J Intell Robot Syst (2016) 84:691–703
DOI 10.1007/s10846-016-0343-2

# GPS Denied UAV Routing with Communication Constraints

**Satyanarayana G. Manyam · Sivakumar Rathinam · Swaroop Darbha · David Casbeer · Yongcan Cao · Phil Chandler**

Received: 11 February 2015 / Accepted: 18 January 2016 / Published online: 9 February 2016


**Abstract** A novel GPS denied routing problem for UAVs is described, where the UAVs cooperatively navigate through a restricted zone deployed with non-communicating Unattended Ground Sensors (UGS). The routing algorithm presenting in this paper ensures the UAVs maintain strict contact with at least one UGS, 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). Two architectures for cooperative navigation of two or three UAVs are considered. For the two UAV problem, a $\frac{9}{2}$-approximation algorithm is developed. The three UAV problem is transformed into a one-in-a-set Traveling Salesman Problem (TSP), which is solved as a regular asymmetric TSP using existing methods after applying a second transformation. Computational results corroborating the performance bounds are presented.

**Keywords** UAV route planning · Cooperative localization · GPS-denied environments · Approximation algorithms

S. Manyam · D. Casbeer · Y. Cao · P. Chandler
Control Science Center of Excellence, Air Force Research Laboratory, WPAFB, Dayton OH, USA

S. Rathinam (✉) · S. Darbha
Department of Mechanical Engineering, Texas A & M University, College Station TX, USA
e-mail: srathinam@gmail.com

## 1 Introduction

This article considers a routing problem involving a team of Unmanned Aerial Vehicles (UAVs) and Unattended Ground Sensors (UGS) in GPS denied environments. The usage of UAVs in civil and military applications has increased rapidly over the past decade. UAV routing is only possible with GPS, which makes UAVs vulnerable to GPS jamming and spoofing [10, 25]. We consider a scenario, where a team of UAVs needs to be routed for patrol in a GPS-denied zone by communicating with UGS deployed in the zone. The UAVs do not have access to GPS but have range sensors that can measure the distance between two UAVs and between an UAV and an UGS based on the strength of their wireless communication link. The UGS on the other hand have limited power, are not networked, but they can communicate with an UAV if it is located relatively close to an UGS, i.e., within communication range.

We consider two different architectures for the navigation of the UAVs aided by the UGS. In both architectures, the geometry and location of the UGS are known by the UAVs. In the first architecture, we assume the UGS are placed uniformly in the restricted zone to aid the UAVs navigate through the zone. Figure 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. We assume that the distance between two UGS along the edges of each square is less than the

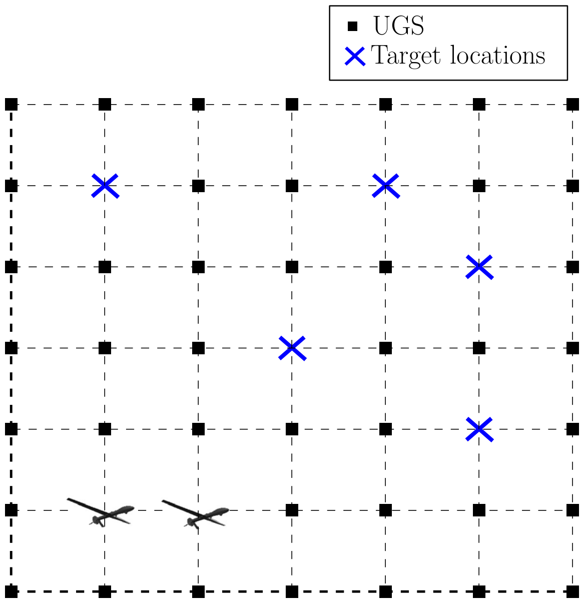



**Fig. 1** Field with UGS deployed

UAV-to-UAV communication range, $R$. At any instance, one of the UAVs (referred to as the first UAV) uses an 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 have controllers on board to orbit around the other UAV while estimating distance between them using range sensors. This type of controllers were developed for the UAVs to orbit around moving or stationary targets [2, 6, 20, 22]. The UAVs navigate through the UGS network by leap-frogging from UGS to UGS.

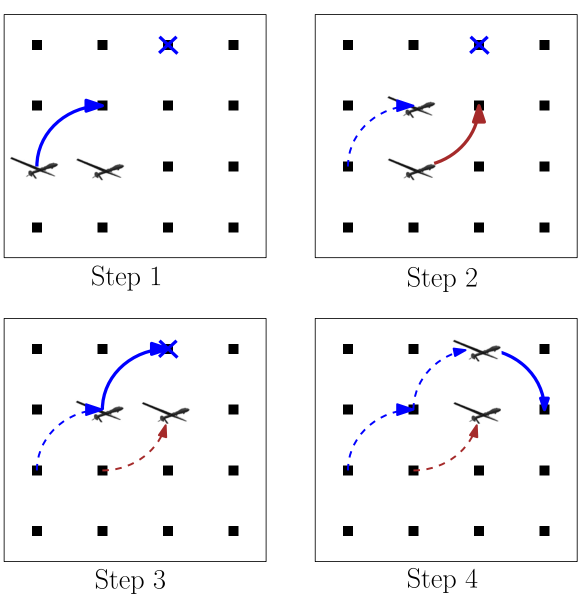



**Fig. 2** Two UAVs navigating an UGS array by leap-frogging/pivoting from UGS to UGS

A subset of the UGS are referred to as *target UGS* (or targets for short); the target UGS are located at critical locations and must be visited by the UAVs to collect information. A target is considered visited if a UAV is located above the target. In the first architecture, the two UAVs are collectively assigned the task of visiting a set of $n$ targets for monitoring [3, 11].

The second architecture differs from the first in that the UGS are not placed in a regular grid; they may be arbitrarily laid out. Cooperative navigation by 2 UAV might not be possible if the inter-UGS distances are greater than the UAV to UAV communication distance, and 3 UAVs might be required. When an UAV is above (orbiting) an UGS, the UAV can communicate with that UGS and localize itself, i.e. it can estimate the co-ordinates of its location in the $xy$-plane. Suppose two UAVs are located on top of two UGS, and a third UAV is located anywhere else within the communication range ($R$ units of distance) from the first two UAVs. The third UAV is able to triangulate its position using $(i)$ $xy$-coordinates of the first two UAVs that are located above two distinct UGS and $(ii)$ range measurements from the first two UAVs. Since the UAV knows its starting location and measures the range to the other UAVs at different points along the path, it can accurately determine its location while traveling along the path. Therefore, the third UAV is able to navigate from one point to another as long as it is within $R$ units of distance from the other two UAVs. Figure 3 illustrates this navigation scenario, where $\text{UAV}_1$, $\text{UAV}_2$ and $\text{UAV}_3$ are, respectively, located at UGSs $a$, $b$ and $c$. UGSs $c$ and $d$ are located within a distance of $R$ units from both the UGSs $a$ and $b$. Once

