STOCHASTIC PROGRAMMING MODELS FOR AUTONOMOUS GROUND VECHICLES

Saravanan Venkatachalam, PhD[1], Manish Bansal, PhD[2], Jonathon M. Smereka, PhD[3]

[1] Department of Industrial and Systems Engineering, Wayne State University
[2] Department of Industrial and Systems Engineering, Virginia Tech
[3] Ground Vehicle Robotics, U.S. Army CCDC Ground Vehicle Systems Center

ABSTRACT

Significant advances in sensing, robotics, and wireless networks have enabled the collaborative utilization of autonomous aerial, ground and underwater vehicles for various applications. However, to successfully harness the benefits of these unmanned ground vehicles (UGVs) in homeland security operations, it is critical to efficiently solve UGV path planning problem which lies at the heart of these operations. Furthermore, in the real-world applications of UGVs, these operations encounter uncertainties such as incomplete information about the target sites, travel times, and the availability of vehicles, sensors, and fuel. This research paper focuses on developing algebraic-based-modeling framework to enable the successful deployment of a team of vehicles while addressing uncertainties in the distance traveled and the availability of UGVs for the mission.

1. INTRODUCTION

Intelligence, surveillance and reconnaissance (ISR) are critical missions within military operations, and modern-day combat zones pose important challenges for ISR [6] [8-9]. ISR operations are maintained through effective and efficient information collection. Unmanned ground vehicles (UGVs) are an important asset for ISR, target engagements, convoy operations for resupply missions, search and rescue, environmental mapping, disaster area surveying and mapping. Depending upon the nature of the missions, UGVs are preferred over other collection assets. Some of the instances where UGVs are preferred include: unsuitable terrain for human or unmanned aerial vehicles (UAVs), harsh and hostile environment, tedious information collection process for humans, and many more. Despite the numerous advantages of UGVs, their size and limited payload capacity lead to fuel constraints and therefore, they are required to make one or more refueling stops in a long mission.

Moreover, these operations encounter unknown terrain or obstacles, resulting in uncertainty in the fuel (or time) required to travel among different points of interests (POIs); for example, in a hostile terrain with improvised explosive devices (IEDs), conducting anti-IED sweeps and explosive ordinance disposal can lead to unexpected delays for UGVs. In fact, in many applications, even the locations of the POIs are not precisely known (uncertain) due to inaccurate a-priori map or imperfect and noisy exteroceptives sensory information or perturbations; for example, in a fire monitoring application, the POIs of UGVs change based on the random propagation of the fire [5]. Likewise, other types of system uncertainties include availability of UGVs with specific attributes such as sensors or terrain-compatible vehicle dynamics.

Due to these challenges, to successfully harness the benefits of the UGVs, it is critical to efficiently solve the UGV path-planning problem (UGVPP). Note that the NP-hard problems such as multiple traveling salesman problem (TSP) and distance-constrained vehicle routing problem are special cases of the UGVPP. In this project, we consider extensions of UGVVPP with aforementioned uncertainties, and refer to this class of problems as Stochastic AVPP (S-AVPP). Motion planning literature [3] for AVs classifies uncertainties into four categories: vehicle dynamics, knowledge of environment, operational environment, and pose information. Uncertainties in operational environment like wind and atmospheric turbulences suite to UAVs and pose information is regarding localization of UGVs. This project focuses on UGVPP with uncertainties in vehicle dynamics and knowledge of environment. Some of the previous works include: analysis of robustness of modular vehicle fleet considering uncertainties in demand of vehicles [7]; path planning for multiple UGVs for deterministic data using heuristic [1]; single vehicle path planning problems for UGVs considering environmental uncertainties [4-5]. The path planning problem for multiple

UGVs considering uncertainties in vehicle dynamics and environmental uncertainties simultaneously using algebraic modelling framework is new to the literature. Furthermore, these algorithms will also be applicable to tackle similar challenges arising in path-planning for UGVs and underwater vehicles, which are used for crop monitoring, ocean bathymetry, forest fire monitoring, border surveillance, and disaster management.
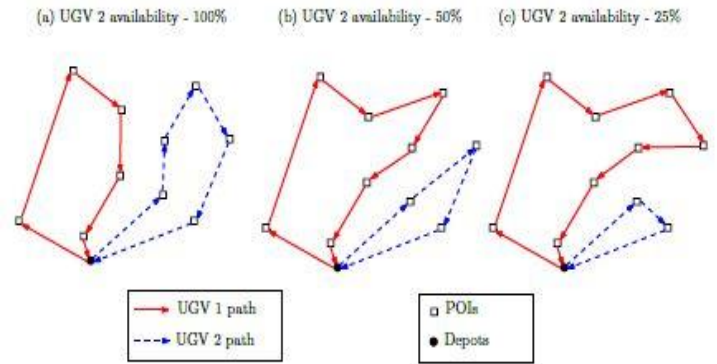

