LA-UR-21-24769
Accepted Manuscript

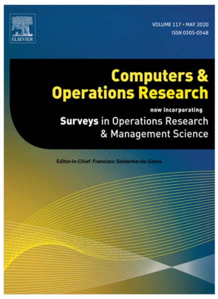
Routing problem for Unmanned Aerial Vehicle patrolling missions—A progressive hedging algorithm

Rajan, Sudarshan
Sundar, Kaarthik
Gautam, Natarajan

To be published in: Computers & Operations Research

DOI to publisher's version: 10.1016/j.cor.2022.105702

Permalink to record:
http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-21-24769



Routing problem for Unmanned Aerial Vehicle patrolling missions—A progressive hedging algorithm

Sudarshan Rajan, Kaarthik Sundar, Natarajan Gautam

PII: S0305-0548(22)00007-7
DOI: https://doi.org/10.1016/j.cor.2022.105702
Reference: CAOR 105702

To appear in: *Computers and Operations Research*

Received date : 14 July 2020
Revised date : 30 November 2021
Accepted date : 7 January 2022

Routing Problem for Unmanned Aerial Vehicle Patrolling Missions - A Progressive Hedging Algorithm

Sudarshan Rajan[a], Kaarthik Sundar[b], Natarajan Gautam[a]

[a] *Texas A& M University, College Station, TX*
[b] *Information Systems & Modeling, Los Alamos National Laboratory, Los Alamos, NM*

Abstract

The paper presents a two-stage stochastic program to model a routing problem involving an Unmanned Aerial Vehicle (UAV) in the context of patrolling missions. In particular, given a set of targets and a set of supplemental targets corresponding to each target, the first stage decisions involve finding the sequence in which the vehicle has to visit the set of targets. Upon reaching each target, the UAV collects information and if the operator of the UAV deems that the information collected is not of sufficient fidelity, then the UAV has to visit all the supplemental targets corresponding to that target to collect additional information before proceeding to visit the next target. The problem is solved using a progressive hedging algorithm and extensive computational results corroborating the effectiveness of the proposed model and the solution methodology is presented.

Keywords: two-stage stochastic program, progressive-hedging, integer programming, routing, unmanned aerial vehicles

1. Introduction

The use of small Unmanned Aerial Vehicles (UAVs) has seen a tremendous increase in the past decade for both civilian applications [1, 2] viz., precision agriculture [3, 4], forest fire monitoring and management [5], ocean bathymetry [6] etc. and military applications [7] viz., monitoring, surveillance, border patrol, intelligence, and reconnaissance (see [8, 9] and references therein). The primary reason for this increase is attributed to the low fixed and operational costs, ease of use, payload capacity, and ability to fly low-altitude missions. There is extensive work in the literature [10] that examines specific applications and develops algorithms ranging from higher level path planning to lower level control algorithms and the integration of the aforementioned applications. Higher level path planning algorithms are offline algorithms that are used to obtain paths for the vehicles a few hours prior to the start of the mission; in the optimization literature, this corresponds to vehicle routing problems (VRP) for UAVs. The lower level control algorithms are online algorithms that perform real-time trajectory adjustments and vehicle speed changes based on the environment at play. Most of the literature in small UAVs have had an application-specific focus because each application tends to impose its own unique set of constraints that need to be handled separately. One application of small UAVs that has received little attention in literature, in terms of higher level path planning, is that of patrolling [11].

Patrolling applications are essentially data collection missions and despite the term patrolling having a military connotation, similar applications exists even in a civilian context e.g., precision agriculture and crop health monitoring [3]. Such missions are typically involve uncertainty in the information collected from pre-specified targets. It also entails re-routing to visit additional nearby locations near the pre-specified targets to improve the fidelity of the information gathered. Hence, higher level planning algorithms that take into account this information uncertainty and plan vehicle routes are needed for such applications.

There have been several attempts in the last few years to build platforms for operating small UAVs in data gathering missions pertaining to precision agriculture [3], mapping disaster recovery [12], etc. To make things concrete, consider the application of precision farming where the UAV visits a particular target to obtain images of the crops. Suppose the images indicate that the crops at that location are damaged or have an unusual pattern in leaf color, then the farmer would want the vehicle to inspect additional nearby locations to gather more information to come up with robust conclusions on the crop health. If nothing unusual is observed then the vehicle is expected to continue to the next pre-specified target location. In this article, we address the higher level planning problem that occurs frequently in almost all data gathering applications and is informally defined as follows:

Given a set of targets, a source, a destination, and a UAV, the objective is to find a path for the UAV that starts at the source, terminates at the destination, and visits each target exactly once, such that the following conditions are satisfied: if the (uncertain) information collected at a particular target is not sufficient, the UAV has to visit one [13] or more supplemental targets corresponding to the visited target, before proceeding towards the next target in the path such that the sum of the cost of the path and the expected additional travel cost to visit the supplemental targets is a minimum.

We shall refer to the above problem as the single vehicle data gathering problem (SVDGP). The uncertainty in the SVDGP is associated with the information collected at a particular target. This uncertainty is modeled using a probability distribution and we do not make assumptions on the exact form of the distribution. Rather, we assume that the distribution permits sampling. This makes the problem setting very general and increases its applicability to a very large pool of civilian and military applications detailed in the previous paragraphs. We formulate the SVDGP as a two-stage decision making problem where the first stage ("here-and-now") decisions are made before the mission under the face of uncertainty and the second stage, recourse decisions are made once the mission starts and uncertainty is revealed. We set up the problem as a two-stage sequential decision making problem instead of a multi-stage problem for the following reason: an algorithm algorithm to solve the two-stage version of the problem provides a reasonable indication of whether the algorithm can scale to a multi-stage version, i.e., if the two-stage version of the problem itself is intractable, then the multi-stage version of the problem is definitely not tractable, and finally (iii) a two-stage approximation combined with a rolling horizon algorithm [14] could be an efficient way to solve the multi-stage version of the stochastic optimization problem. Nevertheless, we note that the formulation and the algorithms presented in this paper can be extended to a multi-stage setting.

1.1. Related Work

The literature contains many variants of UAV routing problems, and algorithms that can obtain optimal solutions and heuristics have been extensively studied for these variants. For ease of exposition, we will analyze the work done in the literature using the following three categories: (1) deterministic UAV routing approaches, (2) stochastic approaches, and (3) progressive-hedging to solve stochastic variants.

As far as deterministic approaches are concerned, plenty of optimization models and algorithms to compute optimal solutions, and fast heuristics to compute good feasible solutions have received extensive attention over the past decade. As mentioned in the introduction, most of the literature concerning deterministic approaches have had an application-specific focus because each application tends to impose its own unique set of constraints that need to be handled separately. In most of the approaches, the concerned routing problems are modeled as variants of the single traveling salesman problem (TSP), multiple TSP, or Vehicle Routing Problems (VRPs) with additional constraints to model the specific mission at hand; for example see [15, 16, 17]. An interested reader is referred to [18] for an extensive survey on optimization approaches for deterministic UAV routing problems. To the best of our knowledge, the only work in the literature that addresses the specific application of patrolling is that of [11] where the authors propose an integer programming formulation to model the patrolling problem as a deterministic multiple TSP and present heuristics to solve the same.

When compared to the deterministic approaches, literature that focuses on modeling uncertainty in the missions and developing algorithms to solve stochastic variants of the problems concerning UAVs are scarce. Nevertheless, there exists a wide range of literature that deals with stochastic optimization problems in the context of the TSPs and VRPs. The interested reader is referred to [19] for a review of all the variants. For the TSP, stochastic variants involving uncertainty in the targets themselves and uncertainty in travel times are the predominant ones that are dealt with in the literature [20, 21]. As for the VRPs, stochasticity arising from uncertain demands [22, 23], capacities, customers [24], travel times [23, 25] and various combinations of those [26] have been rigorously examined both from a modeling and algorithm development standpoint. In the context of problems concerning UAVs, two-stage stochastic programming formulations have been explored in single and multiple UAV delivery problems [27, 28, 29, 30]. The related literature for stochastic TSP and VRPs is summarized in Table 1. To the best of our knowledge, apart from our preliminary conference article that introduces a simpler variant of the SVDGP with only one supplemental location per target [13], there is no other work that explores such formulations and algorithms in the context of data gathering missions with uncertainty in data collected at points of interest. In [13], we formulate the SVDGP with only one supplemental location per target to make the second stage optimization problem a linear program; this in turn enables a solution algorithm to compute the optimal two-stage solution using a Benders decomposition. In this paper, we explore a general and more practical setting where there are multiple supplemental locations per target that need to be visited to obtain high fidelity information on the target area under consideration.

Problem type	Uncertainty	Reference	Approach
TSP	targets	[20]	theoretical bounds
TSP	travel times	[21]	dynamic programming
VRP	demands	[22, 23]	heuristics, Markov Decision Process
VRP	capacity, customers	[24]	heuristics and bounds
VRP	travel times	[25]	heuristics
VRP	travel times, capacity, demands	[26]	nearest neighborhood heuristic
UAV routing	facility location	[27]	heuristic + shortest-path
UAV delivery	reliability of drones	[28]	MILP heuristics
UAV routing	fuel consumption	[29]	SAA with rounding heuristic
UAV delivery	delivery service	[11]	SAA + genetic algorithm

Table 1: Literature related to stochastic variants of the TSP and VRPs.

In this article, we present a progressive hedging algorithm to solve the two-stage stochastic programming formulation for the SVDGP. To that end, we first review the literature that utilize the progressive hedging algorithm to solve stochastic variants of TSP and VRPs. To the best of our knowledge, [31] is the first work to solve a two-stage stochastic programming formulation for a multi-path TSP with uncertain travel times using a progressive hedging algorithm; they showed its the benefits of the proposed methodology in both solution quality and computational effort when compared to solving the deterministic equivalent formulation using a commercial solver. Other recent works also show the algorithm's effectiveness in a stochastic time-window assignment VRP [32], stochastic inventory routing problems [33], and other variants of VRPs [34]. In all these papers, it was observed that the progressive hedging approach was efficient in solving the stochastic variants of TSP and VRPs making it a suitable first choice for the SVDGP.

The contributions of this article are as follows: we develop the first two-stage stochastic programming formulation for a UAV data-gathering mission where uncertainty arises in the actual data collected in the targets. The formulation, as presented in the subsequent sections, contains binary variables in both the first as well as the second stage and we present a progressive hedging algorithm [35] to solve this problem. The motivation behind developing a progressive hedging algorithm to solve this two-stage problem formulation will be detailed in later sections. Finally, extensive computational experiments that corroborate the effectiveness of these models against deterministic counterparts and effectiveness of the algorithm to compute an optimal solution for any instance of the problem, as opposed to solving an extensive form (EF) using sample average approximation (SAA), are presented. The rest of the article is organized as follows: in Sec. 2, we present the formal problem statement after introducing the necessary notations. In Sec. 3, we formulate the SVDGP as a two-stage stochastic program and present an algorithm to solve the same in Sec. 4. Finally in Sec. 5, we present the computational results followed by conclusions and future research directions in Sec. 6.

