

LA-UR-18-29794 (Accepted Manuscript)

## A Two-Stage Approach for Routing Multiple Unmanned Aerial Vehicles with Stochastic Fuel Consumption

Venkatachalam, Saravanan  
Sundar, Kaarthik  
Rathinam, Sivakumar

Provided by the author(s) and the Los Alamos National Laboratory (2019-02-05).

**To be published in:** Sensors

**DOI to publisher's version:** 10.3390/s18113756

**Permalink to record:** <http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-18-29794>

**Disclaimer:**

Approved for public release. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Article

# A Two-Stage Approach for Routing Multiple Unmanned Aerial Vehicles with Stochastic Fuel Consumption

Saravanan Venkatachalam <sup>1,\*</sup> , Kaarthik Sundar <sup>2</sup>  and Sivakumar Rathinam <sup>3</sup> 

<sup>1</sup> Department of Industrial and Systems Engineering, Wayne State University, Detroit, MI 48202, USA

<sup>2</sup> Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA; kaarthik01sundar@gmail.com

<sup>3</sup> Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA; srathinam@tamu.edu

\* Correspondence: saravanan.v@wayne.edu; Tel.: +1-313-577-1821

Received: 29 September 2018 ; Accepted: 31 October 2018 ; Published: 3 November 2018



**Abstract:** The past decade has seen a substantial increase in the use of small unmanned aerial vehicles (UAVs) in both civil and military applications. This article addresses an important aspect of refueling in the context of routing multiple small UAVs to complete a surveillance or data collection mission. Specifically, this article formulates a multiple-UAV routing problem with the refueling constraint of minimizing the overall fuel consumption for all the vehicles as a two-stage stochastic optimization problem with uncertainty associated with the fuel consumption of each vehicle. The two-stage model allows for the application of sample average approximation (SAA). Although the SAA solution asymptotically converges to the optimal solution for the two-stage model, the SAA run time can be prohibitive for medium- and large-scale test instances. Hence, we develop a tabu search-based heuristic that exploits the model structure while considering the uncertainty in fuel consumption. Extensive computational experiments corroborate the benefits of the two-stage model compared to a deterministic model and the effectiveness of the heuristic for obtaining high-quality solutions.

**Keywords:** vehicle routing; unmanned aerial vehicles; fuel constraints; two-stage stochastic optimization; sample average approximation; tabu search

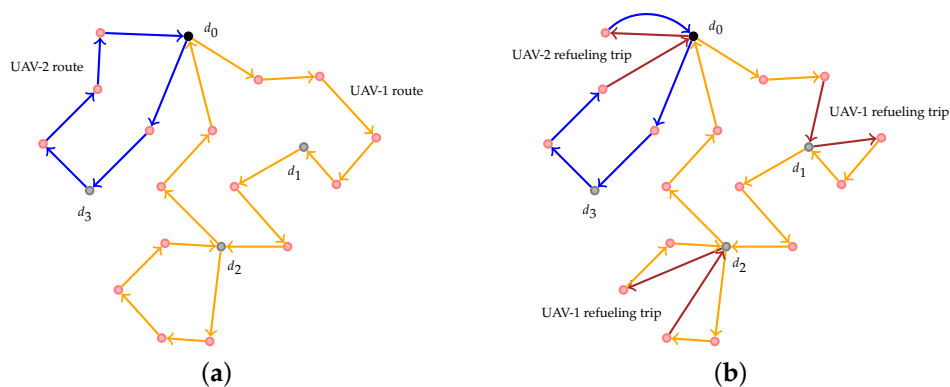
---

## 1. Introduction

Advances in sensing, robotics, and wireless sensor networks have enabled the use of small unmanned aerial vehicles (UAVs) in environmental sensing applications such as crop monitoring [1–3], forest fire monitoring [4], and ecosystem management [5,6], as well as civil security applications such as border surveillance [7,8] and disaster management [9]. In military applications, small UAVs are preferred as they can be hand launched without the need for a runway [10]. More recently, a variety of studies have indicated that small UAVs have a huge potential for package delivery, last-mile delivery, and emergency response [11], and studies have also indicated that the use of small UAVs in emergency response applications would lead to significant cost reductions [12,13] because of their inexpensive operation, maintenance, and labor costs. Moreover, in the emergency response context, they also have the potential to save lives by transporting and delivering much-needed food and water supplies over any terrain. Worldwide, corporate companies such as Amazon and Google have begun field-testing their UAV capabilities through initiatives such as Amazon Prime Air [13] and Project Wing [14], respectively.

Despite the advantages of small UAV platforms, they come with other resource constraints due to their size and limited payload capacity. Small UAVs typically have fuel constraints and are therefore

required to make one or more refueling stops in a long surveillance or data gathering application. For instance, consider the following problem setup involving multiple small UAVs that are supposed to visit a set of targets and collect data from them. We are given a set of depots and a set of targets. The depots are refueling sites, and a team of homogeneous small UAVs is initially stationed at one of the depots. It is safe to assume that any UAV is refueled to its capacity when it visits a depot. The fuel a vehicle consumes to travel between any pair of targets/depots is uncertain, and the description of this uncertainty is known via numerous samples. In this context, the following two-stage Fuel-Constrained Multiple-UAV Routing Problem (FCMURP) naturally arises: In the first stage, the FCMURP aims to compute a route for each UAV that starts and ends at the depot where all the vehicles are initially stationed, such that each target is visited at least once by some vehicle and no vehicle runs out of fuel as it traverses its route. If a vehicle does not have enough fuel, it can make a refueling stop at any depot. This fuel restriction is imposed in the first stage using a nominal value for the fuel consumed by any vehicle as it travels between a pair of targets/depots. The first-stage routing decisions for each vehicle are referred to as ‘here-and-now’ decisions in the parlance of two-stage stochastic programming; these decisions are typically made before the realization of the uncertainty is known, which conforms to most realistic settings where decisions have to be made in the face of uncertain parameters. In the second stage, additional refueling trips are added to the first-stage routes to ensure feasibility for the realization of the fuel consumption. The second-stage decisions are referred to as ‘the recourse actions/decisions’ because these decisions are typically made after the realization of uncertainties. The objective of the FCMURP is to minimize the sum of the travel distances for all the vehicles (first-stage objective function) and the expected travel distance for the additional refueling trips (second-stage objective function). An illustration of feasible first-stage and second-stage solutions is shown in Figure 1.



**Figure 1.** Illustration of a feasible first-stage solution and recourse action for a particular realization of the uncertainty. (a) A feasible first-stage solution to the two-stage FCMURP with two UAVs; the first-stage routes for the UAVs correspond to the ‘here-and-now’ decisions. (b) A feasible recourse action for the two-stage FCMURP with two UAVs; for a given realization of the fuel consumed by any vehicle to traverse an edge, the refuel trips (colored in brown) are added to the first-stage solution.

### 1.1. Related Work

The FCMURP is NP-hard because it is a generalization of the multiple traveling salesman problem (MTSP). The literature contains many variants of vehicle routing problems (VRPs) for UAVs with fuel constraints. The interested reader is referred to [15] for an overview of the formulations and algorithms for many of these variants. A single-vehicle deterministic variant of the FCMURP for ground vehicles was first introduced by Khuller et al. [16], who develop an approximation algorithm for the case where the travel costs are symmetric and satisfy the triangle inequality. Khuller et al. assume that the minimum fuel required to travel from any target to its nearest depot is equal to at

most  $F\alpha/2$  units, where  $\alpha$  is a constant in the interval  $[0, 1)$  and  $F$  is the fuel capacity of the vehicle. This is a reasonable assumption since, in any case, one cannot have a feasible tour if there is a target that cannot be visited from any of the depots. Based on this assumption, Khuller et al. [16] present a  $(3(1 + \alpha))/(2(1 - \alpha))$  approximation algorithm for the problem. Variants of the deterministic FCMURP in the context of routing UAVs are addressed in [17–21]. Specifically, Ref. [17] was the first paper to address the problem in the context of UAVs. The authors formulate the single-vehicle variant as a mixed-integer linear program (MILP) and present  $k$ -opt-based exchange heuristics to obtain feasible solutions within 7% of the optimal solution, on average. The MILP formulations are solved using off-the-shelf commercial solvers to obtain an optimal solution to the single-vehicle variant, and the authors also observe that the solvers are unable to handle their formulation for instances with more than 25 targets. Sundar et al. [18] extend the approximation algorithm in [16] to the asymmetric case and present heuristics to solve this case. Levy et al. [19] present variable neighborhood search heuristics for the deterministic FCMURP with heterogeneous vehicles, i.e., vehicles with different fuel capacities. More recently, Ref. [20] develop an approximation algorithm and heuristics for the deterministic FCMURP. Mitchell et al. [20] extend the MILP proposed in [17] to the multiple-vehicle setting. Like [17], they also report that the off-the-shelf mixed-integer solvers had difficulty computing an optimal solution. More recently, Sundar et al. [22,23] analytically and empirically analyze different MILP formulations for the deterministic FCMURP and develop additional valid inequalities for these.