Figure 1: An illustration of considering the availability of UGVs as uncertain in the UGVPP. (a) Optimal solution for a deterministic UGVPP which is sub-optimal for the UGVs when considering uncertainty for the availability of UGVs. (b-c) Optimal solutions for stochastic UGVPP instances having different chances of availability for UGV2. Note that as the chances of the availability of UGV2 reduces, the number of assigned POIs to UGV2 also reduces.

2. NOTATION

Let $T = \{t_1, \ldots, t_n\}$ denote the set of points of interests (POIs), let d_0 denote the depot where a set of heterogeneous unmanned ground vehicles (UGVs) M: = {1. . . |M|}, each with fuel capacity F_m, $m \in M$, are initially stationed, let $\bar{D} = \{d_1, \ldots, d_k\}$ denote the set of additional k depots or refueling sites, and let $D = \bar{D} \cup \{d_0\}$. All of the |M| UGVs stationed at the depot d_0 are assumed to be fueled to capacity. The model formulations are defined on a directed graph $G = (V, E)$ where $V = T \cup D$ denotes the set of vertices and E denotes the

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 2 of 10

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 $\hat{f}_{ij}^m$ represent the travel cost and the nominal fuel that will be consumed by UGV $m \in M$ while traversing the edge (i, j). We remark that $\hat{f}_{ij}^m$ is directly computed using the length of the edge (i, j) and the fuel economy of the UGV. Additional notations that will be used in the mathematical formulation are as follows: for any set $S \subset V$, $\delta^+(S) = \{(i,j) \in E : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i,j) \in E : j \in S, i \notin S\}$. When $S = \{i\}$, we shall simply write $\delta^+(i)$ and $\delta^-(i)$ instead $\delta^+(\{i\})$ and $\delta^-(\{i\})$, respectively.

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

3. MATHEMATICAL FORMULATION

The first-stage decision variables represent 'here-and now' decisions that are determined before the realization of randomness, and second-stage decisions are determined after scenarios

representing the uncertainties are presented. The first-stage decision variables in the stochastic

Symbol	explanation
$T = \{t_1, \ldots, t_n\}$	set of n POIs
d_0	depot where m UGVs are initially stationed
$\overline{D} = \{d_1, \ldots, d_k\}$	additional depots/refueling sites
$D = \overline{D} \bigcup \{d_0\}$	set of depots
F_m	fuel capacity of an 'm' UGV
$G = (V, E)$	directed graph with $V = T \cup D$
c_{ij}	travel cost for the edge $(i, j) \in E$
$\hat{f}_{ij}^m$	nominal fuel consumed by the UGV to traverse the edge $(i, j) \in E$
f	random variable vector representing the fuel consumed by any UGV
p_j^m	profit or incentive for visiting a POI by an UGV m
Ω	set of scenarios for f
$\omega \in \Omega$	realization of random variable f
$p(\omega)$	probability of occurrence of ω
$\mathbb{E}$	expectation operator

Table 1: Table of notations

program is used to compute the initial set of routes for each of the UGVs such that either each POI is visited by only one of the UGVs or all the UGVs collect maximum incentives from the POIs, while ensuring that no UGV ever runs out of fuel as it traverses its route. The fuel constraint for each UGV in the first stage is enforced using the nominal fuel consumption value $\hat{f}_{ij}^m$ for each edge $(i, j) \in E$. For a realization $w \in \Omega$, the second-stage decision variables are used to compute the recourse costs that must be added to the first-stage routes based on the realized values of $f_{ij}^m(\omega)$ for all $(i, j) \in E$ and $m \in M$.

Specifically, the first-stage decision variables are as follows: each edge $(i, j) \in E$ is associated with a variable x_{ij}^m that equals 1 if the edge (i, j) is traversed by 'm' UGV, and 0 otherwise. We let $x \in \{0, 1\}^{|E| \times |M|}$ denote the vector of all decision variables x_{ij}^m. 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 UGV as it starts from depot i and reaches the vertex SS. Additionally, for any $A \subseteq E$, we let $x^m(A) = \sum_{(i,j)\in A} x_{ij}^m$. Analogous to the variable x_{ij}^m in the first stage, we define a binary variable $y_{ij}^m(\omega)$ for each edge $(i, j) \in E$. The variables $y_{ij}^m(\omega)$ are used

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 3 of 10

to define the refueling trips needed for any vehicle when the route defined by the first-stage feasible solution x is not feasible for the realization $w \in \Omega$.

