2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Routing of Two Unmanned Aerial Vehicles with Communication Constraints

Satyanarayana G. Manyam[1], Sivakumar Rathinam[1], Swaroop Darbha[1],
David Casbeer[2] and Phil Chandler[2]

***Abstract*— A novel GPS denied routing problem for UAVs is described, where two UAVs cooperatively navigate through an array of non-communicating Unattended Ground Sensors (UGS). Contact with UGS is strictly maintained, which allows the UGS act as beacons for relative navigation eliminating the need for dead reckoning. This problem is referred to as the Communication Constrained UAV Routing Problem (CCURP). To solve the CCURP, shortest paths between targets are computed by means of a graph transformation. Given the shortest paths between targets, two solution methods are presented. The first is a $\frac{15}{2}$-approximation algorithm. The second method poses the CCURP as an one-in-a-set Traveling Salesman Problem (TSP), which can then be solved using known methods by transforming the problem into a regular asymmetric TSP. Computational results corroborating the performance bounds in this article are also presented.**

I. Introduction

This article considers a routing problem involving a team of two Unmanned Aerial Vehicles (UAVs) and Unattended Ground Sensors (UGS) in GPS denied environments. This problem arises in an operational scenario considered at the Air Force Research Laboratory where a team of two UAVs and UGS are used to restrict any intruder's access to a GPS-denied restricted zone. The UAVs do not have access to GPS but have range sensors that can estimate the distance between the UAVs based on the strength of their wireless communication link. The UAVs also have an onboard control and navigation software that can maintain the distance between the UAVs at a constant value. The UGS on the other hand have limited power, are not networked and are placed uniformly in the restricted zone to aid the UAVs traverse through the zone. Fig.1 shows an illustration of this scenario where the restricted zone is divided into squares and an UGS (represented by the black dot) is located at each corner of the zone. Each square in the zone has a side equal to the communication range of the UAVs. At any time instant, one of the UAVs (referred to as the first UAV) uses a UGS to localize its position while the second UAV pivots (orbits) about the first UAV from one UGS to another as shown in Fig. 2. The UAVs navigate through the UGS network by leap-frogging from UGS to UGS. A subset of UGS also referred to as targets are located at critical locations of the restricted zone and must be visited by the UAVs to collect any intruder information. A target is considered to be visited if a UAV is located vertically above the target. The two UAVs are collectively assigned the task of visiting a set of n targets for monitoring [1], [2].

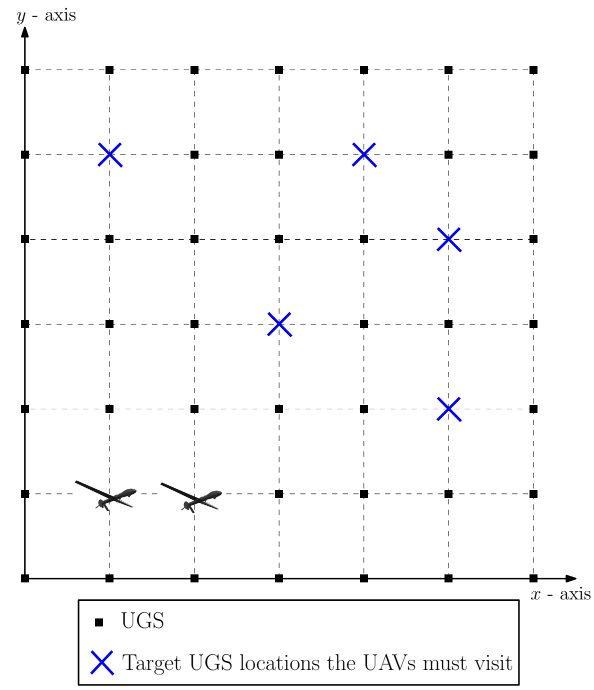


Fig. 1. Field with UGS deployed

Given this operational scenario, the objective of the routing problem is to find an optimal *cyclical* trajectory for each UAV so that

(i) each target is visited by one of the UAVs,
(ii) the UAVs always maintain a fixed distance (equal to their communication range) throughout their motion,
(iii) at least one of the UAVs is located vertically above a UGS at every instant of time, and
(iv) the sum of the distances traveled by the two UAVs is minimized.

Requirements (ii) and (iii) ensure that one of the UAVs localizes while the other navigates relative to the localized UAV. Requirement (i) is necessary for accomplishing the mission while requirement (iv) ensures that the distance or the time spent in visiting all the targets is the smallest possible. We refer to this problem as the Communication Constrained UAV Routing Problem (CCURP). A feasible solution to a problem instance is shown in Fig. 3.

If the UAVs have access to the GPS data, the routing problem for the UAVs is a generalization of the single

[1]Mechanical Engineering, Texas A&M University, College Station, Texas.
[2]Control Science Center of Excellence, Air Force Research Laboratory, WPAFB.