Another line of work examines the deterministic FCMURP in the context of electric vehicles, which is referred to as the green vehicle routing problem (GVRP). The GVRP was first introduced in [24]. The authors of [24] propose an MILP formulation and several heuristics to tackle the GVRP. Since then, many algorithms have been developed to solve the deterministic capacitated and uncapacitated variants of the GVRP; an excellent survey of this work is presented in [25]. Other variants of the VRP that are closely related to the deterministic version of the FCMURP are the distance-constrained VRP [26–30] and the orienteering problem [31,32]. All these variants are well suited for ground vehicles. The distance-constrained VRP is a special case of the FCMURP with a single vehicle and a single depot. The FCMURP is also quite different and more general than the orienteering problem, which focuses on maximizing the number of targets visited by a vehicle subject to its fuel constraints. To the best of our knowledge, fuel-constrained, multiple-vehicle routing problems considering uncertainty in the fuel consumption of the vehicles have not been explored in the framework of stochastic optimization in the literature. Hence, this article can be considered as a first step in that direction. We also remark that the framework and the model that we present for the FCMURP in this article are fairly general and can accommodate many other practical constraints that arise in typical surveillance missions involving multiple UAVs. We use the sample average approximation (SAA) approach [33,34] to solve the two-stage stochastic program and obtain statistical estimates of the upper and lower bounds for the optimal FCMURP solution. The SAA approach is a Monte Carlo simulation-based sampling technique, in which the expected value of the objective function is approximated using a finite number of scenarios; nevertheless, SAA is computationally expensive for larger problem instances. To address the scalability of the SAA algorithm, we also propose a tabu search heuristic [35,36] to compute good-quality suboptimal solutions with relatively less computational effort. We use a tabu search heuristic because of its success with stochastic variants of many VRPs; the interested reader is referred to [37–42] for tabu searches applied to general applications and VRPs, respectively. Specifically, the tabu search algorithm presented in this article is adapted from the work in [39,43].

## 1.2. Contributions

The following are the main contributions of this article: (i) the FCMURP with uncertainty in the fuel consumption of all vehicles is modeled as a two-stage stochastic program with random recourse; (ii) SAA is used to obtain statistical estimates of the lower and upper bounds for the optimal solution of the two-stage stochastic program; (iii) a tabu search heuristic for finding suboptimal solutions for

the two-stage stochastic model is proposed; and finally; (iv) the performance achieved by SAA and the tabu search heuristic is corroborated through extensive computational experiments.

The rest of this paper is organized as follows. Notation is introduced in Section 2, and the two-stage stochastic programming model formulation is presented in Section 3. Details about SAA using an exact method and a heuristic solution are presented in Sections 4 and 5, respectively. Computational results for randomly generated instances are presented in Section 6. Finally, the paper concludes in Section 7.

## 2. Notation

Let  $T = \{t_1, \dots, t_n\}$  denote the set of targets, let  $d_0$  denote the depot where  $m$  homogeneous UAVs, each with fuel capacity  $F$ , are initially stationed, let  $\bar{D} = \{d_1, \dots, d_k\}$  denote the set of additional depots or refueling sites, and let  $D = \bar{D} \cup \{d_0\}$ . All the  $m$  UAVs initially stationed at the depot  $d_0$  are assumed to be fueled to capacity. The FCMURP is defined on a directed graph  $G = (V, E)$ , where  $V = T \cup D$  denotes the set of vertices and  $E$  denotes the set of edges joining any pair of vertices. We assume that  $G$  does not contain any self-loops. For each edge  $(i, j) \in E$ , we let  $c_{ij}$  and  $\hat{f}_{ij}$  represent the travel cost and the nominal fuel that will be consumed by any UAV traversing the edge  $(i, j)$ . We remark that  $\hat{f}_{ij}$  is directly computed using the length of the edge  $(i, j)$  and the fuel economy of the UAVs. Additional notation that will be used in the mathematical formulation is as follows: for any set  $S \subset V$ ,  $\delta^+(S) = \{(i, j) \in E : i \in S, j \notin S\}$  and  $\delta^-(S) = \{(i, j) \in E : i \notin S, j \in S\}$ . When  $S = \{i\}$ , we shall simply write  $\delta^+(i)$  and  $\delta^-(i)$  instead of  $\delta^+(\{i\})$  and  $\delta^-(\{i\})$ , respectively.

The notation introduced next is for describing the uncertainty associated with the vehicles' fuel consumption. Let  $f$  denote a discrete random variable vector representing the fuel consumed by any UAV to traverse any edge in  $E$ . The vector  $f$  has  $|E|$  components, one for each edge, and the random variable in the vector  $f$  corresponding to edge  $(i, j)$  is denoted by  $f_{ij}$ . Let  $\Omega$  denote the set of scenarios for  $f$ , where  $\omega \in \Omega$  represents a random event or realization of the random variable  $f$  with a probability of occurrence  $p(\omega)$ . We use  $f_{ij}(\omega)$  to denote the fuel consumed by any UAV when traversing the edge  $(i, j)$ , and  $f(\omega)$  to denote the random vector for the realization  $\omega \in \Omega$ . Finally, we use  $\mathbb{E}$  to denote the expectation operator, i.e.,  $\mathbb{E}(\alpha) = \sum_{\omega \in \Omega} p(\omega)\alpha$ . Table 1 lists all the notations introduced in this section for ease of reading. In the next section, we formulate the FCMURP as a two-stage stochastic program using the notation introduced in this section.

**Table 1.** Table of notations.

Symbol	Explanation
$T = \{t_1, \dots, t_n\}$	set of $n$ targets
$d_0$	depot where $m$ UAVs are initially stationed
$\bar{D} = \{d_1, \dots, d_k\}$	additional depots/refueling sites
$D = \bar{D} \cup \{d_0\}$	set of depots
$F$	fuel capacity of each UAV
$G = (V, E)$	directed graph with $V = T \cup D$ on which FCMURP is formulated
$c_{ij}$	travel cost for the edge $(i, j) \in E$
$\hat{f}_{ij}$	nominal fuel consumed by the UAV to traverse the edge $(i, j) \in E$
$f$	random variable vector representing the fuel consumed by any UAV
$\Omega$	set of scenarios for $f$
$\omega \in \Omega$	realization of random variable $f$
$p(\omega)$	probability of occurrence of $\omega$
$\mathbb{E}$	expectation operator

## 3. Mathematical Formulation

The first-stage decision variables in the stochastic program are used to compute the initial set of routes for each of the UAVs such that every target is visited at least once by some UAV and no UAV ever runs out of fuel as it traverses its route. The fuel constraint for each UAV in the first-stage

is enforced using the nominal fuel consumption value  $\hat{f}_{ij}$  for each edge  $(i, j) \in E$ . For a realization  $\omega \in \Omega$ , the second-stage decision variables are used to compute the refueling trips that must be added to the first-stage routes based on the realized values of  $f_{ij}(\omega)$  for all  $(i, j) \in E$ .

Specifically, the first-stage decision variables are as follows: each edge  $(i, j) \in E$  is associated with a variable  $x_{ij}$  that equals 1 if the edge  $(i, j)$  is traversed by some UAV, and 0 otherwise. We let  $x \in \{0, 1\}^{|E|}$  denote the vector of all decision variables  $x_{ij}$ . There is also a flow variable  $z_{ij}$  associated with each edge  $(i, j) \in E$  that denotes the total nominal fuel consumed by any vehicle as it starts from depot  $i$  and reaches the vertex  $j$ . Additionally, for any  $A \subseteq E$ , we let  $x(A) = \sum_{(i,j) \in A} x_{ij}$ . Analogous to the variable  $x_{ij}$  in the first stage, we define a binary variable  $y_{ij}(\omega)$  for each edge  $(i, j) \in E$ . The variables  $y_{ij}(\omega)$  are used to define the refueling trips needed for any vehicle when the route defined by  $x$  is not feasible for the realization  $\omega$ . Similarly,  $v_{ij}(\omega)$  is a flow variable analogous to  $z_{ij}$  for every  $(i, j) \in E$  and  $\omega \in \Omega$ . Additional second-stage variables  $q_{ij}(\omega)$  for each  $(i, j) \in E$  are used to compute the cost of a refueling trip for the realization  $\omega$ . Finally, for every  $i, j \in V$  and  $\omega \in \Omega$ , we let  $\hat{d}$  be a depot in the set  $D$  that minimizes the overall refueling trip between  $i$  and  $j$ , i.e.,  $\hat{d} = \operatorname{argmin}_d f_{id}(\omega) + f_{dj}(\omega)$ . We remark that the dependence of  $\hat{d}$  on  $(i, j) \in E$  and  $\omega$  is not explicitly shown and we leave it to the reader to understand the dependence from the context.