4. FORMULATION 1

Given a team of heterogeneous UGVs (each UGV with a different capacity and travel time between POIs) and multiple refueling depots, a set of target POIs to visit and stochastic travel times or fuel consumption, find a path for each UGV such that each POI site is visited by at most one UGV, and the overall distance traveled by the UGVs is minimized.

4.1. OBJECTIVE FUNCTION

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

$$\min \ C, \text{ where } C \triangleq \sum_{\substack{(i,j)\in E \\ m\in M}} \hat{f}_{ij}^m x_{ij}^m + \mathbb{E}_{\Omega}\left[\beta(x, f)\right]$$

$$= \sum_{\substack{(i,j)\in E \\ m\in M}} \hat{f}_{ij}^m x_{ij}^m + \sum_{\omega\in\Omega} p(\omega)\beta(x, f(\omega)). \tag{1}$$

4.2. FIRSTSTAGE ROUTING CONSTRAINT

The constraints for the first-stage enforce the routing constraints, i.e., the requirements that each POI $i \in T$ should be visited by only one of the UGVs and that each UGV 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 UGV to traverse any edge $(i, j) \in E$. The first-stage routing constraints are as follows:

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

$$x^m(\delta^+(d_0)) = 1, \quad \forall m \in M, \tag{2b}$$

$$x^m(\delta^-(d_0)) = 1, \quad \forall m \in M, \tag{2c}$$

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

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

$$x^m(\delta^+(i)) = x^m(\delta^-(i)) \quad \forall i \in T, m \in M, \tag{2f}$$

$$\sum_{j\in V} z_{ij} - \sum_{j\in V} z_{ji} = \sum_{j\in V} \hat{f}_{ij}^m x_{ij}^m \quad \forall i \in T, \forall m \in M, \tag{3a}$$

$$z_{di} = \hat{f}_{di}^m x_{di}^m \quad \forall i \in V, d \in D, m \in M, \tag{3b}$$

$$0 \leqslant z_{ij} \leqslant F_m x_{ij}^m \quad \forall (i,j) \in E, m \in M, \tag{3c}$$

$$x_{ij}^m \in \{0, 1\} \quad \forall (i,j) \in E, m \in M. \tag{3d}$$

Constraint (2a) forces the in-degree and out-degree of each refueling station to be equal. Constraints (2b) and (2c) ensure that all the UGVs leave and return to depot d_0. Constraint (2d) ensures that a feasible solution is connected. For each POI i, the pair of constraints in (2e) require that some UGV visits the POI i. Constraint (2f) forces the in-degree and out-degree of each POI to be equal. Constraint (3a) eliminates subtours of the targets and also defines the flow variables z_{ij} for each edge $(i, j) \in E$ using the nominal fuel consumption values $\hat{f}_{ij}$. Constraints (3b) – (3c) together impose the fuel constraints on the routes for all the UGVs. Finally, constraint (3d) imposes binary restrictions on the decision variables x_{ij}^m.

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 4 of 10

5. FORMULATION 2

Given a team of UGVs and a subset of it is randomly available for the mission, and a set of POIs sites to visit, find a path for each UGV such that each POI is visited by at most by one UGV, and an objective based on the incentives of POIs visited by the UGVs is maximized.

5.1. OBJECTIVE FUNCTION

The objective function for the two-stage stochastic programming model is the sum of the first-stage profit and the expected second-stage profits. The second-stage profit for a realization $\omega \in \Omega$ of the fuel consumption of the UGV is the change in profit for the realization ω. The reduction in profits is a function of the first-stage routing decision x and the realization ω. Letting the recourse cost be denoted by $\beta(x, z, f(\omega))$, the objective function for the two-stage stochastic optimization problem is given by

$$\max\; C, \text{ where } C \triangleq \sum_{\substack{(i,j)\in E: j\in T \\ m\in M}} p_j^m x_{ij}^m + \mathbb{E}_\Omega\left[\beta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{f})\right]$$
$$= \sum_{\substack{(i,j)\in E: j\in T \\ m\in M}} p_j^m x_{ij}^m + \sum_{\omega\in\Omega} p(\omega)\beta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{f}(\omega)). \tag{6}$$

