ROUTING VEHICLES WITH MOTION, RESOURCE AND MISSION CONSTRAINTS: ALGORITHMS AND BOUNDS

A Dissertation

by

SATYANARAYANA GUPTA MANYAM

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Swaroop Darbha
Co-Chair of Committee,	Sivakumar Rathinam
Committee Members,	Rajagopal Kumbakonam
	Shankar Bhattacharyya
Head of Department,	Andreas Polycarpou

May 2015

Major Subject: Mechanical Engineering

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are used for several military and civil applications such as reconnaissance, surveillance etc. The UAVs, due to their design and size limitations, have inherent kinematic constraints, communication constraints etc. This thesis considers the path planning problems for UAVs while satisfying a class of constraints.

We consider a multiple depot UAV routing problem, where the vehicles have motion constraints due to bound on their yaw-rate. For a given set of targets, it is required that each target should be on the path of at least one of the vehicles. This problem is hard to solve and currently there are no algorithm that could find an optimal solution. We aim to find tight lower bounds for this problem via Lagrangian relaxation. The complicating constraints of the problem are relaxed, and the cost function is penalized whenever those constraints are violated. This reduces the original problem to a known problem - a standard multiple traveling salesmen problem (MTSP). Simulation results are presented to show that this method significantly improved the existing lower bounds.

The second problem we consider is the routing of UAVs in GPS denied environments and with limited communication range. Two different architectures for navigation assisted by an array of Unattended Ground Sensors (UGSs) are considered. In the first case, when an UAV localizes itself by communicating with an UGS, the second UAV can orbit around the first UAV. Contact with UGS allows them to act as beacons for relative navigation eliminating the need for GPS. A randomized algorithm with approximation ratio of $\frac{9}{2}$ and a transformation technique are developed to solve this problem. In the second architecture, when two UAVs are

located at two different UGSs, the third UAV localizes by triangulation using range measurements from the first two UAVs. This three UAV case is solved using a graph transformation technique to pose it as an one-in-a-set TSP. The solutions produced by these algorithms were used to simulate the UAV routing on AMASE, a simulation tool for routing UAVs developed by the Air Force Research Laboratories.

DEDICATION

ఎందరో మహానుభావులు, అందరికీ వందనములు.

ACKNOWLEDGEMENTS

I sincerely thank all the people who directly or indirectly helped me complete my Ph.D. I am so grateful to my advisors Prof. Swaroop Darbha and Prof. Sivakumar Rathinam, who guided, motivated and envisioned me throughout my graduate studies with immense patience. I am also fortunate and thankful to have Prof. K. R. Rajagopal and Prof. Shankar Bhattacharyya on my commitee, who inspired me with their immaculate art of teaching.

I want to extend my deepest gratitude to my father M.P Vardhana Rao who planted the thought to become a teacher. I am grateful to my wife Anusha, who has patiently endured and encouraged me through out the journey. I want to extend my wholehearted thanks to my mother, sister and my all season friends, who supported me through the rough tides and serenely waiting me to graduate.

I would like to thank Dr. David Casbeer and Phil Chandler at Air Force Research Laboratories for their inputs to my research. I also want to thank the the Department of Mechanical Engineering at Texas A & M University for their support and assistance.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. MOTIVATION AND INTRODUCTION

Civil and military applications for Unmanned Aerial Vehicles (UAVs) have grown enormously in the past two decades. Due to their portability and lack of need for a pilot, UAVs could be deployed in dangerous and hard to reach terrains. In military applications, UAV are used for border patrolling, reconnaissance missions etc. [1, 2, 3], and in civil applications, they are used in search and rescue missions, traffic monitoring, wild fire monitoring etc. [4, 5, 6]. UAVs are envisaged as data mules to collect the data from sensor nodes in a sensor network [7, 8].

Along with many advantages, UAVs also have certain limitations. One such limitation is the fuel carrying capacity; one needs to plan efficient routes to make use of these limited resources. The fixed wing UAVs cannot rotate on their own axis, and therefore cannot change their heading instantaneously. One needs to take additional lengths resulting from motion constraints into account while solving the path planning problems. Another limitation of the UAVs is the on-board battery capacity, which restricts the communication range of the UAVs. UAVs rely on GPS signals for navigation and therefore, are vulnerable to GPS jamming, in which case UAVs need to rely on communication signals for navigation. Any two UAVs might be able to communicate only if the distance between them is at most a certain limit, which we refer to as the communication range (R) of the UAVs.

In this thesis, we consider routing problems for the UAVs with two different classes of constraints. In a typical surveillance or reconnaissance mission, a team of UAVs start from their initial locations, visit the given set of targets, and return to their starting location. We refer to the initial locations as depots and the path of the UAVs as tours. Once the tour is completed, the UAVs repeat the mission, and this

is done again and again. One needs to find a tour that minimizes the fuel expended for each tour for two reasons: (i) The limited available resources are efficiently used and with the available amount of fuel the UAVs can go for more missions. (ii) This may lead to shorter/faster tours, and hence each target would be visited by a UAV more frequently. Here, we make the following assumptions on the UAVs:

1. The inertia of the UAVs are negligible.
2. There are no obstacles in the paths of the UAVs.
3. The disturbances due to wind is negligible.
4. The UAVs fly at a constant speed and hence the cost of traveling is directly proportional to the length of the path.

We consider the fixed wing UAVs here, and they have finite speed and finite rate of change of heading. We assume that the inertia of the UAVs are negligible, and we can model the UAVs as Dubins vehicles. In this model, the rate of change of heading angle is upper bounded by a constant, Ω. If the speed of the vehicle is u, the minimum turning radius (ρ) is given by $\frac{u}{\Omega}$. This minimum turn radius criteria mandates that the paths of the UAVs need to have bounded curvature. In [9], Dubins solved the problem of finding shortest path of bounded curvature, and gave an algorithm to find the path of minimum length between two points with initial and final heading angles given. We use this result to obtain the length of the shortest paths between a given pair of targets and headings.

Using the Dubins result, the path planning problem reduces to a continuous/discrete optimization problem. We refer to this routing problem with one UAV (modeled as Dubins vehicle) as the Dubins traveling salesman problem (DTSP). We refer to the

routing problem with multiple UAVs (each initially located at different depots) as multiple depot multiple vehicle Dubins traveling salesman problem (MDMVDTSP).

To solve the DTSP/MDMVDTSP, one needs to identify the sequence of targets that are visited by each UAV and the heading angle of the UAVs at each target. This problem is a generalization of the traveling salesman problem (TSP) where a salesman starts from an initial location, visits the given cities, and returns to the starting location in such a way that the total distance traveled by him is a minimum. A generalization of this is a multiple traveling salesman problem (MTSP), where a group of salesman has to visit the given cities, and each city should be visited by at least one salesman. In the UAV routing problem, the underlying problem of assignment of targets to a UAV and identifying the sequence is same as the MTSP. This problem was studied extensively and a lot of results are available in the literature [10, 11, 12, 13, 14, 15, 16].

In the case of DTSP/MDMVDTSP the optimal sequence of targets depends on the length of path between each pair of targets, which depends on the heading angles of the UAV at each target while the optimal headings depends on the sequence of targets. Due to this reason, there is a coupling between two problems: the TSP/MTSP which can be formulated with discrete variables and the problem of finding optimal heading which are continuous variables. This coupling of the discrete and continuous optimization problems makes it hard to solve. One of the possible methods to solve this problem is to discretize the headings, i.e. the heading of the UAV at each target can only belong to a finite discrete set $\Phi = \{\phi_1, \phi_2, \cdots \phi_d\}$. Now one can pose this as an errand scheduling problem, which can be formulated using integer variables to determine the sequence of targets and the heading angle at each target.

There are currently no algorithms that could solve DTSP in polynomial time,

but there are approximation algorithms[1] and heuristics available. To corroborate the performance of the approximation algorithms or heuristics, one need to know a tight lower bound for the given problem. The lower bounds are important for two reasons: (*i*) They corroborate the performance of heuristics. (*ii*) They can be used to efficiently eliminate certain solutions in branch and cut procedures to solve the mixed integer linear programming problems. We are interested in computing tight lower bounds for the DTSP and MDMVDTSP using Lagrangian relaxation. That is, we relax some of the complicating constraints of the problem, and penalize the objective whenever these constraints are violated. This relaxed problem can be posed as an easier related problem, which can be solved using readily available results. The solution of this relaxed problem will be a lower bound to the actual minimization problem.

1.1 Lower bound computation for DTSP and MDMVDTSP

The first part of the thesis deals with the algorithms to compute lower bounds for the DTSP and MDMVDTSP. Here, the main idea is motivated by the method of Held and Karp in [17]; they use the duality to compute the lower bounds for the TSP. They considered the formulation of TSP in [18], where the decision variables indicate which target/vertex has to be visited next after visiting a target. A visit from one target to another target is called as an edge from first target to the second. The TSP induces a constraint (degree constraint) that there should be at least two edges incident on each vertex, which can be interpreted as one incoming edge and one outgoing edge. In [17], the degree constraint is relaxed and penalized the cost function using penalty (or dual) variables. This relaxation reduces the TSP to a known problem of finding

[1] An α-approximation algorithm finds a feasible solution to the problem in polynomial time, and provides a guarantee that the solution is within α times the optimal solution. But in actual computation, it may produce solutions better than the proven *apriori* guarantee.

minimum spanning tree, for which combinatorial algorithms are available to solve. Therefore, for any given set of penalty variables, a lower bound to the TSP could be computed. Held-Karp also found a method to compute the penalty variables that would give the best lower bounds, and it was found to be with in 1% for most instances of TSP.

We follow a similar approach to compute tight lower bounds for the DTSP/ MD-MVDTSP. The complicating constraints for the DTSP are the yaw rate constraints. If the motion constraints (yaw rate) are relaxed, this problem reduces to an Euclidean TSP(ETSP), i.e. TSP with Euclidean distances as the cost of travel between any pair of targets. Hence, the solution to the ETSP is a lower bound to the optimal solution of the DTSP. In this thesis, we relax the motion constraint of the UAV at target locations, (i.e. at the target locations, the incoming heading angle of the UAV may not be equal to the outgoing heading angle), and penalize the cost function whenever this constraint is violated. After the relaxation the DTSP reduces to a regular asymmetric TSP[2] (ATSP). This resulting ATSP is NP hard problem too, but there are algorithms available that would give a very tight lower bound for the ATSP. This in turn is a lower bound to the optimal solution of the DTSP. To find the set of penalty variables that provides the best lower bounds, we used a sub-gradient algorithm. It is an iterative procedure that updates the penalty variables in each iteration and produces better lower bounds after each update. We extended this technique to the multiple depot case (MDMVDTSP) too. In the multiple depot case, after relaxing the motion constraints and penalizing the cost function, MDMVDTSP reduces to a multiple traveling salesman problem (MTSP). We can transform the MTSP into an ATSP using the result in [19], and compute a tight lower bound to the resulting

[2]In an ATSP, the cost of travel from target i to target j may not be equal to the cost of travel from target j to target i.

ATSP, which in turn is a lower bound to the MDMVDTSP.

It was proved in [20] that the length of the Dubins path between two targets (i and j) is convex with respect to the heading angles at the targets (θ_i and θ_j), when the distance between those two targets is at least 4ρ units. Using this property of the Dubins paths, we developed another method to compute lower bounds for those instances of DTSP/MDMVDTSP where the distance between every pair of targets is at least 4ρ units. We discretize the heading angles at each target, i.e. the heading angles are restricted to a discrete set $\Phi = \{\phi_1, \cdots \phi_d\}$, and pose the DTSP as a one-in-a-set TSP. We proved that $C_{ost} - 4N\delta$ is a lower bound to the DTSP where C_{ost} is the optimal solution to the discretized DTSP posed as the one-in-a-set TSP, N is the number of targets and δ is the difference between any two consecutive angles in the set Φ. We extended this technique to the multiple depot case (MDMVDTSP) too.