2. Problem Statement

Before we present the formal problem statement, we introduce some notations that will be used throughout the rest of the article. We are given a set of n targets $\hat{T} = \{t_1, \ldots, t_n\}$, a source s, and a destination d. For ease for exposition, we will assume that $s = d = t_0$ and remark that the formulations and algorithms can be extended easily to the case where they are distinct. We also let $T = \hat{T} \cup \{t_0\}$. As detailed in the previous section, the decision making process is two-staged. The first stage decision for the mission is to compute a path for the UAV that starts and ends at t_0 and visits each target in the set $\hat{T}$ and collects information. Also associated with each target $k \in T$ is a set of supplemental locations S_k whose purpose is as follows: suppose the UAV visits the target k and collects some information, then depending on the this information gathered, it may have to visit some additional set of locations (supplemental location set S_k) to either validate the collected information or to gather more information. The decision on whether the UAV needs to visit these supplemental targets are recourse decisions and are taken after the uncertainty is realized. For the purpose of this article, we assume $|S_k|$, the cardinality of S_k for each $k = 1, \ldots, n$, is m, where $m \geqslant 2$ and $S_0 = \emptyset$. We shall, from here on, refer to the set $V = T \bigcup S_1 \bigcup \cdots \bigcup S_n$ as the set of vertices. Given these notations, the SVDGP is formulated on a graph $G = (V, E)$ where E denotes the edge set; we delegate the definition of the edge set E to the Sec. 3. We note that G can be a directed or an undirected graph and the formulation and the algorithms presented in this article are applicable to both cases. To keep the exposition fairly general, we will assume that the graph is directed and that the cost of traveling between any pair of targets is asymmetric i.e., given $(i, j) \in V$, $c_{ij} \neq c_{ji}$. The cost of traversing an edge $(i, j) \in E$ can be anything ranging from distance to travel time. We also assume that this cost of traversal between any pair of vertices is known a-priori or can be pre-computed. Since the focus of the article is to develop a model to account for uncertainty in a systematic way, in the next section, we detail the uncertainty model associated with the information collected at each target.

2.1. Uncertainty Modeling

The uncertainty in the problem arises from the fidelity of information collected at any target $i \in T$. Based on the information collected at target i, the UAV may or may not have to visit the supplemental locations in the set S_i before proceeding to visit the next target in the route. Hence, we associate with each target $i \in T$, a Bernoulli random variable with probability p_i which denotes the probability that the information collected at target i is not of sufficient fidelity. For the target t_0, we assume that $p_i = 0$, where $i \in \{t_0\}$. For ease of exposition, we assume that the random variables corresponding to any pair of targets, is independent of each other. However, this can be relaxed or changed and the uncertainty can also be modeled using a Markov chain. The uncertainty in this information collected over all the targets is modeled as a binary scenario vector of size $|T|$ where the component corresponding to target $i \in T$ takes a value 1 with a probability p_i. These realizations of uncertainty are captured in a countable set of scenarios Ω, in which each element $\omega \in \Omega$ occurs with probability p_ω. Each element $\omega \in \Omega$ is a $|T|$-dimensional binary vector where each element is 1 with a probability p_i. We let ω_i denote the i^{th} component of ω. When $\omega_i = 1$, the data gathered from the target i is not of sufficient fidelity and additional supplemental targets corresponding to the target i have to be visited by the vehicle. In this particular case, given that the probabilities p_i and p_j associated with any pair of targets (i, j) are independent, the probability of occurrence of the scenario ω is given by

$$p_\omega = \prod_{i=0}^{n} \{\mathbf{1}(\omega_i = 1) \cdot p_i + \mathbf{1}(\omega_i = 0) \cdot (1 - p_i)\} \tag{1}$$

where, the function $\mathbf{1}(\cdot)$ is the indicator function.

If we let $\mathcal{F}$ denote the set of all feasible first stage paths for the UAV, the first stage cost is the cost of the path and for any feasible path in $\mathcal{F}$, the second stage decision is a set of recourse decisions which is a set of routes through the supplemental targets based on the fidelity of information collected at the targets. The second stage decisions are scenario-dependent and the second stage cost is given by the expected additional cost of traversal for the UAV to visit the supplemental targets. The goal of the SVDGP is to find a path that minimizes the sum of the first and second stage travel costs.

3. Mathematical Formulation

We remark that when there is no uncertainty associated with the problem and when the information gathered at every target $i \in T$ is already of sufficient fidelity, then no supplemental targets need to be visited by the vehicle and the first stage problem reduces to a asymmetric TSP on the set T. For the SVDGP, if the information associated with a target $i \in T$ is not of sufficient fidelity, the UAV needs to visits the supplemental targets in the set S_i before proceeding towards the next target. To formulate this problem as a two-stage stochastic program, we first define two edge sets E^1 and E^2 for the first and the second stages respectively. The edge set E^1 includes edges that are allowed for the vehicle in the first stage i.e., any edge between every pair of vertices in the set T. The second stage edge set E^2 includes the edges that the vehicle is allowed to traverse in the second stage i.e., apart from containing all the edges in the set E^1, it includes the edges from every target $i \in T$ to every supplemental target in the set S_i, from every supplemental target in the set $\cup_i S_i$ to every target in the set T and for every $i \in T$, between any pair of supplemental targets in the set S_i. For any edge in the set $(i, j) \in E^1 \cup E^2$, c_{ij} denotes the cost of traversal of that edge. Also, given a subset of vertices $\hat{V}$, we define $\delta^1_+(\hat{V}) = \{(i, j) \in E^1 : j \notin \hat{V} \text{ and } i \in \hat{V}\}$ and $\delta^2_+(\hat{V}) = \{(i, j) \in E^2 : j \notin \hat{V} \text{ and } i \in \hat{V}\}$ as the set of outgoing edges in the first and second stage edge set, respectively. Similarly, given $\hat{V} \subset V$, we let $\delta^1_-(\hat{V}) = \{(j, i) \in E^1 : j \notin \hat{V} \text{ and } i \in \hat{V}\}$ and $\delta^2_-(\hat{V}) = \{(j, i) \in E^2 : j \notin \hat{V} \text{ and } i \in \hat{V}\}$ as the set of incoming edge. Furthermore, when $\hat{V} = \{i\}$ i.e., a singleton, we simply write $\delta^1_+(i)$ instead of to $\delta^1_+(\{i\})$. Finally, given two disjoint subsets of vertices $V_1, V_2 \subset V$, we define $\gamma(V_1 \to V_2) = \{(i, j) \in E^2 : i \in V_1, j \in V_2\}$.

3.1. Objective function

Given the above notations, we introduce binary first stage decision variables x_{ij} for each $(i, j) \in E^1$, denoting the presence of the edge (i, j) in the first stage solution and another set of binary second stage decision variables y^ω_{ij} for each $(i, j) \in E^2$ denoting the presence of the edge (i, j) in the second stage for the scenario $\omega \in \Omega$. We let $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the vector of first and second stage decision variables, respectively. Finally, given a subset of edges $\hat{E}^1 \subset E^1$ and $\hat{E}^2 \subset E^2$, we let $x(\hat{E}^1)$ and $y^\omega(\hat{E}^2)$ denote the sums $\sum_{(i,j)\in\hat{E}^1} x_{ij}$ and $\sum_{(i,j)\in\hat{E}^2} y^\omega_{ij}$, respectively. Now, the objective for the two-stage stochastic programming formulation of the SVDGP is given by:

$$\begin{aligned} \min\ C &\triangleq \sum_{(i,j)\in E^1} c_{ij}x_{ij} + \mathbb{E}_\Omega\left[\beta(\boldsymbol{x},\omega)\right] \\ &= \sum_{(i,j)\in E^1} c_{ij}x_{ij} + \sum_{\omega\in\Omega} p_\omega \beta(\boldsymbol{x},\omega) \end{aligned} \tag{2}$$

Here, $\beta(\boldsymbol{x},\omega)$ is the additional traversal cost required to visit the supplemental targets given a scenario $\omega \in \Omega$.

3.2. First stage constraints

Since the first stage solution is a feasible tour through the set of targets in T, the constraints in this stage are the TSP constraints which are given below:

$$x(\delta^1_+(i)) = 1 \quad \forall i \in T, \tag{3a}$$
$$x(\delta^1_-(i)) = 1 \quad \forall i \in T, \tag{3b}$$
$$x(\delta^1_+(S)) \geqslant 1 \quad \forall S \subset T,\ |S| \geqslant 2,\ \text{and} \tag{3c}$$
$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E^1. \tag{3d}$$

Here, Eqs. (3a) and (3b) are the in-degree and out-degree constraints that enforce exactly one edge to enter and leave each target in the set T. Eq. (3c) ensures that there are no sub-tours in the solution, and finally Eq. (3d) enforces the binary restrictions on the decision variables x_{ij}.

3.3. Second stage formulation

The second stage model for a fixed first stage solution $\boldsymbol{x}$ and the realization of uncertainty $\omega \in \Omega$ is as follows:

$$\beta(\boldsymbol{x},\omega) = \min \quad \sum_{(i,j)\in E^2} c_{ij}y^\omega_{ij} - \sum_{(i,j)\in E^1} c_{ij}y^\omega_{ij} \tag{4a}$$

subject to:

$$y^\omega(\delta^2_+(i)) = 1 \quad \forall i \in T, \tag{4b}$$
$$y^\omega(\delta^2_-(i)) = 1 \quad \forall i \in T, \tag{4c}$$
$$y^\omega(\delta^2_+(j)) = \omega_i \quad \forall j \in S_i, i \in T, \tag{4d}$$
$$y^\omega(\delta^2_-(j)) = \omega_i \quad \forall j \in S_i, i \in T \tag{4e}$$
$$y^\omega_{ij} = x_{ij}(1-\omega_i) \quad \forall (i,j) \in E^1 \tag{4f}$$
$$y^\omega(\gamma(S_i \to \{j\})) = x_{ij}\omega_i \quad \forall i,j \in T, \tag{4g}$$
$$y^\omega(\delta^2_+(S)) \geqslant 1 \quad \forall S \subset V,\ |S| \geqslant 2, S\cap T \neq \emptyset \text{ and} \tag{4h}$$
$$y^\omega_{ij} \in \{0,1\} \quad \forall (i,j) \in E^2. \tag{4i}$$

The objective in Eq. (4a) minimizes the additional cost of visiting the supplemental targets if information collected at a particular target is not of sufficient fidelity. The first summation is the total cost of the route including the additional supplemental target visits and the second term is the cost of the edges that are present both in the first and the second stage solutions. The Eqs. (4b) and (4c) are the in-degree and out-degree constraints for the targets in the set T. The Eqs. (4d) and (4e) ensure that if the information collected at any target $i \in T$ is not of sufficient fidelity i.e., $\omega_i = 1$, then the supplemental targets in the set S_i have to be visited by the vehicle. The Eq. (4f) ensures that if an edge $(i,j) \in E^1$ is traversed by the vehicle in the first stage and if $\omega_i = 0$, then this edge is necessarily used in the second stage by setting $y_{ij}^{\omega} = 1$ and if an edge $(i,j) \in E^1$ is not used by the vehicle in the first stage, then it is also not used in the second stage corresponding to scenario ω. Eq. (4g) ensures that given a target $i \in T$ with $\omega_i = 1$ and given that $j \in T$ is the target that vehicle visits immediately after i in the first stage solution, the vehicle has to collect additional information from all the supplemental targets in the set S_i before visiting the next target j. Together, Eqs. (4f) and (4g) also ensure that the sequence in which the vehicle visits the targets in the first stage solution given by $\boldsymbol{x}$ remains unchanged in the second stage. The Eq. (4h) eliminate sub-tours in the second stage solution and finally, the Eq. (4i) impose binary restrictions on the decision variables y_{ij}^{ω}.

4. Solution Methodology

This section details the description of our approach to solve the two-stage mathematical formulation of the SVDGP presented in Sec. 3. The formulation, as presented, contains binary decision variables both in the first and the second stages. We leverage the Progressive hedging (PH) algorithm proposed by Rockafellar and Wets [36] to solve the SVDGP; PH is also referred to as scenario decomposition in the literature [35] since it decomposes stochastic programs by scenarios i.e., samples of the realization of the random variables in the problem. The other predominantly used method for solving mixed-integer stochastic optimization problems is Stochastic Dual Dynamic Programming (SDDP) [37]. This method relies on outer-approximating the recourse problems using linear constraints obtained from the dual values of linear relaxation of the recourse problems. For the SVDGP, we favour the PH algorithm over the SDDP since each recourse problem itself has exponential number of sub-tour elimination constraints that are added dynamically during the solve and obtaining dual values of the dynamically added constraints requires separate cut-management algorithm. PH is a well-known algorithm for solving multistage stochastic convex optimization problems [38]. In fact, PH possesses theoretical convergence guarantees when all decision variables in the convex multistage stochastic program are continuous, i.e., PH is guaranteed to converge to the global optimal solution of the multistage stochastic program when all decision variables, except for the *here-and-now* decision variables, are continuous. In the presence of discrete variables, a wealth of recent theoretical and empirical research [39, 40, 41] has shown that the PH algorithm can prove to be a very robust heuristic to solve stochastic programs, specifically the case of pure binary programs in both stages of the formulation even in the case that the number of scenarios is prohibitively large. These algorithmic features of PH make its application ideal for the SVDGP. To that end, the subsequent sections detail the techniques involved in the PH approach and present findings that shows its impact as an effective heuristic to solve the SVDGP. In the forthcoming paragraphs, we present an overview of the PH algorithm specifically tailored for the SVDGP.