5.2. FIRST STAGE ROUTING CONSTRAINTS

The constraints for the first-stage enforce the routing constraints, i.e., the requirements that each POI in T can be visited at least once by some UGV and that each UGV 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 UGV to traverse any edge $(i, j) \in$ E. The first-stage routing constraints are as follows:

$$x^m(\delta^+(d)) = x^m(\delta^-(d)) \quad \forall d \in D \setminus \{d_0\}, m \in M, \tag{7a}$$

$$x^m(\delta^+(d_0)) = 1, \quad \forall m \in M, \tag{7b}$$

$$x^m(\delta^-(d_0)) = 1, \quad \forall m \in M, \tag{7c}$$

$$x^m(S) \leqslant |S| - 1 \quad \forall S \subset V \setminus \{d_0\} : S \cap T \cap D \neq \emptyset, \tag{7d}$$

$$\sum_{i\in V, m\in M} x_{ij}^m \leqslant 1 \text{ and } \sum_{i\in V, m\in M} x_{ji}^m \leqslant 1 \quad \forall j \in T, \tag{7e}$$

$$x^m(\delta^+(i)) = x^m(\delta^-(i)) \quad \forall i \in T, m \in M, \tag{7f}$$

$$\sum_{j\in V} z_{ij}^m - \sum_{j\in V} z_{ji}^m = \sum_{j\in V} f_{ij}^m x_{ij}^m \quad \forall i \in T, m \in M, \tag{7g}$$

$$z_{di}^m = f_{di}^m x_{di}^m \quad \forall i \in V, d \in D, m \in M, \tag{7h}$$

$$0 \leqslant \sum_{(i,j)\in E} f_{ij}^m x_{ij}^m \leqslant F_m \quad \forall (i,j) \in E, m \in M, \tag{7i}$$

$$x_{ij}^m \in \{0, 1\}, \; z_{ij}^m \leq F_m \quad \forall (i,j) \in E, m \in M.. \tag{7j}$$

Constraint (7a) forces the in-degree and out-degree of each refueling station to be equal. Constraints (7b) and (7c) ensure that all the UGVs leave and return to depot d_0, where m is the number of UGVs. Constraint (7d) ensures that a feasible solution is connected. For each target i, the pair of constraints in (7e) state that some UGV visits the POI i only once. Constraint (7f) forces the in-degree and out-degree of each POI station to be equal. Constraints (7g) -(7h) eliminates sub-tours of the POIs and also defines the flow variables z_{ij}^m for each edge $(i, j) \in$ E and UV h. Constraints (7i) impose the fuel capacity constraints on the routes for all the UGVs. Finally, constraint (7j) imposes restrictions on the decision variables.

5.3. SECOND – STAGE CONSTRAINTS

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

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 5 of 10

$$\beta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{f}(\omega)) = \max \sum_{\substack{(i,j) \in E \\ m \in M}} -p_j^m v_{ij}^m(\omega) \qquad (8a)$$

subject to:

$$\sum_{j \in V} f_{ij}^m v_{ij}^m(\omega) = \sum_{j \in V} z_{ij}^m - \sum_{j \in V} z_{ji}^m - \alpha_m(\omega) \sum_{j \in V} f_{ij}^m x_i^l$$

$$\forall i \in T, m \in M, \qquad (8b)$$

$$v_{ij}^m(\omega) \leq x_{ij}^m \quad \forall (i,j) \in E, m \in M, \qquad (8c)$$

$$f_{di}^m v_{di}^m(\omega) = z_{di}^m - \alpha_m(\omega).f_{di}^m x_{di}^m \quad \forall i \in V, d \in D,$$

$$\forall m \in M, \qquad (8d)$$

$$v_{ij}^m(\omega) \in \{0,1\} \quad \forall (i,j) \in E, m \in M. \qquad (8e)$$

$$\forall m \in M,$$

In the second-stage, $\alpha_m(\omega)$ takes a value of 1 or 0 denoting the availability of an UGV m for the scenario ω or not. Variable $v_{ij}^m(\omega)$ maintains the feasibility of the constraints (8b) -(8c) for the given first-stage values x and z. Constraint (8c) states the dependence of x_{ij}^m and $v_{ij}^m(\omega)$, and finally binary restrictions for $v_{ij}^m(\omega)$ are presented in (8e). Let the relaxed recourse problem for $\beta(x, z, f(\omega))$ be represented as $\beta_r(x, z, f(\omega))$. In $\beta_r(x, z, f(\omega))$, the constraints (8e) are replaced by $0 \leq v_{ijh}(\omega) \leq 1$.