1.2 GPS denied UAV routing

The second part of the thesis considers a routing problem in GPS-denied environments involving a team of Unmanned Aerial Vehicles (UAVs) and a set of Unattended Ground Sensors (UGSs). Access to GPS is an important requirement for the navigation of the UAVs, which makes UAVs vulnerable to GPS jamming and spoofing [21, 22]. The surveillance missions could possibly be in hostile environments, and the UAVs may be denied the access to the GPS signals. We consider a scenario where a team of UAVs need to be routed for patrol in a GPS-denied zone; the navigation of the UAVs is assisted by the UGSs deployed in the zone. The UAVs are equipped with range sensors on board. Based on the strength of their wireless communication link, they can estimate the distance between two UAVs or between a UAV and an UGS. Since the batteries powering the wireless signal have limited capacity, the UAVs can

estimate the range only if the distance between them is at most the certain limiting distance R. The UGSs have limited power too, are not networked, but an UGS can communicate with a UAV if it is located relatively close to the UGS. Here we assume they can communicate if a UAV is located vertically above the UGS at UAV's flying altitude. We refer to this routing problem of the UAVs with limited communication range as communication constrained UAV routing problem (CCURP).

We have considered two different architectures for the navigation of the UAVs aided by the UGS. In both the architectures, the geometry and the relative location of the UGSs are known by the UAVs. In the first architecture, two UAVs are needed for the mission. The two UAVs have orbiting controllers, which allows them to circumnavigate around the other UAV. When a UAV loiters above an UGS, the other UAV can orbit around the first UAV, and reach out to another UGS. The UAVs cooperatively perform a series of these maneuvers and navigate through zone to fulfill the mission. In the second architecture considered, a UAV estimates its states using the range measurements relative to two other UAVs idling above two different UGSs. If two range measurements between a UAV and two other known locations are available, states of the UAV can be estimated using triangulation. In this scenario, two UAVs loiter above two UGSs, a third UAV can travel from one UGS to other, as long as it is within the communication range from the first two UAVs. To fulfill the mission, the team of three UAVs have to visit all the UGSs/targets using the navigation scheme described.

If the UAVs have access to the GPS data, the routing problem for the UAVs is a generalization of the traveling salesman problem (TSP). A suite of algorithms to find optimal solutions, approximate solutions, lower bounds are available [23, 24, 25, 26, 27] for several generalizations of TSP with multiple vehicles, vehicles with motion constraints etc. The contributions of this section are the following:

- We developed a $\frac{9}{2}$-approximation algorithm for the two UAV CCURP.
- We presented a method to pose the two UAV problem as a one-in-a-set TSP which was solved by transforming it into an asymmetric TSP.
- For the three UAV CCURP, we developed a graph transformation method to transform it into a regular asymmetric TSP.

1.3 Contributions of the dissertation

- We presented an algorithm using Lagrangian relaxation to find tight lower bound to the DTSP. We presented a sub-gradient procedure to find the tightest possible bounds.
- The algorithm to find lower bound is extended to the multiple vehicle, MD-MVDTSP.
- For the instances of DTSP/MDMVDTSP which satisfy a distance criteria, we developed an algorithm to compute lower bounds using the convexity property of the Dubins paths.
- We presented two architectures for routing the UAVs in the absence of GPS, and with a limited communication range. In these two architectures, UAVs rely on range measurements for navigation. First architecture requires two UAVs and the second architecture requires three UAVs.
  - A $\frac{9}{2}$ approximation algorithm was developed to solve the Two-UAV CCURP.
  - A transformation method was developed to pose the Two-UAV CCURP as one-in-a-set TSP, which could be transformed into an ATSP.

– Two UAV CCURP is implemented in AMASE[3] for the feasibility study and simulation of the UAV routing.

– A transformation technique is developed to pose the Three-UAV CCURP as one-in-a-set.

[3]A tool-set developed at AFRL for simulation and demonstration of UAV routing and control technologies.

2. ROUTING UAVS WITH MOTION CONSTRAINTS*

2.1 Introduction

In this section, we present the algorithms to compute tight lower bounds for the DTSP and the MDMVDTSP. The objective of the DTSP is to find a path for the vehicle such that each target is visited at least once by the vehicle, the path satisfies the motion constraints of the vehicle and the length of the path is a minimum. To solve the DTSP, one has to find the heading angle for the vehicle at each target and the sequence in which the targets must be visited. Once the heading angles are known for any two adjacent targets along the tour, the result in [9] can be used to determine the path for the vehicle. Routing problems of this genre were earlier studied in [28],[23],[29],[24] and [25]. References [28] and [23] provide an approximate solution and an associated guarantee of sub-optimality. In [24], a two step approach is prescribed to solve a Multi Depot, Multiple TSP. The sequence of targets to be visited is first solved as a combinatorial problem, and the heading angle at each target is later computed using dynamic programming. The work in [25] deals with a Heterogeneous, Multi Depot, Multiple UAV Routing Problem (HMDMURP), where there are multiple heterogeneous vehicles required to visit a group of targets. HMDMURP is transformed into a standard Asymmetric TSP in [25] and solved using the Lin-Kernighan Helsgaun (LKH) heuristic [30]. The applications of these path

*Part of this section was reprinted with permission from the following articles:
S. Manyam, S. Rathinam, S. Darbha, and K. Obermeyer. "Computation of a lower bound for a vehicle routing problem with motion constraints," in *Proc. ASME Dynamic Systems and Control Conference*, 2012, pp. 695-701. Copyright © 2012 ASME.
S. Manyam, S. Rathinam, and S. Darbha, "Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2013, pp. 2378-2383.
S. Manyam, S. Rathinam, S. Darbha, and K. Obermeyer, "Lower bounds for a vehicle routing problem with motion constraints," Accepted for publication in *International Journal of Robotics and Automation*, Vol. 30, 2015. Copyright © 2015 ACTA Press.

planning problems with the curvature constraints were studied by several authors in [31, 32] and [33]. A study of these path planning algorithms is included in [34].

Even though there are heuristics and approximation algorithms, there are currently no algorithms that can either find an optimal solution or a tight lower bound for the DTSP. Lower bounds are important because they can be used to corroborate the quality of the solutions produced by the heuristics or the approximation algorithms. Currently, there is only one available method to compute lower bounds for the DTSP. One can relax the motion constraints of the vehicles, causing the problem reduced to a regular TSP. Here, the cost to travel between a pair of targets is equal to or proportional to the Euclidean distance between them.

Another approach to solve the DTSP is to restrict the heading angles of the vehicle at each target to a discrete set. This discretized DTSP (DDTSP) can be posed as a One in a Set TSP (OST) and a lower bound can be obtained for the OST [25]. For example, in [25], a transformation method which converts the OST into a standard TSP is used to find feasible solutions and compute lower bounds. However, the transformation method as prescribed in [25] does not provide tight lower bound for every instance of the DDTSP for the following reason: this method relies on modifying the travel costs by a large constant (this constant is generally called the big-M). As a result, the quality of the lower bound depends on the value of M and tends to deteriorate for large values of M (which usually happens as the size of the problem increases). In fact, as observed in the simulations, there are several instances where the transformation method produced a bound that was either negative or lower than the bound obtained by solving the Euclidean TSP (ETSP). In this context, the following are the objectives of this article for the DTSP where the vehicle is allowed to visit each target only at a specified set of heading angles:

- We aim to develop tight lower bounds for every instance tighter than the optimal solution of the corresponding Euclidean TSP.

- We aim to compare the bounds provided by all of the available methods to corroborate their performance.

To address the first objective, we provide a new method for computing a lower bound for the DTSP using Lagrangian relaxation. A Lagrangian relaxation is obtained by removing some of the constraints in the DTSP and penalizing the cost function whenever they are violated. Using the weak duality theorem, it follows that the cost of the solution to the Lagrangian relaxation is a lower bound to the optimal cost of the DTSP. The objective function of the Lagrangian relaxation is posed as an asymmetric TSP where the cost of traveling each edge is computed by solving a variational problem. This asymmetric TSP is solved using the Lin-Kernighan Helsgaun[†] (LKH) heuristic [30], which is one of the best known heuristics for the TSP in the literature. The LKH heuristic also gives a tight lower bound which serves as a lower bound to the DTSP. For any given set of dual/penalty variables, the solution of this relaxation gives a lower bound to the DTSP. Sub-gradient optimization techniques are used in order to obtain the best lower bound. Simulation results seem to corroborate that the proposed method produces a lower bound better than the transformation method [25] in almost all instances. Additionally, this lower bound

[†]LKH [30] is a local search algorithm available for solving the single TSP. It starts with a feasible tour and repeatedly attempts to move to a neighboring tour with a lower cost. Neighboring tours can be found by removing edges from the current tour and adding new edges appropriately. The crux of this search lies in the ability to find suitable edges to be added or removed from the current tour. It has been observed in simulations that the optimal Lagrangian dual cost of a TSP instance is very close to its optimal tour cost (within 1-2%) for practically every instance of the TSPLIB, and most of the edges present in an optimal dual solution are also present in the optimal tours of the TSP. Therefore, while adding new edges in LKH, priority is given to the the edges found in the dual solution as they act as proxies to the edges in the optimal tours. This basic idea was used by Helsgaun in [30], where he developed the LKH heuristic and obtained optimal tours for large TSP instances with high frequency.

is a significant improvement when compared to the solution of the ETSP.

2.2 Lagrangian based lower bounds for the Dubins traveling salesman problem

2.2.1 Problem formulation

Let $N = \{1, 2, ..., n\}$ be the set of given targets and $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ be the set of headings at the targets. Let E denote the set of all the edges joining any two vertices in N. Let x_{ij} be a binary decision variable which equals 1 if there is an edge from i to j in the tour and equals 0 otherwise. Let X be the matrix of decision variables, whose entry in the i^{th} row and j^{th} column is x_{ij}. Let $\mathcal{F}$ represent the set of all feasible solutions, such that in each solution, every target in N is visited. We will say that $X \in \mathcal{F}$, if the matrix corresponds to one of the feasible solutions. The DTSP can be stated as the following:

$$\min_{\theta, X} \sum_{(i,j) \in E} d_{ij}(\theta_i, \theta_j) x_{ij} \tag{2.1}$$

subject to:

$$X \in \mathcal{F}, \tag{2.2}$$

where $d_{ij}(\theta_i, \theta_j)$ is the length of the shortest path of bounded curvature, from vertex i at (x_i, y_i) with an initial heading θ_i to vertex j at (x_j, y_j) with a final heading θ_j. The length d_{ij} can be expressed in terms of the kinematics of the Dubins vehicle as:

$$d_{ij}(\theta_i, \theta_j) = \min_{u_{ij}} \quad (t_{ij}^f - t_{ij}^0), \tag{2.3}$$

subject to:

$$\dot{\zeta}_{ij} = \cos\theta_{ij}, \quad \dot{\eta}_{ij} = \sin\theta_{ij}, \quad \dot{\theta}_{ij} = u_{ij}, \quad |u_{ij}| \leq \Omega, \tag{2.4}$$

$$\zeta_{ij}(t_{ij}^0) = x_i, \quad \eta_{ij}(t_{ij}^0) = y_i, \tag{2.5}$$

$$\zeta_{ij}(t_{ij}^f) = x_j, \quad \eta_{ij}(t_{ij}^f) = y_j, \tag{2.6}$$

$$\theta_{ij}(t_{ij}^0) = \theta_i, \quad \theta_{ij}(t_{ij}^f) = \theta_j. \tag{2.7}$$

Here, ζ_{ij} and η_{ij} are the position coordinates of the Dubins vehicle in x and y directions, $\theta_{ij}(t_{ij})$ is the heading angle of the vehicle at time t_{ij}, t_{ij}^0 and t_{ij}^f are the initial and final times while traveling from vertex i to vertex j. The term $\dot{\theta}_{ij}$ is the yaw rate of the vehicle and it is upper bounded by Ω. When the minimum turning radius of a UAV equals to 1 unit (of distance), the term Ω equals 1. Vehicles with different turning radii can be modeled by changing the corresponding value of Ω.