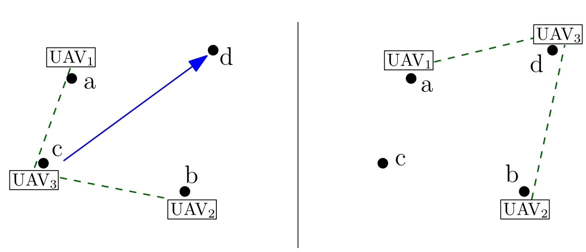



**Fig. 3** $\text{UAV}_3$ travel from UGS $c$ to $d$ while maintaining the communication links with $\text{UAV}_1$ and $\text{UAV}_2$

$\text{UAV}_3$ is off from UGS $c$, it utilizes range measurements with respect to the other two UAVs to localize itself and navigate to UGS $d$. A series of such maneuvers are performed to navigate from target to target to accomplish the mission

If the UAVs have access to GPS, the UAV routing problem is a generalization of the traveling salesman problem (TSP) [21]. A suite of algorithms to find optimal solutions, approximate solutions, lower bounds are available [8, 12, 13, 15, 18, 23, 24, 27] for several generalizations of TSP with multiple vehicles, vehicles with motion constraints, etc. Algorithms for motion planning and navigation using cell phone signal and monocular vision in a GPS denied environment are presented in [5, 17, 26]. Here, we present path planning algorithms in GPS denied environments using cooperative localization. In our previous work [14], we developed a $\frac{15}{2}$-approximation algorithm,[1] and a transformation method for the two UAV problem. The approximation algorithm picks a random configuration at each target and reduces the routing problem to a TSP which is solved using Christofides algorithm [4]. In this article, we present an algorithm with better approximation ratio for the two UAV problem, and generalize the routing framework to three UAV case that could navigate on an arbitrarily placed UGSs network. These contributions are summarized as follows:

- We develop a $\frac{9}{2}$-approximation algorithm for the two UAV CCURP.
- We present a graph transformation method to transform the three UAV problem into a regular asymmetric TSP.

This paper addresses the problem of cooperative UAV routing in an environment where the UAVs must maintain specified UAV-to-UAV communication channels, as well as UAV-to-UGS links. The paper is organized as follows: an approximation algorithm and a heuristic to solve the two UAV CCURP on uniformly placed UGS is presented in Section 2. The CCURP with three UAVs is presented and a graph transformation method is explained in Section 3. The computational results and the simulation of the two UAV CCURP on AMASE is presented in Section 4.

[1] An $\alpha-$approximation algorithm is an algorithm that runs in polynomial time and produces a solution whose cost is at most $\alpha$ times the optimal cost for every instance of the problem.

## 2 Two UAV Routing

In the two UAV architecture, the objective of the routing problem is to find an optimal *cyclical* trajectory for each UAV so that

(i) each target is visited by some UAV,
(ii) the UAVs always maintain a fixed distance (less than or equal to their communication range $R$) throughout their motion,
(iii) at least one of the UAVs is located vertically above an UGS at every instance, 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. 4.

Let $G = (V, E)$ represent a graph where $V$ denotes the set of all UGS and $E$ represents the set of all the

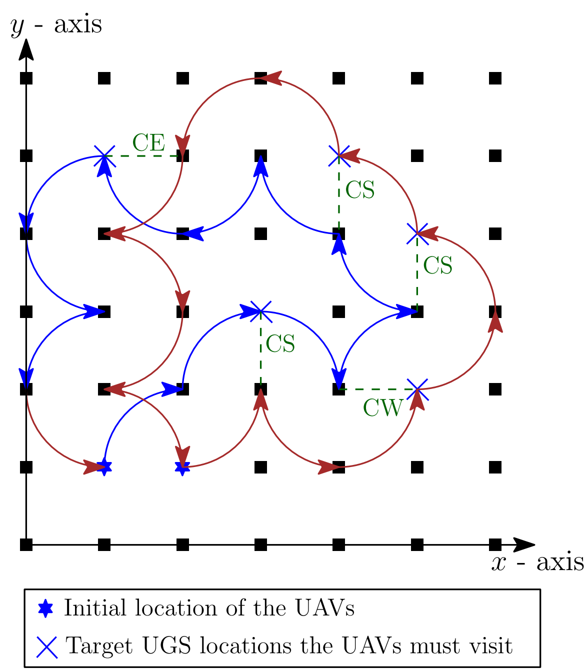
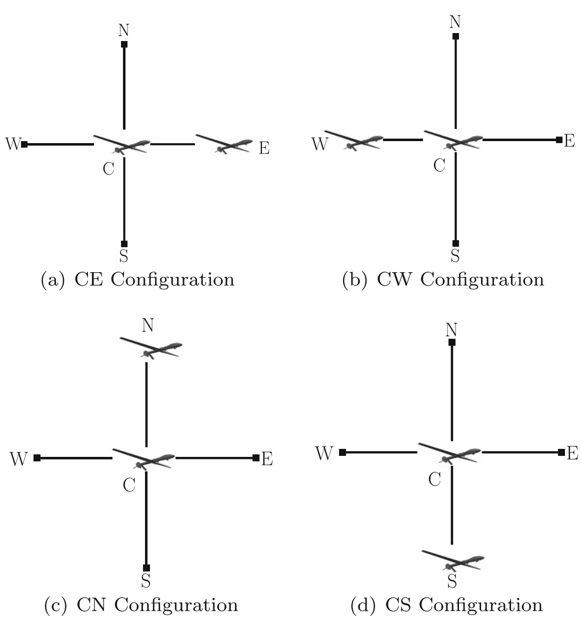



**Fig. 4** An example, feasible communication constrained tour for two UAVs

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 is constant and, without loss of generality, is assumed to be of length $R$. Let $T := \{1, 2, ...n\} \subseteq V$ denote the subset of UGS that must be visited by the UAVs. At the start of the monitoring mission, the UAVs are assumed to occupy a pair of adjacent vertices as shown in Fig. 1. Without loss of generality, we assume there are two adjacent targets in $T$ and we initialize the locations of the UAVs at these two targets.

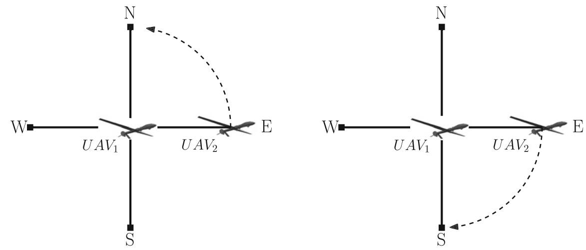
An admissible configuration, or simply the configuration, of UAVs is defined to be the adjacent pair of vertices occupied by the UAVs. Since the two UAVs are assumed to be identical, the definition of configuration of UAVs does not make a distinction as to which of the UAVs occupies which vertex as long 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. 5. 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° as shown in Fig. 6. The UAVs travel $\frac{\pi R}{2}$ units




(a) CE Configuration (b) CW Configuration

(c) CN Configuration (d) CS Configuration

**Fig. 5** Four different configurations available for visiting a target at C




(a) $UAV_2$ rotating in the counter-clockwise direction (b) $UAV_2$ rotating in the clockwise direction

**Fig. 6** One flip of the UAVs

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 $(\zeta_i, \eta_i)$. 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 $(\zeta_i, \eta_i)$ and the other UAV occupies one of the following set of coordinates depending on the configurations $CE, CN, CW, CS$ respectively: $(\zeta_i + R, \eta_i)$, $(\zeta_i, \eta_i + R)$, $(\zeta_i - R, \eta_i)$, $(\zeta_i, \eta_i - R)$.

