

# Path Planning for Cooperative Routing of Air-Ground Vehicles

Satyanarayana G. Manyam<sup>1</sup>, David W. Casbeer<sup>2</sup>, and Kaarthik Sundar<sup>3</sup>

**Abstract**—We consider a cooperative vehicle routing problem for surveillance and reconnaissance missions with communication constraints between the vehicles. We propose a framework which involves a ground vehicle and an aerial vehicle; the vehicles travel cooperatively satisfying the communication limits, and visit a set of targets. We present a mixed integer linear programming (MILP) formulation and develop a branch-and-cut algorithm to solve the path planning problem for the ground and air vehicles. The effectiveness of the proposed approach is corroborated through extensive computational experiments on several randomly generated instances.

## I. INTRODUCTION

In this paper, we present a path planning problem involving an Unmanned Aerial Vehicle (UAV) and a ground vehicle for intelligence, surveillance and reconnaissance (ISR) missions. UAVs are being routinely used in military applications such as border patrol, reconnaissance, and surveillance expeditions, and civilian applications [1]–[3]. They are prime candidates for ISR missions due to their several advantages such as portability and low risk, to name a few. A typical ISR mission would require the UAVs to collect images, videos, or sensor data and transmit them to a ground/base station. The data collected in these ISR missions are most often very time sensitive and ideally, the data would be useful only if it is processed in real-time or near real-time. We propose a framework to meet this objective, and solve the underlying problem of planning paths for the UAV and the ground vehicle.

Cooperative control and path planning for a team of vehicles (UAVs or UAVs together with ground vehicles) has been a problem of interest and has received wide attention during the past decade (see [4]–[10]). The problem of communication-constrained vehicle routing for a team of vehicles has been addressed in [11]–[14]. More specifically, authors in [11] study the problem of deploying a team of mobile agents to periodically monitor several points of interest. Authors in [12] perform an experimental study of the strategies to maintain end-to-end communication links for a team of robots in reconnaissance missions. Heuristic approaches for a variation of the traditional vehicle routing problems with a simple communication model has been studied in [14]. In [10], the authors considered a forest mapping application involving an UAV and a ground vehicle; they develop algorithms to plan routes for the fuel constrained

UAV and the ground vehicle, which can act as refueling station for the UAV during the mission. This is the first work in the literature that proposes an approach using one UAV and one ground vehicle.

In this article, we propose a reconnaissance and data collection methodology using a UAV and a ground vehicle. The two vehicles are subject to communication constraints and they work cooperatively to accomplish the mission. The choice of using a ground vehicle and a UAV for ISR missions is motivated by the lack of roads and the presence of geographical obstacles such as rivers, lakes, mountains, etc. that must be circumvented to reach target locations. In the scenarios where ground vehicle may not be able to reach certain targets, a UAV could be used to gather information. However, due to size, weight, and power restrictions the UAV would be unable to transmit the data from remote targets to the ground station. Furthermore, in ideal situations, the ground vehicle would have a communication link to the ground station, perhaps through satellite or by other means. Hence, in this paper, the UAV is required to stay connected to the ground vehicle, under the thought that data can be instantaneously transmitted to the ground station. We also do not impose any specific constraint on communication for the ground vehicle, and thus in cases where communication is degraded and the ground vehicle cannot maintain contact with the ground station, the cooperative routing problem here will still yield a viable solution. In this case, one could think of this problem as a cooperative routing problem to minimize the effort taken by the vehicles to acquire the data and then ferry it back to the ground station. The problem is formally defined in the following section.

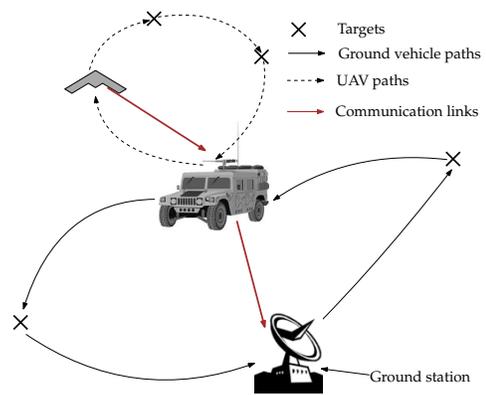


Fig. 1. Air vehicle - Ground vehicles paths

<sup>1</sup>National Research Council Fellow, Air Force Research Laboratory, Dayton-OH, 45433, msgupta@gmail.com.

<sup>2</sup>Research Scientist, Control Science Center of Excellence, Air Force Research Laboratory, WPAFB, OH, 45433, david.casbeer@us.af.mil.