Suppose the desired tour contains the edges (i, j) and (j, k) in the path of the vehicle, the arriving (final) heading of the vehicle while traveling from target i to target j should be equal to the departure (initial) heading of the vehicle while traveling from target j to target k. We do not know which target precedes others in the desired tour, but we know that there is only one incoming and outgoing edge incident on target j. So, we can make use of the binary variables x_{ij} to formally state the heading angle constraint at a target as follows:

$$\sum_{i:(i,j)\in E} \theta_{ij}(t_{ij}^f) x_{ij} - \sum_{k:(j,k)\in E} \theta_{jk}(t_{ij}^0) x_{jk} = 0. \quad \forall j \in N, \tag{2.8}$$

In general, equations (2.7) and (2.8) are difficult constraints to deal with. The domain of θ_{ij} is cylindrical and one has to identify 0 and 2π as one and the same. We

will pose these constraints using sines and cosines of the angles θ_{ij} as shown below:

$$\cos\theta_{ij}(t_{ij}^0) = \cos\theta_i, \quad \sin\theta_{ij}(t_{ij}^0) = \sin\theta_i, \tag{2.9}$$

$$\cos\theta_{ij}(t_{ij}^f) = \cos\theta_j, \quad \sin\theta_{ij}(t_{ij}^f) = \sin\theta_j. \tag{2.10}$$

In summary, the constraint on the heading angles (2.8) at each target j can be re-stated in terms of the sines and cosines as:

$$\sum_{i:(i,j)\in E} \cos\theta_{ij}(t_{ij}^f)x_{ij} - \sum_{k:(j,k)\in E} \cos\theta_{jk}(t_{ij}^0)x_{jk} = 0, \quad \forall j \in N, \tag{2.11}$$

$$\sum_{i:(i,j)\in E} \sin\theta_{ij}(t_{ij}^f)x_{ij} - \sum_{k:(j,k)\in E} \sin\theta_{jk}(t_{ij}^0)x_{jk} = 0, \quad \forall j \in N. \tag{2.12}$$

2.2.2 Computation of lower bounds

To compute a lower bound, the idea is to relax the constraints (2.11) and (2.12) and penalize the objective function (2.1) whenever the constraints are violated via a set of penalty (dual) variables. Let the penalty variable corresponding to the angle constraint of target j in (2.11) is α_j and the penalty variable corresponding to the angle constraint of target j in (2.12) is β_j. Let $\Pi = [\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n]$, where α_j, $\beta_j \in \Re\ \forall j = 1...n$. For any given set of penalty variables Π, a solution to the following

Lagrangian relaxation of (2.1 - 2.2) is lower bound to the DTSP.

$$L(\Pi) = \min_{\theta, X} \sum_{(i,j)\in E} d_{ij}(\theta_i, \theta_j) x_{ij} \tag{2.13}$$
$$- \sum_{j\in N} \alpha_j \left[\sum_{i:(i,j)\in E} \cos\theta_{ij}(t^f_{ij}) x_{ij} - \sum_{k:(j,k)\in E} \cos\theta_{jk}(t^0_{ij}) x_{jk} \right]$$
$$- \sum_{j\in N} \beta_j \left[\sum_{i:(i,j)\in E} \sin\theta_{ij}(t^f_{ij}) x_{ij} - \sum_{k:(j,k)\in E} \sin\theta_{jk}(t^0_{ij}) x_{jk} \right],$$

subject to:

$$X \in \mathcal{F}.$$

By replacing $\cos\theta_{ij}(t^f_{ij})$ and $\cos\theta_{jk}(t^0_{ij})$ with $\cos\theta_j$, and replacing $\sin\theta_{ij}(t^f_{ij})$ and $\sin\theta_{jk}(t^0_{ij})$ with $\sin\theta_j$ and simplifying the objective in (2.13), $L(\Pi)$ can be written as

$$L(\Pi) = \min_{\theta, X} \sum_{(i,j)\in E} [d_{ij}(\theta_i, \theta_j) - \alpha_j \cos\theta_j - \beta_j \sin\theta_j \tag{2.14}$$
$$+ \alpha_i \cos\theta_i + \beta_i \sin\theta_i] x_{ij}.$$

Now, for any set of penalty variables, note that $L(\Pi) \geq J(\Pi)$ where

$$J(\Pi) = \min_{X} \sum_{(i,j)\in E} \min_{\theta} [d_{ij}(\theta_i, \theta_j) - \alpha_j \cos\theta_j - \beta_j \sin\theta_j$$
$$+ \alpha_i \cos\theta_i + \beta_i \sin\theta_i] x_{ij}. \tag{2.15}$$

Theorem 1. *For any given* Π*, the solution to the minimization problem with objective* $J(\Pi)$ *shown in equation (2.15), subject to the constraints in (2.2) is a lower bound to the DTSP (2.1-2.2, 2.11-2.12).*

Proof. Clearly $L(\Pi)$ in (2.13) is the Lagrangian relaxation of the DTSP defined by the

objective in (2.1) and the constraints in (2.2,2.11,2.12). The weak duality theorem states that for a minimization problem, the cost of the Lagrangian relaxation for any set of penalty variables is at most equal to the optimal cost of the DTSP. Therefore, $L(\Pi)$ is a lower bound to the DTSP. One can also note that that $J(\Pi)$ is at most equal to $L(\Pi)$. Therefore, for any given Π, the solution to (2.15) is a lower bound to the DTSP. □

Consider the following variational problem

$$\begin{aligned}\nu_{ij}(\alpha_i, \alpha_j, \beta_i, \beta_j) = &\min_{\theta_i, \theta_j} d_{ij}(\theta_i, \theta_j) - \alpha_j \cos\theta_j \\ &- \beta_j \sin\theta_j + \alpha_i \cos\theta_i + \beta_i \sin\theta_i,\end{aligned} \tag{2.16}$$

where $d_{ij}(\theta_i, \theta_j)$ is given by equations (2.3) to (2.6) and (2.9) to (2.10). $d_{ij}(\theta_i, \theta_j)$ is the minimum Dubins distance required by the vehicle to travel from the configuration (x_i, y_i, θ_i) to (x_j, y_j, θ_j) and can be calculated using the result from Dubins[9]. Given the values of $\alpha_i, \alpha_j, \beta_i, \beta_j$, one can compute ν_{ij} as follows: Discretize the allowable values of the heading angle and obtain a discrete set of heading angles $\Phi = \{\phi_1, \phi_2, ...\phi_d\}$. We assume that this discrete set of heading angles is the same for all the targets, *i.e.* $\Phi_i = \Phi$, $\forall i \in N$. Therefore, for every pair of $\theta_i, \theta_j \in \Phi$, the value of $d_{ij}(\theta_i, \theta_j) - \alpha_j \cos\theta_j - \beta_j \sin\theta_j + \alpha_i \cos\theta_i + \beta_i \sin\theta_i$ can be easily computed. The minimum of all these values, each corresponding to a pair of $\theta_i, \theta_j \in \Phi$ is ν_{ij}.

Now, given $\Pi = (\alpha_1, ...\alpha_n, \beta_1, ...\beta_n)$, ν_{ij} can be computed for every edge $(i, j) \in E$. The minimization problem in (2.15) then reduces to the following:

$$J(\Pi) = \min_{X \in \mathcal{F}} \sum_{(i,j) \in E} \nu_{ij}(\alpha_i, \alpha_j, \beta_i, \beta_j) x_{ij}, \tag{2.17}$$

This is an asymmetric Traveling Salesmen Problem (ATSP) where the weight of each

edge (i, j) is ν_{ij}. This ATSP can be solved using the LKH heuristic and it gives a tight lower bound for the ATSP, which is in turn a lower bound to the DTSP.

In summary, the following is the algorithm for finding a lower bound, given a set of penalizing variables (Π):

1. Select the penalty variables $\Pi^k = (\alpha_1, ..\alpha_n, \beta_1, ..\beta_n)$.
2. Use Π^k to formulate the Lagrangian relaxation and the variational problem (2.16).
3. Solve the variational problem (2.16) for every $(i, j) \in E$.
4. Find a lower bound by solving the ATSP using the LKH heuristic.

For any given Π, $J(\Pi)$ is a lower bound to the DTSP, the best lower bound can be found by maximizing $J(\Pi)$ for all Π, *i.e.*, by solving $J^* = \max_\Pi J(\Pi)$. Since $J(\Pi)$ is a combination of a finite number of linear functions, it is concave in Π. Therefore, we use a subgradient optimization technique to solve for J^*. One can refer to [35] and [36] to understand the details of subgradient optimization.

2.2.2.1 Subgradient procedure

This procedure is explained in the context of a general optimization problem. Consider the problem: $\min_x c'x$ subject to $Ax = b$ where x is a $n \times 1$ vector, b is a $m \times 1$ vector, c is a $n \times 1$ vector and A is a $m \times n$ matrix. A Lagrangian relaxation of this problem is $\min_x c'x + \lambda'(Ax - b)$ where $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ is a vector of penalty variables. The best lower bound can then be obtained by solving the following Lagrangian dual problem: $\max_\lambda \min_x c'x + \lambda'(Ax - b)$. A subgradient procedure for solving this dual problem starts with an initial set of penalty variables (λ^0) and iteratively attempts to update these penalty variables with a goal of improving the

lower bound. After each iteration k, a new penalty vector λ^{k+1} is computed by taking a step along the direction of the subgradient:

$$\lambda^{k+1} = \lambda^k + \delta^k s^k, \tag{2.18}$$

where s^k is a vector (of dimension $m \times 1$) representing the subgradent direction and δ^k is the step size along the subgradient. In general, the subgradient direction s^k is chosen to be $b - Ax^k$ [36] where x^k is the solution of the Lagrangian dual problem for the k^{th} iteration. One can also select a constant step size or a diminishing step size or a Polyak's step size for updating the penalty variables. The convexity of the Lagrangian function then guarantees the convergence of the above subgradient procedure.

2.2.2.2 Implementation details for the DTSP

An important part of the subgradient optimization technique is to find a direction of the subgradient and step size at each iteration k. With respect to the DTSP, as constraints (2.11) and (2.12) are relaxed, one can chose the following as the subgradient:

$$s^k = \begin{bmatrix} \sum_{i:(i,j)\in E} \cos\theta_{ij}^k (t_{ij}^f) x_{ij}^k - \sum_{k:(j,k)\in E} \cos\theta_{jk}^k (t_{ij}^0) x_{jk}^k \\ \sum_{i:(i,j)\in E} \sin\theta_{ij}^k (t_{ij}^f) x_{ij}^k - \sum_{k:(j,k)\in E} \sin\theta_{jk}^k (t_{ij}^0) x_{jk}^k \end{bmatrix}. \tag{2.19}$$

Here, x_{ij}^k and θ_{ij}^k are the solutions for the problems defined in (2.17) and (2.16) respectively in the k^{th} iteration. For the computational experiments, we chose the step size (δ^k) to be 1 initially and reduced it by a factor of 2 after every 50 iterations. This iterative procedure is outlined as below:

1. Initialize $k := 0$. Choose the initial penalty vector Π^0 with all the penalty variables to be equal to zero. Let ϵ be a small number and k_{max} be the maximum number of iterations allowed for the subgradient procedure.

2. Compute ν_{ij} as shown in (2.16), for all $(i, j) \in E$.

3. Solve the ATSP using the LKH heuristic. Set the lower bound, $J(\Pi^k)$, during the k^{th} iteration to be equal to the lower bound computed using the LKH heuristic.

4. If $k > 1$ and $\frac{J(\Pi^{k+1}) - J(\Pi^k)}{J(\Pi^k)} \leq \epsilon$ stop the iterative procedure and output the best lower bound.

5. Compute $\Pi^{k+1} = \Pi^k + \delta^k s^k$, where s^k is given by the equation (2.19). If $k \geq k_{max}$, stop the iterative procedure and output the best lower bound; else, set $k = k + 1$ and return to step 2.

