\hat{R}(\hat{z}, \xi, v) : v \in \mathcal{V}\right\}$ and record the optimal visit plans found $\hat{v}^\xi$.

4: If $\sum_{\xi \in \Xi} \pi^\xi R(\xi, \hat{v}^\xi) > LB$, then update $LB = \sum_{\xi \in \Xi} \pi^\xi R(\xi, \hat{v}^\xi)$ and $\bar{z} = \hat{z}$.

5: If $LB = UB$, terminate with an optimal solution given by $\bar{z}$. Otherwise, update $\hat{\mathcal{V}} = \hat{\mathcal{V}} \cup \left\{\bigcup_{\xi \in \Xi} \hat{v}^\xi\right\}$ and return to Line 2.

we need to be careful when recording an optimal visit plan $\hat{v}^\xi$ at Line 3 to account for the case in which there exist alternative optimal visit plans (see Appendix C).

The major computational challenge for Algorithm 1 is solving RFV($\hat{\mathcal{V}}$), because enforcing the feasibility of the visit plans $v^\xi \in \mathcal{V}$ requires making copies of all the routing constraints (11b)-(11n) for each scenario. As the number of scenarios grows, RFV($\hat{\mathcal{V}}$) becomes prohibitively large and considerably challenging to solve. To alleviate this problem we explore valid inequalities and a scenario-based decomposition.

6.2. Valid inequalities

The value function reformulation (35) can be enhanced by including constraints that give necessary conditions for the optimality of v^ξ on each scenario $\xi \in \Xi$. Lemma 1 provides such conditions; namely, we can use Γ_0 and Γ_N and the corresponding optimal solutions, to provide bounds for the best possible estimated expected reward for each scenario.

In order to introduce the inequalities, recall the definition of $\phi^\xi_{\mathcal{P}}$ and $v^{\xi,\mathcal{P}}$ in Equation (24) and observe that if $\mathcal{P} = \emptyset$ (if no nodes are probed), then $\phi^\xi_\emptyset$ and $v^{\xi,\emptyset}$ do not depend on ξ. In fact, it is readily seen that $\phi^\xi_\emptyset = \Gamma_0$ for any $\xi \in \Xi$. Thus, for notation simplicity we set $\phi_0 = \phi^\xi_\emptyset$ and $v^0 = v^{\emptyset,\xi}$. The proof is in Appendix D.

Proposition 1. *Let z and v^ξ, $\xi \in \Xi$, be optimal in* (35). *Then they satisfy that*

$$\sum_{i \in \mathcal{N}} r_i((1-\xi_i)z_i + (1-p_i)(1-z_i))v_i^\xi \geq \phi^\xi_{\mathcal{N}} + \sum_{i \in \mathcal{N}} (1-z_i) r_i (\xi_i - p_i) v_i^{\xi,\mathcal{N}} \qquad \forall \xi \in \Xi, \tag{36}$$

and

$$\sum_{i \in \mathcal{N}} r_i((1-\xi_i)z_i + (1-p_i)(1-z_i))v_i^\xi \geq \phi_0 + \sum_{i \in \mathcal{N}} z_i r_i (p_i - \xi_i) v_i^0 \qquad \forall \xi \in \Xi. \tag{37}$$

Observe that the inequalities in Proposition 1 require knowing optimal solutions to Γ_0 and Γ_N. Whereas these are challenging problems, because solving them requires solving $N+1$ single-stage joint routing problems (see Eq. (20) and Eq. (23)), they are not bilevel problems. Moreover, computing Γ_0 and Γ_B is required to compute both the value of information and the price of not having full information, which implies that the computational effort to construct the inequalities is justified.

6.3. Scenario-based decomposition

We propose an alternative relaxation for the bilevel problem based on the following reformulation of the value function formulation (35), which introduces copies of the probing variables for each scenario (similar techniques have been used for different domains, e.g., Carøe and Schultz (1999), Wang et al. (2023)):

$$\Gamma_B = \max \sum_{\xi\in\Xi} \pi^\xi \sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i^\xi \tag{38a}$$

$$\text{s.t. } z^\xi = z^{\xi'} \qquad \forall \xi,\xi' \in \Xi \tag{38b}$$

$$\sum_{i\in\mathcal{N}} z_i^\xi \le B \qquad \forall \xi \in \Xi \tag{38c}$$

$$\sum_{i\in\mathcal{N}} r_i((1-\xi_i)z_i^\xi + (1-p_i)(1-z_i^\xi))v_i^\xi \ge \tag{38d}$$

$$\sum_{i\in\mathcal{N}} r_i((1-\xi_i)z_i^\xi + (1-p_i)(1-z_i^\xi))v_i \qquad \forall v\in\mathcal{V}, \forall \xi \in \Xi \tag{38e}$$

$$z^\xi \in \{0,1\}^N,\ v^\xi \in \mathcal{V} \qquad \forall \xi \in \Xi. \tag{38f}$$

Formulation (38) is equivalent to formulation (35) because constraints (38b) ensure that all the copies of the probing variables have the same value. We obtain a relaxation by relaxing constraints (38b). Moreover, after removing constraints (38b), there are no coupling constraints left in the relaxed formulation, which can be decomposed and solved for each scenario independently. We leverage this decomposition approach within a specialized branch-and-bound procedure to solve the bilevel problem to optimality.

At the root node of our branch-and-bound tree we solve the relaxation described above. For each scenario we solve the corresponding problem via Algorithm 1 (i.e., when solving for a given scenario ξ', we set $\Xi = \{\xi'\}$ in Algorithm 1). These problems are considerable easier to solve than the original problem since they only consider one scenario at a time and as a result $\mathrm{RFV}(\hat{\mathcal{V}})$ does not grow too large. On the downside, there could be "disagreements" between the probing variables for different scenarios, as these problems are solved independently, thus requiring a branching strategy on the probing variables.

Our proposed branching scheme works as follows. Consider a node of the branch-and-bound tree and let $\hat{z}$ be the current values of the probing variables. We select the branching variable by computing the average values of $\hat{z}$ as:

$$\hat{z}_i^{\mathsf{avg}} = \frac{\sum_{\xi \in \Xi} \hat{z}_i^{\xi}}{|\Xi|}, \quad \forall i \in \mathcal{N}$$

and then selecting the branching variable index $\hat{\imath}$ given by the most fractional $\hat{z}^{\mathsf{avg}}$

$$\hat{\imath} = \arg\max_{i \in \mathcal{N}} \{\min\{\hat{z}_i^{\mathsf{avg}}, 1 - \hat{z}_i^{\mathsf{avg}}\}\}.$$

Once $\hat{\imath}$ is identified, we create two branches. In the left branch we add a new child node with the updated domain given by $z_{\hat{\imath}}^{\xi} = 0, \ \forall \xi \in \Xi$, while in the right branch we add a new child node with the updated domain given by $z_{\hat{\imath}}^{\xi} = 1, \ \forall \xi \in \Xi$.