<sup>3</sup>Graduate Student, Dept. of Mechanical Engineering, Texas A&M University, College Station, TX 77843.

### A. Path planning problem

The path planning problem considered in this article is formally defined as follows: we are given the locations of a set of targets and a ground station where the ground vehicle is initially stationed. The UAV is being carried by the ground vehicle. For each vehicle, we are given a travel cost between every pair of targets. This cost could be the Euclidean distance between the pair of targets or could also depend on the terrain. Given this, the objective of the problem is to determine paths of minimum cost for the ground vehicle and the UAV such that following conditions are satisfied:

- every target is visited either by the ground vehicle or by the UAV, and
- the UAV can always establish a reliable communication link with the ground vehicle.

The sequence of actions for a feasible mission is as follows. The ground vehicle starts at the ground station and visits the first target in its route, then the UAV is deployed from the first target and it collects the data from a subset of targets according to its plan and returns to the ground vehicle. Then the ground vehicle proceeds to the next stop. This process is repeated until the data is collected from all of the targets. An illustration of a typical route is shown in Fig. 1. Once all the targets are visited, the ground vehicle carrying the UAV returns to the ground station. More specifically, the objective of determining the routes for the vehicles involves (i) identifying the stops and order to visit them for the ground vehicle and (ii) at each stop, identifying the subset of targets and order in which the UAV has to visit them. Furthermore, to ensure that the UAV can always establish a communication link with the ground vehicle, we enforce the constraint that the UAV has to stay within a distance of  $R$  units from the ground vehicle at all times during the mission. We call this problem the *cooperative air-ground vehicle routing problem* (CAGVRP). A feasible solution to the problem is shown in Fig. 2.

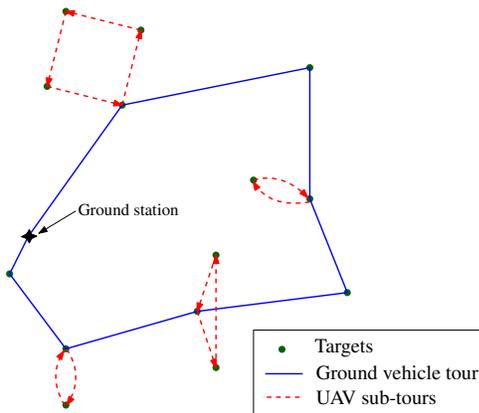


Fig. 2. A sample tour of the CAGVRP

### B. Related Literature

The CAGVRP is NP-hard because it is a generalization of the traveling salesman problem. Authors in [9] propose a

framework similar to the CAGVRP involving a truck, traveling on a road network and a quadrotor for a package delivery problem in urban environments. This problem differs from the CAGVRP in two aspects: (i) the ground vehicle or the truck is restricted to travel along a road network and (ii) the quadrotor has to return to the ground vehicle after each delivery. The authors present a transformation algorithm to transform the problem to a generalized traveling salesman problem, whereas the focus of this paper is to develop algorithms to obtain an optimal solution to the CAGVRP. CAGVRP also resembles the two echelon vehicle routing problem [15]; the key difference is in CAGVRP the locations at which the UAV tours originate are not known, which makes it even harder to solve.

Other problems addressed in the literature that are similar to the CAGVRP are the ring-star problem [16]–[19] and the hierarchical ring-network problem [20]–[25]. The ring-star problem aims to find a minimum cost cycle (or ring) through a subset of targets, and the targets that do not lie on the cycle should be assigned to one of the targets in the cycle. Algorithms to obtain an optimal solution based on the branch-and-cut paradigm are presented for the ring-star problem in [16], for its multiple-depot variant in [17], and for the capacitated variants in [18], [19]. In CAGVRP, not only must the targets that does not lie on the main ring be assigned to the targets on the ring, but there must be sub-tours chosen for all the off-ring targets.

Another closely related problem is the hierarchical ring-network problem (HRNP) which aims to find a hierarchical two-layer ring network. The top layer consists of a federal ring (analogous to the ground vehicle tour) which establishes connection between a number of node-disjoint metro rings (analogous to the UAV sub-tours) in the bottom layer. Authors in [21] present heuristics and an approximation algorithm to solve the HRNP; they assume that the number of metro rings and the nodes at which the metro rings are attached to the federal ring are given. In [24], heuristics and enumeration methods are given to solve the HRNP, where a certain demand must be satisfied between every pair of targets. Heuristics to solve a variant of the HRNP are presented in [20].