Given any two targets $i$ and $j$, let $d_{min}(\theta_i, \theta_j)$ denote the minimum total distance required by the UAVs to travel from configuration $\theta_i$ to $\theta_j$. $d_{min}(\theta_i, \theta_j)$ can be computed by using a shortest path algorithm in the following way: Let $G_s = (V_s, E_s)$ be a new graph, lifted from $G$, where each vertex in $V_s$ corresponds to a configuration in $G$ and $E_s$ denotes all the edges that join any two adjacent configurations in $V_s$. Since each edge in $E_s$ joins two adjacent configurations, a travel cost of $\frac{\pi R}{2}$ (corresponding to one flip) units is assigned to the edge. The Dijkstra algorithm can then be applied on $G_s$ to find the length of the shortest path from $\theta_i$ to $\theta_j$. This shortest path specifies the sequence of flips that are necessary to move the UAVs from $\theta_i$ to $\theta_j$.

A tour for the UAVs is specified by a sequence $(s_1, s_2, \ldots, s_n)$ of targets visited by the UAVs and the corresponding configurations $\theta_{s_1}, \theta_{s_2}, \ldots, \theta_{s_n}$ at the targets. The length of the tour, $D(tour)$, is defined as:

$$D(tour) := \sum_{i=1}^{n-1} d_{min}(\theta_{s_i}, \theta_{s_{i+1}}) + d_{min}(\theta_{s_n}, \theta_{s_1}).$$

The objective of the CCURP is to determine the sequence $(s_1, s_2, \ldots, s_n)$ of targets to be visited and the associated configurations $\theta_{s_1}, \theta_{s_2}, \ldots, \theta_{s_n}$ at the targets such that $D(tour)$ is minimal.

### 2.1 Approximation Algorithm for Solving the 2-UAV Problem

In this section, we present an approximation algorithm for solving the CCURP and prove the approximation ratio of the proposed algorithm. In gist, this algorithm generates two feasible solutions for the CCURP and chooses the best of the two. The first solution is generated by arbitrarily choosing a configuration at each target and applying the Christofides algorithm [4] to generate a tour. The second solution is generated by rotating each of the configurations chosen in the first solution by two flips and applying the Christofides algorithm to generate another tour. The main steps in this algorithm *Approx* are as follows:

**Algorithm 1** Approx

1. For each target $u \in T$, choose any configuration $\theta_u^1$ from the set $\{CE, CW, CN, CS\}$.
2. For each target $u \in T$, assign a new configuration $\theta_u^2$ that is exactly two flips away from $\theta_u^1$. Specifically,

$$\theta_u^2 := \begin{cases} CE, & \text{if } \theta_u^1 = CW, \\ CN, & \text{if } \theta_u^1 = CS, \\ CW, & \text{if } \theta_u^1 = CE, \\ CS, & \text{if } \theta_u^1 = CN. \end{cases}$$

3. For $k = 1:2$, do the following:
   - Compute $d_{min}(\theta_u^k, \theta_v^k)$ for any two distinct targets $u, v \in T$.
   - Define a graph $G^k = (\Theta^k, E^k)$ where $\Theta^k := \{\theta_u^k : u \in T\}$ and $E^k$ denote the set of all the edges that join any two configurations in $\Theta^k$. The cost of traveling the edge joining any two distinct configurations $\theta_u^k, \theta_v^k$ is set to $d_{min}(\theta_u^k, \theta_v^k)$.
   - Use the Christofides algorithm [21] available for the single TSP to find a tour that visits each configuration exactly once. Let the sequence of configurations found by the Christofides algorithm be denoted by $\Pi^k$. Also, let the solution be denoted as $S_k := (\Pi^k, \Theta^k)$.
4. Output the solution with the minimum cost among $S_1$ and $S_2$.

In the above algorithm, $\Theta^1$ and $\Theta^2$ denote the two sets of configurations chosen in steps 1 and 2 of the algorithm. Fixing the configuration to visit each target reduces the CCURP to a single TSP. Given the configurations $\Theta^1$ and $\Theta^2$, let $\Pi_{opt}^1$ and $\Pi_{opt}^2$ be the optimal sequence of targets to visit respectively. For any feasible sequence of targets $(\Pi)$ and a corresponding set of configurations $(\Theta)$, let $C(\Pi, \Theta)$ represent the cost of the corresponding tour. Let $\Pi^*$ be the sequence of targets and let $\Theta^*$ be the corresponding configurations at each target in an optimal solution to the CCURP.

**Lemma 1** *For* $k = 1, 2$, $C(\Pi^k, \Theta^k) \le \frac{3}{2} C(\Pi^*, \Theta^k)$.

*Proof* Choose any $k \in \{1, 2\}$. The Christofides algorithm for the TSP has an approximation ratio of $\frac{3}{2}$ [21]. Given the set of configurations $\Theta^k$, as $\Pi_{opt}^k$ is an optimal solution to the TSP and $\Pi^k$ is the feasible solution obtained using the Christofides algorithm, we must have

$$C(\Pi^k, \Theta^k) \le \frac{3}{2} C(\Pi_{opt}^k, \Theta^k). \tag{1}$$

In addition, as $\Pi^*$ is a feasible sequence of targets to the CCURP, we must also have

$$C(\Pi_{opt}^k, \Theta^k) \le C(\Pi^*, \Theta^k). \tag{2}$$

From inequalities (1) and (2):

$$C(\Pi^k, \Theta^k) \le \frac{3}{2} C(\Pi^*, \Theta^k). \tag{3}$$

□

Given a configuration at a target, note that any other configuration in $\{CE, CN, CW, CS\}$ at the target can be reached through a sequence of at most two flips. For example, $CN$ can be reached from $CE$ using one flip and the UAVs travel $\frac{\pi R}{2}$ units during this flip. Similarly, $CW$ can be reached from $CE$ through a sequence of two flips and the UAVs travel $\pi R$ units during this motion. The following lemma bounds the cost of choosing the configurations in $\Theta^1$ in terms of the optimal cost of the CCURP.

Let $n_1$ denote the number of targets whose configurations in $\Theta^1$ differ from the corresponding optimal set of configurations in $\Theta^*$ by one flip. Let $n_2$ denote the number of targets whose configurations in $\Theta^1$ differ from the corresponding optimal set of configurations in $\Theta^*$ by two flips.