### 3.1. Objective Function

The objective function for the two-stage stochastic programming model for the FCMURP is the sum of the first-stage travel cost and the expected second-stage recourse cost. The second-stage recourse cost for a realization  $\omega \in \Omega$  of the fuel consumption of the vehicles is the cost of the additional refueling trips that are required for the realization  $\omega$ . The recourse cost is a function of the first-stage routing decision  $x$  and the realization  $\omega$ . Letting the recourse cost be denoted by  $\beta(x, f(\omega))$ , the objective function for the two-stage stochastic optimization problem is given by

$$\min C, \text{ where } C \triangleq \sum_{(i,j) \in E} c_{ij}x_{ij} + \mathbb{E}_{\Omega} [\beta(x, f)] = \sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{\omega \in \Omega} p(\omega)\beta(x, f(\omega)). \quad (1)$$

### 3.2. First-Stage Routing Constraints

The constraints for the first stage enforce the routing constraints, i.e., the requirements that each target in  $T$  should be visited at least once by some vehicle and that each vehicle never runs out of fuel as it traverses its route. In the first stage, the fuel constraint is enforced using the nominal value of fuel consumed by any vehicle to traverse any edge  $(i, j) \in E$ . The first-stage routing constraints are as follows:

$$x(\delta^+(d)) = x(\delta^-(d)) \quad \forall d \in D \setminus \{d_0\}, \quad (2a)$$

$$x(\delta^+(d_0)) = m, \quad (2b)$$

$$x(\delta^-(d_0)) = m, \quad (2c)$$

$$x(\delta^+(S)) \geq 1 \quad \forall S \subset V \setminus \{d_0\} : S \cap T \cap D \neq \emptyset, \quad (2d)$$

$$x(\delta^+(i)) = 1 \text{ and } x(\delta^-(i)) = 1 \quad \forall i \in T, \quad (2e)$$

$$\sum_{j \in V} z_{ij} - \sum_{j \in V} z_{ji} = \sum_{j \in V} \hat{f}_{ij}x_{ij} \quad \forall i \in T, \quad (2f)$$

$$z_{di} = \hat{f}_{di}x_{di} \quad \forall i \in V, d \in D, \quad (2g)$$

$$0 \leq z_{ij} \leq Fx_{ij} \quad \forall (i, j) \in E, \quad (2h)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E. \quad (2i)$$

Constraint (2a) forces the in-degree and out-degree of each refueling station to be equal. Constraints (2b) and (2c) ensure that all the UAVs leave and return to depot  $d_0$ , where  $m$  is the number of UAVs. Constraint (2d) ensures that a feasible solution is connected. For each target  $i$ ,

the pair of constraints in (2e) require that some UAV visits the target  $i$ . Constraint (2f) eliminates subtours of the targets and defines the flow variables  $z_{ij}$  for each edge  $(i, j) \in E$  using the nominal fuel consumption values  $\hat{f}_{ij}$ . Constraints (2g) and (2h) together impose the fuel constraints on the routes for all the UAVs. Finally, constraint (2i) imposes binary restrictions on the decision variables  $x_{ij}$ .

### 3.3. Second-Stage Constraints

The second-stage model for a fixed  $x$  and  $f(\omega)$  is given as follows:

$$\beta(x, f(\omega)) = \min \sum_{i \in T, d \in D} \{c_{id}q_{id}(\omega) + c_{di}q_{di}(\omega)\} - \sum_{i, j \in T} c_{ij}q_{ij}(\omega) \quad (3a)$$

subject to:

$$\sum_{j \in V} v_{ij}(\omega) - \sum_{j \in V} v_{ji}(\omega) = \sum_{j \in V} f_{ij}(\omega)y_{ij}(\omega) \quad \forall i \in T, \quad (3b)$$

$$v_{di}(\omega) = f_{di}(\omega)y_{id}(\omega) \quad \forall i \in V, d \in D, \quad (3c)$$

$$0 \leq v_{ij}(\omega) \leq Fy_{ij}(\omega) \quad \forall (i, j) \in E, j \in T, \quad (3d)$$

$$y_{i\hat{d}}(\omega) + y_{\hat{d}j}(\omega) \geq 2(x_{ij} - y_{ij}(\omega)) \quad \forall (i, j) \in E, \quad (3e)$$

$$y_{ij}(\omega) \leq x_{ij} \quad \forall (i, j) \in T, \quad (3f)$$

$$q_{id}(\omega) \geq y_{id}(\omega) - x_{id} \quad \forall i \in T, d \in D, \quad (3g)$$

$$q_{di}(\omega) \geq y_{di}(\omega) - x_{di} \quad \forall i \in T, d \in D, \quad (3h)$$

$$q_{ij}(\omega) = x_{ij}(1 - y_{ij}(\omega)) \quad \forall i, j \in T, \quad (3i)$$

$$x_{id} \leq y_{id}(\omega) \quad \forall i \in T, d \in D, \quad (3j)$$

$$x_{di} \leq y_{di}(\omega) \quad \forall i \in T, d \in D, \quad (3k)$$

$$\sum_{d \in D} y_{id}(\omega) \leq 1 \quad \forall i \in T, \quad (3l)$$

$$0 \leq q_{ij}(\omega), y_{ij}(\omega) \in \{0, 1\}, \quad \forall (i, j) \in E. \quad (3m)$$

The objective function (3a) minimizes the total refueling trips for a given scenario and removes the cost of the edges between the targets which are used in the first stage but not in the second stage due to the additional refueling edges added in the second stage. Constraints (3b)–(3d) are analogous to constraints (2f)–(2h); they prevent subtours of the targets and impose the fuel constraints when the realization of the fuel consumption is given by  $\omega$ . Constraint (3e) defines the actual recourse action for any UAV, i.e., given a realization  $\omega$ , if the route defined by  $x$  is not feasible for  $\omega$ , this constraint allows for adding refueling visits to the route without altering the sequence in which each UAV visits the targets. Furthermore, if the route corresponding to any UAV obtained using the first-stage variables  $x$  includes an edge  $(i, j)$ , we force the refueling-site visit to occur at depot  $\hat{d}$ , where  $\hat{d}$  is precomputed using the vertices  $i, j \in V$ . Constraint (3f) ensures that the sequence of target visits for each UAV is not altered by the recourse action. Constraints (3g)–(3l) ensure that the recourse cost of additional visits to refueling stations is captured by the objective function (3a). Finally, constraint (3m) imposes the binary and continuous restrictions on the second-stage variables. It should be noted that due to constraint (3i), the second-stage model is a mixed-integer nonlinear model.

### 3.4. Tightening the Two-Stage Stochastic Formulation

In this section, we present the results we use to strengthen the constraints in the first and second stages of the formulation presented in the previous section. We say that a constraint is strengthened if it eliminates fractional solutions to the two-stage stochastic formulation of the FCMURP without removing any feasible FCMURP solutions. The following proposition strengthens the inequalities (2h) and (3d). Since the results are similar for (2h) and (3d), we present the proposition only for (2h).

**Proposition 1.** For any  $i, j, k \in V$ , if  $\hat{f}_{ij} + \hat{f}_{jk} \geq \hat{f}_{ki}$ , then the inequalities in (2h) can be strengthened as follows:

$$z_{ij} \leq (F - t_j)x_{ij} \quad \forall j \in T, (i, j) \in E, \quad (4a)$$

$$z_{id} \leq Fx_{id} \quad \forall i \in V \text{ and } d \in D, \quad (4b)$$

$$z_{ij} \geq (s_i + \hat{f}_{ij})x_{ij} \quad \forall i \in T, (i, j) \in E, \quad (4c)$$

where  $t_i = \min_{d \in D} \hat{f}_{id}$  and  $s_i = \min_{d \in D} \hat{f}_{di}$ .

**Proof.** When  $j \in D$  or  $i \in D$ , constraints (2h) and (4b) specify the values of  $z_{id}$  and  $z_{di}$ , respectively. When both  $i, j \in D$ , constraint (2h) bounds the value of  $z_{ij}$ . Hence, we only need to discuss the case where  $i, j \in T$ . When  $x_{ij} = 1$ , the total fuel consumed by any vehicle that traverses the edge  $(i, j)$  cannot be greater than  $(F - t_j)$ , where  $t_j$  is the minimum amount of nominal fuel required by any vehicle to reach a refueling station or the depot from target  $j$ . Therefore, constraint (4a) strengthens the upper bound of  $z_{ij}$  in (2h). Similarly, any vehicle that traverses edge  $(i, j)$  consumes at least  $(s_i + \hat{f}_{ij})$  fuel. As a result, constraint (4c) strengthens the lower bound of  $z_{ij}$  in (2h).  $\square$

By virtue of Proposition 1, throughout the rest of the article, we shall assume that constraints (2h) and (3d) are replaced by their strengthened counterparts.

**Proposition 2.** Using a new set of variables  $0 \leq \widehat{xy}_{ij}(\omega) \leq 1$ , constraint (3i) can be linearized using the following set of constraints.

$$q_{ij}(\omega) = x_{ij} - \widehat{xy}_{ij}(\omega) \quad \forall i, j \in T, \quad (5a)$$

$$\widehat{xy}_{ij}(\omega) \leq x_{ij} \quad \forall i, j \in T, \quad (5b)$$

$$\widehat{xy}_{ij}(\omega) \leq y_{ij}(\omega) \quad \forall i, j \in T, \quad (5c)$$

$$\widehat{xy}_{ij}(\omega) \geq x_{ij} + y_{ij}(\omega) - 1 \quad \forall i, j \in T. \quad (5d)$$

**Proof.** See [44].  $\square$

Using Proposition 2, the second-stage constraints in Section 3.3 can be reformulated as linear constraints with binary and continuous variables, making the two-stage stochastic program an MILP. From here on, unless otherwise stated, we shall assume that the second-stage constraints are linear.

#### 4. Solution Approach: Sample Average Approximation (SAA)

In this section, we present an approach to solving the two-stage stochastic program for the FCMURP presented in Section 3 using sampling. Although the mathematical formulation of the FCMURP presented in Section 3 is an MILP, the number of realizations of the uncertain fuel consumption is typically very large, and hence, it is not computationally viable to directly input the formulation to an off-the-shelf commercial MILP solver. Nevertheless, a wealth of recent theoretical and empirical work in the literature has shown that an SAA approach can often accurately solve stochastic programs when the number of scenarios (realizations of the uncertainty) is prohibitively large [34,45–47]. For the sake of completeness, we present details about SAA. The SAA of the two-stage stochastic program approximates the objective function  $C$  in (1) by its sample average approximation  $C_\Gamma$  given by

$$C_\Gamma = \sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{\omega \in \Gamma} \frac{\beta(x, f(\omega))}{|\Gamma|}, \quad (6)$$

where  $\Gamma \subset \Omega$ , with  $\Omega$  denoting the sample space of realizations for  $f$ . The function  $C_\Gamma$  is known to be an unbiased estimator of  $C$  [48] for all feasible solutions  $x$ , i.e.,  $\mathbb{E}C_\Gamma(x) = C(x)$  for every feasible solution  $x$ . Hence, the approximating problem is now given by

$$v_\Gamma = \min \{C_\Gamma : (2) \text{ and } (3)\}. \quad (7)$$

If  $C^*$  denotes the optimal solution to the full two-stage stochastic program formulated in Section 3, then the value of  $v_\Gamma$  will approach  $C^*$  as the number of scenarios considered in the sample average problem,  $|\Gamma|$ , approaches  $|\Omega|$  [49]. However, the solution  $v_\Gamma$  is biased in the sense that  $\mathbb{E}v_\Gamma \leq C^*$  [48]. In the forthcoming sections, we detail procedures to obtain statistical estimates of lower and upper bounds for the objective value of the full two-stage stochastic program,  $C^*$ , using the solution obtained by solving multiple small problems in (7) for different sample sets  $\Gamma$ .