4.1. Algorithm Overview

Before, we present the overview of the algorithm, we restate the two-stage stochastic program for the SVDGP in a concise manner as follows:

$$\min \quad \sum_{(i,j)\in E^1} c_{ij}x_{ij} + \sum_{\omega\in\Omega} p_\omega \beta(\boldsymbol{x},\omega) \tag{5a}$$

$$\text{subject to:} \quad \boldsymbol{x} \in \mathcal{Q} \tag{5b}$$

where, $\mathcal{Q} = \{\boldsymbol{x} : \boldsymbol{x} \text{ satisfies Eq. (3)}\}$. The concise formulation, as presented in Eq. (5), is the well-known extensive form of the two-stage stochastic program [42] in Sec. 3. The PH is a scenario-decomposition algorithm that uses a separate set of first stage decision variables for each scenario to perform parallel solves i.e., for each $\omega \in \Omega$, it introduces decision variables $\boldsymbol{x}_\omega$ and implicitly enforces the non-anticipativity constraints ($\boldsymbol{x} = \boldsymbol{x}_\omega \quad \forall \omega \in \Omega$) via penalization; these constraints avoid allowing the first stage decision vector $\boldsymbol{x}$ to depend on the scenario. The basic PH algorithm takes as input two parameters (i) a penalty factor, $\rho > 0$ and (ii) a termination threshold, ε. Given ρ and ε, the pseudo-code for the PH algorithm is as follows:

Algorithm 1 Progressive Hedging: a pseudo-code

Initialization:

1: $k \leftarrow 0$ ▷ iteration count

2: For every $\omega \in \Omega$, $\boldsymbol{x}_\omega^k \leftarrow \text{argmin}_{\boldsymbol{x} \in \mathcal{Q}} \sum_{(i,j) \in E^1} c_{ij} x_{ij} + \beta(\boldsymbol{x}, \omega)$

3: $\boldsymbol{x}^k \leftarrow \sum_{\omega \in \Omega} p_\omega \boldsymbol{x}_\omega^k$

4: For every $\omega \in \Omega$, $\boldsymbol{w}_\omega^k \triangleq \rho \cdot (\boldsymbol{x}_\omega^k - \boldsymbol{x}^k)$ ▷ initial weight vector computation $\boldsymbol{w}_\omega^k$

Iteration update:

5: $k \leftarrow k + 1$

Decomposition:

6: For every $\omega \in \Omega$,

$$\boldsymbol{x}_\omega^k \leftarrow \underset{\boldsymbol{x} \in \mathcal{Q}}{\text{argmin}} \left(\sum_{(i,j) \in E^1} c_{ij} x_{ij} + \langle \boldsymbol{w}_\omega^{k-1}, \boldsymbol{x} \rangle + \frac{\rho}{2} \left\| \boldsymbol{x} - \boldsymbol{x}^{k-1} \right\|^2 + \beta(\boldsymbol{x}, \omega) \right)$$

Aggregation:

7: $\boldsymbol{x}^k \leftarrow \sum_{\omega \in \Omega} p_\omega \boldsymbol{x}_\omega^k$

Weight update:

8: For every $\omega \in \Omega$, $\boldsymbol{w}_\omega^k \triangleq \boldsymbol{w}_\omega^{k-1} + \rho \cdot (\boldsymbol{x}_\omega^k - \boldsymbol{x}_\omega)$

Termination criterion check:

9: $\epsilon^k \triangleq \sum_{\omega \in \Omega} p_\omega \left\| \boldsymbol{x}_\omega^k - \boldsymbol{x}^k \right\|$

10: **if** $\epsilon^k > \varepsilon$ **then**

11: Go to Step 5

12: **else**

13: Terminate with $\boldsymbol{x}^k$ as the first stage solution

14: **end if**

In Step 6 of Algorithm 1, the symbol $\langle a, b \rangle$ denotes the dot product of the vectors a and b. Furthermore, we notice that the computations in Steps 2 and 6 involve solving multiple mixed-integer linear programs, one for each scenario in the set Ω and that the solves are completely parallelizable. The quadratic penalty term (or the proximal term [35]) penalizes the non-anticipativity constraints and ensures that it is satisfied as the algorithm terminates. For every iteration k, the term ϵ^k is also referred to as the residual. The values of of ε is a tolerance value and it can be set to any value that is desired depending on the level of accuracy that is required for PH algorithm. As for the value of ρ, the following procedure was used to set its value. This procedure is not new and has been previously used in the literature concerning PH algorithms in [35]. It was observed in [35] that values of $\rho \in (0, 1)$ is a good starting point to fix the value of ρ and that ρ values proportional to the first stage edge costs, i.e., c_{ij} would lead to good convergence behaviour (without oscillations) of the PH algorithm. This technique of fixing a ρ value proportional to the costs is referred to as "cost proportional" heuristic [35]. For the SVDGP, we combined both the suggestions by first scaling all the costs between every pair of targets in the range $(0, 1)$ and then allowing the value of ρ to be the average of all the scaled cost values. This technique worked well for all the computational experiments and hence, no further experiments were done to compute more effective values of ρ.

In general, there is no guarantee that the PH algorithm will converge to a binary first stage solution [36]. But, it has been observed that for a wide variety of problems it is observed to converge to a binary first stage solution [35]. In all our computational experiments, we did not find a single instance for which the algorithm did not converge. Nevertheless, if one encounters instances where convergence is not observed, heuristics like variable-fixing, slamming, or rounding can be used to force convergence. The following sections show the results obtained by adopting this approach to our two-stage formulation of the problem.

5. Computational Results

The PH algorithm for the SVDGP was implemented in C++, and CPLEX 12.8 was used to solve the multiple MILPs that occur in each iteration of the PH algorithm. All the computational experiments were performed using an 2.9 GHz Intel Core i7 processor with 16 GB RAM. The performance of the algorithm was tested on randomly generated test instances. The instances generation procedure is detailed in the subsequent section.

5.1. Instance Generation

All the targets, the supplemental locations, the source, and the destination were all generated on a 100×100 grid. The source and the destination vertices for all the instances were located at $(5, 5)$ and $(95, 95)$, respectively. The number of targets n were varied from 10 to 40 in steps of 5. For each target, the locations of its corresponding supplemental targets were also randomly generated within a maximum pre-specified radius, R, from the target location; the value of R was chosen from the set $\{5, 10\}$ units. The number of supplemental locations per target, m was chosen from the set $\{3, 5, 7, 9\}$. Now, as for the vehicle itself, we assume that the vehicle is a fixed-wing UAV with a minimum turn-radius of 5 units. For this fixed-wing UAV, the cost of traversal between any pair of targets is assumed to be the shortest path taken by the vehicle to go from one target to the other. This path length is in-turn a function of the heading angle of the UAV at each target. To that end, we generate random heading angles for every target and its corresponding supplemental location and then compute the length using the well-known result by Dubins [43]. We remark that the shortest path computed using the result in [43] is asymmetric. Given this instance generation procedure, the total number of instances was 56 and all computational experiments were performed on this set of 56 instances. A subset of these instances are used for each of the computational experiments and the subset of chosen instances for each experiment is presented in the respective sections.

5.2. PH Algorithm Parameters and Scenario Generation

The PH algorithm has two main input parameters, ρ and ϵ. The value of ϵ is set to 1×10^{-5} and the value of ρ for each instance was computed using the heuristic detailed in Sec. 4.1. We note that for majority of the instances the value of ρ computed using the heuristic was close to 0.5. The uncertain scenarios for each run of the problem are generated using a vector of Bernoulli random variable, one for each target. Across a majority of the experiments in the subsequent sections, the probability that the information collected at a particular target is of sufficient fidelity is assumed to be 0.5 for every target. For a few other computation experiments other probability values are used and these values are presented when the corresponding experiment is described. For the source and the destination, we set this probability to 0 and required number of scenarios are generated from this vector of Bernoulli random variables. The number of scenarios was varied from a minimum value of 10 to a maximum value of 200 across all experiments. Each randomly generated scenario, ω is a binary vector indicating if the information collected at a particular target is of sufficient fidelity or not. In the subsequent paragraphs, we present the results for the various computational experiments perform to evaluate the PH algorithm on the SVDGP. Before we present the results obtained using the PH algorithm, we remark that when the full two-stage stochastic formulation in Sec. 3 was provided to CPLEX with a time-limit of three hours, all problem instances with number of targets greater than 50 and number of scenarios greater than 20 timed-out. Hence, we do not present any results that show that CPLEX was not able to solve the full problem as stated in Sec. 3.

Performance of PH algorithm. In this section, we compare the performance of the PH algorithm against solving the full two-stage problem in its extensive form using CPLEX. For this study, we choose only the 10 target instances. The values of m and R are varied in the set $m \in \{2, 3, 4, 5\}$ and $R \in \{5, 10\}$. The values of m are chosen to be $m \in \{2, 3, 4, 5\}$ because the extensive form for the two-stage program does not complete within the stipulated computation time limit when $m > 5$. Finally, the number of scenarios, $|\Omega|$ was varied in the set $\{50, 100, 200\}$ and the probabilities to generate these scenarios were chosen in the set $\{0.2, 0.5, 0.8\}$. Then we generated 10 random instances per each combination of these parameters. In total, the number of test instances for this study was 720. CPLEX was used to solve the extensive form of the two-stage problem with a computation time limit of 7200 seconds. The Table 2 shows the number of instances where solving the extensive form failed to result in an optimal solution for the instance compared to solving the problem using the PH algorithm. As seen from Table 2, it is clear that solving the extensive form using CPLEX is very inefficient since many instances with just 10 targets were not solved to optimality. On the other hand, the PH algorithm is capable of solving all instances to optimality. This justifies the need for the proposed decomposition algorithm for the SVDGP. For all instances that were solved to optimality by both the solution methods, the difference in solution quality between the two methods was found to be 0%. This result provides an empirical proof that despite being an heuristic approach, the PH algorithm is capable of empirically producing the globally optimal solution for the two-stage SVDGP considered in this paper. We remark that this statement is not true in general since it is based on the experiments that were performed for the 720 test instances for this study. Finally, for all the runs where both the approaches provided the optimal solution, the PH algorithm was on an average 2.5 times faster than solving the extensive form using CPLEX. The objective values obtained by the progressive hedging algorithm and CPLEX is shown in Table 10. Furthermore, out of the 720 instances, 48% of the instances had PH algorithm converging to the optimal solution in more than 1 iteration.

$\lvert\Omega\rvert$	total # instances	# timed out instances	
		extensive form	PH algorithm
50	240	20	0
100	240	21	0
200	240	34	0

Table 2: Number of instances where providing the extensive form directly to CPLEX timed out as against applying the PH algorithm to solve the two-stage problem.