Finally, we implement a primal heuristic that is called once after solving the root node and works as follows. Let $\hat{z}$ and $\hat{z}^{\mathsf{avg}}$ be defined as before. Our primal heuristic sorts the locations according to $\hat{z}^{\mathsf{avg}}$, from largest to smallest. Then it selects the first B locations (according to the sorting) to be probed and solves the corresponding routing problems to compute the estimated reward, the actual reward, and update the lower bound. We remark that the branch-and-bound procedure terminates finitely because all the probing variables are binary.

7. Computational results

We conduct a computational study to compare the performance of the proposed algorithms and to measure the value of information over a set of synthetic instances. We code our algorithms in Java using Eclipse SDK version 4.7.1 and all optimization problems are solved using CPLEX 20.1 with a time limit of one hour (3600s). All experiments are conducted on an Intel(R) Xeon(R) CPU E5-1650 v4 at 3.60GHz with 32GB of memory. The source code and problem instances will be publicly available at GitHub.

We generate synthetic problem instances over a two-dimensional space with dimension 100×100. The depot is located at the center of the space at position $(50, 50)$ while the x-and y-coordinates for the rest of the nodes are drawn independently from a discrete uniform distribution $U(0, 100)$. We consider networks of sizes ranging from 20 to 40 nodes, where 50% of the nodes (not including the depot) are randomly designated as UAV-only locations. The actual rewards of serving the nodes is set to 0 for the depot and drawn independently from a discrete uniform distribution $U(50, 100)$ for the rest of the nodes. We create 5 random networks for each size.

Instances are characterized by the maximum time of the operation T and the communication radius U. We set T as a fraction of the time it takes to visit all the locations with the truck $\hat{T}$, which is obtained by solving a TSP considering only the truck for each network. We then set $T = \lambda\hat{T}$ and consider values for λ in $\{0.2, 0.3, 0.4\}$. For U we consider values in $\{20, 25, 30\}$. We fix the UAV travel time multipliers α_{ij} to 0.1 for all arcs and its maximum airborne time T_U to 100, which ensures that maximum airborne time constraint is not binding for this testbed.

We let $\Xi = \{0, 1\}^N$ and, since this would lead to a significantly large number of scenarios, we use sample average approximation (SAA) to estimate the second-stage expected value. We note that SAA is a common approach to estimate expectations in two-stage settings. SAA has an exponentially fast convergence rate in terms of the number of scenarios used (Kleywegt et al. 2002) and has been shown to be highly accurate in routing problems (Verweij et al. 2003).

We denote by $\hat{\Xi} \subseteq \{0, 1\}^N$, the set of scenarios of the SAA, and consider problem configurations with $|\hat{\Xi}| \in \{25, 50, 100\}$. To generate the scenarios in $\hat{\Xi}$, we define for each location a parameter $q_i = {}^{r_i}\!/_{110}$, and then, for each scenario $\xi \in \hat{\Xi}$ and each location $i \in \mathcal{N}$, we generate a uniform random number between 0 and 1 independently at random. If such number is greater than q_i then set $\xi_i = 1$; else $\xi_i = 0$. Observe that this approach ensures that locations with higher reward have a higher probability of suffering a disruption. We set $\pi^{\xi} = {}^{1}\!/_{|\hat{\Xi}|}$ for each $\xi \in \hat{\Xi}$, and compute probabilities p_i using Equation (5). The combination of random networks and instance characterizations results in a total of 405 problem instances.

7.1. Limiting Cases

We first focus on solving the single-level problems of Section 4. We compare the performance of the baseline cutting plane algorithm given in (11) and denoted as `IP` with the decision-diagram based reformulation given in (16) and denoted as `DD`. The problems are solved over the single-level instances obtained by setting the probing budget to zero.

Table 1 presents the solution time in seconds (Time), the number of instances solved to optimality (# Sol), the average optimality gap (Gap), and the number of cuts added (# Cuts), for both approaches and each combination of N, λ, and U. Each row in Table 1 summarizes the results for 15 different problem instances. Execution times include any preprocessing time (e.g., computing bounds, building the decision diagrams) and we record a time of 3600 seconds if an instance is not solved to optimality within the time limit.

Table 1 **Comparing the decision diagram based approach with the baseline cutting-plane approach**

$N-1$	λ	U	DD				IP			
			Time (s)	# Sol	Gap	# Cuts	Time (s)	# Sol	Gap	# Cuts
20	0.2	20	1	15	0%	1	1	15	0%	3070
		25	1	15	0%	0	152	15	0%	80071
		30	1	15	0%	0	1344	10	6%	232,751
	0.3	20	1	15	0%	6	42	15	0%	25539
		25	1	15	0%	4	2160	6	14%	395,056
		30	2	15	0%	2	1684	8	18%	341,695
	0.4	20	1	15	0%	11	23	15	0%	16261
		25	1	15	0%	15	1681	8	9%	336,633
		30	2	15	0%	9	1921	7	7%	386,812
30	0.2	20	3	15	0%	7	721	12	3%	94465
		25	6	15	0%	9	1536	9	48%	238,018
		30	21	15	0%	6	3126	2	41%	508,154
	0.3	20	3	15	0%	63	796	12	2%	110,242
		25	8	15	0%	67	2024	7	11%	403,052
		30	26	15	0%	44	3131	2	22%	626,786
	0.4	20	3	15	0%	72	1221	10	7%	287,143
		25	8	15	0%	81	1785	9	6%	357,439
		30	51	15	0%	71	3553	1	6%	603,475
40	0.2	20	23	15	0%	21	1089	11	2%	86856
		25	73	15	0%	28	1317	10	4%	94864
		30	1463	11	26%	19	2764	4	40%	247,847
	0.3	20	28	15	0%	121	2356	8	4%	219,225
		25	154	15	0%	109	2784	5	6%	278,734
		30	2445	9	59%	93	2951	3	7%	235,298
	0.4	20	27	15	0%	100	2270	7	4%	225,212
		25	263	15	0%	182	2896	4	7%	286,290
		30	3085	4	34%	77	3601	0	10%	217,595
		Total	**285**	**384**	**4%**	**45**	**1812**	**215**	**10%**	**256,985**

Table 1 shows that DD outperforms IP both in terms of average solution times and the number of instances solved to optimality. Over the complete testbed DD is roughly 6 times faster than IP and is able to solve 169 more instances to optimality within the time limit. The difference in performance is more pronounced for instances with 20 and 30 locations or instances with $U = 20$ for which DD is considerably faster than IP. For instances with 40 locations and $U = 30$, the difference in performance is less pronounced because of the increased size of the decision diagrams.

Regarding the optimality Gap, `DD` achieves consistently smaller gaps compared to `IP`, with the exception of instances with 40 locations, λ in $\{0.3, 0.4\}$, and $U = 30$, where `IP` yields smaller optimality gaps but solves fewer instances to optimality within the time limit. The difference in performance between `DD` and `IP` can be explained by examining the number of cuts each approach generates in the solution process. In the case of `DD` cuts are needed only to eliminate subtours for the trucks while all the constraints for the UAV are embedded in the structure of the decision diagrams. In contrast, `IP` has to generate a large amount of cuts to prevent subtours in the truck route and enforce feasibility conditions for UAV-cycles.