#### 4.1. Statistical Estimate of a Lower Bound for $C^*$

From the above, the lower bound for  $C^*$  is given by  $\mathbb{E}v_\Gamma$ . We note that the value  $v_\Gamma$  is a random variable, as it depends on the random sample  $\Gamma \subset \Omega$ . Nevertheless, a confidence interval for the value of  $v_\Gamma$  can be computed using the following procedure, which was suggested in [48]. First,  $N$  independent samples  $\Gamma_1, \Gamma_2, \dots, \Gamma_N$  with the same cardinality ( $|\Gamma_k| = M$ ) are drawn from the same distribution as that of  $\Omega$ . For each of these samples, the associated SAA problem in (7) is solved. The objective values of the associated SAA problems,  $v_{\Gamma_1}, v_{\Gamma_2}, \dots, v_{\Gamma_N}$  are all independent and identically distributed with the mean  $\mathcal{L}$  and standard error  $s_{\mathcal{L}}$  given by

$$\mathcal{L} = \frac{1}{N} \sum_{k=1}^N v_{\Gamma_k} \quad \text{and} \quad s_{\mathcal{L}} = \frac{1}{N-1} \sum_{k=1}^N (v_{\Gamma_k} - \mathcal{L})^2. \quad (8)$$

#### 4.2. Statistical Estimate of an Upper Bound for $C^*$

To construct an upper bound for  $C^*$ , we first observe that for each candidate solution  $x_k^*$  to the SAA problem in (7) with  $|\Gamma_k| = M$ ,

$$x_k^* = \operatorname{argmin}\{C_{\Gamma_k} : (2) \text{ and } (3)\},$$

and that the value of  $C(x_k^*)$  is the *actual* cost of the objective function when the first-stage routes are specified by  $x_k^*$ . Similar to  $v_\Gamma$ , the value  $C(x_k^*)$  is a random variable since it depends on the sample set  $\Gamma_k$ . Furthermore, this quantity  $C(x_k^*)$  is undoubtedly larger than  $C^*$ . Hence, each candidate solution to the  $N$  SAA problems can be used to obtain a statistical estimate of the upper bound using the following procedure. We first take a sample  $\Lambda \subset \Omega$  (where  $\Lambda$  is different from the samples  $\Gamma_k$  used to obtain the lower bound) and compute an estimate of the upper bound  $C(x_k^*)$  for the candidate solution  $x_k^*$  as

$$w_\Lambda(x_k^*) = \sum_{(i,j) \in E} c_{ij}x_{ij}^* + \sum_{\omega \in \Lambda} \frac{\beta(x_k^*, f(\omega))}{|\Lambda|}. \quad (9)$$

$w_\Lambda(x_k^*)$  is then computed for each of the  $N$  candidate SAA solutions, and the  $x^*$  that corresponds to the least upper bound of all of the  $w_\Lambda(x_k^*)$  is chosen as the feasible solution, i.e.,  $x^* = \operatorname{argmin}\{w_\Lambda(x) : x \in x_1^*, x_2^*, \dots, x_N^*\}$ . Typically,  $|\Lambda| \gg |\Gamma_k|$  since the samples in  $\Lambda$  are used only to evaluate a candidate solution as opposed to the  $\Gamma_k$ s that are used to optimize large-scale MILPs that result from the SAAs. The standard error for the candidate upper bound  $x^*$  is computed using an expression similar to the expression used for computing the standard error of the lower bound estimate in (8). We remark that this technique of obtaining the statistical estimate of an upper bound can be used for any feasible FCMURP solution obtained using any heuristic technique; each feasible solution would result in a different estimate for the upper bound.

The challenge arising for the SAA algorithm is that it still requires  $N$  SAA problems, which are MILPs, to be solved to optimality. The optimal solutions to these problems are required to construct the statistical estimates of the upper and lower bounds of the optimal solution of the two-stage stochastic formulation,  $C^*$ , for the FCMURP. When the number of targets  $|T|$  is large, the MILPs can still pose a considerable challenge for commercial MILP solvers. Moreover, the second-stage model is itself an MILP, making it averse to traditional decomposition techniques such as the L-shaped method. For this reason, there is a need to develop heuristics to solve the SAA problems. These heuristics would aid in computing an estimate of the upper bound for  $C^*$ , using the procedure detailed in Section 4.2. However, an estimate of the lower bound for  $C^*$  can only be constructed using the optimal solutions to the SAA problems. In the forthcoming section, we present a tabu search heuristic to solve the individual SAA problems that are used to obtain statistical estimates of an upper bound for  $C^*$  in medium- and large-sized test instances.

## 5. Heuristic Solution Technique

The heuristic solution technique for finding good suboptimal solutions to the two-stage stochastic programming formulation of the FCMURP proceeds in two phases. The first phase is the construction phase, in which an initial feasible solution to the two-stage stochastic program is constructed. The second phase is an improvement phase, in which a tabu search method is applied to the feasible solution from the first phase. The tabu search method performs a dynamic neighborhood search to improve the quality of the objective functions. For the first phase of the heuristic, we present an optimization-based construction algorithm motivated by the fact that the deterministic variant of the FCMURP is computationally not intensive, despite being an MILP itself. The construction heuristic solves multiple, massively parallelizable, instances of the deterministic FCMURP and combines all the solutions to construct a feasible solution for the two-stage stochastic program for the FCMURP. In the following subsection, we present the details of this construction heuristic.

### 5.1. Construction Heuristic

We begin by examining the objective function for the two-stage stochastic program given in (1). The objective function aims to minimize the sum of the total travel cost in the first stage and the expected cost of the refueling trips in the second stage. The input to the construction heuristic is a subset of scenarios  $\Delta \subset \Omega$  and  $|\Delta| = |\Gamma_k|$ . The main task of the construction heuristic is to determine the subset of edges in  $E$  that are most likely to be present in the optimal solution for the two-stage problem. The deterministic version of the FCMURP (which is basically the first-stage problem with the objective of minimizing the travel cost) is solved for each scenario in the set  $\Delta$ , in decreasing order of the probability of the scenarios, with the fuel consumption taking the value  $f(\omega)$  for the solution corresponding to the realization  $\omega \in \Delta$ . Let the optimal solution corresponding to the scenario  $\omega \in \Delta$  be denoted by  $x^\omega$ . The values of  $x^\omega$  for  $\omega \in \Delta$  that are obtained from the deterministic solutions are then used to construct a cumulative weighting function  $d_{ij}$  for each edge  $(i, j) \in E$ , defined by

$$d_{ij} = 1 - \sum_{\omega \in \Delta} p(\omega) x_{ij}^\omega,$$

where  $p(\omega)$  is the probability of the scenario  $\omega$ . These weights are used to transform the travel cost coefficients and the fuel consumption for each edge  $(i, j) \in E$  as

$$\bar{c}_{ij} \leftarrow c_{ij} d_{ij} \quad \text{and} \quad \bar{f}_{ij} \leftarrow \sum_{\omega \in \Delta} p(\omega) f_{ij}(\omega). \quad (10)$$

The intuition behind the construction of the costs  $\bar{c}_{ij}$  for  $(i, j) \in E$  is that if an edge  $(i, j) \in E$  is present in the optimal solution of every deterministic solution  $x^\omega$ ,  $\omega \in \Delta$ , then it is very likely to be a part of the optimal solution for the two-stage formulation. Hence, its transformed cost  $\bar{c}_{ij}$  in this case

becomes zero. A similar argument holds for the other case, where an edge is not a part of the optimal solution for every deterministic solution. A final deterministic solution to obtain the heuristic solution is then constructed by using the transformed cost and fuel consumption values in (10). Pseudocode for the heuristic is given in Algorithm 1.