**Lemma 2** $C(\Pi^*, \Theta^1) \leq C(\Pi^*, \Theta^*) + n_1\pi R + 2n_2\pi R$.

*Proof* Let the sequence of targets visited by the UAVs in $\Pi^*$ be denoted by $(u_1, u_2, \cdots, u_n)$. For all $i = 1, .., n$, let $\theta^1_{u_i}$ represent the UAV configuration at target $u_i$ in $\Theta^1$. Similarly, let $\theta^*_{u_i}$ represent the UAV configuration of target $u_i$ in $\Theta^*_{u_i}$. As the travel costs satisfy the triangle inequality, we have

$$\begin{aligned} d_{min}(\theta^1_{u_i}, \theta^1_{u_{i+1}}) &\leq d_{min}(\theta^1_{u_i}, \theta^*_{u_{i+1}}) + d_{min}(\theta^*_{u_{i+1}}, \theta^1_{u_{i+1}}) \\ &\leq d_{min}(\theta^1_{u_i}, \theta^*_{u_i}) + d_{min}(\theta^*_{u_i}, \theta^*_{u_{i+1}}) \\ &\quad + d_{min}(\theta^*_{u_{i+1}}, \theta^1_{u_{i+1}}). \end{aligned} \tag{4}$$

Adding the above equation for all pairs of adjacent targets in $\Pi^*$, we get,

$$\begin{aligned} C(\Pi^*, \Theta^1) &= \sum_{i=1}^{n-1} d_{min}(\theta^1_{u_i}, \theta^1_{u_{i+1}}) + d_{min}(\theta^1_{u_n}, \theta^1_{u_1}) \\ &\leq \sum_{i=1}^{n-1} d_{min}(\theta^*_{u_i}, \theta^*_{u_{i+1}}) + d_{min}(\theta^*_{u_n}, \theta^*_{u_1}) \\ &\quad + 2\sum_{i=1}^{n} d_{min}(\theta^1_{u_i}, \theta^*_{u_i}) \\ &= C(\Pi^*, \Theta^*) + 2\sum_{i=1}^{n} d_{min}(\theta^1_{u_i}, \theta^*_{u_i}). \end{aligned} \tag{5}$$

Note that $d_{min}(\theta^1_{u_i}, \theta^*_{u_i})$ is equal to $\frac{\pi R}{2}$ if the configurations $\theta^1_{u_i}$ and $\theta^*_{u_i}$ differ by one flip. Also, $d_{min}(\theta^1_{u_i}, \theta^*_{u_i})$ is equal to $\pi R$ if the configurations $\theta^1_{u_i}$ and $\theta^*_{u_i}$ differ by two flips. Therefore,

$$C(\Pi^*, \Theta^1) \leq C(\Pi^*, \Theta^*) + n_1\pi R + 2n_2\pi R.$$

□

Based on the choice of configurations in $\Theta^2$, note that there must be $n_1$ targets whose configurations in $\Theta^2$ differ from the corresponding optimal set of configurations in $\Theta^*$ by one flip. Also, there must be $n - n_2 - n_1$ targets whose configurations in $\Theta^2$ differ from the corresponding optimal set of configurations in $\Theta^*$ by two flips. Using the same reasoning as in the proof of Lemma 2, we can deduce the following result.

**Corollary 1** $C(\Pi^*, \Theta^2) \leq C(\Pi^*, \Theta^*) + n_1\pi R + 2(n - n_1 - n_2)\pi R$.

**Lemma 3** *The cost, C, of any feasible tour to the CCURP must be at least* $\frac{n\pi R}{2}$ *units.*

*Proof* Let $\Pi = (s_1, s_2, ...s_n)$ be the sequence of targets visited by the UAVs and $\Theta = \{\theta_1, ...\theta_n\}$ be the corresponding set of configurations at each target in a feasible tour. The cost of the tour is:

$$C = \sum_{i=1}^{n-1} d_{min}(\theta_{s_i}, \theta_{s_{i+1}}) + d_{min}(\theta_{s_n}, \theta_{s_1}). \tag{6}$$

As the UAVs execute at least one flip while traveling between any two successive configurations along the tour, we must have $d_{min}(\theta_{s_i}, \theta_{s_{i+1}}) \geq \frac{\pi R}{2}$ for $i = 1, \cdots, n-1$ and $d_{min}(\theta_{s_n}, \theta_{s_1}) \geq \frac{\pi R}{2}$. Hence, the lemma follows. □

**Theorem 1** *The approximation factor of Approx is* $\frac{9}{2}$.

*Proof* The computational complexity of $Approx$ is dominated by the Christofides algorithm which runs in polynomial time. $Approx$ chooses the best of the two solutions $(\Pi^1, \Theta^1)$ and $(\Pi^2, \Theta^2)$. Combining Lemmas 1, 2 and Corollary 1, we get,

$$\begin{aligned} &\min(C(\Pi^1, \Theta^1), C(\Pi^2, \Theta^2)) \\ &\leq \frac{3}{2}\min(C(\Pi^*, \Theta^1), C(\Pi^*, \Theta^2)) \\ &\leq \frac{3}{2}\left[C(\Pi^*, \Theta^*) + \pi R \min\{n_1 + 2n_2, 2n - (n_1 + 2n_2)\}\right]. \end{aligned} \tag{7}$$

Given $n_1 + n_2 \leq n$, it is easy to verify that $\min\{n_1 + 2n_2, 2n - (n_1 + 2n_2)\} \leq n$. Therefore,

$$\begin{aligned} &\min(C(\Pi^1, \Theta^1), C(\Pi^2, \Theta^2)) \\ &\leq \tfrac{3}{2}\left[C(\Pi^*, \Theta^*) + n\pi R\right] \\ &\leq \tfrac{3}{2}\left[C(\Pi^*, \Theta^*) + 2C(\Pi^*, \Theta^*)\right] \text{ (using Lemma 3)} \\ &= \tfrac{9}{2}C(\Pi^*, \Theta^*). \end{aligned}$$

□

## 2.2 Heuristic

We also implement a modification of the approximation algorithm referred to as $Approx_{lkh}$ for improving the quality of the solutions produced by $Approx$. In this modification, we use the Lin-Kernigan Heuristic (LKH) [7] instead of the Christofides algorithm to find a tour in *Step 3* of *Approx*. The LKH is the best known heuristic to solve TSP; it is known to produce optimal solutions to most benchmark TSP instances within a few seconds of computation time.

## 3 Three UAV Problem

### 3.1 Problem Statement

In this architecture, the objective of the routing problem is to find trajectories for the three UAVs, such that

– each target UGS is visited at least once by one of the UAVs,
– the traveling UAV is within a distance of $R$ from the other two UAVs, which are located at two UGS,
– the sum of the distances traveled by the three UAVs is minimized.

Let $V$ be the set of targets/vertices, which denote the locations of the UGS in the restricted zone. We define a configuration and adjacent configurations as follows:

*Configuration:* Three UAVs located at three UGS such that the distance between any two of them is less than the communication range $R$. In the Fig. 3, three UAVs are located at UGS $a$, $b$, and $c$ forming an admissible configuration. Even when the UAVs are not located at those UGS, we would refer to those set of three UGS as a possible configuration or simply a configuration.