We remark that to the best of our knowledge and as discussed in the literature review, state-of-the-art exact approaches for single-level UAV and truck routing are able to solve similar problems with up to 50 locations in a few hours depending on the assumptions considered, which suggests that our approaches may be comparable to current state-of-the-art-methods. Since the focus of this work is solving the two-stage problem and gaining insights on the value of information under uncertainty, we did not consider adapting existing single-level UAV and truck routing methods for our problem setting to conduct a head-to-head comparison, which would be an interesting avenue for future research. For the same reason, we did not try to include advanced subtour elimination constraints (Applegate et al. 2011) or to accommodate existing strong valid inequalities (for example Fischetti et al. (1997), Jepsen et al. (2008)) for our problem setting.

We also study the difference in expected actual reward between the two limiting cases, i.e., no probing and full information. For this experiment we focus on instances with intermediate values of $\lambda = 0.3$ and $U = 25$ and solve the problem with full information to optimality using `DD` and ignoring the time limit. Table 2 shows the result of this experiment. The first column presents the number of locations. The second column shows the number of scenarios. The third column presents the optimal expected actual reward with no probing (Γ_0). The fourth column shows the optimal expected actual reward with full information (Γ_N). The final two columns show the average and maximum full information gap I_G computed as

$$I_G = \frac{\Gamma_N - \Gamma_0}{\Gamma_0}. \tag{39}$$

Each row in Table 2 summarizes the results for 5 different problem instances.

Table 2 Measuring the value of full information

$N-1$	$\lvert\hat{\Xi}\rvert$	Γ_0	Γ_N	Avg I_G	Max I_G
	25	222.3	256.3	16%	25%
20	50	225.7	261.6	17%	31%
	100	238.2	274.0	16%	33%
	25	326.8	393.1	21%	31%
30	50	339.1	405.8	20%	30%
	100	345.5	415.8	21%	28%
	25	513.1	607.0	18%	21%
40	50	523.7	623.0	19%	23%
	100	522.8	618.8	19%	22%

Table 2 shows the average gain in objective value associated with full information ranges from 16% to 21% with some problem instances exhibiting up to a 33% increase in value under full information. This behaviour seems to be consistent for different number of locations and scenarios.

7.2. Bilevel Approaches

We now turn our attention to the bilevel approaches to solve the problem with probing. Since these problems are considerably harder to solve, we increase the time limit to 4 hours (14400 seconds) for these experiments. We compare the value-function-based cutting plane approach described in Algorithm 1 and denoted as `VF` with the specialized branch-and-bound approach described in Section 6.3, denoted as `BB`. We focus on instances with intermediate values of $\lambda = 0.3$, $U = 25$, and $|\hat{\Xi}| \in \{10, 20, 50\}$, while considering problem instances with $N \in \{15, 20, 30\}$ and a probing budget $B \in \{3, 5, 10\}$. Tables 3 to 5 compare the computational performance of both approaches. Each row summarizes the results for 5 different problem instances. As before, we record a time of 14400 seconds if an instance is not solved to optimality within the time limit.

Table 3 Comparing the branch-and-bound approach with the value function cutting-plane algorithm for $|\hat{\Xi}| = 10$

N	B	`BB`			`VF`		
		Time (s)	# Sol	Gap	Time (s)	# Sol	Gap
	3	131	5	0.0%	890	5	0%
15	5	221	5	0.0%	682	5	0%
	10	78	5	0.0%	233	5	0%
	3	2102	5	0.0%	11525	1	16%
20	5	8326	4	0.1%	10476	2	16%
	10	8489	3	0.7%	11522	1	16%
	3	12959	1	3.5%	14400	0	31%
30	5	14400	0	9.1%	14400	0	31%
	10	14400	0	8.7%	14400	0	32%
	Total	**6790**	**28**	**2.5%**	**8725**	**19**	**16%**

Table 4 **Comparing the branch-and-bound approach with the value function cutting-plane algorithm for** $|\hat{\Xi}| = 20$

N	B	BB			VF		
		Time (s)	# Sol	Gap	Time (s)	# Sol	Gap
	3	547	5	0.0%	5434	4	0%
15	5	1547	5	0.0%	4517	4	2%
	10	114	5	0.0%	3570	4	1%
	3	9849	4	0.5%	11603	1	21%
20	5	12398	1	1.7%	13750	1	21%
	10	12045	1	2.3%	12385	1	21%
	3	14400	0	10.9%	14400	0	32%
30	5	14400	0	11.8%	14400	0	32%
	10	14400	0	9.2%	14400	0	32%
	Total	**8856**	**21**	**4.0%**	**10496**	**15**	**18%**

Table 5 **Comparing the branch-and-bound approach with the value function cutting-plane algorithm for** $|\hat{\Xi}| = 50$

N	B	BB			VF		
		Time (s)	# Sol	Gap	Time (s)	# Sol	Gap
	3	7660	4	0.7%	8999	2	15%
15	5	8503	3	0.3%	8746	2	16%
	10	632	5	0.0%	8644	2	15%
	3	12095	1	1.9%	14400	0	22%
20	5	14400	0	6.9%	14400	0	25%
	10	13317	1	4.3%	14400	0	25%
	3	14400	0	10.9%	14400	0	32%
30	5	14400	0	9.5%	14400	0	32%
	10	14400	0	8.4%	14400	0	32%
	Total	**11090**	**14**	**4.8%**	**12533**	**6**	**24%**

Tables 3 to 5 show that overall VF is able to solve to optimality 40 out of the 135 instances, and yields relatively weak bounds as shown by the large optimality gaps obtained. Average optimality gap for VF is 19% with values up to 32%. In contrast, BB is able to solve 63 out of the 135 instances and consistently achieves low optimality gaps for instances not solved to optimality. Average optimality gap for BB is 3.77% with values up to 11.8%. As expected, the computational bottleneck for VF is solving relaxed problem RFV($\hat{\mathcal{V}}$), which in some cases is not even solved once within the time limit. Instances with more scenarios are considerably harder to solve for both methods as evidenced by the average computation times, number of instances solved, and optimality gaps obtained. Overall, BB is faster than VF, solves more instances to optimality and achieves considerable lower optimality gaps for the instances not solved within the time limit. We highlight that these are challenging bilevel problems with $|\hat{\Xi}|$ followers, each solving a joint UAV and truck routing problem and therefore the performance of BB being able to consistently find near optimal solutions within less than 5% optimality gap on average is remarkable.