2.2.3 One in a set transformation

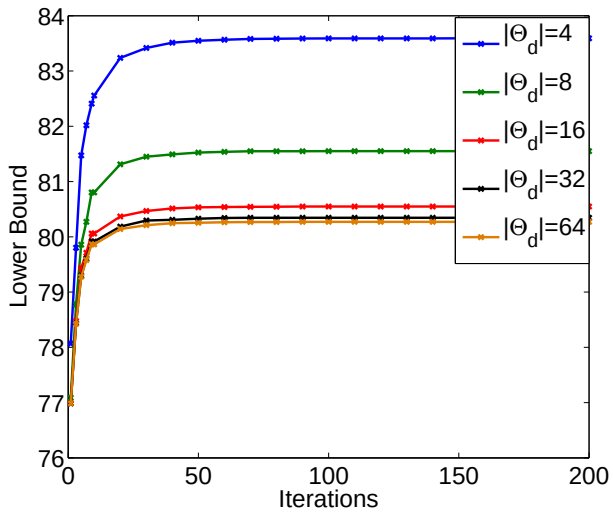
To corroborate the performance of the proposed technique, we also solve the DTSP using the method in Oberlin et. al [25]. In [25], the authors solve the discrete DTSP (where the choice of the heading angle at each target is restricted to a discrete set) by transforming the DDTSP into an ATSP. They replicate each target m times such that each of the m replications correspond to a possible heading angle. Now, the DTSP is posed as a problem of finding a subtour for each vehicle such that exactly one copy of each target is visited once and the total distance traveled by the vehicle is a minimum. This problem is then transformed into an ATSP using the method presented by Noon and Bean in [37]. One can solve this ATSP using the LKH heuristic. This algorithm readily gives a feasible solution to the DTSP, which is an upper bound to the optimum. Also it provides a lower bound which can be

used to compare with the lower bounds obtained using the method proposed in this article.

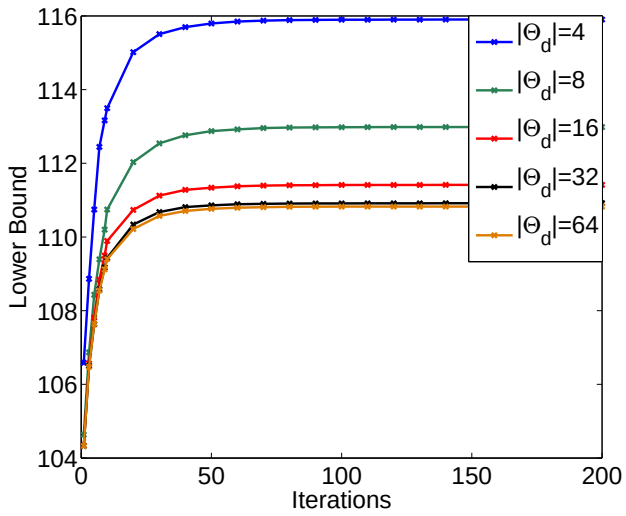
2.2.4 Numerical results

For two instances with 20 targets and 40 targets, targets are randomly located, lower bounds to the DTSP were computed using the Lagrangian relaxation method. The plots of the lower bound versus the number of iterations of the sub-gradient procedure for those two instances are shown in Figure 2.1(a) and Figure 2.1(b) respectively. Each of these figures also show the plots for different discretizations(Θ_d) of the heading angle. As the size of the set Θ_d is increased, the lower bound reduces. This is due to the fact that the solution of the variational problem (2.16) gives a better minimum with more discretizations of the heading angle θ_i. One can also infer from Figure 2.1(a) and Figure 2.1(b) that the final value of lower bound also converges in terms of the size of the set Θ_d. The average of the percentage of improvement compared with the lower bound after 20 iterations with 32 discretizations is plotted in Figure 2.2. Clearly, after 20 iterations, the improvement of the lower bound is minimal even after 200 iterations of sub-gradient procedure. Figure 2.3 shows the average computation time required for all the algorithms as a function of the size of the problem. Even though the computation time significantly increases as the number of iterations increases, the improvement in the lower bound after 20 iterations is minimal. Therefore, one can stop the iterative process after 20 iterations to find a tight lower bound within a reasonable amount of time.

Tables 2.1 and 2.2 compares the lower bounds computed using the proposed method with the optimal cost of the Euclidean TSP (ETSP) and the lower bound obtained using the Transformation Method (TM) with the minimum turning radius $\rho = 4$ and $\rho = 6$. We have computed the lower bounds for two instances of each



(a) Instance with 20 targets



(b) Instance with 40 targets

Figure 2.1: Convergence of sub-gradient procedure

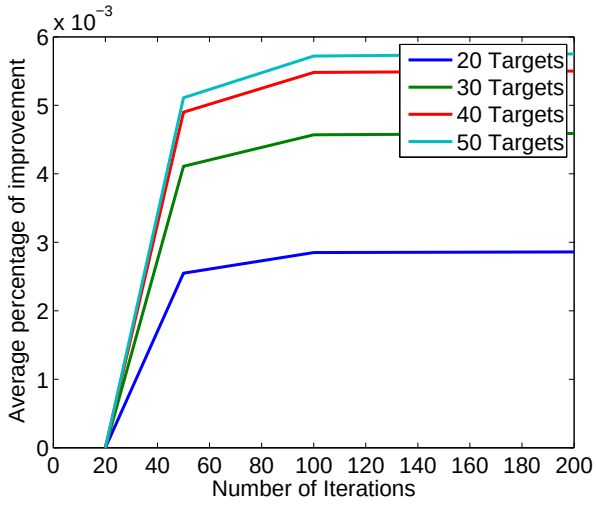


Figure 2.2: Average of the percentage improvement compared to 20 iterations

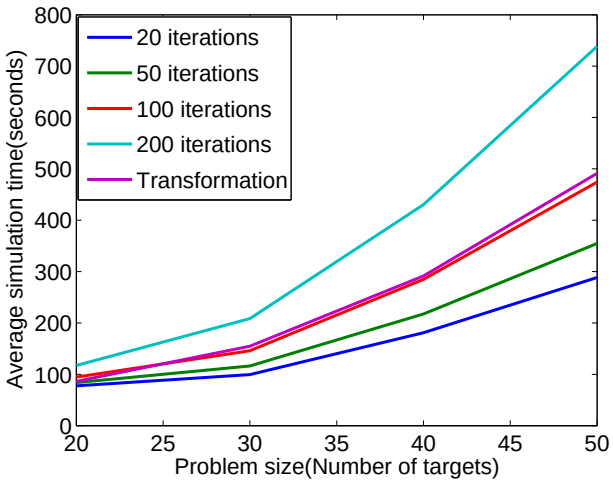


Figure 2.3: Comparison of average simulation time using transformation method and different iteration of proposed algorithm

case with 10, 20, 30, 40 and 50 targets. In these simulations, we have run the sub-gradient procedure for 200 iterations. The ETSP corresponding to each instance was solved using the LKH Heuristic. The LKH algorithm gives a lower bound to the ETSP which is in turn a lower bound to the corresponding DTSP. The first column in the tables indicates the size (number of targets) of an instance. The second column refers to the optimal cost of the Euclidean TSP (ETSP) which is a lower bound to the DTSP. Third and fourth columns refers to the lower bounds calculated using the proposed Lagrangian method and the TM respectively. Fifth and sixth columns refers to the improvement (in %) in the lower bounds calculated using the Lagrangian method and the TM with respect to the optimal cost of the ETSP.

The average improvement in the lower bounds computed using the proposed Lagrangian dual algorithm is 31.5% with $\rho = 4$ and 48.5% with $\rho = 6$. As mentioned in the introduction, there were several instances for which the bounds computed by using the TM were either negative or were inferior compared to the solution of the ETSP. These instances are marked by '*' in the tables. When the transformation method worked, it produced bounds on an average that were better than the bounds found using the proposed method. In summary, choosing the maximum of the bounds found by the proposed method and the transformation method improved the lower bound available for the DTSP significantly.

For an instance with 10 targets, a solution using the Lagrangian relaxation and the heuristic is shown in Figure 2.4. The path shown in blue is the solution from Lagrangian relaxation and the path shown in red is a feasible solution obtained from a heuristic algorithm. As the heading angle constraint is relaxed in the Lagrangian relaxation, the path may not be a feasible Dubins tour. One can see that the incoming and outgoing headings are not same at some targets in Figure 2.4.

Table 2.1: Lower bound comparison with $\rho = 4$

Inst.[a]	**#Targ.**[b]	**ETSP**[c]	**Lag.**[d]	**TM**[e]	**%Diff.**[f]	**%Diff.**[g]
1	10	63.25	73.21	83.26	15.76	31.65
2	10	54.70	60.83	80.53	11.20	47.22
3	20	74.97	93.10	102.70	24.18	36.99
4	20	74.38	91.00	103.55	22.34	39.21
5	30	78.39	115.44	119.73	47.26	52.73
6	30	84.76	111.23	158.21	31.23	86.65
7	40	97.36	135.63	110.97	39.31	13.98
8	40	99.62	142.14	-59.40	42.69	*
9	50	115.26	157.95	-1.60	37.04	*
10	50	113.74	163.84	6.08	44.05	*
Mean					**31.51**	**44.06**

Table 2.2: Lower bound comparison with $\rho = 6$

Inst.[a]	**#Targ.**[b]	**ETSP**[c]	**Lag.**[d]	**TM**[e]	**%Diff.**[f]	**%Diff.**[g]
1	10	58.90	84.30	98.83	43.12	67.78
2	10	66.77	82.77	*	23.96	*
3	20	72.83	115.20	*	58.18	*
4	20	68.75	110.14	*	60.19	*
5	30	93.91	135.60	*	44.40	*
6	30	97.23	151.27	48.18	55.59	*
7	40	99.73	150.40	144.30	50.80	44.68
8	40	97.61	147.37	119.39	50.98	22.32
9	50	110.78	169.49	*	52.99	*
10	50	109.34	158.51	266.53	44.97	143.76
Mean					**48.52**	**69.64**

[a]Instance
[b]Number of targets
[c]Euclidean TSP
[d]Lagrangian algorithm
[e]Transformation method
[f]% Difference using Lagrangian algorithm
[g]% Difference using Transformation method

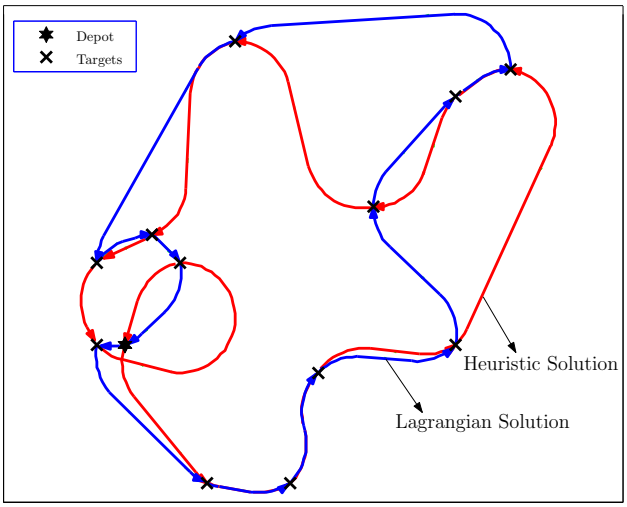


Figure 2.4: Lagrangian solution vs heuristic solution 10 targets

2.3 Extension of the lower bound computation to the multiple depot multiple vehicle DTSP

The DTSP can be extended to a multiple vehicle case and the problem is defined below: given a set of targets and a set of vehicles starting from distinct depots, the objective is to find a path for each of the vehicles such that (i) every target is visited at least once by a vehicle, (ii) the paths satisfy the motion constraints of the vehicles and (iii) the sum of the length of the paths is a minimum. The vehicles here are modeled as Dubins vehicle, and this path planning problem is referred as the Multi-Depot Multiple Vehicle Dubins Traveling Salesman Problem (**MDMVDTSP**). Let $T = \{1, 2, ..., n\}$ be the set of given targets and $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ be the set of headings at the targets. Let $D = \{n+1, n+2, ..., n+m\}$ denote the initial locations (or depots) of the vehicles. Let $N = T \cup D$ and E denote the set of all the edges joining any two vertices in N. Let x_{ij} be a binary decision variable which equals 1 if there is an edge from i to j in the tour and equals 0 otherwise. Let X be the

matrix of decision variables, whose entry in the i^{th} row and j^{th} column is x_{ij}. Let $\mathcal{F}$ represent the set of all feasible solutions, such that in each solution, every target in T is visited atleast once by one of the vehicles in D. We will say that $X \in \mathcal{F}$, if the matrix corresponds to one of the feasible solutions. The MDMVDTSP can be stated as the following:

$$\min_{\theta, X} \sum_{(i,j)\in E} d_{ij}(\theta_i, \theta_j) x_{ij} \tag{2.20}$$

subject to:

$$X \in \mathcal{F}, \tag{2.21}$$

where $d_{ij}(\theta_i, \theta_j)$ is given by the equations (2.3 - 2.7). The Lagrangian relaxation for MDMVDTSP follows similar to the result in section 2.2.2 and it is shown in equation (2.22).

$$J(\Pi) = \min_{X\in\mathcal{F}} \sum_{(i,j)\in E} \nu_{ij}(\alpha_i, \alpha_j, \beta_i, \beta_j) x_{ij}. \tag{2.22}$$

This is an asymmetric Multiple Depot Multiple Traveling Salesman Problem (MDMTSP). This could be transformed to an asymmetric TSP (ATSP) using the result in [19] and the resulting ATSP could be solved using LKH heuristic. The LKH heuristic gives a tight lower bound for the ATSP, which in turn is a lower bound to the MDMVDTPS.