*Adjacent Configurations:* Two configurations are adjacent if they have two UGS in common. In Fig. 3, configurations $(a, b, c)$ and $(a, b, d)$ share two common UGS, $a$ and $b$, and therefore are adjacent.

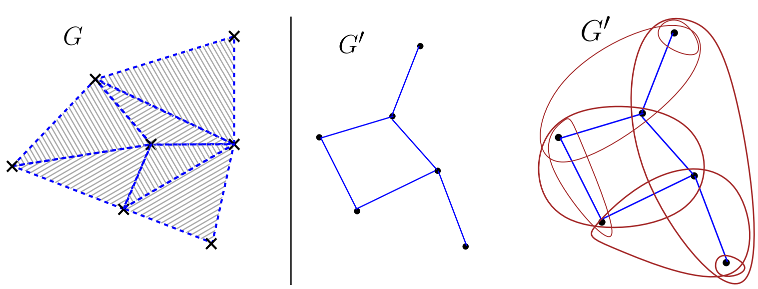
If two configurations are adjacent, the team of three UAVs can move from one configuration to another while satisfying the communication constraints. For example, in Fig. 3, the UAV located at $c$ can go to $d$ while communicating with the UAVs at $a$ and $b$, i.e. the UAVs move from configuration $(a, b, c)$ to $(a, b, d)$. The UAVs have to make these maneuvers to navigate across the zone without losing the communication links from each other. For the feasibility of the routing problem, we make the following assumptions:

*Assumption 1:* For every UGS $i \in V$, there exists at least one configuration $\mathcal{C}_k$ such that, $i \in \mathcal{C}_k$. In other words, there are no UGS in $V$ separated by a distance greater than $R$ from every other UGS.

*Assumption 2:* There are no isolated configurations, i.e. the UAVs can start from any configuration and reach any other configuration.

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent two configurations of UGS $(i, j, k)$ and $(j, k, l)$ respectively. The two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$ are adjacent as they have two common UGS. The UAVs can travel from $\mathcal{C}_1$ to $\mathcal{C}_2$ in a single maneuver and let $D(\mathcal{C}_1, \mathcal{C}_2)$ be the distance traveled by the UAVs during this maneuver. $D(\mathcal{C}_1, \mathcal{C}_2)$ is equal to the euclidean distance between the UGS $i$ and $l$. An UGS can be present in more than one possible configurations. For example, in Fig. 3, UGS $a$ is present in two configurations $(a, b, c)$ and $(a, b, d)$. Let $S_a = \{\mathcal{C}_{a_1}, \ldots \mathcal{C}_{a_l}\}$ be the set of configurations in which UGS $a$ is present in each of the configurations. UGS $a$ is considered to be visited if the UAVs reach any of the configurations $\mathcal{C}_l \in S_a$.

To solve the routing problem, one needs to identify the configurations to visit and the sequence in which they have to be visited, $\mathcal{C}_1 \cdots \mathcal{C}_p$ such that, every target




(a) $G'$ with nodes and edges added (b) Configurations sets

**Fig. 7** Construction of the graph $G'$

**Table 1** Average simulation time in seconds

| Targets | Algorithm *Approx* | Algorithm $Approx_{lkh}$ | OST[a] |
|---|---|---|---|
| 10 | 0.071 | 0.133 | 2.489 |
| 20 | 0.101 | 0.245 | 10.942 |
| 30 | 0.312 | 0.579 | 65.468 |
| 40 | 0.875 | 1.178 | 404.185 |

[a]One-in-a-set transformation

in $V$ is visited at least once by one of the UAVs and the total distance traveled by all the UAVs is minimum.

$$\text{Minimize} \quad \sum_{i=1}^{p-1} D(\mathcal{C}_i, \mathcal{C}_{i+1}) + D(\mathcal{C}_p, \mathcal{C}_1) \tag{8}$$

$$\text{Subject to:} \quad V \subseteq \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_p. \tag{9}$$

## 3.2 Solution Methodology

### *3.2.1 Graph Transformation*

Similar to the two UAV routing problem, we can pose this three vehicle problem as a one-in-a-set TSP using a graph transformation. Let $G = (V, E)$ be a graph, where each vertex represent the location of an UGS. One can construct a graph $G'$, where each vertex in $G'$ corresponds to a configuration in $G$. Construction of Graph $G'$:

- Identify all the possible configurations of UGS in $G$, i.e. triplets of UGS such that the distance between any pair of UGS is less than the communication distance $R$. These configurations are represented by filled triangles in the Fig. 7a.
- Corresponding to each configuration in $G$, add a vertex in $G'$.
- If two configurations are adjacent in $G$, add an edge between the corresponding vertices in $G'$. The weight of this edge is equal to the distance traveled by the UAVs to go from the one configurations to the other. Fig. 7a shows the graph $G'$ with vertices and edges.
- For each UGS $a \in V$, identify the set of configurations (filled triangles) in which $a$ is present, and call this set $S_a$. In Fig. 7b, the sets for each UGS are shown in brown.
- Add an edge between every pair of vertices with weight equal to the distance of the shortest path between them.
- Solve the following one-in-a-set TSP: find a tour on $G'$, such that at least one vertex/configuration from each set $S_a$, $a \in V$ is visited.

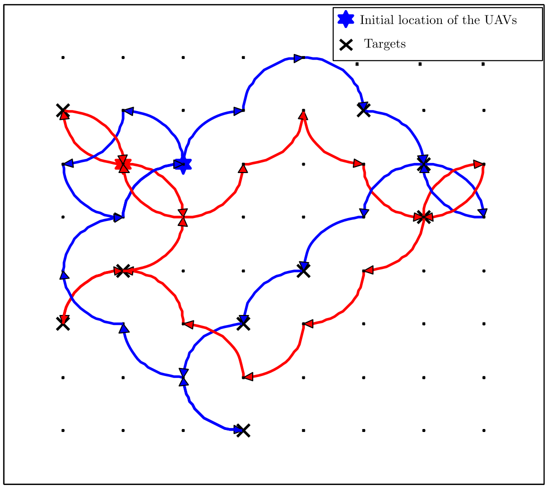
### *3.2.2 One-in-a-Set TSP*

The problem posed as one-in-a-set TSP on $G'$ does not contain mutually exclusive node sets. In [16], a technique is presented to transform this problem into an one-in-a-set TSP instance with mutually exclusive node-sets. Then it is transformed into a regular asymmetric TSP. Here we present an overview of the transformation, but one can refer to [16] for detailed description and proofs. Consider the one-in-a-set TSP on the configurations graph $G'$ constructed in the Section 3.2.1. Let the set of nodes of $G'$ be $V'$ and edges be $E'$. Each node $i \in V'$ belongs to one or more node-sets, and let $\mathcal{M}_i$ be the set of node-sets of which node $i$ is a member, and $|\mathcal{M}_i|$ is the size of this set. We will transform the given graph in four stages into the one-in-a-set TSP with mutually exclusive node-sets.