Comparison with deterministic equivalent. This set of results is focused on comparing the solution of the two-stage stochastic program provided by the PH algorithm against its the Expected Value Problem (EVP) [44]. The expected value problem is solved directly using CPLEX and the value of stochastic solution or VSS is the difference between the objective value of the full two-stage stochastic program and its expected value problem. It is known that this VSS is non-negative [44] and greater its value the greater the value in

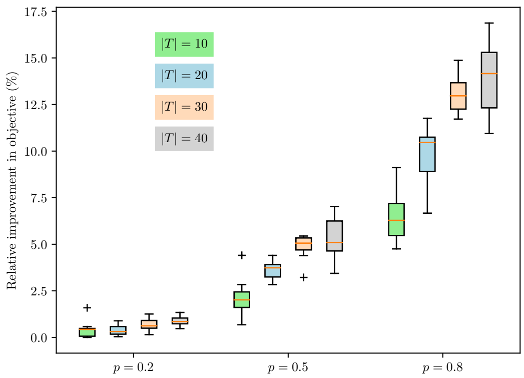


Figure 1: Value of Stochastic Solution (in percentage).

solving the stochastic version as opposed to the expected value problem. To this end, we fix the value of m and R to be 5 units and vary the probability of generating scenarios within the set $\{0.2, 0.5, 0.8\}$. The number of generated scenarios is fixed to 100 for all the runs. The Fig. 1 show the value of stochastic solution as a relative percentage with respect to the objective value of the EVP. As observed from the Fig. 1, there is a substantial quantitative value in solving the two-stage stochastic problem as against an EVP when the value of the probabilities used for generating the scenarios are greater than 0.5.

Effect of increasing the number of scenarios. Here, we present the first set of results to demonstrate the scalability of the PH algorithm for the SVDGP with increasing number of scenarios. To that end, for this experiment, the instances with value of $n \in \{10, 15, 20, 25, 30, 35, 40\}$, $m = 5$, $R = 5$ are chosen. For each of these instances, the number of scenarios are varied in the set $|\Omega| \in \{10, 25, 50, 100, 200\}$. The results for this set of experiments is shown in Tables 3 and 4. It is observed from the tables that the PH algorithm is effectively able to solve all the instances with up to 100 scenarios within two hours of computation time.

$\lvert\Omega\rvert$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
10	0.24	1.05	4.06	281.30	60.40	13.99	351.94
25	0.59	2.71	12.99	564.80	134.28	47.31	1084.65
50	0.95	4.85	21.86	1766.95	262.78	95.25	3121.86
100	2.02	9.94	46.57	5025.31	598.94	191.47	7256.49
200	4.28	19.07	57.45	9304.92	1133.70	336.65	16315.99

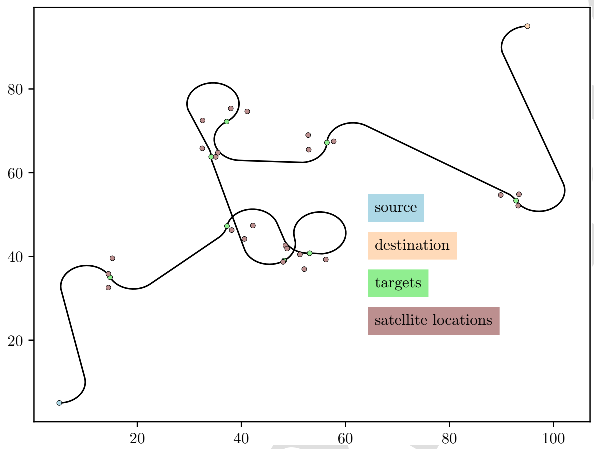
Table 3: Computation time (in seconds) taken by the PH algorithm for increasing number of targets and number of scenarios.

The Fig. 3 shows the feasible first stage (Fig. 2a) and a second stage solution (Fig. 2b) for one particular realization of the random variable for a 10-target instance. Notice that the paths taken by the vehicle are not straight lines because we assume that the vehicle is a fixed-wing aircraft with a minimum turn-radius of 5 units and follows the Dubins path [43].

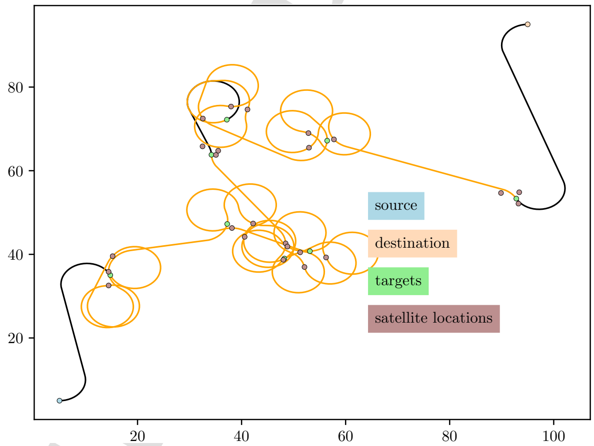
Effect of increasing the number of supplemental locations per target. The second set of results are aimed at demonstrating the scalability of the algorithm with increasing number of supplemental locations per target. To that end, for this experiment, the instances with value of $n \in \{10, 15, 20, 25, 30, 35, 40\}$, $m = \{3, 5, 7, 9\}$, $R = 5$, and $|\Omega| = 100$ (generated using a probability of 0.5) are chosen and the results are reported in Tables 5 and 6. Though the expectation is for the computation time and the number of iterations to increase

$\vert\Omega\vert$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
10	1	1	1	16	9	1	14
25	1	1	1	14	10	1	15
50	1	1	2	26	11	1	22
100	1	1	1	34	12	1	23
200	1	1	1	30	12	1	25

Table 4: Number of iterations taken to obtain convergence of the PH algorithm for increasing number of targets and number of scenarios.



(a) First stage solution for a 10-target instance.



(b) The solution to the two-stage problem for one particular realization of the uncertainty. The black lines are the path that is same as the first stage solution and the orange routes correspond to the additional visits to the supplemental locations corresponding to the targets where the fidelity of information collected was not sufficient.

Figure 2: The solution provided by the PH algorithm for a 10-target instance.

with increasing value of m, it is not really the case for this problem, as observed from Tables 5 and 6. Furthermore, we also observe that for the 40-target instances, the computation time is much greater than two hours. For typical patrolling applications, the number of target locations will be around 100 targets. For such patrolling missions, use of multiple UAVs is preferred. In this case, a simple clustering heuristic can be used to partition the targets into manageable cluster sizes and the algorithm can be used to compute a single vehicle solution for each cluster. This addresses the scalability aspect of the algorithm.

m	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
3	1.68	6.34	21.90	185.49	92.15	66.25	2777.89
5	2.02	9.94	46.57	5025.31	598.94	191.47	7256.49
7	5.82	33.55	63.68	162.63	1605.04	6082.88	9404.69
9	22.38	57.53	870.38	1422.58	412.64	7152.56	2325.88

Table 5: Computation time (in seconds) taken by the PH algorithm for increasing number of targets and number of supplemental locations per target.

m	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
3	1	1	1	11	1	1	20
5	1	1	2	34	12	1	23
7	1	1	1	1	7	19	20
9	1	1	6	6	1	8	1

Table 6: Number of iterations taken to obtain convergence of the PH algorithm for increasing number of targets and number of supplemental locations per target.

Effect of increasing the pre-specifed radius for the supplemental locations. This set of results presents the variation in the computation time and the number of iterations for the PH algorithm to converge when the pre-specified radius in which the supplemental locations are present is varied in the set $\{5, 10\}$. The results present the change in computation time and number of iterations for $n = \{10, 15, 20, 25, 30, 35, 40\}$ and when $m = 5$, $|\Omega| = 100$ (generated using a probability of 0.5). The Tables 7 and 8 present the computation times and the number of iterations for the above instances, respectively. Again, no clear trend exists between the value of R and the computation time or the number of iterations taken for the PH algorithm to converge; nevertheless, the trivial trend that can be observed from the two tables is that greater the number of iterations the greater the computation time.

R	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
5	2.023	9.943	46.569	5025.308	598.943	191.472	7256.488
10	45.824	762.224	23.985	98.602	736.891	1284.960	2022.578

Table 7: Computation time (in seconds) taken by the PH algorithm for values of R.

Effect of changing the probabilities in the scenario generation. This set of results is aimed at examining the effect of changing the probabilities that the information collected at any target is not of sufficient fidelity. We choose a specific instance with $n = 20$, $m = 5$, and $R = 5$ units. There is no specific reason for this choice and the results for all the other instances followed the same trend. To that end, the probabilities that we choose to generate 100 scenarios for this particular instance is given by $\{0.0, 0.2, 0.5, 0.8, 1.0\}$. We remark that when the probability is 0.0, it basically means that the information collected at all the targets

R	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$	$n = 35$	$n = 40$
5	1	1	2	34	12	1	23
10	13	37	1	1	10	12	7

Table 8: Number of iterations taken by the PH algorithm for varying values of R.

is of sufficient fidelity and this reduces the SVDGP to computing a traveling salesman tour through the set of targets. On the other hand, if all the probabilities take a value 1.0, then the information collected at every target is not of sufficient fidelity and the SVDGP reduces to a TSP the set of targets and their supplemental locations with additional sequencing constraints i.e., the supplemental locations have to be visited immediately after their respective target visits. The Table 9 presents both the computation time and the number of iterations for this experiment. The computation time for the case when the probability value is 0.0 is the least as solving a TSP with 20 targets is substantially fast. On the other end of the spectrum, the computation time for case with a probability value of 1.0 is the maximum as it reduces to the problem of solving a TSP with 120 vertices and additional sequencing constraints. Also, for these two cases, the recourse action is obtained directly by solving the TSP or the TSP with additional sequencing constraints and hence, the number of iterations for the PH algorithm to converge for these two cases will always be equal to one.

p	time (seconds)	# iter
0	4.88	1
0.2	39.62	5
0.5	45.79	2
0.8	89.38	1
1	269.26	1

Table 9: Computation time and number of iterations for different values of probabilities. Here p represents the probability that the information collected at the target is not of sufficient fidelity for any target, and # iter is the number of iterations taken by the PH algorithm to converge.

PH algorithm residual statistics. This set of results shows the changes in the residual of the PH algorithm (i.e., ϵ^k in Algorithm 1) for varying number of scenarios. We remark that this residual function is not a strictly decreasing function. For this experiment, the instance with $n = 40$, $m = 5$, and $R = 5$ was chosen. As far as the scenarios are concerned, all of them were created using a probability value of 0.5. The Fig. 3 shows the residual values for varying number of scenarios. All the results presented thus far illustrate the effectiveness of the PH algorithm in computing heuristic solutions to the SVDGP with uncertainty in the fidelity of information collected.

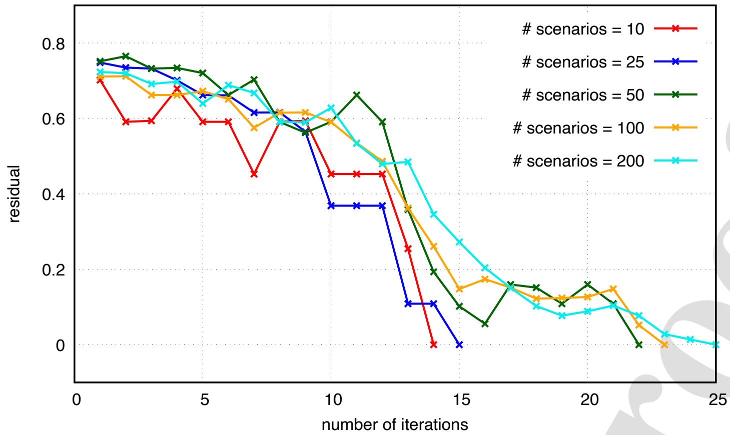


Figure 3: Residual value progress for the PH algorithm. The value of (n, m, R) for the instance for this set of runs is given by $(40, 5, 5)$, respectively.