7.3. Value and price of information

We are also interested in measuring the value of information, both in terms of the added value of having probing as well as the gap with respect to full information. Table 6 presents two measures for the value of information computed with solutions obtained by `BB`. The first measure is a standardized measure of the value of information and is computed as

$$\text{Probing Value} = \frac{\Gamma_B - \Gamma_0}{\Gamma_0}. \tag{40}$$

The second measure standardizes the price of not having full information and is given by

$$\text{Price Gap} = \frac{\Gamma_N - \Gamma_B}{\Gamma_N}. \tag{41}$$

For instances not solved to optimality, we use the best solution found as a proxy for Γ_B in the performance measures, which means that the computations of our performance measures are approximate: the probing values we obtain are a lower bound on the true value whereas the price gap values we obtain provide an upper bound for the true value. For each measure we present the average, minimum, and maximum over the 5 problem instances represented by each row.

Table 6 shows that probing leads to an improvement in objective function value ranging from 0% up to 41%. On average, solutions with probing are 14% better than solutions without probing. This difference is more pronounced for instances with $N = 15$, for which `BB` is able to find optimal solutions more often, resulting in an average gain of 19%. Even for instances with $N = 30$, for which `BB` finds fewer optimal solutions, the average improvement is 9%, with values as high as 23%. Similarly, average probing value is larger for instances having $|\hat{\Xi}| = 10$, for which `BB` is able to find optimal solutions for more instances. We remark that even small probing budgets of $B = 3$ often lead to impressive improvements in the objective function value of up to 32%.

Regarding the price gap, the solutions obtained are on average only 5% away from the full information bound, with gap values ranging from 0% to 24%. For instances with $N = 15$ and $|\hat{\Xi}| = 50$, a small probing budget of $B = 3$ already produces solutions within 5% of Γ_N on average, having a budget of $B = 5$ reduces this gap to 2%, and a budget of $B = 10$ virtually closes the price gap. A similar behaviour is displayed by several instance classes. In some instances we note that the information gap follows a counter intuitive behaviour, as in theory it should be non-increasing with respect to the budget values (e.g., for $N = 20$

Table 6 **Measuring the value of information**

$\lvert\hat{\Xi}\rvert$	N	B	Probing Value			Price Gap		
			avg	min	max	avg	min	max
		3	17%	0%	32%	2%	0%	6%
	15	5	19%	0%	37%	1%	0%	3%
		10	19%	0%	41%	0%	0%	0%
		3	19%	11%	25%	2%	1%	4%
10	20	5	21%	11%	26%	1%	0%	1%
		10	21%	11%	26%	1%	0%	3%
		3	15%	9%	23%	6%	2%	12%
	30	5	10%	4%	15%	10%	6%	19%
		10	12%	4%	21%	9%	6%	14%
		3	16%	0%	25%	3%	0%	7%
	15	5	19%	0%	32%	1%	0%	2%
		10	20%	0%	34%	0%	0%	0%
		3	12%	9%	17%	4%	1%	6%
20	20	5	14%	9%	20%	3%	1%	5%
		10	14%	6%	22%	2%	0%	4%
		3	5%	0%	10%	13%	5%	24%
	30	5	5%	4%	6%	12%	6%	20%
		10	9%	6%	13%	9%	3%	17%
		3	16%	0%	25%	5%	0%	9%
	15	5	21%	0%	31%	2%	0%	4%
		10	23%	0%	36%	0%	0%	1%
		3	9%	4%	16%	6%	4%	12%
50	20	5	8%	2%	17%	8%	3%	11%
		10	11%	4%	23%	4%	0%	6%
		3	5%	4%	7%	13%	6%	18%
	30	5	8%	3%	16%	10%	5%	17%
		10	10%	4%	18%	8%	4%	13%
	Total		**14%**	**0%**	**41%**	**5%**	**0%**	**24%**

and $\lvert\hat{\Xi}\rvert = 50$ the information gap for $B = 3$ is 6% and it increases to 8% for $B = 5$). This behaviour happens because we are not able to obtain optimal solutions for all the instances and use the best solution available to compute our performance measures. From tables 3 to 5 we observe that instances with $B = 5$ are more difficult to solve than instances with $B = 3$ as fewer instances are solved to optimality and as result, our computation of the information gap for $B = 5$ is less accurate than for $B = 3$.

The two main takeaways from Table 6 for our problem instances are: even small probing budgets can result in considerable improvement of the solution quality with respect to not doing any probing; and even small probing budgets can yield solutions within a small gap of the full information bound. Finally, we conduct additional experiments to measure the effect of the valid inequalities on the computation time (see Appendix E).

8. Conclusions

We study a joint UAV and truck routing problem under uncertainty in which the planner has the ability to probe locations to confirm the occurrence or absence of disruptions, which interfere with the UAV operations. After the probing stage, the planner decides the joint UAV and truck routing plan considering common operational constraints from the literature. The main focus of our work is on measuring the value of the information provided by the probing stage.

We represent the problem as a bilevel problem with multiple followers. We explore formulations for the limiting cases in which there is no probing or there is no uncertainty and then develop approaches to solve the joint routing problem and the bilevel problem. To the best of our knowledge, we are the first ones to contribute exact approaches for this type of probing problems. We complete our contributions with valid inequalities and bounds on the value of information.

Computational experiments show that our proposed bilevel approaches are able to find high-quality solutions for instances with up to 30 locations. Moreover, both our theoretical results and computations suggest that even small probing budgets could yield considerable improvements in solution quality when compared to not doing any probing. Additionally, we propose a decision-diagram-based formulation for the routing problem that is consistently faster than a baseline subtour elimination approach.

One future research stream is to explore methodological and computational approaches to accelerate the solution of the joint UAV and truck routing problem. Another research venue would explore advanced decomposition techniques, valid inequalities, and heuristics for the bilevel problem. One of the main challenges is obtaining strong relaxations that can be solved in a reasonable time. A third research avenue could consider more complex settings involving multiples UAVs or recourse actions to be taken once a UAV arrives at a location and faces a disruption as well as more advanced techniques for computing the estimated and actual reward functions.

Acknowledgments

The authors gratefully acknowledge the support of the *Air Force Office of Scientific Research* under grant FA9550-22-1-0236. The second author gratefully acknowledges the support of the *National Science Foundation* under grant CMMI 2145553.

References

Adamczyk M, Sviridenko M, Ward J (2016) Submodular stochastic probing on matroids. *Mathematics of Operations Research* 41(3):1022–1038.

Agatz N, Bouman P, Schmidt M (2018) Optimization approaches for the traveling salesman problem with drone. *Transportation Science* 52(4):965–981.

Amazon (2022) Amazon prime air prepares for drone deliveries. Available at: `https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries`, visited on 05/17/2023.

Applegate DL, Bixby RE, Chvátal V, Cook WJ (2011) *The Traveling Salesman Problem : a Computational Study*. Princeton Series in Applied Mathematics (Princeton: Princeton University Press), ISBN 9781400841103.

Artstein Z (1994) Probing for information in two-stage stochastic programming and the associated consistency. *Asymptotic Statistics: Proceedings of the Fifth Prague Symposium, held from September 4–9, 1993*, 21–33 (Springer).

Artstein Z (1999) Gains and costs of informationin stochastic programming. *Annals of operations research* 85(0):129–152.

Artstein Z, Wets RJ (1993) Sensors and information in optimization under stochastic uncertainty. *Mathematics of Operations Research* 18(3):523–547.