---

**Algorithm 1** Construction heuristic
 

---

**Input:** the instance data,  $\Delta \subset \Omega$

**Output:** a heuristic solution  $x^h$

```

1: for  $\omega \in \Delta$  do                                     ▷ Parallelizable for each  $\omega \in \Delta$ 
2:   solve the deterministic FCMURP with fuel consumption values set to  $f(\omega)$ 
3:    $x^\omega \leftarrow$  optimal solution of the deterministic problem
4: end for
5: for  $(i, j) \in E$  do
6:    $d_{ij} \leftarrow 1 - \sum_{\omega \in \Delta} p(\omega)x_{ij}^\omega$            ▷ weighting function
7:    $\bar{c}_{ij} \leftarrow c_{ij}d_{ij}$                                ▷ travel-cost transformation
8:    $\bar{f}_{ij} \leftarrow \sum_{\omega \in \Delta} p(\omega)f_{ij}(\omega)$    ▷ fuel consumption update
9: end for
10:  $x^h \leftarrow$  optimal solution of deterministic FCMURP solved with  $\bar{c}_{ij}$  and  $\bar{f}_{ij}$ 

```

---

### 5.2. Improvement Heuristic: Tabu Search Method

The tabu search method (TSM) is essentially a dynamic neighborhood search algorithm with a flexible way to restrict the next solution choice to some subset of neighbors of the current solution. Unlike a local descent-based search algorithm, TSM permits moves that degrade the current objective function value. The procedure maintains a modified neighborhood for the current solution that is constructed by maintaining a selective history of the states encountered during the search. Hence, TSM is a dynamic neighborhood search method, i.e., the neighborhood of the solution is not a static set, but rather a set that can change according to the search history. The algorithm presented in this section is based on the work in [39,43], which we adapt to our problem setting. The input to TSM is the feasible solution obtained using the construction heuristic presented in the previous section. At each iteration of TSM, a neighborhood for the current solution is constructed, and within the neighborhood, a new current solution is selected in accordance with a selection criterion. Then the algorithm continues with the new current solution.

Given  $\bar{x}$ , a feasible first-stage route for the FCMURP, TSM computes the cost of  $\bar{x}$  as

$$C(\bar{x}) = \sum_{(i,j) \in E} c_{ij}\bar{x}_{ij} + \sum_{\omega \in \Delta_1} p(\omega)\beta(\bar{x}, f(\omega)) + \sum_{\omega \in \Delta_2} p(\omega)\nu, \quad (11)$$

where  $\Delta_1$  and  $\Delta_2$  represent feasible and infeasible scenarios for the given feasible solution  $\bar{x}$  (notice that  $\Delta_1 \cap \Delta_2 = \emptyset$  and  $\Delta_1 \cup \Delta_2 = \Delta \subset \Omega$ ). The parameter  $\nu$  penalizes the infeasible  $\bar{x}$  and is chosen as the sum of the largest  $\beta(\bar{x}, f(\omega))$  and a large positive number. A dynamic neighborhood of the solution  $\bar{x}$ , denoted by  $g(\bar{x})$ , is then constructed. Given  $\bar{x}$ ,  $g(\bar{x})$  is obtained by performing a one-point exchange of the targets, i.e., two targets  $i$  and  $j$  in the solution  $\bar{x}$  are swapped. Suppose that in the current iteration a solution that is obtained by swapping two targets  $i$  and  $j$  is selected as a new current solution, then this target-target swap is inserted into to the ‘tabu’ list to prevent its selection for the next  $q$  iterations. The purpose of this ‘tabu’ list is to keep track of exchanges that have taken place during the recent past. This prevents certain solutions from the recent past from belonging to the neighborhood  $g(\bar{x})$ . Nevertheless, swaps may also be removed from the ‘tabu’ list if the *aspiration criterion* is satisfied i.e., a solution generated by exchanging the positions of the targets within a route has a better objective value than the current best objective value.

At each iteration, the algorithm selects a non-tabu solution in the neighborhood  $g(\bar{x})$  of the current solution  $\bar{x}$  that has a better objective value than the current solution (lines 3 and 4 in Algorithm 2). If such a solution cannot be found by the algorithm, then the best non-tabu solution in  $g(\bar{x})$  is chosen (line 6 in Algorithm 2). The chosen solution is then checked for feasibility and objective value improvement (line 8 in Algorithm 2) and if satisfied, is set to be the best solution thus far, i.e.,  $x^*$ . The primary termination criterion of the algorithm is given by the number of iterations,  $\theta$ . In addition, the algorithm has a secondary termination criterion: if the best solution  $x^*$  is not updated for  $\tau$  iterations ( $\tau < \theta$ ), then the algorithm is terminated (see line 17 in Algorithm 2). The complete pseudocode for the tabu search heuristic is shown in Algorithm 2.

---

**Algorithm 2** Tabu search heuristic.
 

---

**Input:** initial feasible solution  $\bar{x}$

**Output:** a heuristic solution  $x^*$

```

1: set  $x^* := \bar{x}$ ,  $C(x^*) := C(\bar{x})$ ,  $k := 1$ ,  $\Xi := \emptyset$ , and stop := 0 ▷  $\Xi$  - tabu list
2: while  $k \leq \theta$  and stop = 0 do
3:   generate  $g(\bar{x})$  ▷ neighborhood generation
4:   choose  $x' \in g(\bar{x})$  that satisfies  $C(x') < C(\bar{x})$  and is not in tabu list  $\Xi$  or satisfies aspiration
   criteria
5:   if  $x' = \emptyset$  then
6:     choose a solution  $x' = \operatorname{argmin}\{C(x) : x \in g(\bar{x}), x_{i,j} \notin \Xi\}$  ▷ best non-tabu solution
7:   end if
8:   if  $x'$  is feasible and  $C(x') < C(x^*)$  then ▷ feasibility check
9:     set  $x^* := x'$  and  $C(x^*) := C(x')$ 
10:  end if
11:  if  $x^*$  is not updated for  $\sqrt{k}$  iterations then
12:    set  $\bar{x} = x^*$ ,  $C(\bar{x}) := C(x^*)$  ▷ reset neighborhood search
13:  else
14:    set  $\bar{x} := x'$  and  $C(\bar{x}) := C(x')$ 
15:  end if
16:  update  $\Xi$  based on  $\varrho$  ▷ tabu list update
17:  if  $x^*$  is not updated for  $\tau$  iterations then ▷ secondary termination criterion
18:    set stop := 1
19:  end if
20:  set  $k := k + 1$ 
21: end while

```

---

## 6. Computational Results

All the formulations and algorithms were implemented using the Java programming language, and CPLEX 12.6.2 was used as the underlying MILP solver. All simulations were performed using an Intel(R) Core(TM) i7 processor @2.70 GHz and 80 GB RAM. The performance of the algorithm was tested on randomly generated test instances.

### 6.1. Instance Generation

Two classes of test instances were randomly generated. The first set consisted of five instances with 10 targets. We refer to this set of instances as the small-scale instances for which all the algorithms presented in this article were tested and compared. The second set consisted of 120 instances with  $|T| \in \{20, 30\}$ , which we will refer to as large-scale instances. For all the instances, the coordinates of

all targets were uniformly generated from a square grid of size  $100 \times 100$  and there were four refueling sites. The locations of the depot and the four refueling sites were fixed for all the test instances.

For each of the randomly generated small-scale instances, there were three vehicles in the depot, and the fuel capacity  $F$  of the vehicles was varied linearly with a parameter  $\lambda$  that denoted the maximum distance between any depot and any target.  $F$  was set to  $2.25\lambda$  for all the small-scale instances. For the large-scale instances, the number of targets, the number of vehicles  $m$ , and the value of  $F$  were chosen from the sets  $\{20, 30\}$ ,  $\{3, 4, 5\}$ , and  $\{2.25\lambda, 2.5\lambda, 2.75\lambda, 3\lambda\}$ , respectively. For each combination  $(|T|, m, F)$ , five random instances were generated, resulting in a total of 120 large-scale instances. To define the uncertainty regarding the fuel consumed by any UAV to traverse an edge  $(i, j) \in E$ , we let the mean fuel consumed for traversing the edge  $(i, j)$  be the Euclidean distance between the vertices  $i$  and  $j$  [50]. For the scenario generation, we also followed [50], which suggests a gamma distribution to model the uncertainties regarding fuel consumption. In general, the fuel consumption has the following two properties [50]: (i) the fuel consumed by any UAV to traverse any edge  $(i, j)$  is a higher deviation from the expected value due to external factors such as wind; and (ii) the deviations from the mean that are higher than the mean are much higher when compared to the deviations that are lower than the mean. A gamma distribution is an ideal candidate to exhibit such behavior. Hence, for the purposes of this article, we generated scenarios from the gamma distribution for the random fuel consumption vector,  $f$ , using a scale parameter  $0.25 \cdot \mathbb{E}(f)$  and a shape parameter value of four [50]. We also remark that, since the SAA and heuristic solutions are independent of the distributions used, the algorithm would still work with scenarios generated from any other fitting distribution. Finally, to make the scenarios more realistic, we divided the coordinate space in all test instances into four equal quadrants, and randomly designated one of the quadrants as ‘congested’, one as ‘sparse’, and the other two as ‘mean’. For a scenario  $\omega$ , all the edges incident to a vertex in a congested quadrant were given an  $f_{ij}(\omega)$  that is higher than its mean,  $\mathbb{E}(f_{ij})$ . Similarly, all the edges connected with a vertex in a sparse quadrant were given an  $f_{ij}(\omega)$  that is lower than its  $\mathbb{E}(f_{ij})$ . For the remaining edges, i.e., edges with both vertexes in mean quadrants,  $f_{ij}(\omega)$  was set to  $\mathbb{E}(f_{ij})$ . Instead of randomly assigning  $f_{ij}(\omega)$  for the edges based on the gamma distribution, this design creates correlations among the fuel consumption for different edges, making the scenarios more realistic.

### 6.2. Parameters for SAA and the Heuristic

The statistical estimate of the lower bound for the optimal objective value of the two-stage stochastic program was obtained using SAA as detailed in Section 4.1. The following parameter values were used to obtain the estimate:  $N = 10$  and  $|\Gamma_k| = 10$  for every  $k \in \{1, \dots, 10\}$ . All 10 scenarios in each  $\Gamma_k$  were independently drawn in accordance with the procedure detailed in the previous section. The scenario set  $\Lambda$  that was used to compute an estimate of the upper bound for  $C^*$  from any feasible first-stage solution (the UAV routes) was generated similarly to the sets  $\Gamma_k$ ,  $k = 1, \dots, 10$ , with  $|\Lambda| = 1000$ . Java’s concurrent libraries were used to parallelize the construction heuristic. Construction and improvement heuristics were run  $N$  times, each iteration with  $\Delta = \Gamma_k$ ,  $k \in \{1, \dots, N\}$ .