6. Conclusion

This article presents the first formulation and solution approach for a UAV path planning problem in the context of data gathering missions. We present a Progressive Hedging algorithm to solve the problem and this may be considered as a good first step towards practical implementation. The PH algorithm has been tested on its effectiveness has been corroborated on a large set of test instances. On all the instances, it was observed that no convergence issues have been observed. Nevertheless, future work would focus on improved algorithms to prune the computation time to the order of minutes. This problem can be extended to a multi-stage setting using the same framework. This is the first attempt to solve such a sequential stochastic programming problem in this context and there is scope to work on this further to develop and refine distributed computing algorithms to solve it for a multi-stage setting while reducing the computational times. Furthermore, extension of this problem to a multi-vehicle setting entails addressing an additional complexity of partitioning the targets. This partitioning problem can be solved heuristically using clustering algorithms or one can resort to exact resolution of the partitions by embedding constraints in to the model itself; this presents an interesting avenue for future work.

Acknowledgements

Kaarthik Sundar would like to acknowledge the funding provided by LANL's Directed Research and Development (LDRD) project: "20200603ECR: Distributed Algorithms for Large-Scale Ordinary Differential/Partial Differential Equation (ODE/PDE) Constrained Optimization Problems on Graphs". This work was carried out under the U.S. DOE Contract No. DE-AC52-06NA25396.

Appendix A Progressive Hedging Approach vs CPLEX

The Table 10 shows the objective values obtained by using the CPLEX and the progressive hedging algorithm.

Table 10: Complete results for performance of PH algorithm.

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
50	2	10	0	417.682	417.682	0.772	0.268
50	2	10	1	471.799	471.799	0.612	0.214
50	2	10	2	422.116	422.116	7.744	0.238
50	2	10	3	416.741	416.741	2.173	0.162
50	2	10	4	438.438	438.438	0.485	0.300
50	2	10	5	422.456	422.456	5.085	0.315
50	2	10	6	427.494	427.494	0.452	0.176
50	2	10	7	462.189	462.189	0.540	0.208
50	2	10	8	405.399	405.399	0.401	0.197
50	2	10	9	385.918	385.918	0.539	0.140
50	2	5	0	431.853	431.853	0.577	0.146
50	2	5	1	445.085	445.085	0.640	0.283
50	2	5	2	408.367	408.367	0.647	0.267
50	2	5	3	446.556	446.556	0.543	0.150
50	2	5	4	431.737	431.737	3.377	0.117
50	2	5	5	523.067	523.067	5.933	0.351
50	2	5	6	469.162	469.162	0.583	0.172
50	2	5	7	466.46	466.46	0.588	0.143
50	2	5	8	442.777	442.777	0.531	0.128
50	2	5	9	433.674	433.674	5.458	0.134
50	3	10	0	414.775	414.775	0.756	0.174
50	3	10	1	497.215	497.215	0.718	0.274
50	3	10	2	426.181	426.181	0.684	0.309
50	3	10	3	507.662	507.662	8.547	0.338
50	3	10	4	472.554	472.554	0.738	0.265
50	3	10	5	470.72	470.72	0.825	0.776
50	3	10	6	469.779	469.779	1.266	0.248
50	3	10	7	454.722	454.722	0.727	0.315
50	3	10	8	507.662	507.662	8.343	0.343
50	3	10	9	435.764	435.764	3.548	0.254
50	3	5	0	483.471	483.471	0.637	0.215
50	3	5	1	470.018	470.018	0.817	0.305
50	3	5	2	463.736	463.736	0.679	0.209
50	3	5	3	565.074	565.074	1.013	1.085
50	3	5	4	477.83	477.83	1.239	0.797
50	3	5	5	418.666	418.666	0.619	0.184
50	3	5	6	498.611	498.611	0.662	0.352
50	3	5	7	502.623	502.623	0.528	0.225
50	3	5	8	441.076	441.076	1.317	0.193
50	3	5	9	519.328	519.328	12.642	325.269
50	4	10	0	520.906	520.906	1.129	0.283
50	4	10	1	480.341	480.341	4.346	0.364
50	4	10	2	449.635	449.635	0.948	0.369
50	4	10	3	574.154	574.154	3.919	0.825
50	4	10	4	486.745	486.745	1.057	0.415
50	4	10	5	481.465	481.465	1.107	0.362
50	4	10	6	519.754	519.754	0.949	0.458
50	4	10	7	492.497	492.497	0.821	0.371
50	4	10	8	507.095	507.095	1.024	0.426
50	4	10	9	498.056	498.056	0.837	0.338
50	4	5	0	546.717	546.717	1.543	1.756
50	4	5	1	559.046	559.046	0.796	1.301
50	4	5	2	569.254	569.254	4.658	1.062
50	4	5	3	511.314	511.314	0.804	0.353
50	4	5	4	545.209	545.209	1.412	0.637
50	4	5	5	611.231	611.231	3.029	12.240
50	5	10	0	561.692	561.692	4.001	0.344
50	5	10	1	558.401	558.401	1.127	0.665
50	5	10	2	535.974	535.974	1.172	0.498
50	5	10	3	559.409	559.409	1.506	0.969
50	5	10	4	500.424	500.424	1.483	0.960
50	5	10	5	556.651	556.651	1.762	1.369

Journal Pre-proof

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
50	5	10	6	555.887	555.887	1.960	0.856
50	5	10	7	559.409	559.409	1.908	1.036
50	5	10	8	530.709	530.709	0.989	0.804
50	5	10	9	443.013	443.013	0.834	0.287
50	5	5	0	539.173	539.173	3.442	0.448
50	5	5	1	578.938	578.938	6.941	0.445
50	5	5	2	626.473	626.473	0.931	0.551
50	5	5	3	564.793	564.793	0.848	0.323
50	5	5	4	574.025	574.025	0.765	0.403
50	5	5	5	584.492	584.492	0.128	0.546
50	5	5	6	555.916	555.916	1.064	0.315
50	5	5	7	510.819	510.819	1.247	0.501
50	5	5	8	486.053	486.053	1.001	0.462
50	5	5	9	534.426	534.426	1.148	0.469
50	2	10	0	564.588	564.588	0.733	0.261
50	2	10	1	587.734	587.734	0.847	0.235
50	2	10	2	568.637	568.637	1.001	0.384
50	2	10	3	596.657	596.657	3.166	0.245
50	2	10	4	566.431	566.431	0.819	0.571
50	2	10	5	573.34	573.34	2.939	0.257
50	2	10	6	590.063	590.063	0.465	0.173
50	2	10	7	596.643	596.643	0.688	0.201
50	2	10	8	541.697	541.697	0.453	0.145
50	2	10	9	518.813	518.813	0.451	0.188
50	2	5	0	569.144	569.144	0.547	0.228
50	2	5	1	580.306	580.306	0.734	0.314
50	2	5	2	583.723	583.723	0.875	0.255
50	2	5	3	603.648	603.648	0.655	0.215
50	2	5	4	599.828	599.828	5.716	0.129
50	2	5	5	661.932	661.932	1.063	4.926
50	2	5	6	617.661	617.661	0.662	0.208
50	2	5	7	603.62	603.62	0.619	0.177
50	2	5	8	602.326	602.326	0.575	0.177
50	2	5	9	580.647	580.647	9.171	0.178
50	3	10	0	623.592	623.592	0.875	0.221
50	3	10	1	697.587	697.587	1.137	0.440
50	3	10	2	606.192	606.192	1.079	0.543
50	3	10	3	698.299	698.299	1.644	1.376
50	3	10	4	664.403	664.403	0.995	0.244
50	3	10	5	642.368	642.368	1.534	1.472
50	3	10	6	676.99	676.99	1.666	0.327
50	3	10	7	630.077	630.077	0.996	0.439
50	3	10	8	698.299	698.299	0.636	1.360
50	3	10	9	619.705	619.705	0.384	0.378
50	3	5	0	683.646	683.646	0.934	0.335
50	3	5	1	683.935	683.935	0.787	0.373
50	3	5	2	676.974	676.974	0.886	0.320
50	3	5	4	690.557	690.557	0.326	0.306
50	3	5	5	600.703	600.703	0.849	0.230
50	3	5	6	689.019	689.019	0.927	0.708
50	3	5	7	693.892	693.892	0.710	0.279
50	3	5	8	653.256	653.256	1.290	0.190
50	4	10	0	770.886	770.886	1.318	0.704
50	4	10	1	728.95	728.95	7.449	0.517
50	4	10	2	690.439	690.439	1.837	0.623
50	4	10	3	794.735	794.735	5.722	1.523
50	4	10	4	729.515	729.515	1.762	1.054
50	4	10	5	718.056	718.056	1.575	0.437
50	4	10	6	750.941	750.941	1.155	0.506
50	4	10	7	776.613	776.613	1.672	1.008
50	4	10	8	775.285	775.285	1.554	0.499
50	4	10	9	757.748	757.748	1.246	0.438
50	4	5	0	823.311	823.311	4.744	11.164
50	4	5	1	827.186	827.186	0.745	0.903
50	4	5	2	850.385	850.385	0.841	0.409
50	4	5	3	749.986	749.986	1.144	0.564
50	4	5	4	829.013	829.013	2.146	1.098

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
50	5	10	0	889.759	889.759	2.650	1.328
50	5	10	1	908.227	908.227	1.782	1.241
50	5	10	2	826.431	826.431	1.798	0.708
50	5	10	3	889.42	889.42	54.325	338.499
50	5	10	4	791.85	791.85	2.824	5.650
50	5	10	5	866.095	866.095	2.530	3.470
50	5	10	6	845.644	845.644	2.463	2.048
50	5	10	7	889.42	889.42	56.516	351.462
50	5	10	8	790.15	790.15	1.742	0.941
50	5	10	9	755.779	755.779	1.186	0.363
50	5	5	0	875.86	875.86	0.271	0.718
50	5	5	1	976.848	976.848	0.826	0.875
50	5	5	2	934.133	934.133	2.461	3.989
50	5	5	3	899.895	899.895	0.088	0.511
50	5	5	4	915.169	915.169	0.322	0.916
50	5	5	5	918.459	918.459	1.997	0.601
50	5	5	6	943.349	943.349	1.310	0.412
50	5	5	7	837.063	837.063	1.806	1.346
50	5	5	8	813.896	813.896	1.624	1.253
50	5	5	9	843.011	843.011	1.729	0.534
50	2	10	0	707.404	707.404	1.012	0.242
50	2	10	1	712.45	712.45	1.001	0.274
50	2	10	2	742.132	742.132	0.064	0.379
50	2	10	3	749.457	749.457	0.873	0.199
50	2	10	4	703.348	703.348	0.711	0.404
50	2	10	5	697.452	697.452	2.573	0.304
50	2	10	6	731.009	731.009	0.600	0.179
50	2	10	7	710.035	710.035	0.683	0.227
50	2	10	8	675.702	675.702	0.422	0.144
50	2	10	9	677.487	677.487	0.633	0.149
50	2	5	0	707.94	707.94	0.784	0.161
50	2	5	1	741.628	741.628	0.797	0.231
50	2	5	2	719.653	719.653	0.836	0.218
50	2	5	3	774.369	774.369	1.035	0.280
50	2	5	4	746.002	746.002	0.754	0.126
50	2	5	5	802.944	802.944	1.835	1.029
50	2	5	6	780.306	780.306	0.654	0.163
50	2	5	7	721.766	721.766	0.678	0.182
50	2	5	8	754.708	754.708	0.668	0.184
50	2	5	9	694.717	694.717	3.688	0.219
50	3	10	0	800.8	800.8	0.969	0.232
50	3	10	1	869.324	869.324	1.449	0.983
50	3	10	2	800.098	800.098	1.052	0.483
50	3	10	3	930.755	930.755	3.241	4.917
50	3	10	4	857.544	857.544	1.085	0.274
50	3	10	5	854.733	854.733	2.039	0.794
50	3	10	6	860.336	860.336	1.997	0.345
50	3	10	7	840.869	840.869	1.827	0.782
50	3	10	8	930.755	930.755	3.640	4.926
50	3	10	9	762.409	762.409	0.487	0.508
50	3	5	0	907.978	907.978	0.822	0.359
50	3	5	1	904.377	904.377	0.781	0.334
50	3	5	2	890.432	890.432	1.332	0.476
50	3	5	4	893.92	893.92	0.005	0.690
50	3	5	5	790.395	790.395	0.601	0.223
50	3	5	6	858.508	858.508	0.908	0.534
50	3	5	7	895.249	895.249	1.051	0.277
50	3	5	8	889.208	889.208	1.597	0.219
50	4	10	0	1032.8	1032.8	2.014	4.176
50	4	10	1	954.21	954.21	0.634	0.847
50	4	10	2	976.209	976.209	2.580	2.434
50	4	10	3	1027.12	1027.12	6.839	3.930
50	4	10	4	992.837	992.837	2.713	1.429
50	4	10	5	964.372	964.372	2.086	0.651
50	4	10	6	979.4	979.4	1.215	0.510
50	4	10	7	1012.86	1012.86	2.921	1.444
50	4	10	8	986.333	986.333	1.418	0.571