2.3.1 Numerical results

The lower bound for the MDMVDTSP are computed using the proposed Lagrangian method and one in a set transformation method for 20 instances each with 10, 20, 30 and 40 targets with 2, 3 and 4 vehicles. The minimum turning radius (ρ)

for each of these vehicles is 4. The percentage of improvement in the lower bound compared to the solution of the corresponding EMTSP is computed. The solution of the EMTSP is a lower bound to the MVDTSP. The average of the percentage improvement of the lower bound for 20 instances of each of the case is listed in Table 2.3. We can see a significant improvement of 30 to 40 percent in the lower bounds computed using the proposed method. The proposed method performs better than the transformation method in all of the instances with multiple vehicles. In most of the instances, the transformation method produced either negative value for the lower bound or a value less than the solution of EMTSP and hence the average improvement is negative in Table 2.3. That is because this method relies on modifying the cost of traveling for some of the edges by a large constant(generally called big-M). Hence the lower bound depends on the value chosen for M.

The average computation time (for 20 instances) required to compute the lower bounds using the EMTSP, the proposed method and the transformation method are shown in Table 2.4. We can clearly see that the computation time required by the proposed method is significantly less compared to the transformation method. The transformation method transforms the MVDTSP into an asymmetric TSP with a very large number of nodes and hence the high computation time. The EMTSP is transformed into a symmetric TSP and solved using LKH heuristic, which is the fastest known heuristic to solve TSP. Since this involves only solving a TSP, the computation time required for EMTSP is very less compared to the other two methods.

A dual solution and a heuristic solution are shown for an instance with single depot in Figure 2.5 and for two instances with multiple depots in Figures 2.6 and 2.7. The paths shown in blue are the dual solutions and the paths shown in the red are the feasible heuristic solutions. The dual solution is the solution of the problem in equation (2.22). However, this may not be a feasible path, since the incoming

angle and outgoing angle at some of the targets may not be the same. One can see the discontinuity of the dual path at few targets in Figure 2.5, 2.6 and 2.7. In case of multiple vehicles, two or more vehicles are used to visit the targets, only if the targets are separated into more than one cluster and a depot is present in each cluster as in Figure 2.7. If the targets are not separated as clusters, in most of the instances, it is cheaper to pick one vehicle to visit all the targets. For the instances where the targets are separated into clusters, the distance needed to travel from one cluster to another would dominate and hence choosing two vehicles in each cluster to visit the targets in their clusters could be cheaper.

Table 2.3: Average of the percentage lower bound improvement compared to the solution of EMTSP

# Targets	**% Improvement of the Lowerbound**					
	2 Depots		**3 Depots**		**4 Depots**	
	TM[a]	Lag[b]	TM[a]	Lag[b]	TM[a]	Lag[b]
10	-57.8	27.3	-105.9	34.1	-70.3	38.6
20	-25.5	32.3	-48.9	35.1	-72.6	37.3
30	-65.6	32.4	-74.7	34.1	-133.6	35.4
40	-92.6	38.1	-136.9	38.6	-161.3	39.1

[a]Transformation method
[b]Proposed Lagrangian method

2.4 Lower bound computation using convexity

Let $p_1, ...p_n$ be a sequence of targets, p_i is called a sharp turn if the angle $\angle(p_{i-1}, p_i, p_{i+1})$ is acute and one of the p_i's neighbors is within a distance 4 units from the segment joining p_i to its other neighbor. Let $\theta_1, \theta_2....\theta_n$ be the headings at each target, and let $\mathcal{C}$ is a mapping of the angles vector $(\theta_1, \theta_2....\theta_n)$ to the length

Table 2.4: Average of the computation time required using three different methods

# Targets	Computation time (seconds)		
	EMTSP[a]	**TM**[b]	**Lag**[c]
10	0.07	277.81	113.43
20	0.17	1365.07	357.49
30	0.32	3960.58	745.62
40	0.66	7379.01	1255.56

[a]Euclidean MTSP
[b]Transformation method
[c]Proposed Lagrangian method

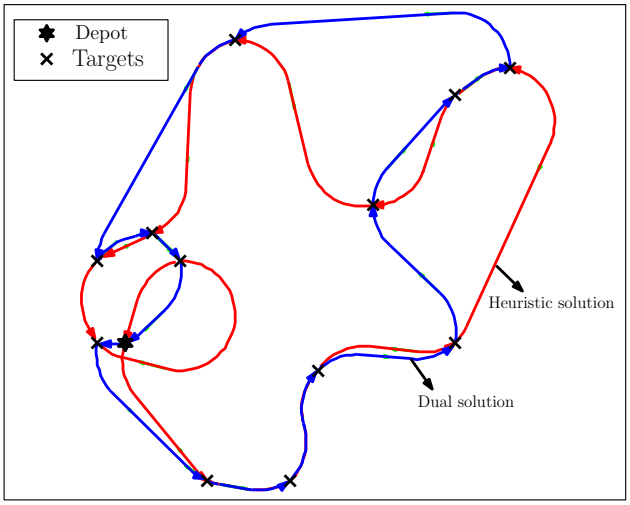


Figure 2.5: Dual solution vs heuristic solution with 12 targets

of the shortest Dubins path, with a minimum turning radius of 1 unit, visiting the targets in the given order. The result in [20] is the following: when there are no sharp turns, all global minima of $\mathcal{C}$ is strictly convex over its lifted domain in $\Re^n$. The shortest distance (d_{ij}) between two targets i and j is a function of the headings

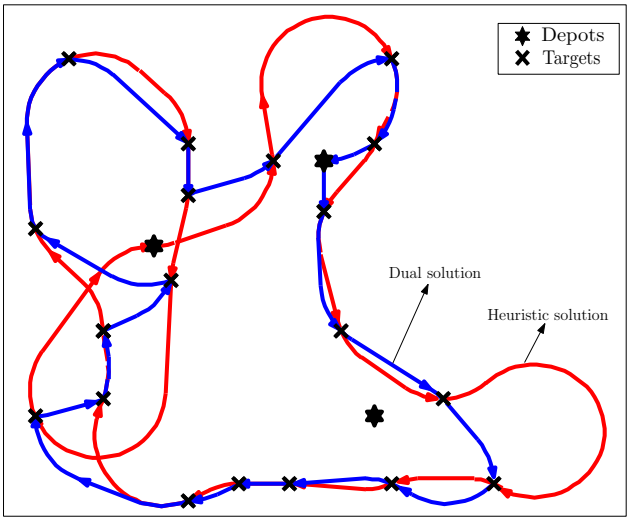


Figure 2.6: Dual solution vs heuristic solution with 20 targets

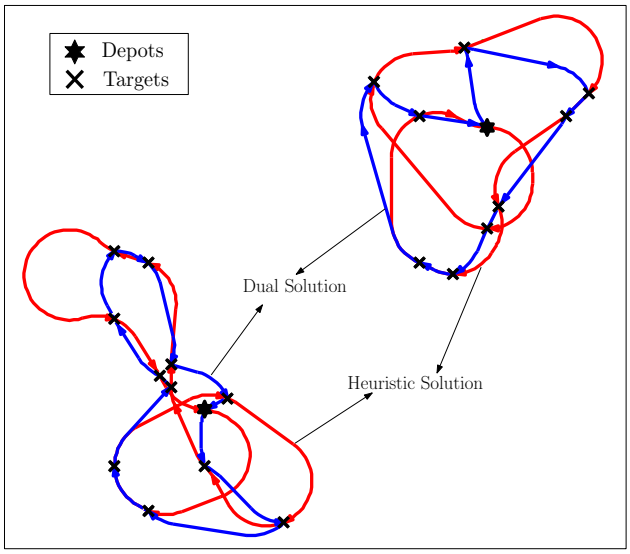


Figure 2.7: Dual solution vs heuristic solution with 20 targets

at the targets (θ_i, θ_j). The partial derivatives of d_{ij} were given in [20] as:

$$\frac{\partial d_{ij}(\theta_i, \theta_j)}{\partial \theta_i} = \pm(1 - cos\alpha_i), \quad \frac{\partial d_{ij}(\theta_i, \theta_j)}{\partial \theta_j} = \pm(1 - cos\alpha_j), \tag{2.23}$$

where α_i and α_j are the length of the circular arcs in the vehicle's path at target i and j respectively. We attempt to compute a lower bound to the single vehicle DTSP using the convexity of $\mathcal{C}$, when the distance between every pair of targets is atleast 4 units.

Consider an instance of DTSP where distance between every pair of vertices is greater than 4 units. To find the optimal solution, one has to find the optimal sequence of targets to be visited and the optimal headings at each target. Let T^* be the optimal sequence of targets, $\theta^*(T^*)$ be the optimal headings at each target and $C(\theta^*(T^*))$ be the cost of the optimal solution. Since this is hard to solve, one can restrict the values of allowable headings at each target to a discrete set Φ_d and pose the DTSP as a one-in-a-set TSP. In general, the optimal sequence and headings of the one-in-a-set TSP can be different from the optimal solution of the DTSP. Let T^1, $\theta^1(T^1)$ be the optimal sequence of targets and headings and $C(\theta^1(T^1))$ be the optimal cost for the corresponding one-in-a-set TSP.

Let $\Phi_d = \{\phi^1, \phi^2, ...\phi^d\}$ be the set of discrete values of headings allowed at each target, where $\phi^{i+1} - \phi^i = \delta$, $i = 1, 2, .., d-1$. Let Θ be the set of vectors of size n, defined as $\Theta = \{\theta : \theta_i \in \Phi_d,\ i = 1, 2, .., n\}$.

Theorem 2. *$C(\theta^1(T^1)) - 4n\delta$ is a lower bound to the optimal solution of the DTSP, $C(\theta^*(T^*))$.*

Proof. For a given sequence T of targets, there are two cases. In the first case, the headings at each target are restricted to a discrete set Φ_d and in the second case, there are no restrictions on the headings. We use two different notations for the optimal solutions of these two cases. Let $\hat{\theta}(T)$ be the optimal vector of headings corresponding to the case where headings at each target are restricted and $\theta^0(T)$ be the vector of optimal headings for the unrestricted case.

Similarly for the optimal sequence T^* of targets , let $\theta^2(T^*)$ be the optimal vector of headings corresponding to the restricted (or discrete) case and $\theta^*(T^*)$ be the vector of optimal headings for the unrestricted case.