Austin R (2010) *Unmanned aircraft systems : UAVs design, development, and deployment*. AIAA Education series (Reston, Va: American Institue of Aeronautics and Astronautics), ISBN 9780470664797.

Axe D (2022) Russia's electronic-warfare troops knocked out 90 percent of ukraine's drones. Available at: `https://www.forbes.com/sites/davidaxe/2022/12/24/russia-electronic-warfare-troops-knocked-out-90-percent-of-ukraines-drones/?sh=1bf477db575c`, visited on 03/20/2023.

Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

Bhattacharya S, Basar T (2010) Game-theoretic analysis of an aerial jamming attack on a UAV communication network. *Proceedings of the 2010 American control conference*, 818–823 (IEEE).

Bouman P, Agatz N, Schmidt M (2018) Dynamic programming approaches for the traveling salesman problem with drone. *Networks* 72(4):528–542.

Boysen N, Briskorn D, Fedtke S, Schwerdfeger S (2018) Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* 72(4):506–527.

Carlsson JG, Song S (2018) Coordinated logistics with a truck and a drone. *Management Science* 64(9):4052–4069.

Carøe CC, Schultz R (1999) Dual decomposition in stochastic integer programming. *Operations Research Letters* 24:37–45.

Congressional Research Service (2022) Unmanned aircraft systems: Roles, missions, and future concepts. Available at: `https://sgp.fas.org/crs/weapons/R47188.pdf`, visited on 03/20/2023.

Cornell A, Kloss B, Presser D, Riedel R (2023) Drones take to the sky, potentially disrupting last-mile delivery. Available at: `https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/future-air-mobility-blog/drones-take-to-the-sky-potentially-disrupting-last-mile-delivery`, visited on 05/17/2023.

CRFS (2023) Integrated vehicle solutions -CRFS- spectrum monitoring and geolocation. Available at: `https://www.crfs.com/product/deployment-options/integrated-vehicle-solutions/`, visited on 04/07/2023.

Curpen R, Bălan T, Micloş IA, Comănici I (2018) Assessment of signal jamming efficiency against LTE UAVs. *2018 International Conference on Communications (COMM)*, 367–370 (IEEE).

Dahm JM (2020) *Electronic Warfare and Signals Intelligence* (The Johns Hopkins University Applied Physics Laboratory LLC), available at: `https://www.jhuapl.edu/sites/default/files/2022-12/EWandSIGINT.pdf`.

Dasdemir E, Batta R, Köksalan M, Öztürk DT (2022) UAV routing for reconnaissance mission: A multi-objective orienteering problem with time-dependent prizes and multiple connections. *Computers & Operations Research* 145:105882.

Dasdemir E, Köksalan M, Öztürk DT (2020) A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. *Computers & Operations Research* 114:104811.

Dempe S (2002) *Foundations of Bilevel Programming* (Boston: Kluwer Academic Publishers).

Development Aid (2020) Experts' opinions: How drones are transforming the humanitarian aid sector. Available at: `https://www.developmentaid.org/news-stream/post/58299/experts-opinions-how-drones-are-transforming-the-humanitarian-aid-sector`, visited on 05/17/2023.

DHS (2019) Radio frequency detection,spectrum analysis, and direction finding equipment. Available at: `https://www.dhs.gov/sites/default/files/saver-msr-rf-detection_cod-508_10july2019.pdf`, visited on 04/07/2023.

Du B, Sun D, Manyam SG, Casbeer DW (2020) Cooperative air-ground vehicle routing using chance-constrained optimization. *2020 American Control Conference (ACC)*, 392–397 (IEEE).

Evers L, Barros AI, Monsuur H, Wagelmans A (2014a) Online stochastic UAV mission planning with time windows and time-sensitive targets. *European Journal of Operational Research* 238(1):348–362.

Evers L, Dollevoet T, Barros AI, Monsuur H (2014b) Robust UAV mission planning. *Annals of Operations Research* 222(1):293–315.

Evers L, Glorie K, Van Der Ster S, Barros AI, Monsuur H (2014c) A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research* 43:248–260.

Farlik J, Kratky M, Casar J (2016) Detectability and jamming of small UAVs by commercially available low-cost means. *2016 International Conference on Communications (COMM)*, 327–330 (IEEE).

Fischetti M, Salazar González JJ, Toth P (1997) A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research* 45(3):378–394.

Gendreau M, Laporte G, Semet F (1998a) A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks: An International Journal* 32(4):263–273.

Gendreau M, Laporte G, Semet F (1998b) A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research* 106(2-3):539–545.

Goel A, Guha S, Munagala K (2010) How to probe for an extreme value. *ACM Transactions on Algorithms (TALG)* 7(1):1–20.

Golden BL, Levy L, Vohra R (1987) The orienteering problem. *Naval Research Logistics (NRL)* 34(3):307–318.

Greenwood F, Joseph D (2020) Aid from the air: A review of drone use in the rcrc global network. *Red Cross Red Crescent* 3.

Guha S, Munagala K (2012) Adaptive uncertainty resolution in bayesian combinatorial optimization problems. *ACM Transactions on Algorithms (TALG)* 8(1):1–23.

Gupta A, Nagarajan V (2013) A stochastic probing problem with applications. *Integer Programming and Combinatorial Optimization: 16th International Conference, IPCO 2013, Valparaíso, Chile, March 18-20, 2013. Proceedings 16*, 205–216 (Springer).

Gupta A, Nagarajan V, Singla S (2016) Algorithms and adaptivity gaps for stochastic probing. *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, 1731–1747 (SIAM).

Ha QM, Deville Y, Pham QD, Hà MH (2018) On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies* 86:597–621.

Harbaugh M (2018) *Unmanned Aerieal Systems (UAS) for Intelligence, surveillance, and reconnaissance (ISR)* (DSIAC), available at: `https://dsiac.org/wp-content/uploads/2018/05/UNMANNED-AERIAL-SYSTEMS-UAS-FOR-INTELLIGENCE-SURVEILLANCE-AND-RECONNAISSANCE-ISR.pdf`.

Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56(2):497–511.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on optimization* 12(2):479–502.

Kress M, Royset JO (2008) Aerial search optimization model (ASOM) for UAVs in special operations. *Military Operations Research* 13:23–33.

Li A, Wu Q, Zhang R (2018) UAV-enabled cooperative jamming for improving secrecy of ground wiretap channel. *IEEE Wireless Communications Letters* 8(1):181–184.

Li H, Wang F (2022) Branch-price-and-cut for the truck–drone routing problem with time windows. *Naval Research Logistics (NRL)* .

Liu Y, Liu Z, Shi J, Wu G, Chen C (2019) Optimization of base location and patrol routes for unmanned aerial vehicles in border intelligence, surveillance, and reconnaissance. *Journal of Advanced Transportation* 2019.

Lozano L, Smith JC (2017) A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research* 65(3):768–786.