THEOREM5.1 The objective values of $\beta(x, z, f(\omega))$ and $\beta_r(x, z, f(\omega))$ are same.

6. ALGORITHM

The constraints (7i) are the typical knapsack constraints and the formulation will resemble 'orienteering problem'. We will refer the formulation with and without knapsack constraints as TS-OP and TS, respectively. In this section, we present a decomposition algorithm to solve problem TS and TS-OP. The formulations TS and its variants can be provided to any commercial branch-and-cut solvers to obtain an optimal solution. However, observe that the formulations will contain constraint (7d) to ensure any feasible solution is connected. The number of such constraints is exponential and it may not be computationally efficient to enumerate all these constraints and provide them upfront to the solvers. Additionally, stochastic integer programs are large in scale due to the variables and constraints in the scenarios and they require decomposition algorithms to exploit the special structure of the problem. These challenges and opportunities motivated us to design a decomposition algorithm to solve the instances for TS and its variants.

6.1. DECOMPOSITION ALGORITHM

The decomposition algorithm is a variant of L-shaped algorithm where the deterministic parameters are used to obtain first-stage solutions, and then the second-stage programs are solved based on the obtained first-stage solutions. Then the optimality cuts are generated and added to the first-stage program to approximate the value function of the second-stage cost. The dual information of all the realization of the random data are used to generate the optimality cuts for the first-stage. The use of L-shaped method for TS is possible only due to the theorem (5.1). Otherwise due to the binary restrictions for second-stage variables, the value function will be non-convex and lower semi-continuous in general, and a direct use of L-shaped method is not possible. The first-stage problem is solved as a mixed-integer program with binary restrictions for x variables and by theorem (5.1), the second-stage programs are solved as linear programs.

6.1.1 PROBLEM REFORMULATION

For the sake of decomposition, the first-stage problem (6) -(7j) is reformulated as the following master problem (MP) and we add an unrestricted variable θ to the first-stage program. In the formulation TS-MP, let $h(\omega)$ and $\mu(\omega)$ represent the right-hand side and dual values for the second-stage constraints (8b) - (8e) and $v_{ijh}(\omega) \in \{0,1\}$ are replaced by $0 \leq v_{ijh}(\omega) \leq 1$. Similarly, $T(\omega)$ and $T^I(\omega)$ represent the co-efficient matrices for the

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 6 of 10

first-stage variables x and z in the second stage constraint (8b) -(8e), respectively. The first-stage master program for the decomposition algorithm TSMP is given as follows:

$$z^k = \max \sum_{\substack{(i,j)\in E:j\in T \\ m\in M}} p_j^m x_{ij}^m + \theta \qquad (9a)$$

Subject to:

$$(7a) - (7j), \qquad (9b)$$

$$\sum_{\omega\in\Omega} \sum_{\substack{(i,j)\in E, \\ m\in M}} (((\pi_1(\omega)^{t\top} T_1) + (\pi_2(\omega)^{t\top} T_2)$$

$$+ (\pi_3(\omega)^{t\top} T_3))x_{ij}^m + ((\pi_1(\omega)^{t\top} S_1) + (\pi_3(\omega)^{t\top} S_3))z_{ij}^m)$$

$$+ \theta \le \sum_{\omega\in\Omega} \pi(\omega)^t h(\omega), \ t \in 1, ..., k \qquad (10a)$$

$$x_{ij}^m \in \{0,1\}, \ z_{ij}^m \le F_m \quad \forall (i,j) \in E, m \in M, \ \theta \in \mathbb{R}. \qquad (10b)$$

In the master problem (10), for a scenario ω, $\pi_1(\omega)$, $\pi_2(\omega)$, and $\pi_3(\omega)$ are the dual vectors of the constraints (8b), (8c), and (8d), respectively. Similarly, T1, T2, and T3 represent the co-efficient matrices for the variables x_{ij}^m in the constraints (8b), (8c), and (8d), respectively. Also, S1 and S3 represent the co-efficient matrices for the variables z_{ij}^m in the constraints (8b), and (8d), respectively. Finally, $\pi(\omega)$ and $h(\omega)$ represent the dual vector and right-hand side for the constraints (8b) -(8e). θ is an unrestricted decision variable. Constraints (10a) are the optimality cuts, which are computed based on the optimal dual solution of all the subproblems given as the second-stage problem $\beta_r(x, z, f(\omega))$. Optimality cuts approximate the value function of the second stage subproblems $\beta_r(x, z, f(\omega))$. It should be noted that the model TS-MP has relatively complete recourse property, i.e., $\beta_r(x, z, f(\omega)) < \infty$ for any x_{ij}^m and z_{ij}^m.