Another variant of HRNP, where the bottom layer could be rings or ring-stars is solved in [22]. The authors assume the number of possible local rings are given. The authors in [25] use branch-and-price algorithm to solve the HRN problem with demands to be satisfied between every pair of targets. Their solution procedure involved two steps: in the first step, they solve a modified HRNP which ignores the cost of the federal ring and design the metro rings; in the second step, they solve a generalized traveling salesman problem on the metro rings. In all the aforementioned papers, the cost of the federal ring is either ignored or assumed to be equal to that of the metro ring. The CAGVRP differs from the HRNP in the following aspects: (i) the travel costs for the ground vehicle and the UAV are different, (ii) we have an additional distance constraint *i.e.*, the distance between the UAV and the ground vehicle has to be within  $R$  units throughout the mission, and

(iii) we attempt to solve the coupled problem of finding paths for the ground vehicle and the UAV to minimize the total travel cost.

This article is organized as follows: We present the problem formulated as mixed-integer linear program (MILP) in Section II. In Section III, we explain the branch-and-cut framework used to solve the MILP formulation. Computational results are reported in Section IV and conclusions are made in Section V.

## II. PROBLEM FORMULATION

This section presents a mixed integer linear programming formulation for the CAGVRP. Let  $T$  denote the set of targets  $\{t_0, \dots, t_n\}$ ;  $t_0$  is the ground station. The CAGVRP is defined on a mixed graph  $G = (T, E \cup A)$ , where  $E$  and  $A$  are a set of undirected edges and directed arcs, respectively, joining any two targets in  $T$ . Each edge  $e = (i, j) \in E$  is associated with a non-negative cost  $c_e$  required for the ground vehicle to traverse the edge  $e$  (if  $e$  connects the vertices  $i$  and  $j$ , then  $(i, j)$  and  $e$  will be used interchangeably to denote the same edge in  $E$ ). Similarly, each arc  $[i, j] \in A$  is associated with a non-negative cost  $d_{ij}$  required for the UAV to travel from target  $i$  to target  $j$ . We also associate with each arc  $[i, j] \in A$ , an auxiliary cost  $f_{ij}$  to enforce the mission constraint that the UAV should always be within a distance of  $R$  units from the ground vehicle. For any  $i, j \in T$ ,  $f_{ij}$  is set to 0 if the Euclidean distance between targets  $i$  and  $j$  is less than  $R$  units, and is set to an arbitrarily large value otherwise.

We associate vectors  $\mathbf{x} \in \mathbb{R}^{|E|}$ ,  $\mathbf{w} \in \mathbb{R}^{|A|}$  and  $\mathbf{y} \in \mathbb{R}^{|A|}$  with each feasible solution  $\mathcal{F}$ . The component  $x_e$  of  $\mathbf{x}$ , associated with edge  $e \in E$ , is a binary variable which takes a value 1 if the edge  $e$  is traversed by the ground vehicle, and 0 otherwise. Each component  $w_{ij}$  of  $\mathbf{w}$ , associated with the directed arc  $[i, j] \in A$  is a binary variable; it denotes if the arc is present in a UAV sub-tour. Similarly,  $y_{ij}$ , a component of  $\mathbf{y}$  is a binary assignment variable that takes a value 1 if the target  $i$  is assigned to the UAV sub-tour that originates from the target  $j$ , and 0 otherwise. Note that if a target  $i$  is visited by a ground vehicle then the assignment variable  $y_{ii}$  is equal to 1 (the target is assigned to itself).

For any  $S \subset T$ , we define  $\gamma(S) = \{(i, j) \in E : i, j \in S\}$ ,  $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$ ,  $\delta^+(S) = \{[i, j] \in A : i \in S, j \notin S\}$ , and  $\delta^-(S) = \{[j, i] \in A : i \in S, j \notin S\}$ . If  $S = \{i\}$ , we simply write  $\delta(i)$ ,  $\delta^+(i)$  and  $\delta^-(i)$  instead of  $\delta(\{i\})$ ,  $\delta^+(\{i\})$  and  $\delta^-(\{i\})$  respectively. A mixed-integer program formulated using the above variables for the CAGVRP is as follows:

$$(\mathcal{F}_1) \quad \text{Minimize} \quad \sum_{e \in E} c_e x_e + \sum_{(i,j) \in A} d_{ij} w_{ij} + \sum_{(i,j) \in A} f_{ij} y_{ij}$$