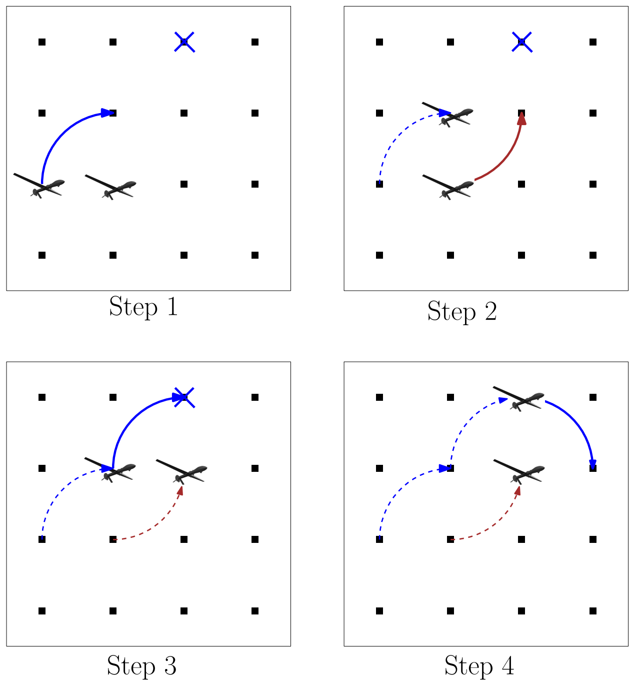


Fig. 2. Navigation of the UAVs

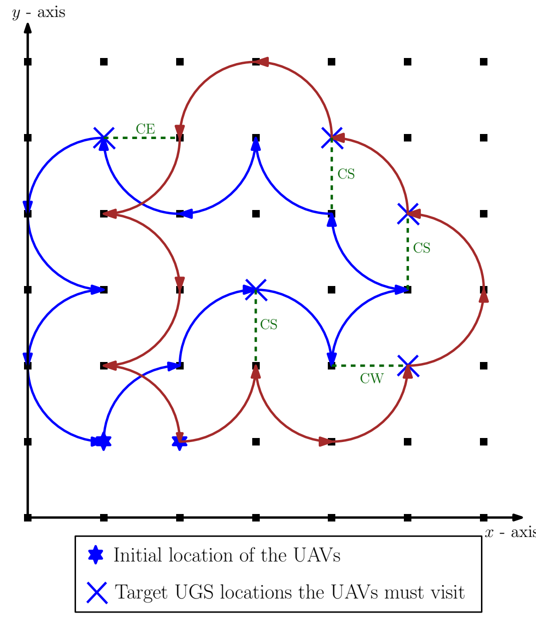


Fig. 3. A tour of the UAVs

Traveling Salesman Problem (TSP)[3] for which several algorithms are available[4], [5], [6]. In this article, we consider a *novel* operational scenario where the GPS information is not available and the UAVs are required to localize their current positions based on the onboard range sensors and UGS.

We are not aware of any algorithms in the literature that addresses the CCURP. The contributions of this article are as follows:

- We develop a $\frac{15}{2}$-approximation algorithm for the CCURP. A $\alpha-$approximation algorithm is an algorithm that runs in polynomial time and produces a solution whose cost is at most α times the optimal cost for every instance of the problem.
- Computational results show that the proposed algorithms can find high quality solutions to the CCURP in the order of a second for instances with 40 targets.

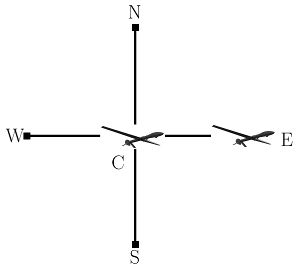
II. Problem Statement

Let $G = (V, E)$ represent a graph where V denotes the set of all UGS (which are the vertices of the restricted zone) and E represents the set of all the edges joining any two vertices in the graph that lie within the communication range. As shown in Fig. 1, four edges are incident on every vertex. The length of each edge or distance between any two adjacent vertices is constant and we assume that to be R. Let $N := \{1, 2, ...n\} \subseteq V$ be the set of targets to be visited, shown as blue cross ('X') in Fig. 1. The initial positions of the two UAVs are given. We assume the initial positions are at two adjacent vertices of the graph G.

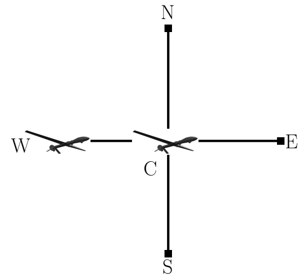
An admissible configuration, or simply the configuration, of UAVs is defined to be the adjacent pair of vertices occupied by the UAVs. Since the two UAVs are assumed to be identical, the definition of configuration of UAVs does not make a distinction as to which of the UAVs occupies which vertex as long as the UAVs occupy the pair of vertices specified by the configuration. If the location of one of the UAVs is fixed at a vertex, there are four possible configurations for the UAVs as shown in Fig. 4. A target is said to be visited if the UAVs reach the target at any one of these four configurations. The UAVs can move between any two adjacent configurations using a flip, *i.e.*, one of the UAVs pivots and rotates around the other fixed UAV by 90 degrees as shown in Fig. 5. The UAVs travel $\frac{\pi R}{2}$ units during each flip. The UAVs can travel from an initial configuration to any final configuration by executing a sequence of flips.

Let the (x, y) coordinates of vertex $u \in V$ be denoted by (ζ_i, η_i). Target u can be visited by the UAVs using any of the configurations present in the set $\{CE, CW, CN, CS\}$. In all these configurations, one of the UAV positions is fixed at (ζ_i, η_i) and the other UAV occupies one of the following set of coordinates depending on the configurations CE, CN, CW, CS respectively: $(\zeta_i + R, \eta_i), (\zeta_i, \eta_i + R), (\zeta_i - R, \eta_i), (\zeta_i, \eta_i - R)$.

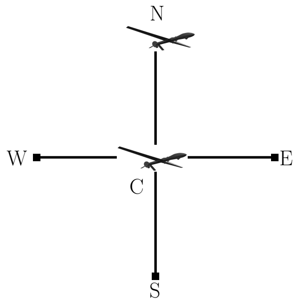
Let Φ be the set of four configurations at which the UAVs can visit a target, $\Phi = \{CE, CW, CN, CS\}$. Let θ_i be the configuration at which target i is visited by the two UAVs. For $i = 1, ..n$, θ_i can be one of the values from the set Φ. Let $d(i, \theta_i, j, \theta_j)$ be the length of the shortest path of the UAVs to travel from target i to target j, where θ_i and θ_j are the configurations of the UAVs at the targets i and j respectively. In a tour, let $\{s_1, ...s_n\}$ be the sequence of targets visited and $\{\theta_{s_1}, ...\theta_{s_n}\}$ be the configurations at the corresponding targets. The length of the tour is $D(tour) = \sum_{i=1}^{n-1} d(s_i, \theta_{s_i}, s_{i+1}, \theta_{s_{i+1}}) + d(s_n, \theta_{s_n}, s_1, \theta_{s_1})$. Without loss of generality, we assume that one of the UAVs