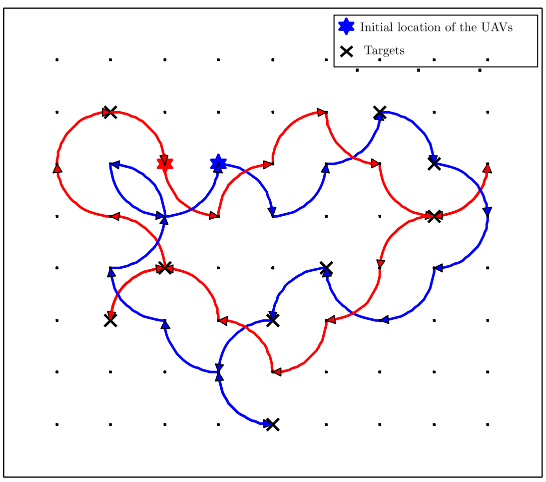
- *Stage 1:* Construct a graph $G'_1$ with same set of nodes ($V'_1 = V'$) as of $G'$, but different edges. Add only the edges in $G'$ which enters at least one new node-set, and the cost of these edges are same as the cost of the edges in $G'$.
- *Stage 2:* Construct graph $G'_2$ with same set of nodes ($V'_2 = V'_1$) and edges as in $G'_1$. Let $c_{ij}$ be

**Table 2** A posterior bound using the proposed algorithms

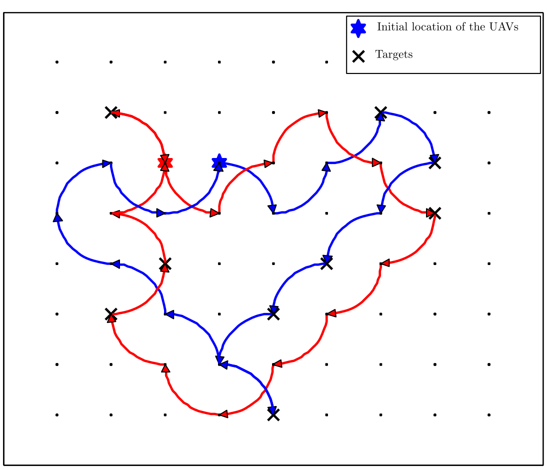
| Targets | Algorithm *Approx* | | Algorithm $Approx_{lkh}$ | |
|---|---|---|---|---|
| | Avg. Solution quality | Max. Solution quality | Avg. Solution quality | Max. Solution quality |
| 10 | 1.089 | 1.230 | 1.051 | 1.114 |
| 20 | 1.140 | 1.249 | 1.088 | 1.144 |
| 30 | 1.176 | 1.271 | 1.119 | 1.156 |
| 40 | 1.220 | 1.346 | 1.142 | 1.185 |




(a) Tour obtained using algorithm *Approx*




(b) Tour obtained using algorithm $Approx_{lkh}$




(c) Optimal tour

**Fig. 8** An instance of the CCURP with 10 targets solved using different algorithms

the cost of the edge $(i, j) \in E'$. Define the cost of each edge in $G'_2$ as $c_{ij} + m\alpha$, where $\alpha$ is a positive number, strictly greater than the sum of the cost of all edges in $G'_1$ and $m$ is the number of new node sets, the edge $(i, j)$ enters.

– *Stage 3:* For each node $i \in V'_2$ with multiple node-set membership, create $|\mathcal{M}_i|$ replicas of the node, each corresponding to a node set. Add edges between each of these nodes (replicas) to the other nodes with cost same as the cost in $G'_2$. Add edges between the replicas of each node with zero cost and refer to this new graph as $G'_3 = (V'_3, E'_3)$.
– *Stage 4:* In each node-set, remove the edges between nodes belonging to the same set, and refer to this graph as $G'_4$.

After this transformation through the four stages, the problem is posed as one-in-a-set TSP on graph $G'_4$ with mutually exclusive node-sets. Now this can be transformed into a regular asymmetric TSP using the result in [16].

## 4 Computational Results

To test the algorithms for the two UAV problem, a restricted zone of size 30 × 30 units is chosen. This zone is divided into squares of size equal to 1 unit. The targets are located randomly in this area. 50 instances are generated for each problem size with 10, 20, 30 and 40 targets. The LKH program [7] by Helsgaun available at http://www.akira.ruc.dk/~keld/research/LKH/ is used to solve the single asymmetric TSP in $Approx_{lkh}$. The LKH program was run without changing any of its default settings. All the simulations are run on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @ 2.53GHz, 12GB RAM).

For a given problem instance $I$, the bound on the *a posterior* guarantee provided by an algorithm is defined as $\frac{C^I_{sol_f}}{C^I_{opt}}$ where $C^I_{sol_f}$ is the cost of the feasible solution found by the algorithm and $C^I_{opt}$ is the optimal cost.

The optimal cost for an instance is found by posing the CCURP as an one-in-a-set TSP and then transforming it into a single symmetric TSP in the following way: Given $n$ sets of vertices with each set containing at least one target, the objective of the one-in-a-set

TSP is to determine a selection of a target from each set and the sequence in which the selected targets must be visited so that the total distance traveled by the salesman is a minimum. The CCURP can be readily cast as a one-in-a-set TSP by associating the sets of configurations $\Theta_i = \{CE, CN, CW, CS\}$, $i = 1, 2, \ldots, n$ as the given $n$ sets and the configurations as be the targets of the one-in-a-set TSP. We then transform the one-in-a-set TSP to a regular asymmetric TSP using the result in [16], and finally, transform the asymmetric TSP into a symmetric TSP using the result in [9]. The resulting symmetric TSP is solved using the *Concorde* solver [1].

The simulation results are shown in Tables 1 and 2. The algorithms $Approx$ and $Approx_{lkh}$ are relatively very fast compared to the transformation method used to find the optimal solution to the CCURP. We are able to solve every instance using $Approx$ and $Approx_{lkh}$ within two seconds. On the other hand, finding an optimal solution using the *Concorde* solver required more than 60 minutes for some instances. The average and worst case *posterior* guarantee of the solutions found by $Approx$ and $Approx_{lkh}$ is shown in Table 2. These results show that the proposed algorithms are able to find high quality solutions for the tested instances relatively fast. Solutions obtained for an instance using the proposed algorithms are shown in Fig. 8a, b. An optimal solution for this instance is shown in Fig. 8c.