Manyam SG, Sundar K, Casbeer DW (2019) Cooperative routing for an air-ground vehicle team—exact algorithm, transformation method, and heuristics. *IEEE Transactions on Automation Science and Engineering* 17(1):537–547.

Mathew N, Smith SL, Waslander SL (2015) Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering* 12(4):1298–1308.

Mitsos A (2010) Global solution of nonlinear mixed-integer bilevel programs. *Journal of Global Optimization* 47(4):557–582.

Moskal MD, Batta R (2017) A macrogrid approach for routing UAVs in support of information gathering. *Military Operations Research* 22(4):35–54.

Moskal MD, Dasdemir E, Batta R (2022) Unmanned aerial vehicle information collection missions with uncertain characteristics. *INFORMS Journal on Computing* .

Mototolea D, Stolk C (2018) Software defined radio for analyzing drone communication protocols. *2018 International Conference on Communications (COMM)*, 485–490 (IEEE).

Mufalli F, Batta R, Nagi R (2012) Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Computers & Operations Research* 39(11):2787–2799.

Multerer T, Ganis A, Prechtel U, Miralles E, Meusling A, Mietzner J, Vossiek M, Loghi M, Ziegler V (2017) Low-cost jamming system against small drones using a 3d mimo radar based tracking. *2017 European Radar Conference (EURAD)*, 299–302 (IEEE).

Murray CC, Chu AG (2015) The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* 54:86–109.

Nguyen D, Sahin C, Shishkin B, Kandasamy N, Dandekar KR (2014) A real-time and protocol-aware reactive jamming framework built on software-defined radios. *Proceedings of the 2014 ACM workshop on Software radio implementation forum*, 15–22.

Otto A, Agatz N, Campbell J, Golden B, Pesch E (2018) Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* 72(4):411–458.

Pärlin K, Alam MM, Le Moullec Y (2018) Jamming of UAV remote control systems using software defined radio. *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, 1–6 (IEEE).

Pietz J, Royset JO (2013) Generalized orienteering problem with resource dependent rewards. *Naval Research Logistics (NRL)* 60(4):294–312.

Poikonen S, Golden B, Wasil EA (2019) A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing* 31(2):335–346.

Poikonen S, Wang X, Golden B (2017) The vehicle routing problem with drones: Extended models and connections. *Networks* 70(1):34–43.

Polaris Market Research (2021) Anti-drone market share, size, trends, industry analysis report, by product (ground-based C-UAV, hand-held C-UAV and UAV-based C-UAV); by system (detection systems, detection disruption systems); by end-use; by technology; by region; segment forecast, 2022 - 2029. Available at: `https://www.polarismarketresearch.com/industry-analysis/anti-drone-market`, visited on 03/20/2023.

Rajan S, Sundar K, Gautam N (2022) Routing problem for unmanned aerial vehicle patrolling missions-a progressive hedging algorithm. *Computers & Operations Research* 142:105702.

Reed S, Campbell AM, Thomas BW (2022) The value of autonomous vehicles for last-mile deliveries in urban environments. *Management Science* 68(1):280–299.

Roberti R, Ruthmair M (2021) Exact methods for the traveling salesman problem with drone. *Transportation Science* 55(2):315–335.

Royset JO, Reber DN (2009) Optimized routing of unmanned aerial systems for the interdiction of improvised explosive devices. *Military Operations Research* 14:5–19.

Trevithick J (2018) The russians are jamming us drones in syria because they have every reason to be. Available at: `https://www.thedrive.com/the-war-zone/20034/the-russians-are-jamming-us-drones-in-syria-because-they-have-every-reason-to-be`, visited on 03/20/2023.

Tsiligirides T (1984) Heuristic methods applied to orienteering. *Journal of the Operational Research Society* 35(9):797–809.

U S Army UAS Center of Excellence (2010) “Eyes of the Army” U. S. army roadmap for unmanned aircraft systems 2010-2035. Available at: `https://irp.fas.org/program/collect/uas-army.pdf`, visited on 03/20/2023.

Unicef (2023) Drones: Addressing transport, connectivity and better emergency preparedness. Available at: `https://www.unicef.org/innovation/drones`, visited on 05/17/2023.

USAF (2016) Wasp III fact sheet. Available at: https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104480/wasp-iii/, visited on 07/28/2022.

USAF (2017) RQ-11B raven fact sheet. Available at: https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104533/rq-11b-raven/, visited on 07/28/2022.

Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: A survey. *European Journal of Operational Research* 209(1):1–10.

Vásquez SA, Angulo G, Klapp MA (2021) An exact solution method for the TSP with drone based on decomposition. *Computers & Operations Research* 127:105127.

Verbeeck C, Vansteenwegen P, Aghezzaf EH (2016) Solving the stochastic time-dependent orienteering problem with time windows. *European Journal of Operational Research* 255(3):699–718.

Verbeeck C, Vansteenwegen P, Aghezzaf EH (2017) The time-dependent orienteering problem with time windows: a fast ant colony system. *Annals of Operations Research* 254(1):481–505.

Verweij B, Ahmed S, Kleywegt AJ, Nemhauser G, Shapiro A (2003) The sample average approximation method applied to stochastic routing problems: a computational study. *Computational optimization and applications* 24:289–333.

Wang J, Liu Y, Song H (2021) Counter-unmanned aircraft system (s)(c-uas): State of the art, challenges, and future trends. *IEEE Aerospace and Electronic Systems Magazine* 36(3):4–29.

Wang K, Lozano L, Cardonha C, Bergman D (2023) Optimizing over an ensemble of trained neural networks. *INFORMS Journal on Computing* 0(0):Article in press, URL https://doi.org/10.1287/ijoc.2023.1285.

Wang X, Poikonen S, Golden B (2017) The vehicle routing problem with drones: several worst-case results. *Optimization Letters* 11(4):679–697.

WFP (2018) Using drones to deliver critical humanitarian aid. Available at: https://drones.wfp.org/updates/using-drones-deliver-critical-humanitarian-aid, visited on 05/17/2023.

Yang Y, Yan C, Cao Y, Roberti R (2023) Planning robust drone-truck delivery routes under road traffic uncertainty. *European Journal of Operational Research* 309(3):1145–1160.

Yurek EE, Ozmutlu HC (2018) A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies* 91:249–262.