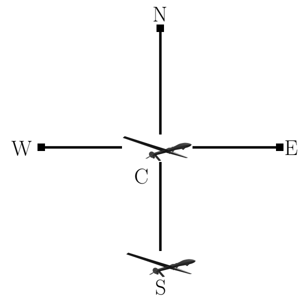
(a) CE Configuration



(b) CW Configuration

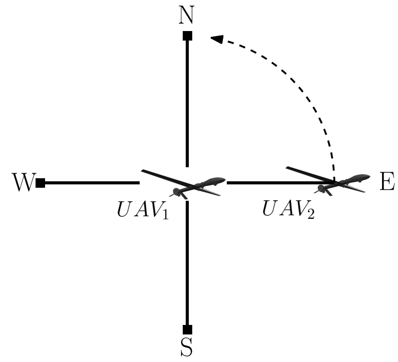


(c) CN Configuration

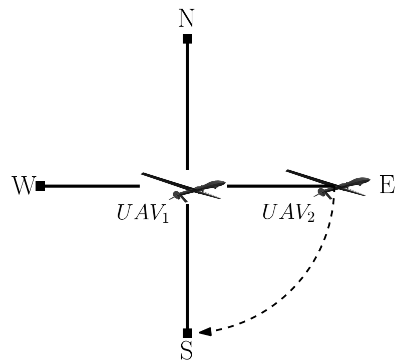


(d) CS Configuration

Fig. 4. Four different configurations at target C



(a) UAV_2 rotating in the counter-clockwise direction



(b) UAV_2 rotating in the clockwise direction

Fig. 5. One flip of the UAVs

is located at target 1 and the initial configuration of the UAVs (θ_1) is given. As we can compute the shortest path between any two given configurations, the CCURP reduces to find the sequence of targets to be visited and the configuration at each target such that the length of the tour, $D(tour)$ is minimum.

One can pose the CCURP as a one-in-a-set traveling salesman problem, which is defined as follows: Given p sets, each set contains q number of targets, the salesman has to visit one target from each set and return to the initial location such that the total distance traveled is a minimum. In CCURP, if we consider each target as a set and the four configurations at each target as its elements, the CCURP can be posed as a one-in-a-set TSP. The two UAVs has to start from their initial locations (we call them depots), visit at least one of the four configurations at each target and return to their depots such that the total distance traveled is minimized. A feasible solution for the CCURP posed as a one-in-a-set TSP is shown in Fig. 6.

III. Problem Formulation

We construct a graph $G_o(V_o, E_o)$ where V_o denotes the set of vertices representing all the configurations corresponding to the targets and E_o is the set of all edges connecting every pair of vertices in V_o. To simplify the formulation, we let target 1 to denote the initial location (or depot) of one of the UAVs. We also let V_o to include a replica (denoted by n+1) of the first target. The vehicles are required to travel from target 1 at a given initial configuration, visit each of the targets exactly once at some configuration and reach the target (n+1) at the same initial configuration.

An optimal solution for the CCURP can be found by formulating the problem as a Mixed Integer Linear Program (MILP). The requirement that the UAVs must visit at least one configuration at each target can be formulated using the single commodity flow constraints. The single commodity flow model is well known for enforcing connectivity constraints in network synthesis problems [7]. In the context of the CCURP, the connectivity constraints are enforced in the following way: Each target is required to receive a unit of commodity from any one of the vehicles. The vehicles can collect the commodities only at their initial locations and are required to deliver exactly one unit of commodity along a configuration at each target. If $\bar{n}$ targets must be visited, the vehicles collect $\bar{n}$ commodities and deliver one unit of commodity at each of the $\bar{n}$ targets.

Binary variables x_{ipjq} and f_{ipjq} are used to formulate the CCURP. $x_{ipjq} = 1$ if and only if there is an edge from the p-th configuration of the target i to the q-th configuration of the target j. The variable f_{ipjq} denotes the amount of commodity flowing from the p-th configuration of the target i to the q-th configuration of the target j. p, q can be of any value in the set $\{1, 2, 3, 4\}$. The values $(1, 2, 3, 4)$ corresponds to the (CE, CW, CN, CS) configurations in the set Φ respectively. The initial configuration at the target 1 which is denoted by θ_1 is known a priori and is set to be equal to the final configuration of the vehicles at target $n + 1$.

Let c_{ipjq} be the cost of the edge connecting the nodes corresponding to the $p-$th configuration at target i and the $q-$th configuration at target j. The values of c_{ipjq} are defined

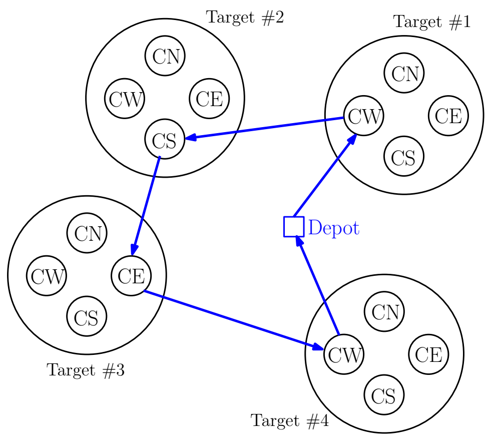


Fig. 6. One-in-a-set TSP: feasible solution

as follows:

1) For $i, j \in \{2, ...n\}$, c_{ipjq} is defined as the length of the shortest path between the nodes corresponding to the $p-$th configuration at target i and the $q-$th configuration at target j. An algorithm to compute this cost is explained in section IV.
2) For $i = n+1, j \in \{1, ...n\}$, $c_{(n+1)1jq} = c_{11jq}$.
3) For $i \in \{1, ...n\}, j = n+1$, $c_{ip(n+1)1} = c_{ip11}$.
4) For $i, j \in \{1, n+1\}$, $c_{i1j1} = M$, where M is a very large constant.