Subject to

$$\sum_{e \in \delta(i)} x_e = 2y_{ii} \quad \forall i \in T, \quad (1)$$

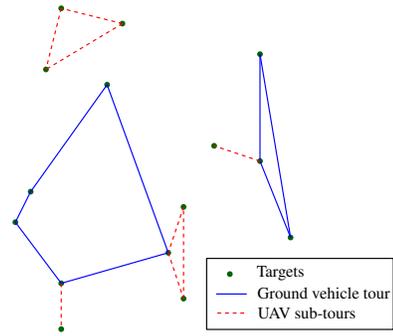


Fig. 3. Infeasible solution to the CAGVRP because the ground vehicle tour consists of two disjoint sub-tours and there exists a separated UAV sub-tour.

$$\sum_{j \in T} w_{ij} = 1, \quad \forall i \in T, \quad (2)$$

$$\sum_{i \in T} w_{ij} = 1, \quad \forall j \in T, \quad (3)$$

$$w_{ij} \leq \sum_{k \in T} y_{ik} y_{jk}, \quad \forall [i, j] \in A, \quad (4)$$

$$\sum_{e \in \delta(S)} x_e \geq 2 \sum_{j \in S} y_{ij}, \quad \forall i \in S, t_0 \notin S, S \subset T, \quad (5)$$

$$\sum_{[i,j] \in \delta^+(S)} w_{ij} \geq 1 - \sum_{j \in S} y_{ij}, \quad \forall i \in S, S \subset T, \quad (6)$$

$$\sum_{[i,j] \in \delta^-(S)} w_{ij} \geq 1 - \sum_{j \in S} y_{ij}, \quad \forall i \in S, S \subset T, \quad (7)$$

$$x_e \in \{0, 1\}, \quad \forall e \in E, \quad \text{and} \quad (8)$$

$$y_{ij}, w_{ij} \in \{0, 1\} \quad \forall [i, j] \in A. \quad (9)$$

The objective of the problem is to minimize the total cost of travel for the ground vehicle and the UAV. In the above formulation, the constraints in (1) ensure that the number of undirected ground vehicle edges incident on any target  $i \in T$  is equal to 2 if and only if the target  $i$  is assigned to itself ( $y_{ii} = 1$ ). The UAV sub-tours connectivity constraints are enforced using the equations (2) and (3). For each target, these constraints ensure that there are two UAV sub-tour edges incident on the target, one incoming and one outgoing edge. This is in contrast to the ground vehicle routes. The UAV has to visit every target  $i \in T$  irrespective of the value of  $y_{ii}$ . When the value of the  $y_{ii} = 1$ , the constraints in (2) and (3) would result in a trivial UAV sub-tour *i.e.*,  $w_{ii} = 1$ . The constraints in Eq. (4) ensure that a sub-tour edge of the UAV can exist between two targets only if those two targets are assigned to the same target. One can observe that these constraints are non-linear. We also note that the degree constraints in Eq. (1) are not sufficient to formulate the route for the ground vehicle.

With just the degree constraints and the assignment con-

straints, the minimization problem may return a solution with separated tours for the ground vehicle and/or isolated sub-tours for the UAV. An instance of such an infeasible solution is shown in Fig. 3. In general, there are three different ways to eliminate these isolated tours/sub-tours. This is done by introducing a new set of constraints *viz.* *MTZ* constraints or flow constraints or sub-tour elimination constraints (SEC) [26]. For vehicle routing problems, the cut based sub-tour elimination (SEC) constraints are known to outperform *MTZ* and flow constraints computationally [27]. Therefore we use SEC constraints (5), (6) and (7) to eliminate such solutions containing separated tours for the ground vehicle and for the UAV tours. These constraints ensure that for any sub-set of targets  $S$ , such that all the targets in  $S$  are assigned to a target not in  $S$ , then a UAV sub-tour containing only the targets in  $S$  cannot exist. The constraints (6) and (7) mandate that there should be an arc coming into the set  $S$  and an arc going out of the set  $S$ . The novelty of this model is it addresses the connectivity of the tours without *MTZ* or flow constraints, rather this model uses SEC constraints to ensure a connected tours. The exponential number of SEC constraints are not explicitly enumerated here but are addressed in the branch-and-cut framework only when they are violated.

As mentioned previously, the formulation  $\mathcal{F}_1$  is non-linear due to the constraints in (4). These constraints can be linearized by introducing an auxiliary variable  $z_{ijk}$  for every triplet of targets  $i, j, k \in T$ .