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
50	4	10	9	1018.67	1018.67	2.088	0.633
50	4	5	0	1073.47	1073.47	6.007	8.475
50	4	5	1	1113.53	1113.53	2.972	11.613
50	4	5	2	1128.44	1128.44	9.017	0.605
50	4	5	3	1079.34	1079.34	2.247	1.231
50	4	5	4	1080.17	1080.17	3.118	2.451
50	5	10	0	1239.76	1239.76	1.643	3.828
50	5	10	1	1195.8	1195.8	2.571	1.209
50	5	10	2	1153.12	1153.12	2.325	0.803
50	5	10	4	1076.29	1076.29	3.282	2.197
50	5	10	5	1156.66	1156.66	3.132	3.448
50	5	10	6	1175.23	1175.23	40.006	801.837
50	5	10	8	1077.74	1077.74	2.465	0.896
50	5	10	9	1039.68	1039.68	1.314	0.427
50	5	5	0	1199.6	1199.6	8.439	0.910
50	5	5	1	1366.28	1366.28	3.479	3.684
50	5	5	2	1268.07	1268.07	64.848	583.397
50	5	5	3	1242.9	1242.9	6.462	0.608
50	5	5	4	1234.11	1234.11	0.811	0.916
50	5	5	5	1239.96	1239.96	2.462	0.980
50	5	5	6	1242.47	1242.47	1.361	0.441
50	5	5	7	1192.41	1192.41	2.242	1.292
50	5	5	8	1215.18	1215.18	2.545	1.754
50	5	5	9	1243.59	1243.59	2.562	0.715
100	2	10	0	432.656	432.656	1.605	0.528
100	2	10	1	473.369	473.369	1.358	0.480
100	2	10	2	426.553	426.553	0.118	0.475
100	2	10	3	416.797	416.797	4.148	0.345
100	2	10	4	440.704	440.704	0.999	0.742
100	2	10	5	419.272	419.272	9.340	0.505
100	2	10	6	422.386	422.386	0.974	0.303
100	2	10	7	460.78	460.78	1.314	0.416
100	2	10	8	402.133	402.133	0.955	0.271
100	2	10	9	387.248	387.248	0.798	0.268
100	2	5	0	424.957	424.957	0.881	0.343
100	2	5	1	449.513	449.513	1.259	0.396
100	2	5	2	401.989	401.989	1.239	0.485
100	2	5	3	447.654	447.654	1.056	0.441
100	2	5	4	438.401	438.401	6.923	0.222
100	2	5	5	512.268	512.268	0.274	0.625
100	2	5	6	464.289	464.289	2.851	0.349
100	2	5	7	465.361	465.361	1.129	0.315
100	2	5	8	438.544	438.544	1.024	0.235
100	2	5	9	431.062	431.062	0.171	0.268
100	3	10	0	416.347	416.347	1.505	0.482
100	3	10	1	485.414	485.414	1.484	0.550
100	3	10	2	434.133	434.133	1.622	0.556
100	3	10	3	511.886	511.886	0.311	0.795
100	3	10	4	461.796	461.796	1.401	0.443
100	3	10	5	458.9	458.9	1.575	1.380
100	3	10	6	460.888	460.888	2.554	0.455
100	3	10	7	457.245	457.245	1.569	0.577
100	3	10	8	511.886	511.886	0.608	0.817
100	3	10	9	428.272	428.272	7.049	0.450
100	3	5	0	475.114	475.114	1.065	0.546
100	3	5	1	467.074	467.074	1.546	0.587
100	3	5	2	475.708	475.708	1.164	0.501
100	3	5	3	550.841	550.841	1.649	1.742
100	3	5	4	501.554	501.554	0.634	0.531
100	3	5	5	424.969	424.969	1.264	0.385
100	3	5	6	478.91	478.91	1.137	0.639
100	3	5	7	494.186	494.186	1.055	0.418
100	3	5	8	448.686	448.686	3.708	0.393
100	4	10	0	535.409	535.409	1.994	0.831
100	4	10	1	479.954	479.954	8.365	0.658
100	4	10	2	468.614	468.614	2.177	0.765
100	4	10	3	556.685	556.685	7.116	1.493

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
100	4	10	4	470.401	470.401	2.234	0.764
100	4	10	5	502.999	502.999	2.272	0.762
100	4	10	6	535.541	535.541	1.994	1.154
100	4	10	7	490.162	490.162	1.645	0.913
100	4	10	8	486.675	486.675	2.044	0.906
100	4	10	9	489.617	489.617	1.693	0.793
100	4	5	0	545.184	545.184	2.878	11.292
100	4	5	1	540.314	540.314	2.075	1.329
100	4	5	2	559.409	559.409	8.406	0.680
100	4	5	3	501.997	501.997	1.573	0.760
100	4	5	4	548.954	548.954	2.780	1.392
100	4	5	5	590.569	590.569	3.575	31.176
100	5	10	0	565.472	565.472	1.743	0.944
100	5	10	1	552.005	552.005	2.251	1.252
100	5	10	2	526.907	526.907	2.284	0.959
100	5	10	3	548.901	548.901	2.224	2.737
100	5	10	4	515.092	515.092	2.994	2.949
100	5	10	5	551.203	551.203	3.091	2.306
100	5	10	6	567.268	567.268	40.828	39.188
100	5	10	7	548.901	548.901	2.560	3.105
100	5	10	8	546.874	546.874	1.888	1.015
100	5	10	9	478.012	478.012	1.837	0.961
100	5	5	0	535.254	535.254	7.844	0.875
100	5	5	1	616.894	616.894	1.769	1.029
100	5	5	2	609.796	609.796	1.536	1.448
100	5	5	3	578.15	578.15	0.221	0.724
100	5	5	4	568.82	568.82	0.983	0.865
100	5	5	5	575.168	575.168	2.198	0.874
100	5	5	6	564.321	564.321	1.936	0.683
100	5	5	7	503.147	503.147	2.427	1.161
100	5	5	8	504.655	504.655	2.374	0.964
100	5	5	9	535.562	535.562	2.273	0.810
100	2	5	0	576.226	576.226	1.163	0.291
100	3	5	0	676.132	676.132	1.587	0.544
100	2	10	0	569.202	569.202	1.907	0.594
100	2	10	1	593.716	593.716	1.690	0.628
100	2	10	2	577.225	577.225	0.404	0.993
100	2	10	3	588.66	588.66	6.586	0.415
100	2	10	4	568.424	568.424	1.146	1.416
100	2	10	5	582.548	582.548	5.999	0.478
100	2	10	6	580.525	580.525	1.284	0.400
100	2	10	7	585.163	585.163	1.007	0.436
100	2	10	8	544.692	544.692	0.880	0.339
100	2	10	9	526.901	526.901	1.301	0.320
100	2	5	0	576.226	576.226	1.410	0.415
100	2	5	1	585.064	585.064	1.490	0.467
100	2	5	2	570.801	570.801	1.363	0.566
100	2	5	3	595.692	595.692	1.249	0.477
100	2	5	4	608.7	608.7	1.229	0.222
100	2	5	5	656.636	656.636	12.780	12.181
100	2	5	6	623.611	623.611	1.381	0.428
100	2	5	7	597.357	597.357	1.385	0.371
100	2	5	8	596.272	596.272	1.226	0.354
100	2	5	9	563.011	563.011	0.684	0.541
100	3	10	0	617.571	617.571	2.088	0.515
100	3	10	1	691.075	691.075	2.359	1.936
100	3	10	2	608.269	608.269	2.121	0.930
100	3	10	3	709.671	709.671	1.173	2.435
100	3	10	4	655.832	655.832	1.908	0.528
100	3	10	5	635.801	635.801	2.685	10.306
100	3	10	6	670.698	670.698	3.372	0.648
100	3	10	7	637.934	637.934	2.137	0.818
100	3	10	8	709.671	709.671	4.815	2.437
100	3	10	9	603.37	603.37	0.988	0.744
100	3	5	0	676.132	676.132	1.480	0.574
100	3	5	1	692.941	692.941	1.502	0.799
100	3	5	2	687.168	687.168	1.658	0.762

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
100	3	5	4	702.186	702.186	4.658	1.676
100	3	5	5	606.65	606.65	1.339	0.433
100	3	5	6	668.473	668.473	1.603	0.834
100	3	5	7	698.02	698.02	1.803	0.556
100	3	5	8	668.222	668.222	3.269	0.443
100	4	10	0	774.683	774.683	2.755	2.951
100	4	10	1	728.794	728.794	15.227	1.095
100	4	10	2	696.011	696.011	3.705	5.664
100	4	10	3	786.965	786.965	1.598	8.449
100	4	10	4	719.109	719.109	3.515	1.920
100	4	10	5	730.417	730.417	3.341	1.111
100	4	10	6	779.048	779.048	2.512	1.319
100	4	10	7	777.077	777.077	3.161	1.931
100	4	10	8	772.975	772.975	2.833	1.192
100	4	10	9	743.483	743.483	2.330	0.948
100	4	5	0	817.752	817.752	16.730	890.002
100	4	5	1	805.32	805.32	2.673	1.472
100	4	5	2	852.735	852.735	1.175	1.017
100	4	5	3	789.891	789.891	2.757	1.174
100	4	5	4	820.692	820.692	4.255	32.472
100	5	10	0	910.113	910.113	10.600	16.746
100	5	10	1	896.893	896.893	3.436	2.767
100	5	10	2	828.857	828.857	3.397	1.488
100	5	10	4	810.489	810.489	5.061	1440.943
100	5	10	5	863.814	863.814	4.651	4.150
100	5	10	6	869.06	869.06	5.256	7.078
100	5	10	8	818.688	818.688	3.502	1.773
100	5	10	9	759.632	759.632	2.661	0.842
100	5	5	0	876.803	876.803	1.196	1.692
100	5	5	1	989.245	989.245	2.172	2.142
100	5	5	2	938.069	938.069	12.207	15.219
100	5	5	3	893.014	893.014	0.230	1.428
100	5	5	4	919.365	919.365	1.873	2.286
100	5	5	5	908.658	908.658	3.724	1.540
100	5	5	6	919.93	919.93	2.484	0.880
100	5	5	7	849.715	849.715	3.923	1.792
100	5	5	8	858.344	858.344	3.450	5.100
100	5	5	9	839.428	839.428	3.587	1.115
100	2	10	0	707.001	707.001	1.892	0.563
100	2	10	1	712.804	712.804	2.159	0.464
100	2	10	2	735.307	735.307	0.408	0.860
100	2	10	3	758.187	758.187	0.944	0.439
100	2	10	4	701.116	701.116	1.442	1.010
100	2	10	5	713.093	713.093	4.582	0.662
100	2	10	6	720.4	720.4	1.027	0.365
100	2	10	7	705.035	705.035	1.137	0.463
100	2	10	8	682.206	682.206	0.964	0.303
100	2	10	9	680.525	680.525	1.034	0.326
100	2	5	0	709.175	709.175	1.395	0.381
100	2	5	1	739.829	739.829	1.492	0.502
100	2	5	2	723.281	723.281	1.486	0.406
100	2	5	3	756.794	756.794	1.792	0.665
100	2	5	4	751.943	751.943	1.314	0.247
100	2	5	5	811.268	811.268	2.595	4.394
100	2	5	6	786.252	786.252	2.372	0.347
100	2	5	7	712.548	712.548	1.523	0.400
100	2	5	8	749.396	749.396	1.370	0.404
100	2	5	9	687.199	687.199	0.979	0.409
100	3	10	0	792.986	792.986	2.236	0.630
100	3	10	1	876.181	876.181	2.908	2.648
100	3	10	2	798.934	798.934	2.225	1.946
100	3	10	3	932.584	932.584	61.359	182.254
100	3	10	4	865.569	865.569	2.285	0.623
100	3	10	5	838.039	838.039	4.049	11.072
100	3	10	6	859.863	859.863	3.447	0.663
100	3	10	7	842.426	842.426	3.386	1.259
100	3	10	8	932.584	932.584	61.856	183.319