The MILP is as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n+1}\sum_{p=1}^{4}\sum_{j=1}^{n+1}\sum_{q=1}^{4} c_{ipjq}x_{ipjq} \tag{1}$$

Subject to:

$$\sum_{j=2}^{n+1}\sum_{q=1}^{4} f_{11jq} - \sum_{j=2}^{n+1}\sum_{q=1}^{4} f_{jq11} = n, \tag{2}$$

$$\sum_{q=1}^{4}\left[\sum_{i=1}^{n+1}\sum_{p=1}^{4} f_{ipjq} - \sum_{i=1}^{n+1}\sum_{p=1}^{4} f_{jqip}\right] = 1, \; j = 2, \cdots, n+1, \tag{3}$$

$$\sum_{i=1}^{n+1}\sum_{p=1}^{4}\sum_{j=1}^{n+1}\sum_{q=1}^{4} x_{ipjq} = n+1, \tag{4}$$

$$\sum_{i=1}^{n+1}\sum_{p=1}^{4} x_{ipjq} - \sum_{i=1}^{n+1}\sum_{p=1}^{4} x_{jqip} = 0, \quad j = 1, \cdots, n+1, \tag{5}$$

$$x_{ipiq} = 0, \; i = 2, \cdots, n, \; p = 1,2,3,4, \; q = 1,2,3,4, \tag{6}$$

$$f_{ipjq} \leq (n+1)x_{ipjq}, \; \text{for all } i, p, j, q, \tag{7}$$

$$x_{ipjq} \in \{0, 1\}, \; \text{for all } i, p, j, q. \tag{8}$$

Equation (2) states that the number of commodities leaving the initial target is n. At each set of four configurations representing a target, equation (3) states that the incoming flow minus the outgoing flow is 1. Equation (4) states that there should be exactly $n+1$ edges in the solution. Equation (5) is the Eulerian constraint which ensures the in-degree and the out-degree at each intermediate target is 1. Equation (6) states that there should be no edges connecting the configurations within a set. Equation (7) states that there can be flow between any two configurations only if there is an edge between them.

To solve the CCURP, one needs to first solve the following sub-problem: Given any two configurations, find the shortest path between the two given configurations.

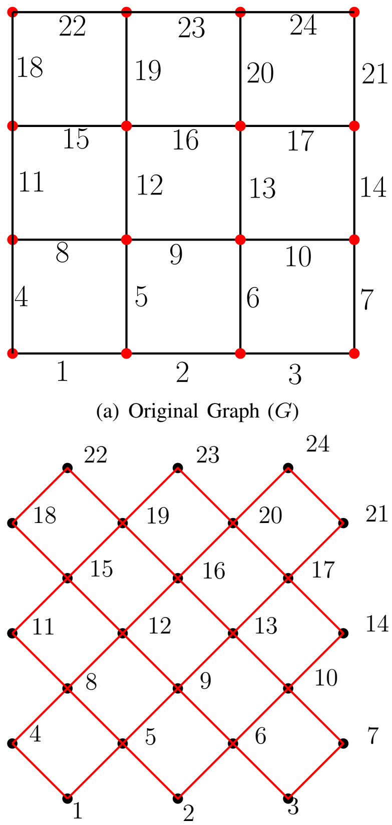
IV. Algorithm to find the shortest path between two configurations

The shortest path problem between two given configurations of UAVs can be solved by transforming the given graph G as explained here. We will explain this with reference to an example. Let the graph shown in Fig. 7(a) be the original graph given. Construct a new graph which has as many vertices as the number of edges in the graph G and let us call this G'. Each edge in G, numbered 1 to 24 in Fig. 7(a) corresponds to a vertex in G' as shown in Fig. 7(b). In the original graph G, each edge corresponds to a configuration of the UAVs. Let us consider any two configurations of the UAVs, which corresponds to two edges in G and which in-turn corresponds to two vertices in G'. If the UAVs could travel from one configuration to another in just one flip, then add an edge between those two vertices in G'. Let the length of the edge be $\frac{\pi R}{2}$ units. For example, the UAVs can travel from a configuration represented by edge #1 to edge #4. Therefore, there is an edge present in G' between vertices numbered 1 and 4. Add all possible edges between the vertices to complete the construction of G'.

For a given initial and final configurations, let a and b be the edges they represent in G. The two UAVs are initially located at the either ends of the edge a and the final locations of the UAVs should be at either ends of the edge b. In the transformed graph G', those two edges corresponds to two vertices a' and b' as shown in Fig. 8(b). One can compute a shortest path between the two vertices in graph G' using Dijkstra's or Bellman-Ford algorithm. In the solution of the shortest path between vertices a' and b', each intermediate vertex corresponds to an edge in G. In the solution, each step between consecutive vertices of G' corresponds to a flip in G. In Fig. 8(b), the shortest path from a' to b' is shown in blue color. The intermediate vertices are 12, 16, 20 and 17. The first step from vertex a' to 12 in G' corresponds to the flip a to 12 in G. The path from edge a to b in G is shown in Fig. 8(a), where the red and green dashed lines represent the paths of the two UAVs.

V. Algorithms for solving the CCURP

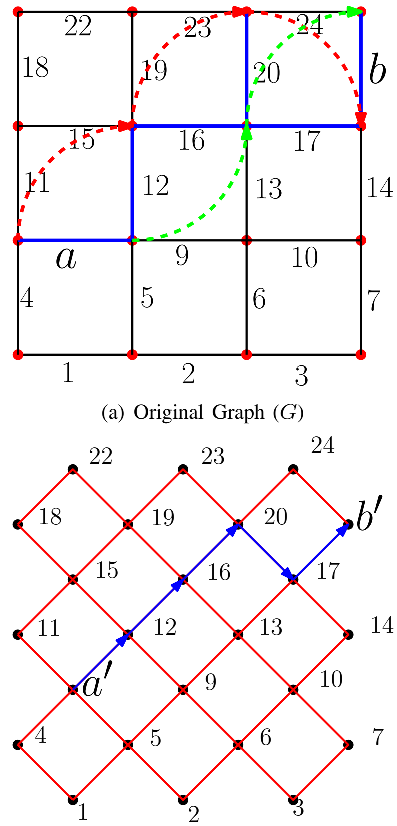
In this section, we present two algorithms to solve the CCURP. The first is an approximation algorithm and we prove the approximation ratio to be $\frac{15}{2}$. In the second algorithm, we present a transformation method to the CCURP. CCURP posed as an one-in-a-set TSP is transformed into a regular asymmetric traveling salesman problem (ATSP) using the result from [8].