The three UAV CCURP is solved using the graph transformation method explained in Section 3.2.1. Four benchmark traveling salesman problem instances are chosen from TSPLIB [19] for the three UAV CCURP. The city locations given in the TSPLIB instances are chosen as the UGS locations for the CCURP. The communication radius of the UAVs needs to be chosen for each instance such that there exists at least one feasible solution. For a given instance and communication radius $R$, for each UGS, there should be at least two other UGS within $R$. For each UGS $u \in V$, we compute the second shortest of all the distances $d_u^2$ between itself and other UGS. We choose the communication radius $R$ to be $\alpha$ times the maximum of all those distances, $R = \alpha \max_{u \in V} d_u^2$, where $\alpha \geq 1$ is a scaling factor. The computational results for four instances with four different values of $\alpha$ are presented in Table 3. The higher the value of $\alpha$, the communication radius is larger and hence more configurations are feasible, and the tour cost is lesser. This is as predicted since with more feasible

**Table 3** Computational results for three UAV CCURP

| Instance | # Targets | $\alpha$ | R | # configs | Tour Cost | Time (secs) |
|---|---|---|---|---|---|---|
| burma14 | 14 | 1.1 | 4.47 | 75 | 53.0993 | 3 |
| | 14 | 1.15 | 4.67 | 89 | 52.54 | 4 |
| | 14 | 1.2 | 4.88 | 122 | 47.22 | 11 |
| | 14 | 1.275 | 5.2 | 138 | 46.59 | 14 |
| ulysses16 | 16 | 1.1 | 17.56 | 458 | 99.34 | 147 |
| | 16 | 1.15 | 18.36 | 458 | 99.34 | 147 |
| | 16 | 1.2 | 19.16 | 465 | 88.66 | 147 |
| | 16 | 1.3 | 20.75 | 491 | 87.47 | 168 |
| bays29 | 29 | 1.1 | 681.83 | 289 | 14811.3 | 84 |
| | 29 | 1.15 | 712.82 | 320 | 13996 | 124 |
| | 29 | 1.2 | 743.81 | 382 | 13152.6 | 182 |
| | 29 | 1.3 | 805 | 498 | 12400.5 | 308 |
| eil51 | 51 | 1.15 | 15.13 | 197 | 857.6 | 59 |
| | 51 | 1.2 | 15.79 | 238 | 836.98 | 83 |
| | 51 | 1.25 | 16.45 | 312 | 736.46 | 145 |
| | 51 | 1.3 | 17.11 | 374 | 734.73 | 213 |

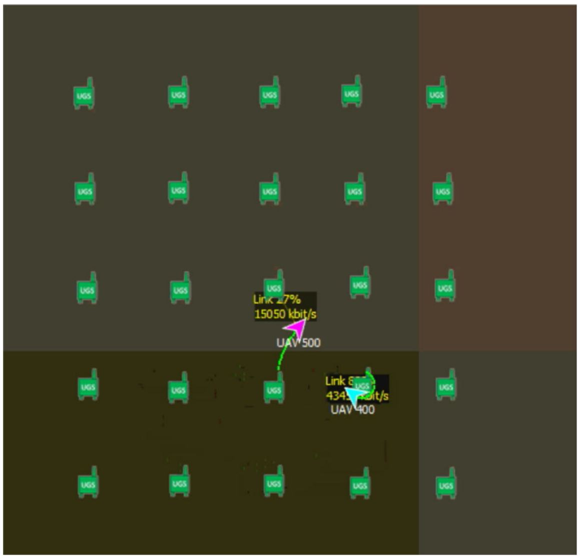



**Fig. 9** Simulation of the two UAV CCURP on AMASE

configurations, there can exist more tours on the configuration graph and minimum of those could be less than or equal to the instances with fewer configurations.

### 4.1 AMASE Simulation

AMASE is a simulation environment developed at the Air-Force Research Laboratory for the cooperative control study and analysis of the UAVs. It is used to study the effectiveness and feasibility of control algorithms necessary for (multi) UAV missions. A circumnavigation controller was developed at AFRL for the navigation of UAVs using range and range-rate measurements relative to a stationary or moving object. Using this controller, an instance of the two UAV CCURP is simulated successfully on AMASE. For the simulation, 25 UGS were deployed uniformly as a square grid and five target locations are selected randomly. A snapshot of the simulation is shown in Fig. 9. The green arrows represent the idling UAV and the pink arrow represents the orbiting UAV. The UAVs considered here are fixed wing aircrafts and therefore cannot loiter at a stationary location in the air. We make the following assumptions for this simulation: $(i)$ The idling UAV orbits the UGS with the least turning radius possible, and is able to communicate with the UGS and localize itself while orbiting. $(ii)$ Using the range measurements from the idling UAV, the second UAV travels from one UGS to another.

## 5 Conclusion

We considered a problem of routing UAVs with communication constraints in a GPS denied environment. We looked at two different architectures routing two and three UAVs communicating with each other and the UGS. We developed a $\frac{9}{2}$-approximation algorithm for the two UAV routing problems and a transformation method for the two and three UAV routing problems. These algorithms for two UAV problem were tested on 50 instances with 10, 20, 30 and 40 targets. The computation time needed by the approximation algorithm is less than a second, and it 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. A similar approach could be used for the three UAV case, however the $\frac{9}{2}$ approximation ratio may not be guaranteed in this case. Computational experiments were performed using the transformation algorithm for the three UAV problem; we chose four benchmark instances from TSPLIB, and tested the algorithm with different communication radii. The computation time required was less than six minutes for all of the instances.

An interesting extension for future work would be to consider the $n$-UAV communication constrained routing problem, where given the increased number of degrees of freedom, it would be important to address the issue of how to choose the "best" routing protocol. We did not investigate this problem here because, as the number of UAVs increases the routing problem turns into a sensor placement problem, which is beyond the scope of this paper.