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
100	3	10	9	765.422	765.422	1.164	1.008
100	3	5	0	900.363	900.363	1.836	0.701
100	3	5	1	902.04	902.04	1.427	0.716
100	3	5	2	899.594	899.594	2.692	1.301
100	3	5	4	897.205	897.205	4.525	3.545
100	3	5	5	802.701	802.701	1.565	0.504
100	3	5	6	860.798	860.798	1.969	1.868
100	3	5	7	901.548	901.548	1.816	0.645
100	3	5	8	894.13	894.13	2.815	0.461
100	4	10	0	1020	1020	3.856	6.781
100	4	10	1	972.162	972.162	2.538	1.743
100	4	10	2	978.306	978.306	5.242	6.510
100	4	10	3	1039.78	1039.78	13.580	14.396
100	4	10	4	991.23	991.23	5.272	2.305
100	4	10	5	973.524	973.524	4.090	1.574
100	4	10	6	1014.34	1014.34	2.563	1.198
100	4	10	7	1011.5	1011.5	5.806	19.511
100	4	10	8	991.42	991.42	3.038	1.058
100	4	10	9	1017.75	1017.75	4.034	1.453
100	4	5	0	1080.57	1080.57	7.666	40.230
100	4	5	1	1110.54	1110.54	79.409	505.641
100	4	5	2	1105.53	1105.53	1.732	1.255
100	4	5	3	1088.86	1088.86	4.959	2.002
100	4	5	4	1056.85	1056.85	6.239	2.875
100	5	10	0	1240.62	1240.62	3.973	13.811
100	5	10	1	1184.54	1184.54	4.885	2.694
100	5	10	2	1143.63	1143.63	4.700	2.024
100	5	10	4	1091.45	1091.45	7.251	14.944
100	5	10	5	1148.67	1148.67	6.204	5.487
100	5	10	6	1169.86	1169.86	2.956	4.551
100	5	10	8	1099.12	1099.12	5.252	2.226
100	5	10	9	1044.01	1044.01	2.449	1.020
100	5	5	0	1200.65	1200.65	1.578	2.388
100	5	5	1	1344.58	1344.58	2.350	3.860
100	5	5	2	1253.95	1253.95	33.520	37.455
100	5	5	3	1216.94	1216.94	1.628	1.407
100	5	5	4	1263.94	1263.94	2.210	2.613
100	5	5	5	1233.66	1233.66	4.843	2.157
100	5	5	6	1245.75	1245.75	3.143	1.017
100	5	5	7	1184.62	1184.62	4.387	2.333
100	5	5	8	1214.32	1214.32	5.211	2.796
100	5	5	9	1250.68	1250.68	4.988	1.561
200	2	10	0	430.323	430.323	2.837	1.154
200	2	10	1	470.386	470.386	2.561	1.148
200	2	10	2	431.11	431.11	3.434	1.363
200	2	10	3	424.036	424.036	9.004	0.828
200	2	10	4	434.655	434.655	2.117	2.215
200	2	10	5	417.482	417.482	1.973	1.132
200	2	10	6	427.649	427.649	1.600	0.782
200	2	10	7	461.034	461.034	2.094	0.908
200	2	10	8	402.024	402.024	1.746	0.629
200	2	10	9	392.12	392.12	1.453	0.581
200	2	5	0	418.937	418.937	1.710	0.715
200	2	5	1	454.263	454.263	2.541	1.037
200	2	5	2	398.946	398.946	2.494	1.065
200	2	5	3	444.359	444.359	2.177	0.944
200	2	5	4	450.321	450.321	1.950	0.611
200	2	5	5	513.475	513.475	2.441	1.683
200	2	5	6	467.345	467.345	5.222	0.887
200	2	5	7	458.307	458.307	2.103	0.832
200	2	5	8	440.563	440.563	1.964	0.637
200	2	5	9	431.836	431.836	0.328	0.703
200	3	10	0	422.481	422.481	2.347	1.020
200	3	10	1	487.119	487.119	3.135	2.226
200	3	10	2	428.66	428.66	2.998	1.914
200	3	10	3	513.781	513.781	2.018	2.598
200	3	10	4	464.164	464.164	2.665	0.996

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
200	3	10	5	462.31	462.31	3.242	3.274
200	3	10	6	460.729	460.729	4.687	1.069
200	3	10	7	453.533	453.533	2.818	1.277
200	3	10	8	513.781	513.781	3.440	2.576
200	3	10	9	439.259	439.259	1.722	1.196
200	3	5	0	470.642	470.642	2.112	1.063
200	3	5	1	466.082	466.082	2.697	1.295
200	3	5	2	461.173	461.173	2.337	1.267
200	3	5	3	548.7	548.7	3.032	8.144
200	3	5	4	500.859	500.859	2.805	1.222
200	3	5	5	413.71	413.71	2.729	0.908
200	3	5	6	476.522	476.522	2.574	1.332
200	3	5	7	486.592	486.592	2.638	1.014
200	3	5	8	454.193	454.193	6.466	0.887
200	4	10	0	536.266	536.266	3.638	2.364
200	4	10	1	475.184	475.184	1.137	1.580
200	4	10	2	463.97	463.97	3.894	2.407
200	4	10	3	559.014	559.014	1.418	4.256
200	4	10	4	468.755	468.755	3.822	1.958
200	4	10	5	500.602	500.602	4.532	1.984
200	4	10	6	530.524	530.524	4.125	2.880
200	4	10	7	484.741	484.741	3.197	2.389
200	4	10	8	481.035	481.035	3.809	1.908
200	4	10	9	475.389	475.389	3.238	1.749
200	4	5	0	538.411	538.411	6.013	4130.433
200	4	5	1	537.522	537.522	3.417	3.115
200	4	5	2	556.661	556.661	1.072	2.438
200	4	5	3	508.884	508.884	3.249	1.816
200	4	5	4	546.355	546.355	5.524	2.961
200	5	10	0	580.622	580.622	3.188	2.854
200	5	10	1	539.728	539.728	4.543	2.967
200	5	10	2	525.531	525.531	4.654	2.532
200	5	10	3	562.542	562.542	12.999	16.888
200	5	10	4	497.213	497.213	5.530	16.740
200	5	10	5	544.778	544.778	5.353	4.357
200	5	10	6	562.491	562.491	43.318	48.735
200	5	10	7	562.542	562.542	9.429	17.277
200	5	10	8	550.036	550.036	3.676	2.974
200	5	10	9	466.014	466.014	3.282	1.726
200	5	5	0	526.455	526.455	1.164	1.850
200	5	5	1	620.442	620.442	3.349	2.856
200	5	5	2	578.037	578.037	0.564	3.457
200	5	5	3	571.327	571.327	2.942	2.104
200	5	5	4	572.053	572.053	1.549	2.248
200	5	5	5	564.504	564.504	5.318	2.468
200	5	5	6	570.108	570.108	3.852	1.587
200	5	5	8	523.885	523.885	4.157	6.132
200	5	5	9	534.297	534.297	4.676	1.981
200	2	10	0	572.289	572.289	3.450	1.355
200	2	10	1	584.908	584.908	3.094	1.381
200	2	10	2	580.91	580.91	3.737	2.253
200	2	10	3	589.598	589.598	0.792	0.962
200	2	10	4	568.144	568.144	2.322	2.451
200	2	10	5	577.251	577.251	1.772	1.328
200	2	10	6	578.346	578.346	1.929	0.881
200	2	10	7	584.846	584.846	2.056	1.009
200	2	10	8	541.518	541.518	1.717	0.677
200	2	10	9	532.508	532.508	1.688	0.716
200	2	5	0	562.973	562.973	2.440	0.775
200	2	5	1	586.315	586.315	2.764	1.145
200	2	5	2	560.097	560.097	2.836	1.096
200	2	5	3	595.43	595.43	2.687	1.109
200	2	5	4	615.532	615.532	0.421	0.525
200	2	5	5	655.606	655.606	25.783	29.471
200	2	5	6	624.543	624.543	6.282	0.969
200	2	5	7	596.324	596.324	2.594	0.943
200	2	5	8	597.28	597.28	2.439	0.817

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
200	2	5	9	563.391	563.391	0.668	1.185
200	3	10	0	613.557	613.557	3.499	1.247
200	3	10	1	683.163	683.163	4.398	12.124
200	3	10	2	612.226	612.226	4.202	3.799
200	3	10	3	714.252	714.252	30.010	35.743
200	3	10	4	659.395	659.395	3.494	1.284
200	3	10	5	642.614	642.614	5.174	14.079
200	3	10	6	660.957	660.957	6.509	1.660
200	3	10	7	644.967	644.967	3.927	2.037
200	3	10	8	714.252	714.252	30.385	35.530
200	3	10	9	608.159	608.159	1.917	1.824
200	3	5	0	677.564	677.564	2.493	1.561
200	3	5	1	689.419	689.419	2.923	1.787
200	3	5	2	679.725	679.725	3.088	1.714
200	3	5	4	707.412	707.412	82.361	4.636
200	3	5	5	604.313	604.313	2.615	1.032
200	3	5	6	662.897	662.897	3.892	3.264
200	3	5	7	695.753	695.753	3.016	1.350
200	3	5	8	669.528	669.528	6.645	1.080
200	4	10	0	781.15	781.15	6.599	8.308
200	4	10	1	712.016	712.016	3.227	2.658
200	4	10	2	697.41	697.41	7.159	26.191
200	4	10	3	787.011	787.011	22.655	34.308
200	4	10	4	713.895	713.895	6.623	4.537
200	4	10	5	735.285	735.285	6.522	2.944
200	4	10	6	772.189	772.189	5.251	4.097
200	4	10	7	752.304	752.304	5.956	5.349
200	4	10	8	742.825	742.825	5.541	3.054
200	4	10	9	747.236	747.236	5.042	2.558
200	4	5	1	807.408	807.408	47.798	7.141
200	4	5	2	858.192	858.192	33.222	2.522
200	4	5	3	794.932	794.932	5.277	3.027
200	5	10	1	867.801	867.801	6.567	8.061
200	5	10	2	839.964	839.964	6.769	4.015
200	5	10	5	844.825	844.825	8.876	9.721
200	5	10	6	873.012	873.012	118.478	1972.559
200	5	10	8	825.688	825.688	6.694	6.603
200	5	10	9	749.316	749.316	4.879	2.204
200	5	5	0	865.256	865.256	3.453	3.858
200	5	5	1	986.241	986.241	5.483	5.434
200	5	5	2	931.815	931.815	25.550	30.254
200	5	5	3	893.308	893.308	1.365	5.551
200	5	5	4	930.931	930.931	3.884	16.070
200	5	5	5	897.562	897.562	7.002	3.494
200	5	5	6	916.807	916.807	5.376	2.179
200	5	5	8	873.806	873.806	6.972	30.354
200	5	5	9	868.503	868.503	7.270	2.745
200	2	10	0	712.196	712.196	4.014	1.287
200	2	10	1	711.116	711.116	4.098	1.279
200	2	10	2	733.829	733.829	4.544	2.417
200	2	10	3	750.674	750.674	1.365	0.984
200	2	10	4	698.329	698.329	2.847	2.772
200	2	10	5	711.767	711.767	9.253	1.148
200	2	10	6	725.46	725.46	2.205	0.897
200	2	10	7	704.704	704.704	2.478	1.026
200	2	10	8	679.669	679.669	1.788	0.650
200	2	10	9	676.586	676.586	2.038	0.630
200	2	5	0	708.969	708.969	2.936	0.862
200	2	5	1	730.989	730.989	3.031	1.163
200	2	5	2	714.036	714.036	3.064	0.966
200	2	5	3	759.818	759.818	3.386	1.381
200	2	5	4	760.043	760.043	0.256	0.579
200	2	5	5	804.643	804.643	11.001	16.763
200	2	5	6	785.9	785.9	4.791	0.846
200	2	5	7	716.329	716.329	2.783	0.961
200	2	5	8	753.784	753.784	2.675	0.787
200	2	5	9	683.99	683.99	0.905	0.928

Table 10: Complete results (continued).