(a) Original Graph (G)

(b) Transformed Graph (G')

Fig. 7. Graph Transformation



(a) Original Graph (G)

(b) Transformed Graph (G')

Fig. 8. Shortest path between two configurations

A. $\frac{15}{2}$-Approximation Algorithm

First we present an algorithm **A-1** to solve the CCURP with n targets. We assume that the initial locations of the UAVs are at one of the targets. For each target, there can be four different configurations of the UAVs such that one of the UAV is at the given target as explained in section III. The algorithm **A-1** is as follows:

Algorithm A-1:

1) For each target, assign a configuration from the set $\{CE, CW, CN, CS\}$ randomly.
2) Compute the shortest distance between each pair of targets with their assigned configuration as explained in section IV.
3) Solve regular TSP with distances computed in step 2 as costs, using Christofides algorithm [3]
4) Output the sequence of targets from the solution of the TSP in step 3 and the configurations assigned in step 1.

To solve the CCURP, one needs to find the sequence of targets to be visited and the configurations at each target such that the total distance traveled is minimum. In the optimal solution, let Π^* be the sequence of targets and let Θ^* be the vector of configurations at each target. Let Θ^0 be any vector of configurations at each target. With the configurations at each target given, the CCURP reduces to a regular TSP. In the optimal solution of this TSP, let Π^0_{opt} be the optimal sequence of tours and Π^0 be the sequence of tours given by the Christofides algorithm. For a given sequence Π and configurations Θ, let $C(\Pi, \Theta)$ be the cost of the corresponding tour.

Lemma 1: $C(\Pi^0, \Theta^0) \leq \frac{3}{2}C(\Pi^*, \Theta^0)$.

Proof: The Christofides algorithm for a TSP has an approximation ratio $3/2$ [3]. Since $C(\Pi^0_{opt}, \Theta^0)$ is the cost of the optimal solution of the TSP (defined above) and $C(\Pi^0, \Theta^0)$ is the cost of the solution from Christofides algorihtm, we have

$$C(\Pi^0, \Theta^0) \leq \frac{3}{2}C(\Pi^0_{opt}, \Theta^0). \tag{9}$$

With the configurations fixed at each target to Θ^0, consider the sequence of targets Π^*. Now (Π^*, Θ^0) is a feasible solution to the corresponding TSP and (Π^0_{opt}, Θ^0) is the optimal solution. From optimality, we have

$$C(\Pi^0_{opt}, \Theta^0) \leq C(\Pi^*, \Theta^0). \tag{10}$$

From inequalities (9) and (10):

$$C(\Pi^0, \Theta^0) \leq \frac{3}{2}C(\Pi^*, \Theta^0). \tag{11}$$

■

Let R be the distance between the two UAVs. At each target there can be four configurations as explained in section III and in the optimal solution, the UAVs can be at one of these four possible configurations. For example, at a given target, one of the configurations (say CE) is the optimal configuration. Among the other three configurations, two of them (CN and CS) are $\frac{\pi R}{2}$ units of distance away from the optimal configurations. One of them (CW) is πR units of distance away from the optimal. In the vector Θ^0, a configuration is prescribed for each target.

Lemma 2: Let the number of targets at which the configuration in Θ^0 is $\frac{\pi R}{2}$ units of distance away from the configuration in Θ^* be n_1. And at all the other targets, the configurations in Θ^0 and Θ^* are the same. Then, $C(\Pi^*, \Theta^0) \leq C(\Pi^*, \Theta^*) + n_1 \pi R$.

Proof: Let the sequence of targets $\Pi^* = \{s_1^*, s_2^*, ...s_n^*\}$. Let the vectors of configurations $\Theta^* = \{\theta_1^*, ...\theta_n^*\}$ and $\Theta^0 = \{\theta_1^0, ...\theta_n^0\}$

Consider a sub-sequence $\Pi_{ijk}^* = \{i, j, k\}$, where $\theta_i^* = \theta_i^0$ and $\theta_k^* = \theta_k^0$. θ_j^* and θ_j^0 are the configuration at target j and they are $\frac{\pi R}{2}$ units distance apart from each other. Let $d(s_i^*, \theta_i^0, s_j^*, \theta_j^0)$ be the distance from (s_i^*, θ_i^0) to (s_j^*, θ_j^0). From the triangular inequality, we have:

$$\begin{aligned} d(s_i^*, \theta_i^0, s_j^*, \theta_j^0) &\leq d(s_i^*, \theta_i^0, s_j^*, \theta_j^*) + d(s_j^*, \theta_j^*, s_j^*, \theta_j^0) \\ &\leq d(s_i^*, \theta_i^*, s_j^*, \theta_j^*) + \frac{\pi R}{2} \end{aligned} \quad (12)$$

Similarly we can write

$$d(s_j^*, \theta_j^0, s_k^*, \theta_k^0) \leq d(s_j^*, \theta_j^*, s_k^*, \theta_k^*) + \frac{\pi R}{2} \quad (13)$$

The cost of the sub-sequence (Π_{ijk}^*) is defined as $C(\Pi_{ijk}^*, \Theta^0) = d(s_i^*, \theta_i^0, s_j^*, \theta_j^0) + d(s_j^*, \theta_j^0, s_k^*, \theta_k^0)$. From inequalities (12) and (13), we can write:

$$C(\Pi_{ijk}^*, \Theta^0) \leq C(\Pi_{ijk}^*, \Theta^*) + \pi R \quad (14)$$

The cost of the tour is a summation of n terms, $C(\Pi^*, \Theta^0) = \sum_{i=1}^{n-1} d(s_i^*, \theta_i^0, s_{i+1}^*, \theta_{i+1}^0) + d(s_n^*, \theta_n^0, s_1^*, \theta_1^0)$. Since there are n_1 targets at which the configurations in Θ^0 and Θ^* are $\frac{\pi R}{2}$ units apart, it can be induced that $C(\Pi^*, \Theta^0) \leq C(\Pi^*, \Theta^*) + n_1 \pi R$. ■

Remark 1: If at n_2 targets, the configurations in Θ^0 are πR units of distance away from the configurations in Θ^*, then, $C(\Pi^*, \Theta^0) \leq C(\Pi^*, \Theta^*) + 2n_2 \pi R$

Lemma 3: For a given instance of CCURP with n targets, the cost (C) of any feasible tour is such that $C \geq \frac{n\pi R}{2}$, where R is the distance between the two UAVs.