We would like to emphasize that the number of optimality cuts generated from second-stage dual values can be a single cut or multi-cut. In single-cut, a cut is generated across all the second-stage problems and in multi-cut, each second-stage program will be approximated by a cut in the first-stage program. In our computational experiments, we adopted a single cut approach as we do not want to stress the first-stage problem as it already consists of binary variables and sub-tour elimination constraints (7d). Also, the presented algorithm can be extended to instances of TS-OP as the changes occur only in first-stage constraints.

Algorithm 1 Branch and Cut Algorithm

1: **Step 0.** Initialize.

2: $n \leftarrow 0$, $lb \leftarrow -\infty$, ϵ is an user defined parameter and $ub \leftarrow \infty$, and x^0, z^0 are obtained as follows:
$\text{argmax}\{\sum_{(i,j)\in E, m\in M} p_j^m x_{ij}^m | x, z \in (7a) - (7j)\}$.
▷ Initial solution

3: **Step 1.** Solve second-stage programs.

4: Solve $\beta_r(x, z, f(\omega))$ for each $\omega \in \Omega$, and obtain dual solution $\pi(\omega)$ for each second-stage program.
▷ Solve sub-problems for given first-stage solution

5: **Step 2.** Optimality cut. Based on the dual values $\pi(\omega)$s from the sub-problems, generate an optimality cut (10a) and add it to the master problem. ▷ Sub-problems' objective function are approximated

6: **Step 3.** Obtain upper bounds.

7: Solve the master program TS-MP with the new optimality cut and let the objection function value be u^n. Set $ub \leftarrow min\{u^n, ub\}$. ▷ Sub-problems' objective function are approximated

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 7 of 10

8: **Step 4.** Add sub-tour elimination constraints.

9: Check for strongly connected components and if there are any sub-tours, add the constraint (7d). Go to Step 3. ▷ Check whether the paths are connected

10: **Step 5.** Update bounds.

11: $v^n \leftarrow \{\sum_{(i,j)\in E, m\in M} p_j^m x_{ij}^m | x, z \in (7a)-(7j)\} + \mathbb{E}_\Omega \left[\beta(x, z, f)\right]$, and set $lb \leftarrow max\{v^n, lb\}$. If lb is updated, set incumbent solution to $x^* \leftarrow x^n$, and $z^* \leftarrow z^n$. ▷ Update the incumbent solution

12: **Step 6.** Termination. If $ub - lb < \epsilon|ub|$ then stop, x^*, and z^* are the optimal solutions, else set $n \leftarrow n + 1$ and return to step 1. ▷ termination condition

6.2. SUB-TOUR ELIMINATION CONSTRAINTS

In our algorithm, we relax the constraints (7d) from the formulation, and whenever the first-stage problem obtains an integer feasible solution to this relaxed problem, we check if any of the constraints (7d) are violated by the integer feasible solution. If so, we add the infeasible constraint to the first-stage problem. This process of adding constraints to the problem sequentially has been observed to be computationally efficient for the TSP, VRP and a huge number of their variants.

Now, we will detail the algorithm used to find a constraint (7d) that is violated for a given integer feasible solution to the relaxed problem. A violated constraint (7d) can be described by a subset of vertices S ⊂ V\{d_0} such that S ∩ D ≠ ∅ and x(S) = |S| for every d ∈ S ∩ D. We find the strongly connected components of S. Every strongly connected component that does not contain the depot is a subset S of V\{d_0} which violates the constraint (7d). We add all these infeasible

constraints and continue solving the original problem. Many off-the-shelf commercial solvers provide a feature called "solver callbacks" to implement such an algorithm into its branch-and-cut framework.

7. COMPUTATIONAL EXPERIMENTS

In this section, we discuss the computational performance of the branch-and-cut algorithm for formulations presented in the Sec. 3. The mixed-integer linear programs were implemented in Java, using the traditional branch-and-cut framework and the solver callback functionality of CPLEX version 12.6.2. All the simulations were performed on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @2.53 GHz, 12 GB RAM). The computation times reported are expressed in seconds, and we imposed a time limit of 3,600 seconds for each run of the algorithm. The performance of the algorithm was tested with randomly generated test instances.