### 6.3. Performance Evaluation Criteria

We compare the objective values obtained using SAA and the heuristic solution with the traditional “Expected Value Problem” (EVP). The EVP is a single-stage deterministic problem that computes minimum travel costs for UAV routes with the fuel consumption set to their mean, i.e., the objective value for the EVP is given by  $\sum_{(i,j) \in E} c_{ij}x_{ij}$ . We denote the objective value for the EVP—i.e., the travel cost for the UAV routes obtained by solving the EVP—as EV in the following tables. Given  $\bar{x}$ , the optimal solution to the EVP, the expected cost of using  $\bar{x}$  in the two-stage stochastic program is given by  $C(\bar{x})$ . In the following tables, this estimate is denoted as EEV (Expected cost of the Expected Value solution). Finally, we use LB-SAA, UB-SAA, and H to respectively denote the statistical estimates of the lower and upper bounds obtained by SAA and the estimate of the upper bound obtained using the heuristic.  $C(\bar{x})$  is estimated using the same scenarios that were used in obtaining the lower bound

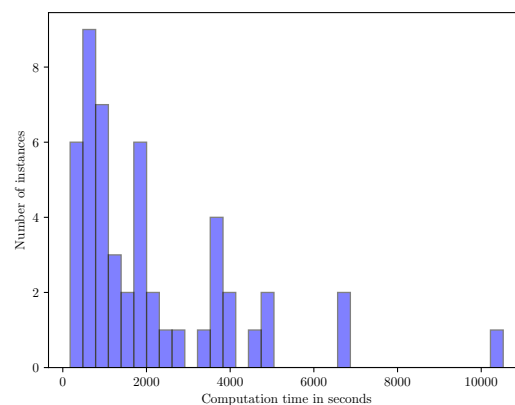
for the SAA. Given a subset of realizations of  $\Omega$ , the difference between the optimal solution to the two-stage stochastic program and the EEV is referred to as the value of the stochastic solution (VSS) in the parlance of stochastic optimization. The VSS was first introduced in [51] and is a standard means for quantifying the usefulness of the stochastic programming approach. As is shown in the literature, for a minimization problem the cost of the optimal solution to the two-stage stochastic program is always less than or equal to the cost of using the expected value solution in a two-stage setting [33]. For a given instance and a subset of realizations, the VSS quantifies the gain from solving a two-stage stochastic program instead of a deterministic EVP.

#### 6.4. Results for Small-Scale Test Instances

The purpose of the small-scale instances was to examine the run times for the SAA and heuristic approaches and to benchmark the objective values obtained using them. Table 2 presents the complete results for the 10-target instances. From this table, it can be observed that the estimates obtained using the tabu search heuristic are very close to the SAA upper bound estimates in the column UB-SAA. Furthermore, the heuristic objective values are also strictly greater than the EEV, indicating that there is value in using the heuristic to solve the two-stage problem rather than solving the corresponding deterministic EVP. A histogram of the run times for solving each problem instance in the SAA approach is shown in Figure 2. The figure indicates that even for 10-target instances, solving each SAA problem to optimality can be time consuming, and in the worst case it takes around three hours. This indicates the computational difficulty of solving the SAA problems to optimality. Nevertheless, the solutions obtained using the SAA approach are useful for evaluating the quality of the solutions obtained using the heuristic approach for the small-scale instances.

**Table 2.** Computational results for the small-scale instances. The values in brackets for the columns EEV, LB-SAA, UB-SAA, and H denote the standard deviation of these estimates.

#	EV	EEV	LB-SAA	UB-SAA	H
1	391.00	513.20 (3.36)	451.70 (1.70)	455.24 (1.74)	477.68 (2.24)
2	432.00	559.25 (1.79)	503.34 (2.04)	504.65 (4.01)	526.44 (4.59)
3	429.00	578.77 (2.11)	553.53 (1.62)	564.70 (2.74)	558.95 (2.97)
4	396.00	483.50 (2.69)	443.50 (2.56)	446.31 (2.74)	449.05 (3.58)
5	405.00	541.33 (2.07)	487.00 (1.62)	490.60 (2.38)	489.38 (2.64)



**Figure 2.** Histogram of run times to compute the optimal solution of each SAA instance for the small-scale problems.

### 6.5. Results for Large-Scale Instances

For the large-scale set of instances with  $|T| \in \{20, 30\}$ , we compare the quality of the solutions produced by the heuristic with the EEV using VSS. We do not use the SAA approach for these instances due to the computational scalability issues of the SAA approach for these test instances. Tables 3–5 present the expected value solution’s objective (EV), the EEV, and the heuristic solution objective value (H) for all instances in the large-scale set. Similar to Table 2, the values in brackets denote the standard deviations of the corresponding estimates. It is clear from Tables 3–5 that the heuristic is consistently able to provide a solution whose mean cost is strictly less than the mean EEV for every instance in this class, indicating that the value of VSS that is computed using the mean values is strictly greater than 0. The average and maximum values of the VSS (in percentages) for the large-scale set of instances computed using the mean values are 6.4% and 29.69%, respectively. A scatter plot of the mean EEV and mean heuristic solution cost is shown in Figure 3. The fact that all the points lie below the line with slope one indicates that the VSS is strictly greater than zero for all the instances. Finally, Figure 4 shows the box plot of the computation time taken by the tabu search heuristic to compute a feasible solution for the FCMURP on all the instances in the large-scale set. It can be observed from the box plot that the computation time for the heuristic solution approach is well within six minutes for all these instances.

**Table 3.** Results for  $m = 2$  vehicles.

$F$	$ T  = 20$				$ T  = 30$			
	#	EV	EEV	H	#	EV	EEV	H
2.25 $\lambda$	1	442.00	510.12 (1.87)	479.63 (1.58)	1	575.00	642.95 (2.69)	596.00 (3.06)
2.25 $\lambda$	2	467.00	564.83 (1.28)	492.50 (1.04)	2	549.00	603.40 (2.60)	585.00 (3.43)
2.25 $\lambda$	3	442.00	511.65 (1.72)	483.75 (2.24)	3	549.00	576.00 (2.51)	566.00 (2.51)
2.25 $\lambda$	4	482.00	506.00 (1.47)	494.00 (0.53)	4	530.00	586.00 (2.42)	561.00 (2.71)
2.25 $\lambda$	5	471.00	578.57 (1.40)	521.15 (0.71)	5	541.00	610.00 (2.33)	588.00 (3.22)
2.50 $\lambda$	1	467.00	565.45 (0.72)	489.00 (0.96)	1	524.00	615.30 (2.24)	547.00 (2.69)
2.50 $\lambda$	2	442.00	512.15 (1.16)	482.25 (0.25)	2	516.00	600.00 (2.15)	560.00 (2.22)
2.50 $\lambda$	3	442.00	502.73 (2.68)	484.45 (1.81)	3	505.00	584.30 (2.06)	535.00 (2.78)
2.50 $\lambda$	4	471.00	588.17 (3.31)	529.63 (3.12)	4	539.00	611.00 (1.97)	604.00 (2.66)
2.50 $\lambda$	5	467.00	564.18 (0.83)	491.25 (0.43)	5	534.00	606.00 (1.88)	585.00 (1.95)
2.75 $\lambda$	1	442.00	512.77 (2.47)	481.63 (2.83)	1	523.00	584.00 (1.79)	540.00 (2.18)
2.75 $\lambda$	2	442.00	493.58 (3.04)	485.68 (2.93)	2	518.00	573.00 (1.70)	539.90 (2.55)
2.75 $\lambda$	3	487.00	587.85 (2.01)	544.00 (3.70)	3	546.00	580.00 (1.61)	576.00 (2.18)
2.75 $\lambda$	4	485.00	584.98 (0.90)	516.25 (1.25)	4	542.00	587.00 (1.52)	571.00 (1.97)
2.75 $\lambda$	5	460.00	523.65 (1.02)	512.20 (4.13)	5	509.00	555.00 (1.43)	535.00 (2.37)
3.00 $\lambda$	1	505.00	584.85 (7.45)	545.40 (1.59)	1	509.00	543.00 (1.34)	539.00 (1.87)
3.00 $\lambda$	2	438.00	534.05 (5.3)	476.28 (2.95)	2	562.00	650.00 (1.25)	629.55 (1.80)
3.00 $\lambda$	3	433.00	509.52 (7.34)	473.73 (4.59)	3	540.00	618.30 (1.16)	579.00 (1.17)
3.00 $\lambda$	4	442.00	513.62 (1.83)	477.12 (2.02)	4	525.00	598.30 (1.07)	540.00 (1.12)
3.00 $\lambda$	5	451.00	534.68 (4.86)	505.18 (8.74)	5	505.00	586.00 (0.98)	580.00 (1.50)