Proof: Let $\Pi = \{s_1, s_2, ...s_n\}$ be the sequence of targets visited and $\Theta = \{\theta_1, ...\theta_n\}$ be the vector of configurations at each target in the tour. The cost of the tour is a summation of n terms:

$$C = \sum_{i=1}^{n-1} d(s_i, \theta_i, s_{i+1}, \theta_{i+1}) + d(s_n, \theta_n, s_1, \theta_1). \quad (15)$$

Here, each term in the summation is distance between two given configurations. From the shortest path computation between two configurations explained in section IV, it is evident that the shortest path possible between any two different configurations is $\frac{\pi R}{2}$. Therefore, each term in the equation (15) should be greater than or equal to $\frac{\pi R}{2}$,

$$d(s_i, \theta_i, s_{i+1}, \theta_{i+1}) \geq \frac{\pi R}{2}. \quad (16)$$

There are n such terms in the right hand side of the equation (15), and summing over all those terms gives:

$$C \geq \frac{n\pi R}{2}. \quad (17)$$

■

Theorem 1: The algorithm **A-1** solves the CCURP with an approximation factor $\frac{15}{2}$.

Proof: In the step 1 of algorithm **A-1**, let the number of targets at which the configuration assigned is $\frac{\pi R}{2}$ units distance away from optimal configuration be n_1. Let the number of targets at which the configuration assigned is πR units distance away from optimal configuration be n_2. Therefore, at the remaining $(n - n_1 - n_2)$ targets, the optimal configuration and the assigned configuration are the same. Using the result from Lemma 2, we can write:

$$C(\Pi^*, \Theta^0) \leq C(\Pi^*, \Theta^*) + n_1 \pi R + 2n_2 \pi R. \quad (18)$$

Since $n_1 + n_2 \leq n$, we have:

$$C(\Pi^*, \Theta^0) \leq C(\Pi^*, \Theta^*) + 2n\pi R. \quad (19)$$

Using Lemma 1 and inequality (19), we have the inequality

$$C(\Pi^0, \Theta^0) \leq \frac{3}{2} \left[C(\Pi^*, \Theta^*) + 2n\pi R \right]. \quad (20)$$

Here, we can replace $2n\pi R$ in the RHS with $4C(\Pi^*, \Theta^*)$ using inequality (17),

$$\begin{aligned} C(\Pi^0, \Theta^0) &\leq \frac{3}{2} \left[C(\Pi^*, \Theta^*) + 4C(\Pi^*, \Theta^*) \right] \\ &\leq \frac{15}{2} C(\Pi^*, \Theta^*). \end{aligned} \quad (21)$$

■

B. Transforming an one-in-a-set TSP to an ATSP

CCURP can be posed as an one-in-a-set TSP. In this section we present a method to transform the one-in-a-set TSP to a regular ATSP using the result from [8]. The UAVs needs to start from the depot, visit one of the configurations at each target and return to the depot such that the total distance traveled is the minimum. A target in CCURP corresponds to a set and a configuration corresponds to a vertex in one-in-a-set TSP. A feasible solution of the problem is shown in Fig. 6. Let P be the graph on which the one-in-a-set TSP is defined. The idea here is to modify the topology of P and transform it into a new graph (P') such that the optimal solution of a single vehicle ATSP on P' is same as the optimal solution of one-in-a-set TSP in P.

To do this, first we number the vertices in each set (s) as s_1, s_2, s_3 and s_4. For example, we name the vertices in set a as a_1, a_2, a_3 and a_4. Since only one of the vertex needs to be visited in each set, adding directed zero cost edges between the vertices of a set does not change the optimal solution. We add zero cost directed edges in each set (s) from s_i to s_{i+1} for $i = 1, 2, 3$ and from s_4 to s_1 as illustrated in Fig. 9. We want the vehicle to enter a set, visit every vertex in the set traveling through the zero cost edges and go to the next set. We would want the optimal cost of the one-in-a-set TSP as same as that of the ATSP and to ensure that, we make some changes to the cost of the edges connecting vertices belonging to different sets. For example if a vehicle enters the set $\#b$ and visits vertex b_1 first, then it would visit b_2, b_3 and b_4 and leaves the set from b_4.

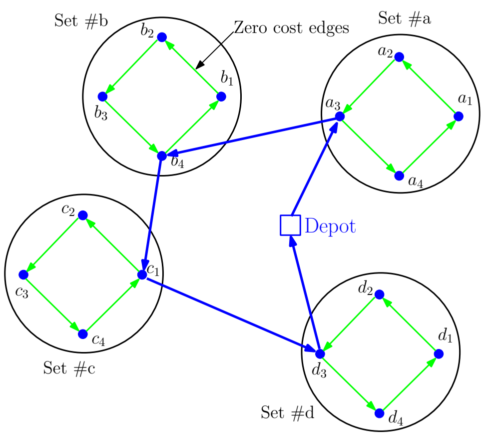


Fig. 9. Transformation of one-in-a-set TSP to ATSP: zero cost edges added.

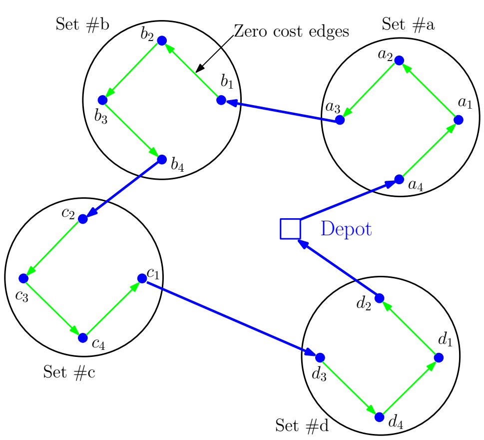
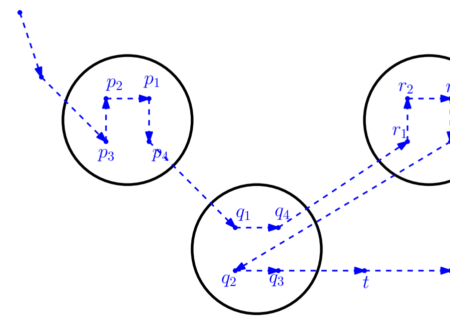


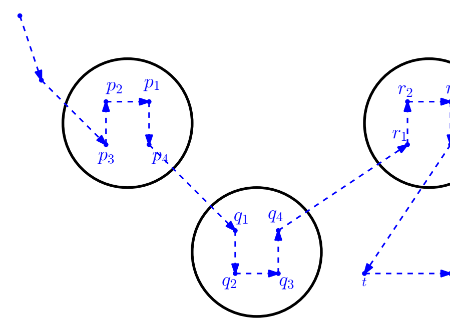
Fig. 10. Transformation of one-in-a-set TSP to ATSP: vehicle entering a set at one vertex exits from the successive vertex.