For the given sequence T of targets, $C(\theta(T))$ is convex in $\theta(T)$ and therefore

$$\begin{aligned} C(\theta^0(T)) &\geq C(\hat{\theta}(T)) + \nabla_\theta C(\hat{\theta}(T)) \cdot (\theta^0(T) - \hat{\theta}(T)) \\ &\geq C(\hat{\theta}(T)) + \\ &\sum_{(i,j):x_{ij}\in T} \left[\frac{\partial d_{ij}(\theta_i,\theta_j)}{\partial \theta_i}(\theta_i^0 - \hat{\theta}_i) + \frac{\partial d_{ij}(\theta_i,\theta_j)}{\partial \theta_j}(\theta_j^0 - \hat{\theta}_j) \right] x_{ij} \end{aligned} \tag{2.24}$$

From equations (2.23), one can see that the maximum value of $\frac{\partial d_{ij}}{\partial \theta_i}$ and $\frac{\partial d_{ij}}{\partial \theta_j}$ is 2 and the maximum value of $(\theta_i^0 - \hat{\theta}_i)$ is δ. The inequality in (2.24) becomes

$$\begin{aligned} C(\theta^0(T)) &\geq C(\hat{\theta}(T)) - \sum_{(i,j):x_{ij}\in T} (2\delta + 2\delta) x_{ij} \\ C(\theta^0(T)) &\geq C(\hat{\theta}(T)) - 4n\delta \end{aligned} \tag{2.25}$$

For the sequence T^*, we can write the inequality similar to (2.25) as:

$$C(\theta^*(T^*)) \geq C(\theta^2(T^*)) - 4n\delta. \tag{2.26}$$

Here the elements of the vector $\theta^2(T^*)$ belongs to the discrete set Φ_d and therefore $(T^*, \theta^2(T^*))$ is a feasible solution to the one-in-a-set TSP. Since $(T^1, \theta^1(T^1))$ is the optimal solution to the one-in-a-set TSP, we can write

$$C(\theta^1(T^1)) \leq C(\theta^2(T^*)). \tag{2.27}$$

From inequalities (2.26) and (2.27)

$$C(\theta^1(T^1)) - 4n\delta \leq C(\theta^*(T^*)). \tag{2.28}$$

□

Theorem 2 holds for the multiple vehicle case too and the proof follows along similar lines to the proof for the single vehicle case. The lower bounds computed using this method are compared with the bounds from the Lagrangian relaxation method and transformation method in Table 2.5 for the single vehicle DTSP and the MDMVDTSP. This method produced better bounds than the other two methods in 8 out of 10 instances of the single vehicle case and all the instances of the multiple vehicles case.

2.5 Conclusions

1. We have considered a routing problem for vehicles with motion constraints, which are modeled as Dubins vehicle. The motion constraints require that the yaw rate of the vehicles is upper bounded by a constant anywhere along the path.

2. This problem is posed as a minimization problem similar to a Traveling Salesman Problem, where the cost to travel between a pair of targets depends on the initial and final headings of the vehicle.

3. We relax the motion constraints of the vehicle at target locations and penalize the objective. This reduces the problem to a regular asymmetric TSP, the solution of which is a lower bound to the DTSP.

4. The ATSP is solved using LKH heuristic, which also gives a lower bound, and

this in turn is a lower bound to the DTSP.

5. The best lower bounds possible using this approach are computed by choosing appropriate penalty variables, which are computed using an iterative subgradient algorithm.

6. This method to compute the lower bounds using the Lagrangian relaxation is extended to the multiple vehicle case (MDMVDTSP). Here, after relaxing the motion constraints, the problem reduces to a Multiple Traveling Salesman Problem, which is solved by transforming into a single TSP.

7. For the instances which satisfy the 4 units distance criteria, we have developed a method to compute the lower bounds using the convexity property of the Dubins paths.

8. The lower bounds computed using the proposed algorithms are compared with the lower bound computed using the transformation method in [25] and the optimal solution of corresponding Euclidean TSP. The proposed methods produced bounds significantly better than the bounds computed using the other two methods.

Table 2.5: Comparing the lower Bounds computed using convexity, Lagrangian and transformation methods

Lower bounds for the single vehicle DTSP				
# Targets	**ETSP**[a]	**Lag**[b]	**OST**[c]	**Conv**[d]
10	106.67	108.19	-380.14	107.94
10	116.48	118.43	-177.20	116.88
20	163.63	167.57	13.49	169.64
20	175.64	177.61	-1512.35	182.80
30	211.10	212.94	-445.53	222.69
30	197.61	200.87	-6.73	202.49
40	237.20	240.97	83.49	246.78
40	236.89	240.64	-3134.59	242.70
50	271.71	275.83	-315.26	289.11
50	272.24	276.53	63.58	283.95
Lower bounds for the MDMVDTSP with 3 vehicles				
10	100.58	103.01	-81.11	125.78
10	125.17	125.30	-1290.98	134.91
20	167.93	169.74	29.45	180.42
20	165.44	168.31	-3271.64	169.32
30	210.28	214.04	-3137.66	245.38
30	213.71	216.04	-3859.22	242.82
40	240.79	245.95	14.68	257.80
40	239.24	242.99	24.64	262.75

[a]Optimal solution of Euclidean TSP
[b]Lower bound using proposed Lagrangian algorithm
[c]Lower bound using transformation method
[d]Lower bound computed using convexity

3. UAV ROUTING IN GPS DENIED ENVIRONMENTS*

3.1 Two UAV CCURP

3.1.1 Introduction

This section deals with the GPS-denied UAV routing; here we consider two different architectures for the navigation of UAVs aided by the unattended ground sensors (UGSs). In the first architecture, we assume the UGS are placed uniformly across the restricted zone. Figure 3.1 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 the four corners of each square. We assume that the distance between two UGS along the edges of each square is less than the 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; an illustration of this is shown in Figure 3.2. The UAVs have controllers on board to orbit around another UAV, which is done by estimating the distance between them using the range sensors. This type of controllers were developed in [38, 39, 40, 41] for the UAVs to orbit around moving or stationary targets. The UAVs navigate through the UGS network by leap-frogging from UGS to UGS.

A set of targets, which is a subset of UGS, are located at critical locations of the restricted zone. The UAVs must visit the targets periodically 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

*Part of this section was reprinted with permission from S. Manyam, S. Rathinam, S. Darbha, D. Casbeer, and P. Chandler, "Routing of two Unmanned Aerial Vehicles with communication constraints," in *Proc. IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 140-148. Copyright © 2014 IEEE.

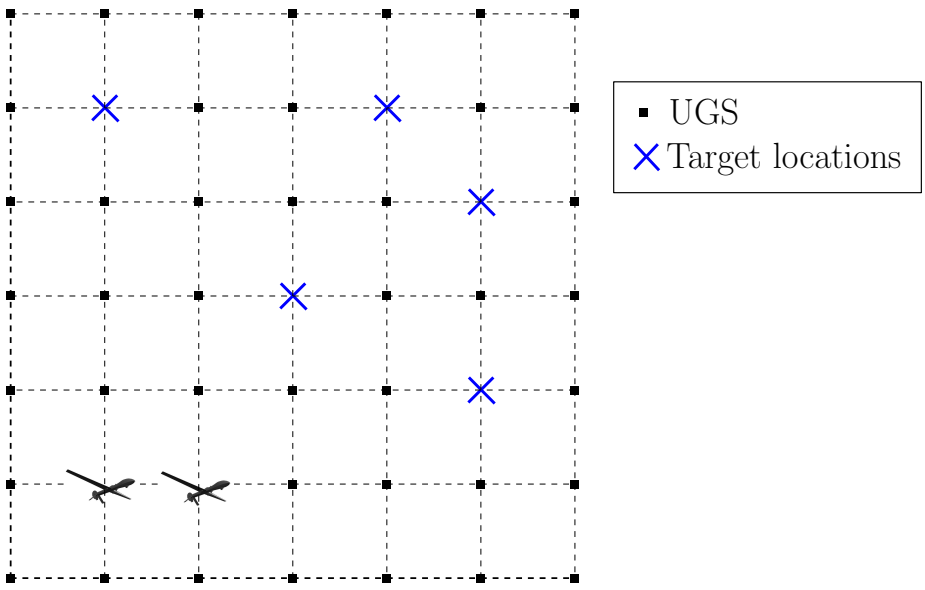


Figure 3.1: Field with UGS deployed

targets. The objective of the routing problem is to find an optimal *cyclical* trajectory for each UAV such that

(i) each target is visited by one of the UAVs,

(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,

(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 traveled or the time spent in visiting all the targets is minimized. We refer to this problem as the Com-

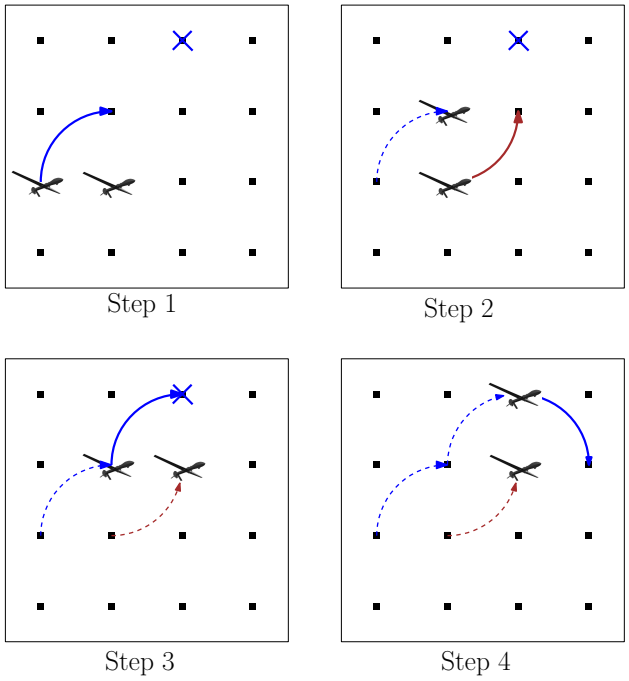


Figure 3.2: Navigation of the UAVs

munication Constrained UAV Routing Problem (CCURP). A feasible solution to an instance of the CCURP is shown in Figure 3.3.

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 Figure 3.1, four edges are incident on every vertex. The length of each edge is a constant and is denoted by 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 Figure 3.1. 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

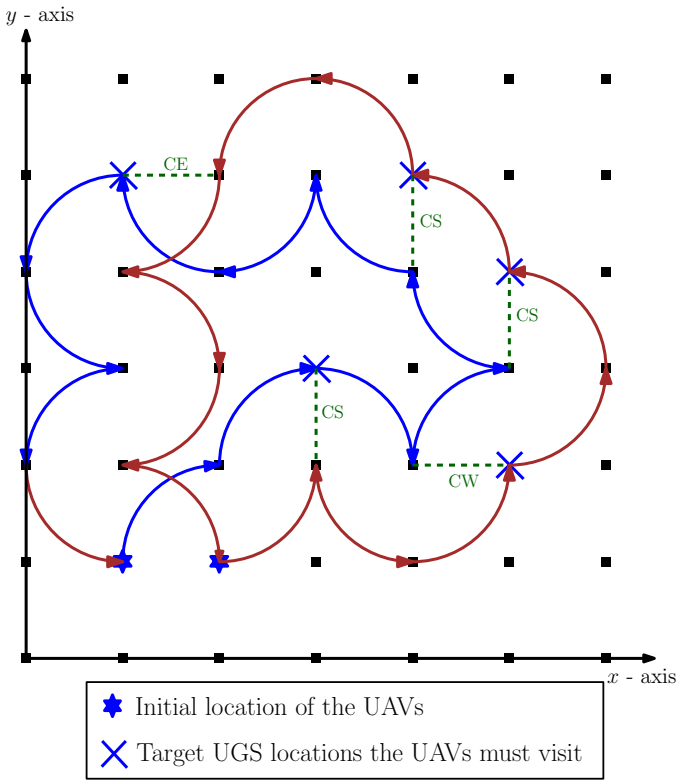
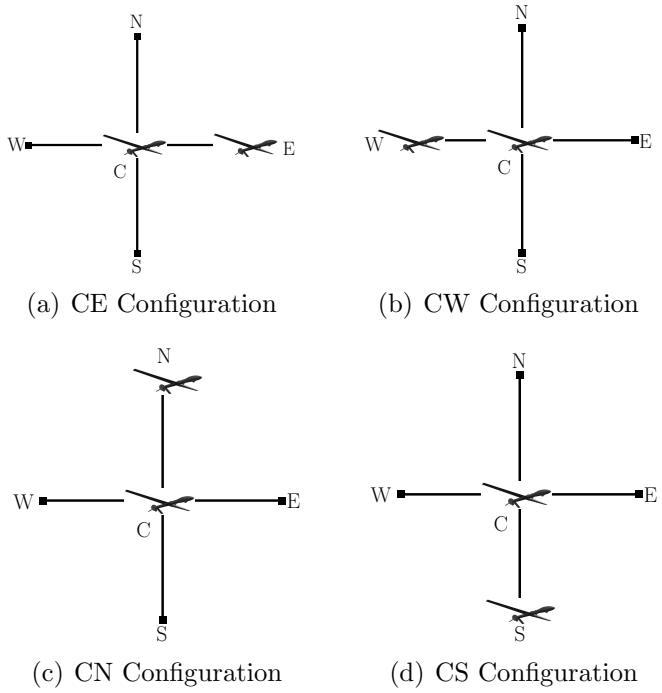


Figure 3.3: A tour of the UAVs

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 Figure 3.4. 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 Figure 3.5. 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.