**Proposition II.1.** *The constraints in (4) can be reformulated using the following constraints:*

$$w_{ij} \leq \sum_{k \in T} z_{ijk}, \quad \forall [i, j] \in A, \quad (10)$$

$$z_{ijk} \leq y_{ik} \quad \forall i, j, k \in T, \quad (11)$$

$$z_{ijk} \leq y_{jk} \quad \forall i, j, k \in T, \text{ and} \quad (12)$$

$$z_{ijk} \geq y_{ik} + y_{jk} - 1 \quad \forall i, j, k \in T. \quad (13)$$

*Proof.* We prove the proposition by considering the following two cases (i)  $y_{ik} = y_{jk} = 1$  and (ii)  $y_{ik} = 0$  or  $y_{jk} = 0$ . For case (i), observe that  $z_{ijk}$  will take a value of 1 due to the constraints (11)–(13). Similarly for case (ii), suppose  $y_{ik} = 1$  and  $y_{jk} = 0$ , then  $z_{ijk} = 0$ . A similar argument holds when  $y_{jk} = 1$  and  $y_{ik} = 0$ .  $\square$

Now, the mixed integer linear programming formulation for the CAGVRP is given by

$$(\mathcal{F}_2) \quad \text{Minimize} \quad \sum_{e \in E} c_e x_e + \sum_{(i,j) \in A} d_{ij} w_{ij} + \sum_{(i,j) \in A} f_{ij} y_{ij}$$

Subject to (1)–(3), (10)–(13), and (5)–(9).

If the integrality restrictions in the constraints (8) and (9) are relaxed, then we call that model a linear programming relaxation of the original MILP. In the following subsection, we shall strengthen the linear programming relaxation of the formulation  $\mathcal{F}_2$  by introducing additional valid inequalities

(a constraint is called a valid inequality if it does not remove any feasible solution).

### A. 2-matching inequalities

This section introduces a class of valid inequalities for the CAGVRP. These inequalities are derived from the 2-matching inequalities for the traveling salesman problem [28]–[30] and is also valid for the ring-star problem and its variants [16], [17], [31]. Specifically, we consider the following inequality:

$$\sum_{e \in \gamma(H)} x_e + \sum_{e \in I} x_e \leq \sum_{i \in H} y_{ii} + \frac{|I| - 1}{2}, \quad (14)$$

for all  $H \subseteq T$  and  $I \subseteq \delta(H)$ . Here  $H$  is called the handle and  $I$ , the teeth.  $H$  and  $I$  satisfy the following conditions: (i) no two edges in the teeth are incident on the same target and (ii) the number of edges in the teeth is odd and greater than equal to 3. The proof of validity of the above inequality is given by the following proposition. One can refer to [16], [17], [31] for the proof of this proposition.

**Proposition II.2.** *The 2-matching inequalities in Eq. (14) is valid of any feasible solution to the CAGVRP.*

The constraints in (14) are equivalent to the *blossom's inequality* [32] for the 2-matching problem and a special case of the comb inequalities for the symmetric traveling salesman problem.

## III. BRANCH-AND-CUT ALGORITHM

In this section, we outline the main components of the branch-and-cut algorithm. As mentioned previously, the branch-and-cut algorithm is an intelligent enumeration technique to find an optimal solution to an MILP. Let  $\bar{\tau}$  denote the optimal solution for a problem instance.

1. *Initialization:* Set the iteration count to  $k \leftarrow 1$  and the initial upper bound  $\bar{\alpha}$  on the optimal objective value as  $+\infty$ . The initial linear programming subproblem is then defined by the constraints (1)–(3), (10)–(13), and  $0 \leq x_e \leq 1$ ,  $0 \leq y_{ij}$ ,  $w_{ij} \leq 1$  for every  $e \in E$  and  $[i, j] \in A$  respectively. This initial linear programming subproblem is solved and inserted into a list  $\mathcal{L}$ .
2. *Termination check and subproblem selection:* If the list  $\mathcal{L}$  is empty, then stop. Otherwise, select a subproblem from the list with the lowest objective value; this choice of subproblem is called the best-first policy [33].
3. *Subproblem solution:* Set  $k \leftarrow k+1$ . Let  $\alpha$  be the solution objective value. If  $\alpha \geq \bar{\alpha}$ , then go to step 2. Otherwise, if the solution is feasible for the CAGVRP, set  $\bar{\alpha} \leftarrow \alpha$ , update  $\bar{\tau}$  and go to step 2; if the solution is infeasible, then go to step 4.
4. *Constraint separation and generation:* Introduce violated sub-tour elimination constraints (5), connectivity constraints (6) and (7) and 2-matching constraints (14). If no constraints can be generated using the current fractional solution, then go to 5, else go to step 3.