Lets say the vehicle next visits the vertex c_2 in set $\#c$. Now, the cost of edge going from b_4 to c_2 is not same as the cost of the edge going from b_1 to c_2. Therefore we would replace the cost of edges going out from b_4 with the cost of the edges going out from b_1. In every set s, for each vertex s_i, by following the cycle of zero cost edges, we can identify the successive vertex s_j. In the new graph, we set the cost of the edge going out from s_j to s'_k, where s'_k is in a different set, to the cost of edge going form s_i to s'_k in the original graph. We do this for all the edges connecting vertices of different sets.

Solving an ATSP on this new graph, there is a possibility that one may not be able to construct an optimal solution to the one-in-a-set TSP from the optimal solution of the ATSP. Sometimes, it may not be cheaper to visit all the vertices in a set at once. For example, the vehicle may visit two vertices in a set, go to another set and come back to the first set and visit the remaining vertices. If there are n sets, there can be more than $n+1$ edges connecting vertices from different sets. To overcome this problem, in the optimal solution of ATSP, we need to make sure there are only n edges connecting vertices from different sets. To ensure that, in P', we increase the cost of each edge connecting vertices from different sets by a constant value (M). If M is chosen such that it is greater than the sum of all the edges in P', then in the optimal solution of the ATSP, there will be only n edges connecting vertices from different sets.



(a) Sub-tour in the assumed optimal solution



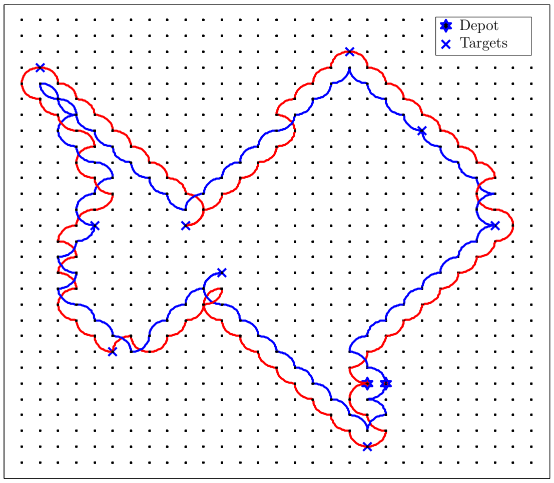
(b) Modified sub-tour

Fig. 11. Sub-tour in the optimal solution of the ATSP

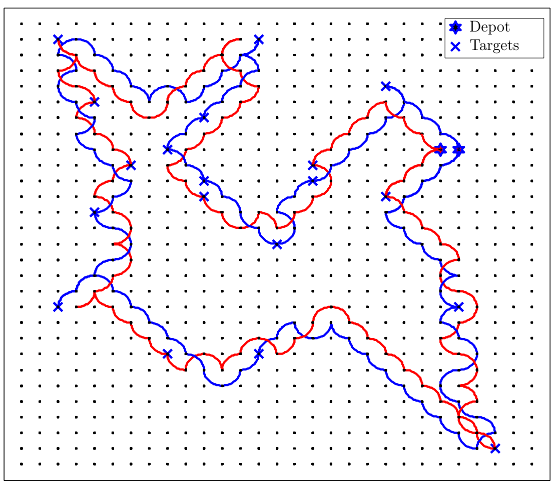
VI. Computational Results

To test the algorithms that we have discussed in this paper, we have generated 50 problem instances with 10, 20, 30 and 40 targets. Each instance is such that the targets are chosen randomly over an area of size 30×30 units. This area is divided as a square grid and an UGS is present at every node. The targets are located at randomly chosen nodes on this area. Tours of the UAVs obtained from the optimal solution of three instances with 10, 20 and 30 targets are shown in Figs. 12(a), 12(b) and 12(c).

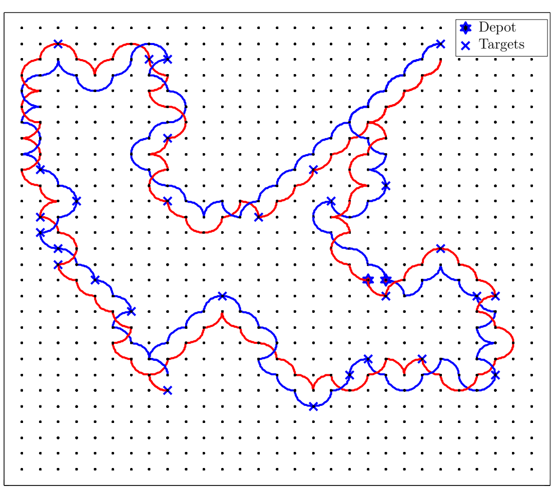
The instances with 10, 20 and 30 targets are solved for optimal solution with the formulation in Section III. This was done using the integer program solver CPLEX. This required a very high computing time to solve the instances with 40



(a) CCURP tour with 10 targets



(b) CCURP tour with 20 targets



(c) CCURP tour with 30 targets

Fig. 12. Optimal tours of the CCURP

targets and hence we did not solve for the optimal solution for these instances. All the instances are solved using the approximation algorithm presented in Section V-A and the transformation algorithm presented in Section V-B. All the simulations were run on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @ 2.53GHz, 12GB RAM). The average of the time required to run these algorithms for the different problem sizes is shown in Fig. 13. The solution quality of an instance is defined as $\frac{C_{algo}}{C_{opt}}$, where C_{algo} is the cost of the tour solved using the proposed algorithms and C_{opt} is the optimal cost of the instance. The average of the solution qualities for approximation algorithm and the transformation algorithm is show in Fig. 14. The transformation method gives the optimal solution for every instance and hence the solution quality is 1 for all the instances.