**Table 4.** Results for  $m = 3$  vehicles.

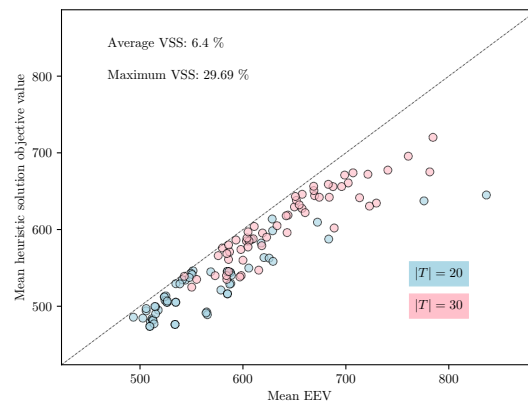
$F$	$ T  = 20$				$ T  = 30$			
	#	EV	EEV	H	#	EV	EEV	H
2.25 $\lambda$	1	460.00	525.10 (3.33)	513.48 (4.33)	1	502.00	597.03 (2.63)	538.23 (3.53)
2.25 $\lambda$	2	485.00	587.18 (1.20)	528.75 (1.64)	2	477.00	550.23 (2.89)	524.85 (3.87)
2.25 $\lambda$	3	460.00	526.27 (1.50)	505.53 (1.01)	3	561.00	706.97 (3.71)	673.95 (3.17)
2.25 $\lambda$	4	460.00	517.65 (2.51)	495.00 (3.05)	4	542.00	598.00 (1.85)	574.00 (2.23)
2.25 $\lambda$	5	552.00	775.90 (1.69)	637.38 (0.89)	5	552.00	729.65 (2.39)	634.43 (2.03)
2.50 $\lambda$	1	497.00	628.82 (1.50)	598.43 (0.92)	1	539.00	688.65 (2.24)	602.00 (2.64)
2.50 $\lambda$	2	460.00	524.38 (1.05)	507.60 (1.88)	2	527.00	619.00 (1.21)	595.40 (1.31)

Table 4. Cont.

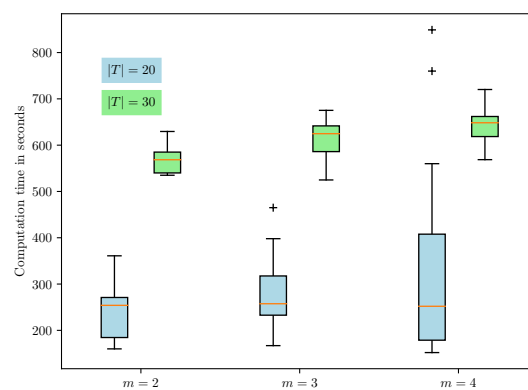
$F$	$ T  = 20$				$ T  = 30$			
	#	EV	EEV	H	#	EV	EEV	H
2.50 $\lambda$	3	492.00	605.67 (2.91)	549.73 (2.07)	3	524.00	604.92 (1.29)	577.27 (1.59)
2.50 $\lambda$	4	486.00	620.45 (1.98)	563.45 (2.52)	4	575.00	713.32 (3.88)	641.48 (3.35)
2.50 $\lambda$	5	486.00	608.18 (2.00)	587.20 (1.99)	5	568.00	669.62 (3.39)	644.20 (2.67)
2.75 $\lambda$	1	460.00	526.30 (0.75)	504.95 (1.20)	1	566.00	651.95 (2.23)	638.00 (2.38)
2.75 $\lambda$	2	460.00	515.05 (2.24)	499.23 (2.23)	2	556.00	604.00 (1.52)	589.00 (2.26)
2.75 $\lambda$	3	487.00	587.85 (2.01)	544.00 (2.43)	3	588.00	702.30 (4.77)	660.80 (4.76)
2.75 $\lambda$	4	485.00	584.98 (0.90)	516.25 (1.46)	4	567.00	657.38 (4.29)	627.53 (3.31)
2.75 $\lambda$	5	460.00	523.65 (1.02)	512.20 (1.57)	5	549.00	660.47 (1.71)	622.05 (1.69)
3.00 $\lambda$	1	460.00	515.15 (1.16)	490.12 (0.75)	1	549.00	605.00 (3.10)	597.00 (2.38)
3.00 $\lambda$	2	438.00	534.05 (1.29)	476.28 (1.90)	2	574.00	781.55 (3.11)	675.17 (3.73)
3.00 $\lambda$	3	433.00	509.52 (1.42)	473.73 (0.80)	3	558.00	674.20 (2.05)	642.08 (1.24)
3.00 $\lambda$	4	460.00	526.55 (0.49)	506.75 (0.42)	4	543.00	654.70 (3.37)	632.23 (2.48)
3.00 $\lambda$	5	460.00	514.40 (1.49)	499.75 (0.77)	5	511.00	586.32 (0.87)	545.30 (0.05)

Table 5. Results for  $m = 4$  vehicles.

$F$	$ T  = 20$				$ T  = 30$			
	#	EV	EEV	H	#	EV	EEV	H
2.25 $\lambda$	1	480.00	551.45 (3.65)	546.25 (2.71)	1	618.00	784.63 (5.31)	720.20 (5.59)
2.25 $\lambda$	2	505.00	568.72 (2.91)	544.92 (2.21)	2	583.00	698.80 (3.51)	670.95 (2.69)
2.25 $\lambda$	3	480.00	549.62 (0.89)	543.00 (0.50)	3	583.00	657.60 (2.02)	645.95 (2.98)
2.25 $\lambda$	4	520.00	544.00 (2.09)	534.00 (1.83)	4	566.00	633.15 (3.32)	605.00 (2.44)
2.25 $\lambda$	5	574.00	836.52 (1.17)	645.00 (1.36)	5	573.00	740.77 (4.20)	677.45 (3.28)
2.50 $\lambda$	1	519.00	672.40 (4.96)	609.55 (4.18)	1	561.00	683.90 (2.37)	642.07 (1.73)
2.50 $\lambda$	2	519.00	628.45 (3.52)	613.75 (3.27)	2	549.00	643.75 (2.12)	618.77 (1.39)
2.50 $\lambda$	3	514.00	617.72 (1.26)	582.35 (0.45)	3	520.00	584.32 (3.58)	568.70 (4.57)
2.50 $\lambda$	4	523.00	683.30 (3.62)	587.52 (3.85)	4	593.00	721.37 (4.79)	672.08 (5.64)
2.50 $\lambda$	5	505.00	588.75 (2.75)	540.28 (1.88)	5	586.00	695.88 (2.84)	656.03 (3.72)
2.75 $\lambda$	1	493.00	541.20 (2.04)	534.55 (1.21)	1	582.00	668.90 (3.69)	650.83 (3.41)
2.75 $\lambda$	2	480.00	534.80 (1.93)	529.40 (2.49)	2	572.00	642.05 (4.54)	618.00 (4.12)
2.75 $\lambda$	3	505.00	625.50 (4.16)	562.70 (3.72)	3	603.00	722.95 (5.76)	630.48 (5.91)
2.75 $\lambda$	4	505.00	584.85 (3.73)	545.40 (3.29)	4	585.00	668.70 (1.59)	656.40 (2.58)
2.75 $\lambda$	5	480.00	550.57 (1.62)	541.33 (0.98)	5	567.00	650.95 (0.97)	643.10 (1.28)
3.00 $\lambda$	1	480.00	506.00 (3.50)	497.00 (4.14)	1	567.00	623.00 (2.02)	590.00 (2.82)
3.00 $\lambda$	2	505.00	629.20 (4.06)	558.50 (4.30)	2	583.00	760.67 (5.70)	695.58 (6.15)
3.00 $\lambda$	3	451.00	534.68 (4.86)	505.18 (4.66)	3	576.00	687.48 (3.19)	655.98 (3.04)
3.00 $\lambda$	4	480.00	547.93 (2.42)	537.50 (2.33)	4	565.00	682.97 (3.34)	658.93 (3.00)
3.00 $\lambda$	5	480.00	538.73 (2.21)	529.20 (3.04)	5	532.00	593.25 (1.89)	586.38 (2.24)



**Figure 3.** Scatter plot of the mean EEV vs. mean of the estimated solution cost from the heuristic.



**Figure 4.** Box plot of the heuristic run times for the second set of instances. The mean run times, in seconds, for  $m = 2$ ,  $m = 3$ , and  $m = 4$  for  $|T| \in \{20, 30\}$  are given by  $(243, 310)$ ,  $(275, 354)$ , and  $(327, 360)$ , respectively.

## 7. Conclusions and Future Work

In this paper, we have introduced a new approach to address route planning for multiple UAVs under uncertainty (FCMURP) regarding the fuel consumption between any two targets. The new approach is a two-stage stochastic model with a first-stage objective function to minimize the route costs and a second-stage recourse cost for additional visits to refueling depots when the fuel is inadequate to complete the tour proposed by the first stage. We adopted the sample average approximation (SAA) method to solve the two-stage FCMURP model, with the second-stage expectation function approximated using a sampling-based approach. The SAA objective value will asymptotically converge to the optimal expected cost of the two-stage model, but it is still computationally expensive for large instances. Therefore, we developed a tabu search heuristic, which is very conducive to parallelization. Our heuristic provides good-quality solutions and is computationally efficient. For our computational experiments, we used randomly generated instances and created uncertainty that reflects more realistic environments. The results indicate that the solutions obtained with the heuristic are computationally efficient and not very different from the solutions for deterministic models and that they are also very robust in the face of uncertainty. Potential future research can consider uncertainty in the availability of UAVs for the mission, and a possible focus for the two-stage model would be to consider a risk-averse model where additional risk-measuring weights are added to the recourse objective function.