5. *Branching*: Create two subproblems by branching on a fractional variable and insert both the subproblems in the list  $\mathcal{L}$ .

At the end of the branch-and-cut algorithm  $\bar{\tau}$  will have the optimal solution. In the following sections, we detail the constraint separation procedure used in the step 4. This procedure finds violated sub-tour elimination constraints, connectivity constraints and 2-matching constraints from a fractional solution to the subproblems defined in the algorithm, if any.

#### A. Separation of connectivity constraints in (5)–(7)

In this section, we discuss exact separation procedures for separating out the constraints in (5)–(7) given a fractional solution. To this end, let  $(x^*, y^*, w^*)$  denote a fractional solution. We first construct a support graph  $(G^*)$  based on the fractional solution;  $G^* = (V^*, E^*)$ ,  $V^* = \{i \in T : 0 < y_{ii}^* < 1\}$  and  $E^* = \{e \in E : 0 < x_e^* < 1\}$ . Then, we check if the graph  $G^*$  is connected; if it is not connected, each vertex set  $S$  corresponding to an individual connected component, such that  $t_0 \notin S$ , generates a violated constraint (5) for each  $i \in S$ . If  $G^*$  is connected, we find the subset of nodes  $S$  with minimum value of  $\sum_{e \in \delta(S)} x_e^*$ . This is done by solving a problem of computing the max-flow on  $G^*$ , where the capacity of each edge  $e$  is set to  $x_e^*$ . We compute the minimum cut of all pairs of nodes on  $G^*$  and consider the node partition  $S$  that does not contain  $t_0$ . We check if this set violates constraint (5) for each  $i \in S$ . If the constraint is violated for any  $i$ , we add the corresponding inequality to the existing pool of inequalities.

We use a similar procedure to find violated connectivity constraints (6)–(7) for the UAV sub-tours. Based on the fractional solution  $w^*$ , we construct graph  $G_u^* = (V^*, A^*)$ ,  $A = \{[i, j] : 0 < w_{ij} < 1\}$ . Similar to the previous procedure we find the violated constraints (6)–(7) defined by  $S$ , such that the subset  $S$  does not contain any targets that are visited by the ground vehicle.

#### B. 2-matching Constraints

To find the violated 2-matching constraints, we follow the procedure described in [16], [30]. We consider each connected component  $H$  of  $G^*$  as a handle of a possible violated 2-matching inequality, whose two-target teeth correspond to edges in  $e \in \delta(H)$  with  $x_e^* = 1$ . We reject the inequality if the number of teeth is even. If the inequality is violated, then we add this inequality to the pool of violated inequalities.

### IV. COMPUTATIONAL RESULTS

In this section, we discuss the computational results of the branch-and-cut algorithm. The algorithm was implemented in Julia [34] using JuMP [35] (a mathematical modelling framework for the Julia programming language) and CPLEX 12.6. The internal CPLEX cut-generation routines were disabled and CPLEX was used only to manage the enumeration (branch-and-bound) tree. All the simulations were performed on a Macbook Pro with an Intel Core i5, 2.7 GHz processor.

The performance of the algorithm was tested on several randomly generated instances.

*Instance generation*: We generated twenty instances, five for each value of  $|T| \in \{10, 20, 30, 40\}$ . The coordinates of the targets are generated randomly from a uniform distribution in a  $100 \times 100$  grid. For every pair of targets  $i, j$ , the travel cost for the ground vehicle to traverse the edge  $(i, j)$  is chosen to be equal to the Euclidean distance between them  $(l_{ij})$ , and the cost of travel by the UAV is chosen to be  $\alpha \cdot l_{ij}$ , where  $\alpha$  is a scaling factor and  $\alpha \in \{0.1, 0.2, 0.3\}$ . We tested the algorithm for a total of 60 instances. Tables I and II summarize the computational results of the algorithm. In Table I, the first column refers to the problem size, second column refers to the scaling factor  $\alpha$ , the third, fourth and the fifth columns refers to the cost of the optimal tour, cost of the ground vehicle tour and cost of the UAV tour averaged over five instances. In Table II, the average number of SEC cuts generated and the number of nodes explored in the branch-and-cut tree are listed in third and fourth columns. The last column indicates the number of instances out of five that were solved to optimality with in the allowed computation time of 9000 seconds.