Appendix A: Considering other estimating procedures

To update the probability vector p after the probing stage, the planner could use a function for the estimated reward given by

$$\tilde{R}(z,\xi,v) = \mathbb{E}[R(\xi,v)|J_z = \xi_z] \tag{42}$$

where for any z we have $J_z = (J_i \colon z_i = 1)$ and $\xi_z = (\xi_i \colon z_i = 1)$. We have that

$$\begin{aligned}\mathbb{E}[R(\xi,v)|J_z = \xi_z] &= E\Big[\sum_{i\in\mathcal{N}} r_i(1-J_i)\Big| J_z = \xi_z\Big] \\ &= \sum_{i\in\mathcal{N}} r_i(1-\xi_i)z_i + \sum_{i\in\mathcal{N}} r_i(1-E[J_i|J_z = \xi_z])(1-z_i)\end{aligned} \tag{43}$$

For any $i \in \mathcal{N}$ such that $z_i = 0$ we have that

$$\begin{aligned}E[J_i|J_z = \xi_z] &= P[J_i = 1|J_z = \xi_z] \\ &= \frac{P[J_i = 1, J_z = \xi_z]}{P[J_z = \xi_z]}.\end{aligned} \tag{44}$$

Note that

$$P[J_z = \xi_z] = \sum_{\hat{\xi}\in\Xi, \hat{\xi}_i = \xi_i \ \forall i\colon\, z_i = 1} \pi^{\hat{\xi}}. \tag{45}$$

Algebraically, if $\mathbb{I}\{A\}$ is the indicator of event A, we can write the denominator in (45) as

$$P[J_z = \xi_z] = \sum_{\hat{\xi}\in\Xi} \pi^{\hat{\xi}} \prod_{\ell\in\mathcal{N}} (\mathbb{I}\{\hat{\xi}_\ell = \xi_\ell\})^{z_\ell}. \tag{46}$$

A similar procedure applies to $P[J_i = 1, J_z = \xi_z]$. Therefore,

$$E[J_i|J_z = \xi_z] = \frac{\sum_{\hat{\xi}\in\Xi} \pi^{\hat{\xi}} \mathbb{I}\{\hat{\xi}_i = 1\} \prod_{\ell\in\mathcal{N}} (\mathbb{I}\{\hat{\xi}_\ell = \xi_\ell\})^{z_\ell}.}{\sum_{\hat{\xi}\in\Xi} \pi^{\hat{\xi}} \prod_{\ell\in\mathcal{N}} (\mathbb{I}\{\hat{\xi}_\ell = \xi_\ell\})^{z_\ell}.}. \tag{47}$$

By plugging (47) into (43) it is clear that estimating using $\tilde{R}$ makes the 'lower-level' problem a optimization problem that has exponential and fractional terms as a function of z. We note, however, that if in the probability space $(\Omega, \Psi, \mathbb{P})$ the random variables J_i, $i \in \mathcal{N}$, are independent, then $E[J_i|J_z = \xi_z] = E[J_i] = p_i$ (for any $i \in \mathcal{N}$ such that $z_i = 0$), and therefore, in this case it is readily seen that $\tilde{R}(z,\xi,v) = \hat{R}(z,\xi,v)$. Also, it is readily seen that $\tilde{R}(\mathbf{0},\xi,v) = \hat{R}(\mathbf{0},\xi,v)$ and that $\tilde{R}(\mathbf{1},\xi,v) = \hat{R}(\mathbf{1},\xi,v)$, where $\mathbf{0}$ and $\mathbf{1}$ are a vector of zeroes and ones, respectively.

Appendix B: Constructing the Decision Diagrams

We build a DD for each hub location $k \in \mathcal{K}$ based on the following dynamic program dynamic program (DP) that sequentially decides which location to visit next. We first compute the locations that are in range of k as:

$$\mathcal{Q}(k) = \big\{i \in \mathcal{N} \colon \ t_{ki} \le U\big\}. \tag{48}$$

Let $\Lambda^q = \{\mathcal{V}^q, i^q, T^q\}$ be the state of the system at stage q, i.e., having fixed routing decisions over $q-1$ steps. State variable $\mathcal{V}^q \subseteq \mathcal{N}$ records the nodes already visited by the UAV-cycle. State variable $i^q \in \mathcal{N}$ denotes the last node visited, and state variable T^q records the time that the UAV has been airborne after arriving to node i^q. Let Θ^k be the state space of the DP for hub k, where $\Lambda^1 = \{\{k\}, k, 0\}$ is the root state,

$\{\}$ is the infeasible state, and we define a set of terminal states, Σ, which includes the infeasible state along with every state for which $i^q = k$, i.e., the UAV returned to the starting hub.

The DP transitions from a given state Λ^q to a new state according to a transition function $\phi\colon (\Theta \setminus \Sigma) \times \mathcal{N} \to \Theta$ such that

$$\Lambda^{q+1} = \phi(\Lambda^q, j), \tag{49}$$

where $j \in \mathcal{Q}(k)$ is the node to be visited immediately after node i^q. We define ϕ as

$$\phi(\Lambda^q, j) = \begin{cases} \{\mathcal{V}^q \cup \{j\}, j, T^q + \alpha_{ij} t_{i^q,j}\} & \text{if } f(\Lambda^q, j) = 1 \\ \{\} & \text{otherwise} \end{cases} \tag{50}$$

where $f\colon (\Theta \setminus \Sigma) \times \mathcal{N} \to \{0,1\}$ is a function that checks for feasibility defined as

$$f(\Lambda^k, j) = \begin{cases} 1 \text{ if } j \notin \mathcal{V}^q,\ T^q + \alpha_{ij} t_{i^q,j} \le T_U \\ 0 \quad \text{otherwise.} \end{cases} \tag{51}$$

Feasibility function f ensures that every node is visited at most once and that the UAV-cycle respects the maximum airborne time limit. Note that the UAV is always within U units of the truck since we only consider nodes in $\mathcal{Q}(k)$ for the state transitions.

We also adapt a standard dominance rule for routing problems. Consider states Λ^q and Λ^p in Θ. If $\mathcal{V}^q = \mathcal{V}^p$ and $T^q < T^p$, then Λ^q dominates Λ^p. Alternatively, if $\mathcal{V}^q = \mathcal{V}^p$ and $T^q = T^p$, then Λ^q and Λ^p are *equivalent* states. We safely remove dominated states and keep only one among multiple equivalent states without changing the optimal objective value obtained by the DP.

Finally, we generate exact DDs from the state transition graph of the DPs described above by removing nodes corresponding to the infeasible state and merging all the nodes corresponding to terminal states in Σ.

Appendix C: Enforcing the optimistic assumption

To make sure that the optimistic assumption is satisfied we need to make a simple solution check in Line 3 of Algorithm 1.

Let v^1 be the visit plans obtained from solving the RVF in Line 2 and let v^2 be the visit plans obtained by solving the lower-level problem in Line 3. For each scenario $\xi \in \Xi$ we check if $\hat{R}(\hat{z}, \xi, v^{\xi,1}) = \hat{R}(\hat{z}, \xi, v^{\xi,2})$, that is, the visit plan obtained by solving RVF is an alternative optimal solution to the lower-level problem. If this is the case, we record $\hat{v}^\xi = v^{\xi,1}$; otherwise, we record $\hat{v}^\xi = v^{\xi,2}$.