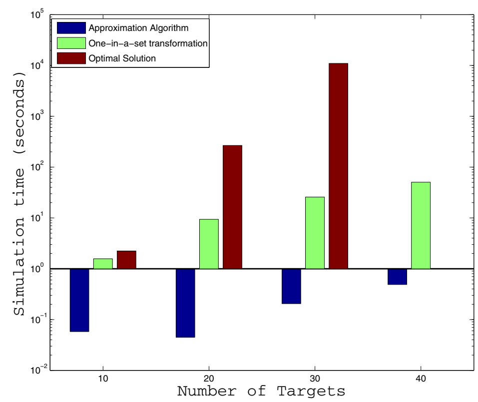


Fig. 13. Simulation time of the proposed algorithms for CCURP

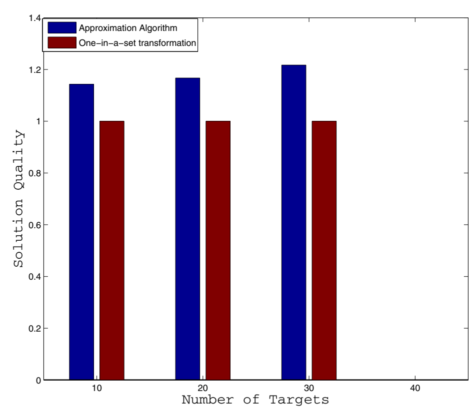


Fig. 14. Solution quality of the proposed algorithms for CCURP

Table I presents the computational results with 10, 20, 30 and 40 targets. The first column refers to the problem size (number of targets). The second, third and fourth columns refers to the tour length, time required to run the algorithm and the solution quality solved using the approximation algorithm. The fifth, sixth and seventh columns refers to the tour length, time required to run the algorithm and solution quality using the transformation algorithm. The eighth and

TABLE I

SIMULATION RESULTS OF THE CCURP USING THE THREE ALGORITHMS.

#Targets	Approximation Algorithm			One-in-a-set transformation			Optimal solution	
	Tour length	Time (secs)	Sol'n quality	Tour length	Time (secs)	Sol'n quality	Tour length	Time (secs)
10	263.893	0.642	1.069	226.195	1.787	1.000	226.195	1
10	267.035	0.123	1.118	238.761	1.245	1.000	238.761	2
10	194.779	0.077	1.051	185.354	1.977	1.000	185.354	4
10	229.336	0.020	1.123	204.204	1.447	1.000	204.203	1
10	273.319	0.032	1.160	235.619	1.306	1.000	235.619	1
20	314.159	0.039	1.099	285.885	8.936	1.000	285.885	82
20	345.575	0.030	1.158	298.451	11.466	1.000	298.452	444
20	289.027	0.031	1.057	273.319	9.495	1.000	273.319	29
20	326.726	0.051	1.195	273.319	11.089	1.000	273.319	7567
20	304.735	0.032	1.155	263.894	9.866	1.000	263.894	61
30	405.266	0.314	1.330	304.735	23.675	1.000	304.735	894
30	348.717	0.233	1.144	304.735	27.536	1.000	304.735	2419
30	383.274	0.078	1.089	351.858	30.708	1.000	351.858	45772
30	373.850	0.043	1.240	301.593	21.420	1.000	301.593	1323
30	402.124	0.114	1.164	345.575	23.851	1.000	345.576	54710
40	461.814	0.961	-	370.708	54.957	-	NC	NC
40	480.664	0.135	-	386.416	54.604	-	NC	NC
40	442.965	0.108	-	361.283	53.401	-	NC	NC
40	392.699	0.166	-	329.867	50.001	-	NC	NC
40	414.690	0.120	-	326.726	49.335	-	NC	NC

ninth columns refers to the tour length and time needed to solve for optimal solution using the solver CPLEX. We expect the transformation method would provide the optimal solution of the original one-in-a-set TSP. Hence one can see that, for the instances with 10, 20 and 30 targets, the transformation algorithm provided the optimal solution. The tour lengths obtained using the approximation algorithm are with in 1.25 times of the optimal solution for all the instances.

VII. CONCLUSION

We considered a problem of routing two UAVs with communication constraints in a GPS denied environment. We posed the problem as a one-in-a-set TSP and solved for an optimal solution using a single commodity flow formulation. We also developed a $\frac{15}{2}$-approximation algorithm and a transformation method to solve this problem. These algorithms were tested on 50 instances with 10, 20, 30 and 40 targets. The approximation algorithm ran in a fraction of a second and produced solutions within 1.25 times the optimal solution for all the instances. On the other hand, the transformation method was relatively time consuming but found optimal solutions for most of the instances.

REFERENCES

[1] D. W. Casbeer, K. Krishnamoorthy, A. Eggert, P. Chandler, and M. Pachter, "Optimal search for a random moving intruder," in *AIAA Infotech@Aerospace Conference*, 2012.

[2] K. Krishnamoorthy, D. W. Casbeer, P. Chandler, M. Pachter, and S. Darbha, "UAV search & capture of a moving ground target under delayed information," in *IEEE Conference on Decision and Control*, 2012.

[3] K. Steiglitz and C. H. Papadimitriou, *Combinatorial optimization: Algorithms and complexity*. Prentice Hall, New Jersey., 1982.

[4] S. Rathinam, R. Sengupta, and S. Darbha, "Resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Trans. Automation Science and Engineering*, vol. 4, no. 1, pp. 98 – 104, 2007.

[5] S. Yadlapalli, W. Malik, S. Darbha, and M. Pachter, "A lagrangian based algorithm for a multiple depot, multiple traveling salesman problem," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 1990 – 1999, August 2009.

[6] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *IEEE Robotics and Automation Magazine*, vol. 17, no. 4, pp. 70 – 77, December 2010.

[7] T. L. Magnanti and L. A. Wolsey, "Chapter 9 optimal trees," in *Network Models*, ser. Handbooks in Operations Research and Management Science, C. M. M.O. Ball, T.L. Magnanti and G. Nemhauser, Eds. Elsevier, 1995, vol. 7, pp. 503 – 615.

[8] C. E. Noon and J. C. Bean, "An efficient transformation of the generalized traveling salesman problem." *INFOR*, vol. 31, no. 1, pp. 39 – 44, 1993.