TABLE I  
AVERAGE OF OPTIMAL TRAVEL COSTS

$ T $	$\alpha$	Avg. opt. cost	Avg. GV cost	Avg. UAV cost
10	0.1	255.90	243.27	12.63
	0.2	266.74	246.89	19.85
	0.3	275.50	256.03	19.48
20	0.1	303.24	276.79	26.45
	0.2	328.68	280.08	48.60
	0.3	351.19	290.80	60.40
30	0.1	283.53	246.85	36.68
	0.2	318.54	254.30	64.24
	0.3	349.39	259.52	89.86
40	0.1	310.44	264.19	46.24
	0.2	357.79	268.86	88.94
	0.3	397.91	281.71	116.19

TABLE II  
BRANCH-AND-CUT ALGORITHM STATISTICS

$ T $	$\alpha$	SEC cuts	B&C nodes	I
10	0.1	167.00	5.80	5
	0.2	138.80	5.00	5
	0.3	170.80	6.20	5
20	0.1	2206.60	46.00	5
	0.2	3213.20	54.20	5
	0.3	6263.40	113.20	5
30	0.1	12764.80	247.60	5
	0.2	29796.40	576.80	5
	0.3	39399.40	761.40	5
40	0.1	66680.60	1017.20	5
	0.2	102671.00	1372.80	2
	0.3	126649.80	1517.80	1

The results tabulated in Tables I and II indicate that the proposed branch-and-cut algorithm can solve instances involving up to 40 targets with modest computation times. In summary we were able to solve 53/60 instances to optimality. For the remaining 7 instances, the algorithm produced

feasible solutions that were within 3% of the upper bound to the optimal (computed by CPLEX by solving corresponding dual problem). The plot in Table III shows the average computation times for all the instances.

TABLE III

AVERAGE COMPUTATION TIME NEEDED TO FIND OPTIMAL SOLUTION

$ T $	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$
10	0.98	0.89	0.89
20	10.53	15.12	29.27
30	220	665	853
40	4588	7643	8156

With  $\alpha$  equal to 0.1, the maximum computation time needed to solve instances with 30 targets is 420 seconds, which is quite reasonable. In general, we observe that the instances with a scale factor of  $\alpha = 0.3$  are more difficult to solve, and need more computation time. This is expected for the following reason: suppose the cost of travel by the UAV is zero, then the algorithm would try to assign as many targets as possible to the UAV. As the cost of travel by the UAV is increased, it has to find the right partitioning of the targets to be assigned to the ground vehicle and the UAV, that minimizes the total cost. This makes the problem combinatorially more difficult.

## V. CONCLUSION

We presented a path planning problem (CAGVRP) that arises in cooperative routing of a ground vehicle and an UAV with communication constraints. A MILP formulation is presented along with separation algorithms to find violating inequalities. The formulation and separation algorithms are integrated into a branch-and-cut framework, and implemented using CPLEX callable libraries. The algorithm was tested on several randomly generated instances with 10, 20, 30 and 40 targets. The algorithm could find optimal solutions for instances up to 40 targets. The limitation of this algorithm is that it may not be able to solve larger instances and one might have to rely on heuristics. Future directions for this work includes finding better separation algorithms, addressing the nonlinear constraints in the branch-and-cut sub-routines instead of linearizing, and developing efficient heuristics. Also the problem can be generalized for multiple air vehicles and/or multiple ground vehicles.

## REFERENCES

- [1] E. W. Frew and T. X. Brown, "Networking issues for small unmanned aircraft systems," *Journal of Intelligent and Robotic Systems*, vol. 54, pp. 21–37, March 2009.
- [2] J. A. Curry, J. Maslanik, G. Holland, and J. Pinto, "Applications of aerosondes in the arctic." *Bulletin of the American Meteorological Society*, vol. 85, no. 12, pp. 1855 – 1861, 2004.
- [3] E. J. Zajkowski T, Dunagan S, "Small UAS communications mission," in *Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference, Salt Lake City, UT*, 2006.
- [4] T. Shima and S. J. Rasmussen, *UAV cooperative decision and control: challenges and practical approaches*. SIAM, 2009, vol. 18.
- [5] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 3, pp. 16–25, 2006.