**Author Contributions:** Conceptualization, S.V., K.S. and S.R.; Methodology, S.V., K.S. and S.R.

**Funding:** This research received no external funding.

**Acknowledgments:** Kaarthik Sundar gratefully acknowledges support for this work from the Centre for Nonlinear Studies.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Willers, J.L.; Jenkins, J.N.; Ladner, W.L.; Gerard, P.D.; Boykin, D.L.; Hood, K.B.; McKibben, P.L.; Samson, S.A.; Bethel, M.M. Site-specific approaches to cotton insect control. Sampling and remote sensing analysis techniques. *Precis. Agric.* **2005**, *6*, 431–452. [[CrossRef](#)]
2. Willers, J.L.; Milliken, G.A.; Jenkins, J.N.; O'Hara, C.G.; Gerard, P.D.; Reynolds, D.B.; Boykin, D.L.; Good, P.V.; Hood, K.B. Defining the experimental unit for the design and analysis of site-specific experiments in commercial cotton fields. *Agric. Syst.* **2008**, *96*, 237–249. [[CrossRef](#)]
3. Willers, J.L.; Jenkins, J.N.; McKinion, J.M.; Gerard, P.; Hood, K.B.; Bassie, J.R.; Cauthen, M.D. Methods of analysis for georeferenced sample counts of tarnished plant bugs in cotton. *Precis. Agric.* **2009**, *10*, 189–212. [[CrossRef](#)]
4. Casbeer, D.W.; Beard, R.W.; McLain, T.W.; Li, S.M.; Mehra, R.K. Forest fire monitoring with multiple small UAVs. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; Volume 5, pp. 3530–3535. [[CrossRef](#)]
5. Corrigan, C.E.; Roberts, G.C.; Ramana, M.V.; Kim, D.; Ramanathan, V. Capturing vertical profiles of aerosols and black carbon over the Indian Ocean using autonomous unmanned aerial vehicles. *Atmos. Chem. Phys.* **2008**, *8*, 737–747. [[CrossRef](#)]
6. Zajkowski, T.; Dunagan, S.; Eilers, J. Small UAS communications mission. In Proceedings of the Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference, Salt Lake City, UT, USA, 24–28 April 2006.
7. Maza, I.; Caballero, F.; Capitan, J.; de Dios, J.R.M.; Ollero, A. Firemen Monitoring with Multiple UAVs for Search and Rescue Missions. In Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics, Bremen, Germany, 26–30 July 2010.
8. Krishnamoorthy, K.; Casbeer, D.; Chandler, P.; Pachter, M.; Darbha, S. UAV Search and Capture of a Moving Ground Target under Delayed Information. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012.
9. Maza, I.; Caballero, F.; Capitan, J.; de Dios, J.R.M.; Ollero, A. Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications. *J. Intell. Robot. Syst.* **2011**, *61*, 563–585. [[CrossRef](#)]
10. Comeaux, M. Soldiers Train with Raven UAVs. 2007. Available online: [https://www.army.mil/article/5644/soldiers\\_train\\_with\\_raven\\_uavs](https://www.army.mil/article/5644/soldiers_train_with_raven_uavs) (accessed on 25 June 2018).
11. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 70–85. [[CrossRef](#)]
12. D'Andrea, R. Guest editorial can drones deliver? *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 647–648. [[CrossRef](#)]
13. Welch, A. *A Cost-Benefit Analysis of Amazon Prime Air*; Technical Report; University of Tennessee at Chattanooga: Chattanooga, TN, USA, 2015.
14. Stewart, J. Google Tests Drone Deliveries in Project Wing Trials. 2014. Available online: <http://www.bbc.com/news/technology-28964260> (accessed on 25 June 2018).
15. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2001.
16. Khuller, S.; Malekian, A.; Mestre, J. To fill or not to fill: The gas station problem. In *Algorithms—ESA 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 534–545.
17. Sundar, K.; Rathinam, S. Route planning algorithms for unmanned aerial vehicles with refueling constraints. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 3266–3271.
18. Sundar, K.; Rathinam, S. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 287–294. [[CrossRef](#)]
19. Levy, D.; Sundar, K.; Rathinam, S. Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. *Math. Probl. Eng.* **2014**, *2014*, 131450. [[CrossRef](#)]

20. Mitchell, D.; Corah, M.; Chakraborty, N.; Sycara, K.; Michael, N. Multi-robot long-term persistent coverage with fuel constrained robots. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1093–1099.
21. Sundar, K.; Rathinam, S. An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 366–371.
22. Sundar, K.; Venkatachalam, S.; Rathinam, S. Analysis of Mixed-Integer Linear Programming Formulations for a Fuel-Constrained Multiple Vehicle Routing Problem. *Unmanned Syst.* **2017**, *5*, 197–207. [[CrossRef](#)]
23. Sundar, K.; Venkatachalam, S.; Rathinam, S. Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 6489–6494.
24. Erdogan, S.; Miller-Hooks, E. A green vehicle routing problem. *Transp. Res. Part E-Logist. Transp. Rev.* **2012**, *48*, 100–114. [[CrossRef](#)]
25. Lin, C.; Choy, K.L.; Ho, G.T.; Chung, S.H.; Lam, H. Survey of green vehicle routing problem: Past and future trends. *Expert Syst. Appl.* **2014**, *41*, 1118–1138. [[CrossRef](#)]
26. Laporte, G.; Desrochers, M.; Nobert, Y. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks* **1984**, *14*, 161–172. [[CrossRef](#)]
27. Li, C.L.; Simchi-Levi, D.; Desrochers, M. On the distance constrained vehicle routing problem. *Oper. Res.* **1992**, *40*, 790–799. [[CrossRef](#)]
28. Kara, I. On the Miller-Tucker-Zemlin Based Formulations for the Distance Constrained Vehicle Routing Problems. In Proceedings of the ICMS International Conference on Mathematical Science, Bolu, Turkey, 23–27 November 2010; Volume 1309, pp. 551–561.
29. Kara, I. Arc based integer programming formulations for the distance constrained vehicle routing problem. In Proceedings of the 3rd IEEE International Symposium on Logistics and Industrial Informatics, Budapest, Hungary, 25–27 August 2011; pp. 33–38.
30. Nagarajan, V.; Ravi, R. Approximation algorithms for distance constrained vehicle routing problems. *Networks* **2012**, *59*, 209–214. [[CrossRef](#)]
31. Fischetti, M.; Gonzalez, J.J.S.; Toth, P. Solving the orienteering problem through branch-and-cut. *Inf. J. Comput.* **1998**, *10*, 133–148. [[CrossRef](#)]
32. Vansteenwegen, P.; Souffriau, W.; Van Oudheusden, D. The orienteering problem: A survey. *Eur. J. Oper. Res.* **2011**, *209*, 1–10. [[CrossRef](#)]
33. Birge, J.R.; Louveaux, F. *Introduction to Stochastic Programming*; Springer Science & Business Media: New York, NY, USA, 2011.
34. Verweij, B.; Ahmed, S.; Kleywegt, A.J.; Nemhauser, G.; Shapiro, A. The sample average approximation method applied to stochastic routing problems: A computational study. *Comput. Optim. Appl.* **2003**, *24*, 289–333. [[CrossRef](#)]
35. Glover, F. Tabu search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [[CrossRef](#)]
36. Glover, F. Tabu search—Part II. *ORSA J. Comput.* **1990**, *2*, 4–32. [[CrossRef](#)]
37. Gendreau, M.; Hertz, A.; Laporte, G. A tabu search heuristic for the vehicle routing problem. *Manag. Sci.* **1994**, *40*, 1276–1290. [[CrossRef](#)]
38. Gendreau, M.; Laporte, G.; Séguin, R. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Oper. Res.* **1996**, *44*, 469–477. [[CrossRef](#)]
39. Cordeau, J.F.; Laporte, G.; Mercier, A. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* **2001**, *52*, 928–936. [[CrossRef](#)]
40. Garcia, B.L.; Potvin, J.Y.; Rousseau, J.M. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Comput. Oper. Res.* **1994**, *21*, 1025–1033. [[CrossRef](#)]
41. Taillard, É.; Badeau, P.; Gendreau, M.; Guertin, F.; Potvin, J.Y. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transport. Sci.* **1997**, *31*, 170–186. [[CrossRef](#)]
42. Rochat, Y.; Taillard, É.D. Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics* **1995**, *1*, 147–167. [[CrossRef](#)]
43. Taş, D.; Dellaert, N.; Van Woensel, T.; De Kok, T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Comput. Oper. Res.* **2013**, *40*, 214–224. [[CrossRef](#)]

44. Manyam, S.G.; Casbeer, D.W.; Sundar, K. Path planning for cooperative routing of air-ground vehicles. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 4630–4635.
45. Kleywegt, A.J.; Shapiro, A.; Homem-de Mello, T. The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* **2002**, *12*, 479–502. [[CrossRef](#)]
46. Linderoth, J.; Shapiro, A.; Wright, S. The empirical behavior of sampling methods for stochastic programming. *Ann. Oper. Res.* **2006**, *142*, 215–241. [[CrossRef](#)]
47. Shapiro, A.; Homem-de Mello, T. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM J. Optim.* **2000**, *11*, 70–86. [[CrossRef](#)]
48. Mak, W.K.; Morton, D.P.; Wood, R.K. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Oper. Res. Lett.* **1999**, *24*, 47–56. [[CrossRef](#)]
49. Dupacová, J.; Wets, R. Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. *Ann. Stat.* **1988**, *16*, 1517–1549. [[CrossRef](#)]
50. Evers, L.; Glorie, K.; Van Der Ster, S.; Barros, A.I.; Monsuur, H. A two-stage approach to the orienteering problem with stochastic weights. *Comput. Oper. Res.* **2014**, *43*, 248–260. [[CrossRef](#)]
51. Birge, J.R. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Math. Program.* **1982**, *24*, 314–325. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).