Instance generation
The problem instances were randomly generated in a square grid of size [100,100]. The number of refueling stations was set to 4 and the locations of the depot and all the refueling stations were fixed a priori for all the test instances. The number of POIs varies from 10 to 40 in steps of five, while their locations were uniformly distributed in the square grid; for each $|T| \in \{10, 15, 20, 25, 30, 25, 40\}$, we generated five random instances. For each of the above generated instances, the number of UGVs in the depot was 3, and the fuel capacity of the UGVs, F, was varied linearly with a parameter λ. λ is defined as the maximum distance between the depot and any POI. The fuel capacity F was assigned a value from the set $\{2.25\ \lambda, 2.5\ \lambda, 2.75\ \lambda, 3\ \lambda\}$. The travel costs and the fuel consumed to travel between any pair POIs vertices were assumed to be directly proportional to the Euclidean distances between the pair and rounded down to the nearest integer.

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.
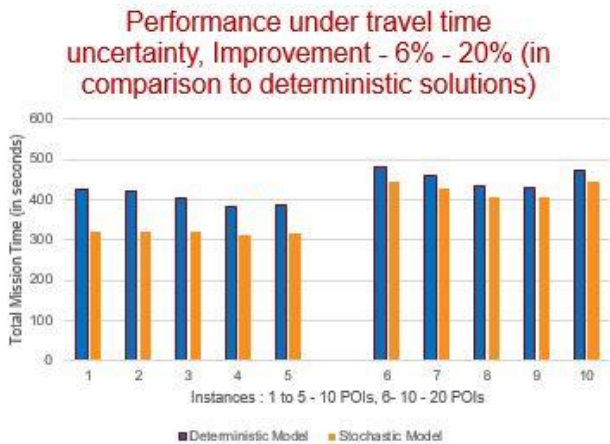
Page 8 of 10

Figure 2: This figure demonstrates the use of two-stage model when uncertainties are considered for travel time among depots and points of interests. Gamma distribution is used to characterize uncertainty in travel time and continuous beta distribution for uncertainty in fuel consumption. The results from stochastic model are compared with deterministic model. For the overall performance under travel time uncertainty, the average improvement is between 6% and 20% (in comparison to deterministic solutions)

8. CONCLUSION

Path planning problem for manned and unmanned UGVs is an important area of research for efficient use of UGVs. This paper presents two different stochastic programming models to address uncertainties in travel time and availability of UGVs. To overcome the computational complexity, a decomposition algorithm is a presented. Computational experiments are performed to demonstrate the usefulness of stochastic models over their deterministic versions.
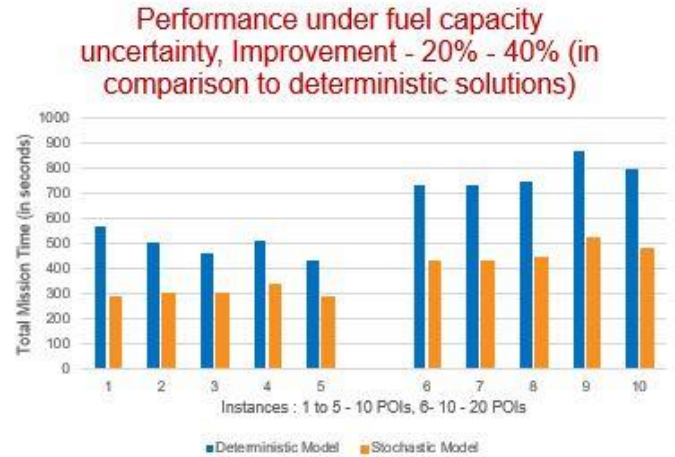

Figure 3: This figure demonstrates the use of two-stage model when uncertainties are considered for fuel capacity. Gamma distribution is used to characterize uncertainty in travel time and continuous beta distribution for uncertainty in fuel consumption. The results from stochastic model are compared with deterministic model. For the overall performance under fuel capacity uncertainty, the average improvement is between 20% and 40% (in comparison to deterministic solutions)
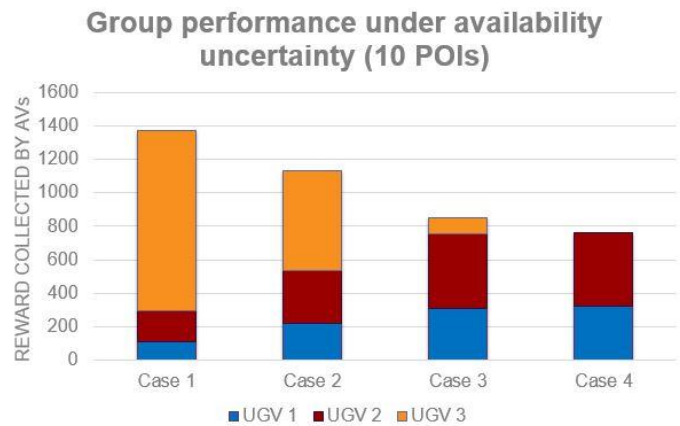