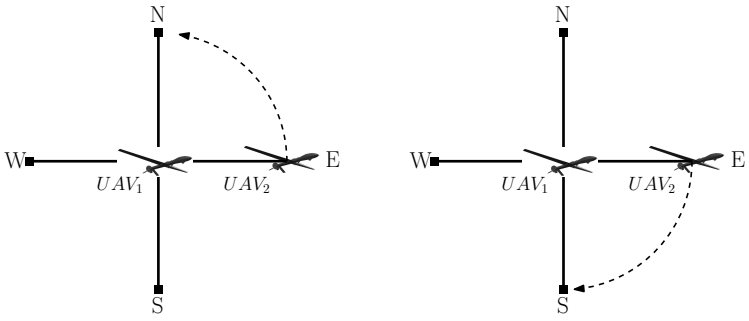
(a) CE Configuration (b) CW Configuration

(c) CN Configuration (d) CS Configuration

Figure 3.4: Four different configurations available for visiting a target at C

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: $(\zeta_i + R, \eta_i), (\zeta_i, \eta_i + R), (\zeta_i - R, \eta_i), (\zeta_i, \eta_i - R)$.

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 θ_i to θ_j. $d_{min}(\theta_i, \theta_j)$ can be computed by using a shortest path algorithm in the following way: Let $G_s = (V_s, E_s)$



(a) UAV_2 rotating in the counter-clockwise direction

(b) UAV_2 rotating in the clockwise direction

Figure 3.5: One flip of the UAVs

denote a new graph where V_s is a set of vertices that denotes all the configurations corresponding to the vertices in V 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 θ_i to θ_j. This shortest path specifies the sequence of flips that are necessary to move the UAVs from θ_i to θ_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.

3.1.2 Posing two UAV routing problem as one-in-a-set TSP

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 refer to them as depot), visit at least one of the four configurations at each target and return to their initial location, while the total distance traveled is minimized. A feasible solution for the CCURP posed as a one-in-a-set TSP is shown in Figure 3.1.2. For any two given configurations, we need to know the shortest path between them to pose the one-in-a-set TSP. In section 3.1.2.1, we present an algorithm to compute these shortest paths. Using the result from [37], one can transform the one-in-a-set TSP into an asymmetric TSP, which could be solved using the LKH heuristic [30]. LKH is the best known heuristic to solve TSP, it is known to produce optimal solutions for most instances of TSP within a few seconds. A detailed description to transform the on-in-a-set TSP into regular asymmetric TSP is provided in *Appendix A*.

3.1.2.1 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 Figure 3.7(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 Figure 3.7(a) corresponds to a vertex in G' as shown in Figure 3.7(b). In the

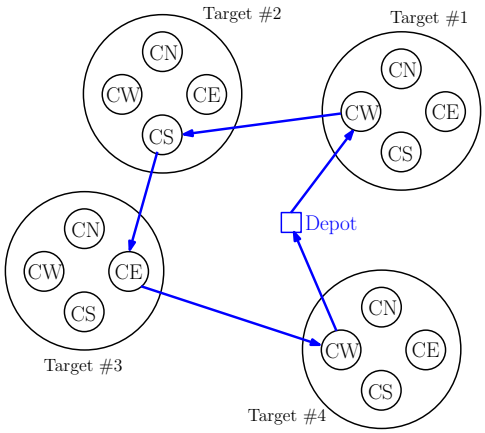
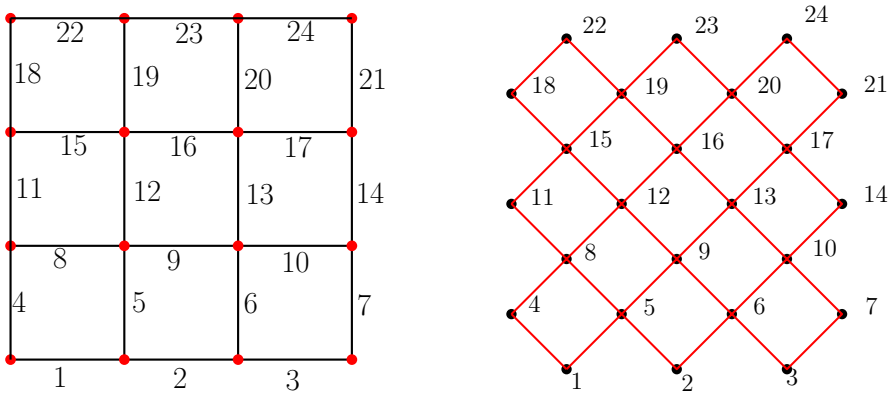


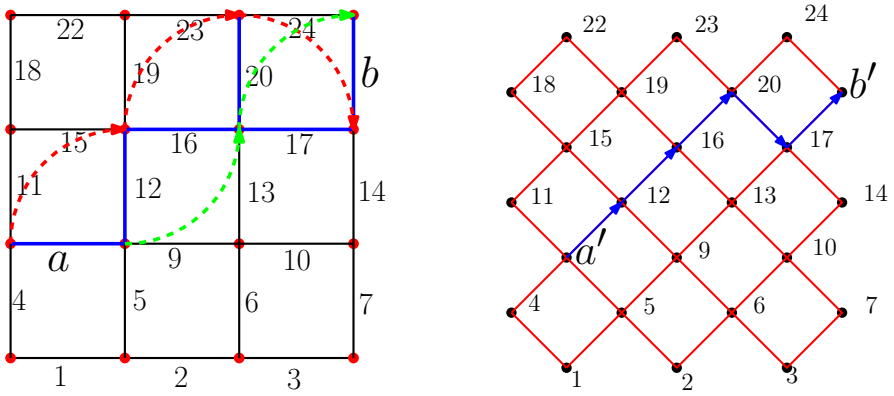
Figure 3.6: One-in-a-set TSP: feasible solution

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'.



(a) Original Graph (G) (b) Transformed Graph (G')

Figure 3.7: Graph transformation for shortest path computation



(a) Original Graph (G) (b) Transformed Graph (G')

Figure 3.8: Shortest path between two configurations

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 Figure 3.8(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 Figure 3.8(b), the shortest path from a' to b' is shown in blue. 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 Figure 3.8(a), where the red and green dashed lines represent the paths of the two UAVs.

3.1.3 Approximation algorithm

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 [42] 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 *Approx*:

1. For each target $u \in T$, choose any configuration θ_u^1 from the set $\{CE, CW, CN, CS\}$.

2. For each target $u \in T$, assign a new configuration θ_u^2 that is exactly two flips away from θ_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 Θ^k. The cost of traveling the edge joining any two distinct configurations θ_u^k, θ_v^k is set to $d_{min}(\theta_u^k, \theta_v^k)$.
   - Use the Christofides algorithm [42] 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 Π^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, Θ^1 and Θ^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 Θ^1 and Θ^2, let Π^1_{opt} and Π^2_{opt} be the optimal sequence of targets to visit respectively. For any feasible sequence of targets (Π) and a corresponding set of configurations (Θ), let $C(\Pi, \Theta)$ represent the cost of the corresponding tour. Let Π^* be the sequence of targets and let Θ^* be the corresponding configurations at each target in an optimal solution to the CCURP.

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

Proof. Choose any $k \in \{1, 2\}$. The Christofides algorithm for the TSP has an approximation ratio of $\frac{3}{2}$ [42]. Given the set of configurations Θ^k, as Π^k_{opt} is an optimal solution to the TSP and Π^k is the feasible solution obtained using the Christofides algorithm, we must have

$$C(\Pi^k, \Theta^k) \leq \frac{3}{2} C(\Pi^k_{opt}, \Theta^k). \tag{3.1}$$

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

$$C(\Pi^k_{opt}, \Theta^k) \leq C(\Pi^*, \Theta^k). \tag{3.2}$$

From inequalities (3.1) and (3.2):

$$C(\Pi^k, \Theta^k) \leq \frac{3}{2} C(\Pi^*, \Theta^k). \tag{3.3}$$

□

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 πR units during this motion. The following lemma bounds the cost of choosing the configurations in Θ^1 in terms of the optimal cost of the CCURP.

Let n_1 denote the number of targets whose configurations in Θ^1 differ from the corresponding optimal set of configurations in Θ^* by one flip. Let n_2 denote the number of targets whose configurations in Θ^1 differ from the corresponding optimal set of configurations in Θ^* 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 Π^* 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 Θ^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

$$\begin{aligned} & d_{min}(\theta^1_{u_i}, \theta^1_{u_{i+1}}) \\ & \leq d_{min}(\theta^1_{u_i}, \theta^*_{u_{i+1}}) + d_{min}(\theta^*_{u_{i+1}}, \theta^1_{u_{i+1}}) \\ & \leq d_{min}(\theta^1_{u_i}, \theta^*_{u_i}) + d_{min}(\theta^*_{u_i}, \theta^*_{u_{i+1}}) + d_{min}(\theta^*_{u_{i+1}}, \theta^1_{u_{i+1}}). \end{aligned} \tag{3.4}$$

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

$$\begin{aligned} C(\Pi^*, \Theta^1) &= \sum_{i=1}^{n-1} d_{min}(\theta^1_{u_i}, \theta^1_{u_{i+1}}) + d_{min}(\theta^1_{u_n}, \theta^1_{u_1}) \\ &\leq \sum_{i=1}^{n-1} d_{min}(\theta^*_{u_i}, \theta^*_{u_{i+1}}) + d_{min}(\theta^*_{u_n}, \theta^*_{u_1}) + 2\sum_{i=1}^{n} d_{min}(\theta^1_{u_i}, \theta^*_{u_i}) \\ &= C(\Pi^*, \Theta^*) + 2\sum_{i=1}^{n} d_{min}(\theta^1_{u_i}, \theta^*_{u_i}). \end{aligned} \tag{3.5}$$

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 πR if the configurations $\theta^1_{u_i}$ and $\theta^*_{u_i}$ differ by two flips. Therefore,

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

□

Based on the choice of configurations in Θ^2, note that there must be n_1 targets whose configurations in Θ^2 differ from the corresponding optimal set of configurations in Θ^* by one flip. Also, there must be $n - n_2 - n_1$ targets whose configurations in Θ^2 differ from the corresponding optimal set of configurations in Θ^* 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:

$$C = \sum_{i=1}^{n-1} d_{min}(\theta_{s_i}, \theta_{s_{i+1}}) + d_{min}(\theta_{s_n}, \theta_{s_1}). \tag{3.6}$$

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 3. *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 (Π^1, Θ^1) and (Π^2, Θ^2). Combining Lemma 1,2 and Corollary 1, we get,

$$\begin{aligned}
&\min(C(\Pi^1, \Theta^1), C(\Pi^2, \Theta^2)) \\
&\leq \frac{3}{2} \min(C(\Pi^*, \Theta^1), C(\Pi^*, \Theta^2)) \\
&\leq \frac{3}{2} \left[C(\Pi^*, \Theta^*) + \pi R \ \min\{n_1 + 2n_2, 2n - (n_1 + 2n_2)\}\right].
\end{aligned} \tag{3.7}$$

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,

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

□

3.1.4 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) [43] instead of the Christofides algorithm to find a tour in *Step 3* of *Approx*. The LKH is considered to be one of the best algorithms for solving a single vehicle TSP.