- [6] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative path planning of unmanned aerial vehicles*. John Wiley & Sons, 2010, vol. 32.
- [7] A. E. Gil, K. M. Passino, S. Ganapathy, and A. Sparks, "Cooperative scheduling of tasks for networked uninhabited autonomous vehicles," in *IEEE Conference on Decision and Control*, 2003., vol. 1, 2003, pp. 522–527.
- [8] J. Las Fargeas, P. Kabamba, and A. Girard, "Cooperative surveillance and pursuit using unmanned aerial vehicles and unattended ground sensors," *Sensors*, vol. 15, no. 1, pp. 1365–1388, 2015.
- [9] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *Trans. on Automation Science and Engineering*, no. 99, pp. 1–11, 2015.
- [10] P. Maini and P. Sujit, "On cooperation between a fuel constrained uav and a refueling ugv for large scale mapping applications," in *International Conference on Unmanned Aircraft Systems*. IEEE, 2015, pp. 1370–1377.
- [11] A. Ghaffarkhah and Y. Mostofi, "Dynamic networked coverage of time-varying environments in the presence of fading communication channels," *ACM Trans. on Sensor Networks*, vol. 10, p. 45, 2014.
- [12] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.
- [13] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of uavs for improved connectivity of mobile ground nodes," in *IEEE Military Communications Conference (MILCOM)*, vol. 3, 2004, pp. 1628–1634.
- [14] C. Sabo, D. Kingston, and K. Cohen, "A formulation and heuristic approach to task allocation and routing of uavs under limited communication," *Unmanned Systems*, vol. 2, no. 1, pp. 1–17, Jan 2014.
- [15] G. Perboli, R. Tadei, and D. Vigo, "The two-echelon capacitated vehicle routing problem: Models and math-based heuristics," *Transportation Science*, vol. 45, no. 3, pp. 364–380, 2011.
- [16] M. Labbe, G. Laporte, I. R. Martin, and J. J. S. Gonzalez, "The ring star problem: Polyhedral analysis and exact algorithm," *Networks*, vol. 43, no. 3, pp. 177 – 189, May 2004.
- [17] K. Sundar and S. Rathinam, "Multiple depot ring star problem: A polyhedral study and exact algorithm," *preprint arXiv:1407.5080*, 2014.
- [18] R. Baldacci and M. Dell'Amico, "Heuristic algorithms for the multi-depot ring-star problem," *European Journal of Operational Research*, vol. 203, no. 1, pp. 270–281, 2010.
- [19] R. Baldacci, M. Dell'Amico, and J. S. Gonzalez, "The capacitated m-ring-star problem," *Operations Research*, vol. 55, no. 6, pp. 1147–1162, 2007.
- [20] O. Goldschmidt, A. Laugier, and E. V. Olinick, "Sonet/sdh ring assignment with capacity constraints," *Discrete Applied Mathematics*, vol. 129, no. 1, pp. 99–128, 2003.
- [21] K. Altinkemer, "Topological design of ring networks," *Computers & operations research*, vol. 21, no. 4, pp. 421–431, 1994.
- [22] P. Carroll and S. McGarraghy, "Investigation of the ring spur assignment problem," in *International Network Optimization Conference (INOC)*, 2009, pp. 26–29.
- [23] J. G. Klincewicz, "Hub location in backbone/tributary network design: a review," *Location Science*, vol. 6, no. 1, pp. 307–335, 1998.
- [24] J. Shi and J. P. Fonseka, "Hierarchical self-healing rings," *IEEE/ACM Trans. on Networks*, vol. 3, no. 6, pp. 690–697, 1995.
- [25] T. Thomadsen and T. Stidsen, "Hierarchical ring network design using branch-and-price," *Telecommunication systems*, vol. 29, no. 1, pp. 61–76, 2005.
- [26] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton university press, Princeton, NJ, 2011.
- [27] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications*. SIAM, Philadelphia, 2014, vol. 18.
- [28] M. Grotschel and M. Padberg, "Polyhedra theory," in *The traveling salesman problem*, E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, Eds. John Wiley & Sons, New York, 1985, pp. 251–305.
- [29] M. Grotschel and O. Holland, "Solution of large-scale symmetric travelling salesman problems," *Mathematical Programming*, vol. 51, no. 1-3, pp. 141–202, 1991.
- [30] M. Fischetti, J. J. Salazar Gonzalez, and P. Toth, "A branch-and-cut algorithm for the symmetric generalized traveling salesman problem," *Operations Research*, vol. 45, no. 3, pp. 378–394, 1997.
- [31] K. Sundar and S. Rathinam, "An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem," in *International*

- Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 366–371.
- [32] J. Edmonds, “Paths, trees, and flowers,” *Canadian Journal of mathematics*, vol. 17, no. 3, pp. 449–467, 1965.
  - [33] L. A. Wolsey, *Integer Programming*. Wiley, New York, 1998, vol. 42.
  - [34] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *preprint arXiv:1411.1607*, 2014.
  - [35] M. Lubin and I. Dunning, “Computing in operations research using julia,” *INFORMS Journal on Computing*, vol. 27, no. 2, pp. 238–248, 2015.