$\|\Omega\|$	m	R	instance	PH objective	CPLEX objective	PH time (sec.)	CPLEX time (sec.)
200	3	10	0	790.988	790.988	4.009	1.363
200	3	10	1	876.743	876.743	6.090	8.151
200	3	10	2	798.531	798.531	4.758	3.886
200	3	10	3	934.879	934.879	114.484	163.436
200	3	10	4	871.718	871.718	4.271	1.559
200	3	10	5	835.789	835.789	8.248	81.919
200	3	10	6	858.62	858.62	6.975	1.657
200	3	10	7	845.752	845.752	6.823	2.862
200	3	10	8	934.879	934.879	116.864	161.715
200	3	10	9	762.304	762.304	26.987	2.356
200	3	5	0	899.309	899.309	2.795	1.781
200	3	5	1	904.497	904.497	2.519	1.789
200	3	5	2	891.14	891.14	5.481	2.890
200	3	5	4	905.235	905.235	2.422	3.655
200	3	5	5	810.276	810.276	2.555	1.105
200	3	5	6	863.766	863.766	4.339	3.982
200	3	5	7	904.073	904.073	3.152	1.520
200	3	5	8	890.285	890.285	7.764	1.153
200	4	10	0	1020.82	1020.82	7.263	19.245
200	4	10	1	949.508	949.508	78.161	4.896
200	4	10	2	970.624	970.624	10.413	62.099
200	4	10	3	1029.53	1029.53	27.255	50.408
200	4	10	4	986.736	986.736	10.574	5.628
200	4	10	5	969.737	969.737	7.870	3.349
200	4	10	6	1009.77	1009.77	5.343	3.519
200	4	10	7	999.784	999.784	10.891	71.527
200	4	10	8	978.985	978.985	5.664	3.959
200	4	10	9	1018.07	1018.07	8.530	5.583
200	4	5	2	1113.48	1113.48	2.675	3.977
200	4	5	3	1080.04	1080.04	1.010	6.532
200	5	10	0	1251.88	1251.88	6.813	43.920
200	5	10	1	1173.31	1173.31	8.938	8.118
200	5	10	2	1146.43	1146.43	9.249	5.262
200	5	10	5	1131.14	1131.14	12.038	14.293
200	5	10	6	1183.08	1183.08	1.594	9.362
200	5	10	8	1112.86	1112.86	10.407	7.804
200	5	10	9	1039.64	1039.64	5.036	2.646
200	5	5	0	1197.04	1197.04	4.045	5.898
200	5	5	1	1337.7	1337.7	9.827	9.219
200	5	5	2	1270.55	1270.55	26.121	51.549
200	5	5	3	1225.28	1225.28	1.487	3.626
200	5	5	4	1278.85	1278.85	4.854	6.672
200	5	5	5	1233.25	1233.25	9.684	5.275
200	5	5	6	1246.91	1246.91	6.071	2.492
200	5	5	7	1209.05	1209.05	7.747	6.403
200	5	5	8	1205.22	1205.22	10.449	19.159
200	5	5	9	1242.72	1242.72	9.713	3.803

References

[1] J. Curry, J. Maslanik, G. Holland, J. Pinto, Applications of Aerosondes in the Arctic, Bulletin of the American Meteorological Society 85 (2004) 1855–1862.

[2] C. Corrigan, G. Roberts, M. Ramana, D. Kim, V. Ramanathan, Capturing vertical profiles of aerosols and black carbon over the Indian Ocean using autonomous unmanned aerial vehicles, Atmospheric Chemistry and Physics Discussions 7 (2007) 11429–11463.

[3] P. Tokekar, J. Vander Hook, D. Mulla, V. Isler, Sensor planning for a symbiotic UAV and UGV system for precision agriculture, IEEE Transactions on Robotics 32 (2016) 1498–1511.

[4] J. F. Shanahan, J. S. Schepers, D. D. Francis, G. E. Varvel, W. W. Wilhelm, J. M. Tringe, M. R. Schlemmer, D. J. Major, Use of remote-sensing imagery to estimate corn grain yield, Agronomy Journal 93 (2001) 583–589.

[5] D. W. Casbeer, R. W. Beard, T. W. McLain, S.-M. Li, R. K. Mehra, Forest fire monitoring with multiple small UAVs, in: Proceedings of the 2005, American Control Conference, 2005., IEEE, 2005, pp. 3530–3535.

[6] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias, A. Dias, E. Silva, Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle, in: Oceans 2009-Europe, IEEE, 2009, pp. 1–6.

[7] T. Zajkowski, S. Dunagan, J. Eilers, Small UAS communications mission, 2006. URL: https://www.fs.fed.us/eng/rsac/RS2006/presentations/zajkowski2.pdf.
[8] S. G. Manyam, K. Sundar, D. W. Casbeer, Cooperative routing for an air-ground vehicle team–exact algorithm, transformation method, and heuristics, IEEE Transactions on Automation Science and Engineering (2019).
[9] K. Sundar, S. Venkatachalam, S. G. Manyam, Path planning for multiple heterogeneous unmanned vehicles with uncertain service times, in: Unmanned Aircraft Systems (ICUAS), 2017 International Conference on, IEEE, 2017, pp. 480–487.
[10] T. Shima, S. Rasmussen, UAV cooperative decision and control: challenges and practical approaches, SIAM, 2009.
[11] Y. Liu, Z. Liu, J. Shi, G. Wu, C. Chen, Optimization of Base Location and Patrol Routes for Unmanned Aerial Vehicles in Border Intelligence, Surveillance, and Reconnaissance, Journal of Advanced Transportation 2019 (2019).
[12] S. M. Adams, C. J. Friedland, A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management, in: 9th International Workshop on Remote Sensing for Disaster Response, 2011, p. 8.
[13] S. Rajan, K. Sundar, N. Gautam, Routing problems for reconnaissance patrolling missions, in: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2019, pp. 213–220.
[14] J. Balasubramanian, I. Grossmann, Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty, Industrial & engineering chemistry research 43 (2004) 3695–3713.
[15] K. Sundar, S. Rathinam, Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots, IEEE Transactions on Automation Science and Engineering 11 (2013) 287–294.
[16] K. Sundar, S. Rathinam, Algorithms for heterogeneous, multiple depot, multiple unmanned vehicle path planning problems, Journal of Intelligent & Robotic Systems 88 (2017) 513–526.
[17] D. Levy, K. Sundar, S. Rathinam, Heuristics for routing heterogeneous unmanned vehicles with fuel constraints, Mathematical Problems in Engineering 2014 (2014).
[18] A. Otto, N. Agatz, J. Campbell, B. Golden, E. Pesch, Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey, Networks 72 (2018) 411–458.
[19] M. Gendreau, G. Laporte, R. Séguin, Stochastic vehicle routing, European Journal of Operational Research 88 (1996) 3–12.
[20] P. Jaillet, A priori solution of a traveling salesman problem in which a random subset of the customers are visited, Operations research 36 (1988) 929–936.
[21] R. L. Carraway, T. L. Morin, H. Moskowitz, Generalized dynamic programming for stochastic combinatorial optimization, Operations Research 37 (1989) 819–829.
[22] F. A. Tillman, The multiple terminal delivery problem with probabilistic demands, Transportation Science 3 (1969) 192–204.
[23] M. Dror, G. Laporte, P. Trudeau, Vehicle routing with stochastic demands: Properties and solution frameworks, Transportation science 23 (1989) 166–176.
[24] D. Bertsimas, Probabilistic combinatorial optimization problems, Ph.D. thesis, Massachusetts Institute of Technology, 1988.
[25] P. Jaillet, A. Odoni, Vehicle routing: Methods and studies, chapter the probabilistic vehicle routing problem, 1988.
[26] A. Jezequel, Probabilistic vehicle routing problems, Ph.D. thesis, Massachusetts institute of technology, 1985.
[27] S. M. Shavarani, M. G. Nejad, F. Rismanchian, G. Izbirak, Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of amazon prime air in the city of san francisco, The International Journal of Advanced Manufacturing Technology 95 (2018) 3141–3153.
[28] M. Torabbeigi, G. J. Lim, S. J. Kim, Drone delivery schedule optimization considering the reliability of drones, in: 2018 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2018, pp. 1048–1053.
[29] S. Venkatachalam, K. Sundar, S. Rathinam, A two-stage approach for routing multiple unmanned aerial vehicles with stochastic fuel consumption, Sensors 18 (2018) 3756.
[30] M. Liu, X. Liu, M. Zhu, F. Zheng, Stochastic drone fleet deployment and planning problem considering multiple-type delivery service, Sustainability 11 (2019) 3871.
[31] G. Perboli, L. Gobbato, F. Maggioni, A progressive hedging method for the multi-path travelling salesman problem with stochastic travel times, IMA Journal of Management Mathematics 28 (2017) 65–86.
[32] M. Jalilvand, M. Bashiri, E. Nikzad, An effective progressive hedging algorithm for the two-layers time window assignment vehicle routing problem in a stochastic environment, Expert Systems with Applications 165 (2021) 113877.
[33] A. M. Campbell, L. W. Clarke, M. W. Savelsbergh, Inventory routing in practice, in: The vehicle routing problem, SIAM, 2002, pp. 309–330.
[34] L. M. Hvattum, A. Løkketangen, G. Laporte, Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic, Transportation Science 40 (2006) 421–438.
[35] J.-P. Watson, D. L. Woodruff, Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems, Computational Management Science 8 (2011) 355–370.
[36] R. T. Rockafellar, R. J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of operations research 16 (1991) 119–147.
[37] A. Shapiro, Analysis of stochastic dual dynamic programming method, European Journal of Operational Research 209

(2011) 63–72.

[38] S. Atakan, S. Sen, A progressive hedging based branch-and-bound algorithm for mixed-integer stochastic programs, Computational Management Science 15 (2018) 501–540.

[39] Y. Fan, C. Liu, Solving stochastic transportation network protection problems using the progressive hedging-based method, Networks and Spatial Economics 10 (2010) 193–208.

[40] O. Listes, R. Dekker, A scenario aggregation–based approach for determining a robust airline fleet composition for dynamic capacity allocation, Transportation Science 39 (2005) 367–382.

[41] A. Løkketangen, D. L. Woodruff, Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming, Journal of Heuristics 2 (1996) 111–128.

[42] S. W. Wallace, W. T. Ziemba, Applications of stochastic programming, SIAM, 2005.

[43] L. E. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, American Journal of mathematics 79 (1957) 497–516.

[44] J. R. Birge, F. Louveaux, Introduction to stochastic programming, Springer Science & Business Media, 2011.

1. The article develops a systematic way to formulate stochastic optimization problems in information gathering or patrolling missions.

2. A two-stage stochastic programming formulation that is solved using a Progressive Hedging algorithm

3. Extensive computational experiments that corroborate the efficacy of the proposed Progressive Hedging algorithm to solve a large number of test instances.

The authors declare that they have no competing interests.