#### Compliance with Ethical Standards

**Conflict of interests** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## References


1. Applegate, D., Bixby, R., Chvatal, V., Cook, W.: Concorde tsp solver. See: http://www.tsp.gatech.edu/concorde.html (2005)
2. Cao, Y., Muse, J., Casbeer, D., Kingston, D.: Circumnavigation of an unknown target using uavs with range and range rate measurements. In: IEEE 52nd Annual Conference on Decision and Control (CDC), 2013, pp. 3617–3622 (2013)
3. Casbeer, D.W., Krishnamoorthy, K., Eggert, A., Chandler, P., Pachter, M.: Optimal search for a random moving intruder. In: AIAA Infotech@Aerospace Conference (2012)
4. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. Rep. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh PA (1976)
5. Gebre-Egziabher, D., Taylor, B.: Impact and mitigation of gps-unavailability on small uav navigation, guidance and control. Tech. rep., Report 2012-2, University of Minnesota Department of Aerospace Engineering and Mechanics (2012)
6. Hashemi, A., Cao, Y., Casbeer, D.W., Yin, G.: Unmanned aerial vehicle circumnavigation using noisy range-based measurements without global positioning system information. J. Dyn. Syst. Meas. Control. **137**, 31,009–31,019 (2015)
7. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. European Journal of Operational Research **126**(1), 106–130 (2000). doi:10.1016/S0377-2217(99)00284-2. http://www.sciencedirect.com/science/article/pii/S0377221799002842
8. Oberlin, P., Rathinam, S., Darbha, S.: Today's traveling salesman problem. IEEE Robot. Autom. Mag. **17**(4), 70–77 (2010)
9. Jonker, R., Volgenant, T.: Transforming asymmetric into symmetric traveling salesman problems. Oper. Res. Lett. **2**(4), 161–163 (1983)
10. Kerns, A.J., Shepard, D.P., Bhatti, J.A., Humphreys, T.E.: Unmanned aircraft capture and control via GPS spoofing. J. Field Rob. **31**(4), 617–636 (2014)
11. Krishnamoorthy, K., Casbeer, D.W., Chandler, P., Pachter, M., Darbha, S.: UAV Search Andamp; Capture of a Moving Ground Target under Delayed Information. In: IEEE Conference on Decision and Control (2012)
12. Malik, W., Rathinam, S., Darbha, S.: An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem. Oper. Res. Lett. **35**(6), 747–753 (2007)
13. Manyam, S., Rathinam, S., Darbha, S.: Computation of Lower Bounds for a Multiple Depot, Multiple Vehicle Routing Problem with Motion Constraints. In: IEEE 52Nd Annual Conference on Decision and Control (CDC), 2013, pp. 2378–2383 (2013)
14. Manyam, S., Rathinam, S., Darbha, S., Casbeer, D., Chandler, P.: Routing of Two Unmanned Aerial Vehicles with Communication Constraints. In: International Conference on Unmanned Aircraft Systems (ICUAS), 2014, pp. 140–148 (2014)
15. Manyam, S.G., Rathinam, S., Darbha, S.: Computation of lower bounds for multiple depot, multiple vehicle routing problem with motion constraints. ASME J. Dyn. Syst. Meas. Control. **137**(9), 094,501–094,501-5 (2015)
16. Noon, C.E., Bean, J.C.: An efficient transformation of the generalized traveling salesman problem. INFOR **31**(1), 39–44 (1993)
17. Potyagaylo, S., Rand, O., Kanza, Y.: Motion Planning for an Autonomous Helicopter in a Gps-Denied Environment. In: Proceedings of the American Helicopter Society 66Th Annual Forum, Phoenix, AZ, USA (2010)
18. Rathinam, S., Sengupta, R., Darbha, S.: Resource allocation algorithm for multivehicle systems with nonholonomic constraints. IEEE Trans. Autom. Sci. Eng. **4**(1), 98–104 (2007)
19. Reinelt, G.: Tsplib—a traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991)
20. Shames, I., Dasgupta, S., Fidan, B., Anderson, B.: Circumnavigation using distance measurements under slow drift. IEEE Trans. Autom. Control **57**(4), 889–903 (2012)
21. Steiglitz, K., Papadimitriou, C.H.: Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, New Jersey (1982)
22. Summers, T.H., Akella, M.R., Mears, M.J.: Coordinated standoff tracking of moving targets: control laws and information architectures. J. Guid. Control. Dyn. **32**(1), 56–69 (2009)
23. Sundar, K., Rathinam, S.: Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. IEEE Trans. Autom. Sci. Eng. **11**(1), 287–194 (2014)
24. Sundar, K., Rathinam, S.: Generalized multiple depot traveling salesman problem - polyhedral study and exact algorithm. Journal of Computers and Operations Research (2015). doi:10.1016/j.cor.2015.12.014
25. Warwick, G.: Lightsquared tests confirm GPS jamming. Aviation Week (2011)
26. Wu, A.D., Johnson, E.N., Kaess, M., Dellaert, F., Chowdhary, G.: Autonomous flight in gps-denied environments using monocular vision and inertial sensors. Journal of Aerospace Information Systems **10**(4), 172–186 (2013)
27. Yadlapalli, S., Malik, W., Darbha, S., Pachter, M.: A lagrangian based algorithm for a multiple depot, multiple traveling salesman problem. Nonlinear Anal. Real World Appl. **10**(4), 1990–1999 (2009)

**Satyanarayana Gupta Manyam** is NRC Research Associate at Air Force Research Laboratory, Wright-Patterson Air Force Base, OH. He received the Ph. D degree in Mechanical Engineering from Texas A & M University, College Station, TX in 2015. He also holds the M.S degree in Mechanical Engineering from State University of New York at Buffalo, Buffalo, NY, and the B.E degree in Mechanical Engineering from BITS, Pilani, India. His main research interests include cooperative routing of multi-vehicle systems, optimal path planning for autonomous vehicles, approximation algorithms and servo regulators for nonlinear systems.

**Sivakumar Rathinam** is currently an associate professor in mechanical engineering at the Texas A&M University, College Station. Prior to joining A&M, he was an associate research scientist at the University of California-NASA Ames Research Center. He received his Ph.D. degree in civil-systems engineering and M.S. in electrical engineering and computer science from the University of California in 2007 and 2006, respectively. His research interests include motion planning, combinatorial optimization, air traffic control and vision-based control.

**Swaroop Darbha** received his B.Tech. from the Indian Institute of Technology - Madras in 1989 and M.S. and Ph.D. degrees from the University of California in 1992 and 1994, respectively. He was a postdoctoral researcher at the California PATH program from 1995 to 1996. He has been on the faculty of mechanical engineering at Texas A&M University since 1997, where he is currently a professor. His current research interests lie in the development of diagnostic systems for air brakes in trucks, development of planning, control and resource allocation algorithms for a collection of UAVs.

**David Wellman Casbeer** received the B.S. and Ph.D. degrees in Electrical Engineering from Brigham Young University, Provo, UT, in 2003 and 2009, respectively. He was at the University of Sydney in 2007 as a visiting researcher studying distributed estimation and Gaussian processes. He is currently a Research Engineer in the Control Science Center of Excellence within the Air Force Research Laboratory with a focus on control and estimation for multiple UAVs under uncertainty. Dr. Casbeer is on the editorial board for the Journal of Intelligent & Robotic Systems. He is a Senior member of the IEEE and the AIAA. He is the chair elect for the AIAA Intelligent Systems Technical Committee.

**Yongcan Cao** received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree from Utah State University, Logan, in 2010, all in electrical engineering. He worked as a Postdoc Researcher at Utah State University from September 2010 to September 2011 and a Postdoc Scholar at University of California, Riverside, from October 2011 to February 2012. He worked at the Air Force Research Laboratory, Wright-Patterson AFB, from March 2012 to July 2015. Since August 2015, he is an Assistant Professor in the Department of Electrical and Computer Engineering, University of Texas at San Antonio. His research interest focuses on autonomous vehicles, cyber-physical systems, sensor networks, and human-robot interaction. He is currently an Associate Editor on the IEEE Control Systems Society Conference Editorial Board and the IEEE Robotics and Automation Society Conference Editorial Board.

**Phillip R. Chandler** is a researcher and research team leader in UAVs. The team works in the development of autonomous and cooperative control at the Air Force Research Laboratory at Wright-Patterson AFB. Current research focus is on the cooperative control of a team of vehicles performing joint tasks with partial information over asynchronous networks. Mr. Chandler is technical lead on the Cooperative Operations in Urban Terrain flight demonstration program which includes small and micro air vehicles. Previous research efforts have included real time parameter identification and on-line control design. This work transitioned to the RESTORE and Self Designing Controller programs. Past experience also includes position of chief engineer on the Self-Repairing Flight Control System flight test program.