Figure 4: The results are for the formulation presented in the section 5. Each POI has a reward and is visited by at most one UGV, and the objective is to maximize the total reward collected by the UGVs. UGV 3's availability with probability are 1, 0.75, 0.25, and 0 in Cases 1, 2, 3, and 4, respectively. The results are for 10 POIs.

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.
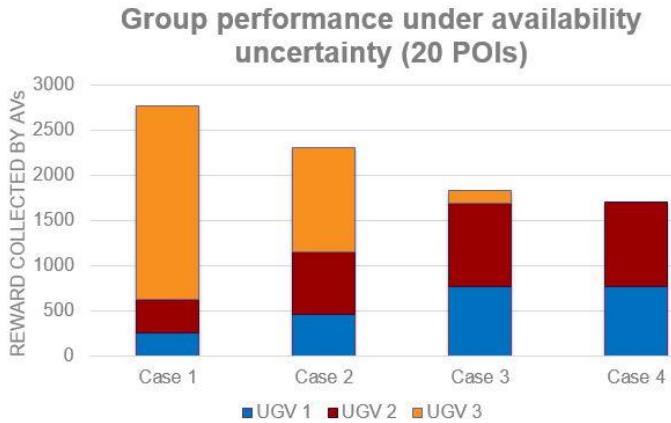
Page 9 of 10

Figure 5: The results are for the formulation presented in the section 5. Each POI has a reward and is visited by at most one UGV, and the objective is to maximize the total reward collected by the UGVs. UGV 3's availability with probability are 1, 0.75, 0.25, and 0 in Cases 1, 2, 3, and 4, respectively. The results are for 20 POIs.

9. LEGAL STATEMENT

DISTRIBUTION A. Approved for public release: distribution unlimited. OPSEC# 2818.

1. REFERENCES

[1] Bellingham, J.; Tillerson, M.; Richards, A.; HOW, J. P. Multi-task allocation and path planning for cooperating uavs. In: Cooperative control: models, applications and algorithms. [S.l.]: Springer, 2003. p. 23–41.

[2] Casbeer, D. W.; Beard, R. W.; Mclain, T. W.; LI, S.-M.; Mehra, R. K. Forest fire monitoring with multiple small uavs. In: IEEE. Proceedings of the 2005, American Control Conference, 2005. [S.l.], 2005. p. 3530–3535.

[3] Dadkhah, N.; Mettler, B. Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance. Journal of Intelligent & Robotic Systems, Springer, v. 65, n. 1-4, p. 233–246, 2012.

[4] Evers, L.; Barros, A. I.; Monsuur, H.; Wagelmans, A. Online stochastic uav mission planning with time windows and time-sensitive targets. European Journal of Operational Research, Elsevier, v. 238, n. 1, p. 348–362, 2014.

[5] Evers, L.; Dollevoet, T.; Barros, A. I.; Monsuur, H. Robust uav mission planning. Annals of Operations Research, Springer, v. 222, n. 1, p. 293–315, 2014.

[6] Krishnamoorthy, K.; Casbeer, D.; Chandler, P.; Pachter, M.; Darbha, S. Uav search & capture of a moving ground target under delayed information. In: IEEE. 2012 IEEE 51st IEEE Conference on Decision and Control (CDC). [S.l.], 2012. p. 3092–3097.

[7] Li, X.; Epureanu, B. I. Robustness and adaptability analysis of future military modular fleet operation system. In: American Society of Mechanical Engineers. ASME 2017 Dynamic Systems and Control Conference. [S.l.], 2017. p. V002T05A003–V002T05A003.

[8] Sammuelson, D. A. Changing the war with analytics. OR/MS Today, v. 37, n. 1, p. 30–35, 2010.

[9] Zaloga, S. J. Unmanned aerial vehicles: robotic air warfare 1917–2007. [S.l.]: Bloomsbury Publishing, 2011.

Stochastic Programming Models for Autonomous Ground Vehicles, Venkatachalam, et al.

Page 10 of 10