Doing this ensures that the optimistic assumption is satisfied by the optimal solution obtained at the termination of the algorithm. To show this consider an optimal solution $\bar{z}$ obtained via Algorithm 1, and its corresponding visit plan $\bar{v}$ and optimal objective value Γ_B. Assume by contradiction that the optimistic assumption is not satisfied, i.e., there exists an alternative visit plan v' such that $\hat{R}(\bar{z}, \xi, \bar{v}^\xi) = \hat{R}(\bar{z}, \xi, v'^\xi)$ for all scenarios, and $R(\xi, v'^\xi) > R(\xi, \bar{v}^\xi)$ for at least one scenario $\xi \in \Xi$. This contradicts that Γ_B is the optimal objective value, because visit plan v' is a feasible solution to RVF that yields an upper bound strictly greater than Γ_B. In turn, following the update rule described above after solving the lower-level problems for $\bar{z}$ would yield a lower bound strictly greater than Γ_B as well.

Appendix D: Proofs of selected results

Proof of Lemma 1. By definition,

$$\phi^{\xi}_{\mathcal{Q}} = \sum_{i\in\mathcal{Q}} r_i(1-\xi_i)v_i^{\xi,\mathcal{Q}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(1-p_i)v_i^{\xi,\mathcal{Q}} \tag{52a}$$

$$\geq \sum_{i\in\mathcal{Q}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(1-p_i)v_i^{\xi,\mathcal{P}} \tag{52b}$$

$$\geq \sum_{i\in\mathcal{P}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(1-p_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} \tag{52c}$$

$$\geq \sum_{i\in\mathcal{P}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(1-p_i)v_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(1-p_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} \tag{52d}$$

$$\geq \phi^{\xi}_{\mathcal{P}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(\xi_i-p_i)v_i^{\xi,\mathcal{P}}, \tag{52e}$$

where the second line follows from the optimality of $v^{\xi,\mathcal{Q}}$, the third by adding and subtracting $\sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}}$, and the last one from the definition of $\phi^{\xi}_{\mathcal{P}}$. By doing similar steps, starting from $\phi^{\xi}_{\mathcal{P}}$ rather than $\phi^{\xi}_{\mathcal{Q}}$ it can be shown that $\phi^{\xi}_{\mathcal{P}} \geq \phi^{\xi}_{\mathcal{Q}} + \sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{Q}}$. The combination of both inequalities gives the result. ∎

Proof of Theorem 1. Note that

$$\sum_{i\in\mathcal{N}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} = \phi^{\xi}_{\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(1-\xi_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(1-p_i)v_i^{\xi,\mathcal{P}} \tag{53a}$$

$$= \phi^{\xi}_{\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}}, \tag{53b}$$

and therefore

$$\gamma(\mathcal{P}) = \sum_{\xi\in\Xi} \pi^{\xi}(\phi^{\xi}_{\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}}). \tag{54}$$

Because an analogous result holds for $\mathcal{Q}$, we conclude that

$$\gamma(\mathcal{P})-\gamma(\mathcal{Q}) = \sum_{\xi\in\Xi} \pi^{\xi}\Big(\phi^{\xi}_{\mathcal{P}} - \phi^{\xi}_{\mathcal{Q}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{Q}}\Big) \tag{55a}$$

$$\leq \sum_{\xi\in\Xi} \pi^{\xi}\Big(\sum_{i\in\mathcal{P}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}} + \sum_{i\in\mathcal{N}\setminus\mathcal{P}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{Q}}\Big) \tag{55b}$$

$$\leq \sum_{\xi\in\Xi} \pi^{\xi}\Big(\sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{P}} - \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)v_i^{\xi,\mathcal{Q}}\Big), \tag{55c}$$

where (55b) follows from the upper-bound in Lemma 1, and thus the upper-bound in (28) follows. The lower-bound in (28) follow from a similar procedure, using the lower-bound in Lemma 1. ∎

Proof of Corollary 1. By Theorem (1) we have that,

$$\gamma(\mathcal{P})-\gamma(\mathcal{Q}) \leq \sum_{\xi\in\Xi} \pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i(p_i-\xi_i)(v_i^{\xi,\mathcal{P}} - v_i^{\xi,\mathcal{Q}})\Big] \tag{56a}$$

$$\leq \sum_{\xi\in\Xi} \pi^{\xi}\Big[\sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i \max\{p_i-\xi_i, 0\}(v_i^{\xi,\mathcal{P}} - v_i^{\xi,\mathcal{Q}})\Big] \tag{56b}$$

$$\leq \sum_{\xi\in\Xi} \pi^{\xi} \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i \max\{p_i-\xi_i, 0\} \tag{56c}$$

$$\leq \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i \sum_{\xi\in\Xi} \pi^\xi \max\{p_i - \xi_i, 0\} \tag{56d}$$

$$\leq \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i p_i \sum_{\xi\in\Xi:\ \xi_i=0} \pi^\xi \tag{56e}$$

$$\leq \sum_{i\in\mathcal{N}\setminus\mathcal{Q}} r_i p_i (1-p_i). \tag{56f}$$

The final inequality in (29) follows by noticing that $x(1-x)\leq 1/4$ for any $0\leq x\leq 1$. ∎

Proof of Proposition 1. The inequality in (36) follows from the right inequality in Lemma 1, by setting $\mathcal{P}=\mathcal{N}$, $\mathcal{Q}=\{i\in\mathcal{N}: z_i=1\}$, and by noting that under these definitions v^ξ is the optimal visit plan associated with $\phi^\xi_{\mathcal{Q}}$, i.e., $\phi^\xi_{\mathcal{Q}}=\sum_{i\in\mathcal{N}} r_i((1-\xi_i)z_i+(1-p_i)(1-z_i))v^\xi_i$. Inequality (37) follows by similar arguments from the left inequality in Lemma 1, by setting $\mathcal{P}=\{i\in\mathcal{N}: z_i=1\}$ and $\mathcal{Q}=\emptyset$. ∎

Appendix E: Sensitivity Analysis

We study the effect of the valid inequalities (36)–(37) when solving RFV($\hat{\mathcal{V}}$). We compare two versions of the `VF` approach, one without the valid inequalities (`VF-NO-VI`) and one with the valid inequalities (`VF-VI`). We focus on the 45 instances with 15 nodes and report the average computation time for solving problem RFV($\hat{\mathcal{V}}$) and the optimality gap for instances not solved within the time limit.

Table 7 Measuring the effect of the valid inequalities

$\|\Xi\|$	B	`VF-NO-VI`		`VF-VI`	
		Time (s)	Gap (s)	Time	Gap
	3	102	0.0%	105	0.0%
10	5	85	0.0%	87	0.0%
	10	43	0.0%	43	0.0%
	3	1858	0.7%	1289	0.5%
20	5	3265	4.5%	1877	1.6%
	10	1560	0.2%	1785	0.8%
	3	8675	14.9%	8675	14.9%
50	5	8658	15.4%	8663	15.5%
	10	8645	15.3%	8644	15.5%
	Total	**3655**	**5.7%**	**3463**	**5.4%**

Table 7 shows that overall adding the valid inequalities results in a roughly 5% decrease in the average computation time needed to solve RFV($\hat{\mathcal{V}}$) as well as marginal improvement in the average optimality gap. In some cases the effect of the valid inequalities is negligible (see instances with $|\Xi|=10$) while in other cases the effect is considerable (see instances with $|\Xi|=20$ and $B=5$).