3.2 Three UAV CCURP

In the second architecture, the UGS may be arbitrarily located within the environment; there is no need for a regular grid. In this scenario, three UAVs are needed for the cooperative navigation. When a UAV is located above 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. If two UAVs are located at two UGSs, the third UAV triangulates its position using (i) xy-coordinates of the first two UAVs that are located above distinct UGS and (ii) range measurements from the first two UAVs. It was shown in [44], two measurements from two different known locations is sufficient for the system to be observable. As long as the third UAV is within the communication range (R units of distance) from the other two UAVs (located at two different UGS), it can navigate from one point to another. In Figure 3.9, UAV_1 and UAV_2 are, respectively, located at UGS a and b. Since, UGS c and d are within a distance of R units from a and b, UAV_3 is able to travel from UGS c to d while maintaining communication link with UAV_1 and UAV_2.

In a more general scenario with more than three UAVs, the UAV that is traveling from one UGS to another might be able to communicate with more than two UAVs within its communication range. Due to more range measurements available, this would allow more precise estimation of the traveling UAV's location. In this thesis, we consider the routing problem with three UAVs, which is a minimum requirement for this navigation scheme.

3.2.1 Problem statement

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

- each UGS is visited atleast once by one of the UAVs,

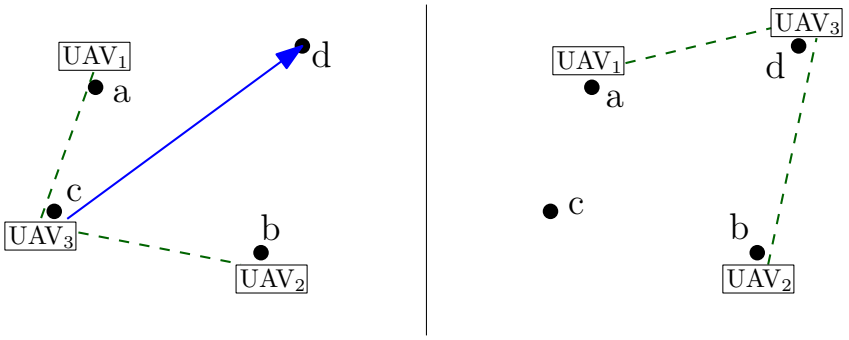


Figure 3.9: UAVs traveling while maintaining the communication links

- the traveling UAV is within a distance of R from the other two UAVs, which are located at two UGSs,
- 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 UGSs in the restricted zone. We define a configuration and adjacent configurations as the following:

Configuration: Three UAVs located at three UGSs such that the distance between any two of them is less than the communication range R. In the Figure 3.9, three UAVs are located at UGSs a, b, and c forming an admissible configuration. Even when the UAVs are not located at those UGSs, we would refer to those set of three UGSs as a possible configuration or simply a configuration.

Adjacent Configurations: Two configurations are adjacent if they have two UGSs in common. In Figure 3.9, configurations (a, b, c) and (a, b, d) share two common UGSs, 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 exam-

ple, in Figure 3.9, the UAV located at c can go to d while using range measurements relative to the positions of 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. An UGS may present in more than one configuration as shown in Figure 3.9, UGS a is present in two configurations (a, b, c) and (a, b, d). An UGS is considered to be visited, if the UAVs visit atleast one of the configurations in which the UGS is present. For the feasibility of the routing problem, we make the following assumptions:

A1: For every UGS $i \in V$, there exists atleast 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.

A2: 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 UGSs i and l. An UGS can be present in more than one configurations. For example, in Figure 3.9, 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 these 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 in V is visited atleast once by one of the UAVs and the total distance traveled by all

the UAVs is minimum.

$$\text{Minimize} \sum_{i=1}^{p-1} D(\mathcal{C}_i, \mathcal{C}_{i+1}) + D(\mathcal{C}_p, \mathcal{C}_1) \tag{3.8}$$

$$\text{Subject to: } V \subseteq \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_p. \tag{3.9}$$

3.2.2 Solution methodology

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. The following is the outline of construction of Graph G':

- Identify all the possible configurations in G, i.e. triplets of UGS such that the distance between any pair of UGS is less than the UAVs' communication radius R.
- 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. Figure 3.10 shows an example of the graph G and G' constructed as explained here.
- For each UGS $a \in V$, identify the set (S_a) of configurations in which a is present. In Figure 3.11, a target is represented in red, the configurations in which it is present and corresponding nodes in G' are shown in red.
- Complete the graph G' by adding the edges between every pair of vertices with weight equal to the distance of the shortest path between them.

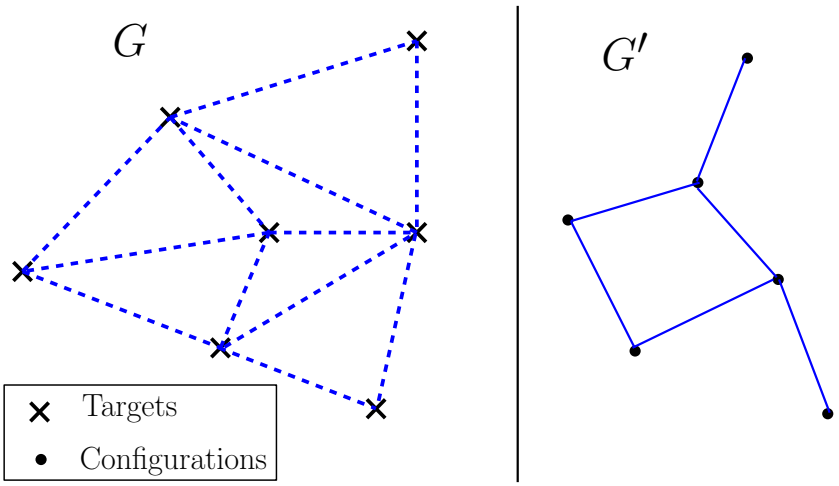


Figure 3.10: Graph G' constructed based on the graph of the target locations G

- Solve the following one-in-a-set TSP: find a tour on G', such that atleast one vertex/configuration from each set S_a, $a \in V$ is visited, such that the length of the tour is a minimum.

The resulting one-in-a-set TSP may not have mutually exclusive node-sets. An algorithm to transform this into an instance with mutually exclusive node-sets, and in-turn into an asymmetric TSP is presented in *Appendix A*.

3.3 AMASE simulation

AMASE is a simulation environment developed at Air-Force Research Labs for the cooperative control study and analysis of the UAVs. It is used to study the effectiveness and feasibility to implement the control algorithms necessary for the 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 this simulation, 25 UGS were deployed uniformly as a

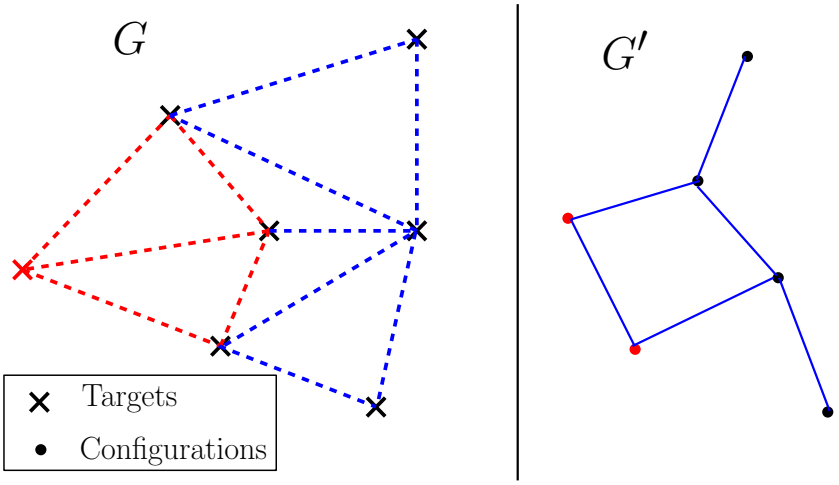


Figure 3.11: A target present in more than one configuration shown in G and its corresponding nodes in G'

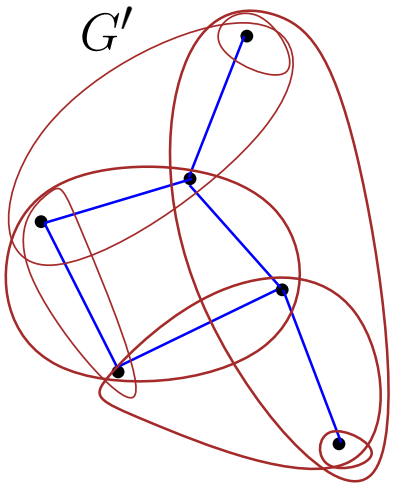


Figure 3.12: Node sets in G', each set containing the configurations correspond to each target

squre grid and five target locations are selected randomly. A snapshot of the simulation is shown in Figure 3.13. The green arrow represent the idling UAV and the pink arrow represents the orbiting UAV. Since the UAV considered here were fixed wing aircrafts, the idling UAV orbits the UGS with the least turning radius possible while the orbiting UAV hops from one UGS to another.

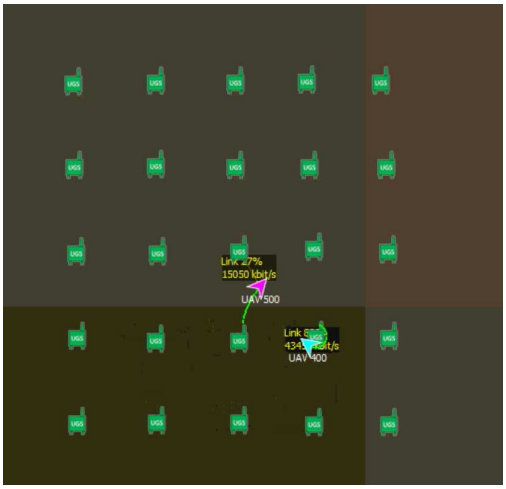


Figure 3.13: Simulation of the two UAV CCURP on AMASE

3.4 Computational results

To test the algorithms for the two UAV problem, a restricted zone of size 30 × 30 units was chosen. This zone was divided into squares of size equal to 1 unit. The targets were located randomly in this area. Fifty instances were generated for each case with 10, 20, 30 and 40 targets. The LKH program [43] by Helsgaun available at http://www.akira.ruc.dk/ keld/research/LKH/ was 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 were 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 was 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 [37], and finally, transform the asymmetric TSP into a symmetric TSP using the result in [45]. The resulting symmetric TSP was solved using the *Concorde* solver [46].

The simulation results are shown in Tables 3.1 and 3.2. The algorithms *Approx* and $Approx_{lkh}$ were relatively very fast compared to the transformation method used for solving the CCURP. We were 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 *a posterior* guarantee of the solutions found by *Approx* and $Approx_{lkh}$ is shown in table 3.2. These results show that the proposed algorithms were able to find high quality solutions for the tested instances relatively fast. Solutions obtained for an instance using the proposed algorithms are shown in Figure 3.4 and 3.4. An optimal solution for this instance is shown in Figure 3.4.

The three UAV CCURP is solved using the graph transformation method explained in Section 3.2.2. Four benchmark traveling salesman problem instances are chosen from TSPLIB [47] for the three UAV CCURP. The city locations given in the TSPLIB instances are chosen as the UGS locations for the CCURP. We have to choose the communication radius of the UAVs for each instance such that there exists atleast one feasible CCURP tour. With the communication radius R, for each UGS, there should be atleast two other UGSs within R. For each UGS $u \in V$, we have computed the second shortest of all the distances d_u^2 between itself and other UGSs. We chose the communication radius R to be α 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 of TSPLIB with four different values for α are presented in Table 3.3. The higher the value of α, the communication radius is larger and hence more configurations are feasible. This would lead to find a tour with lower cost than the instance with a lower communication radius. This is as predicted since with more feasible 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.

Table 3.1: 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

Table 3.2: A posterior bound using the proposed algorithms

Targets	**Algorithm** *Approx*		**Algorithm** $Approx_{lkh}$	
	Solution quality (Avg.)	**Solution quality (Max.)**	**Solution quality (Avg.)**	**Solution quality (Max.)**
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

Table 3.3: Computational results for three UAV CCURP

Instance	**#Targets**	α	**R**	**#Config -urations**	**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

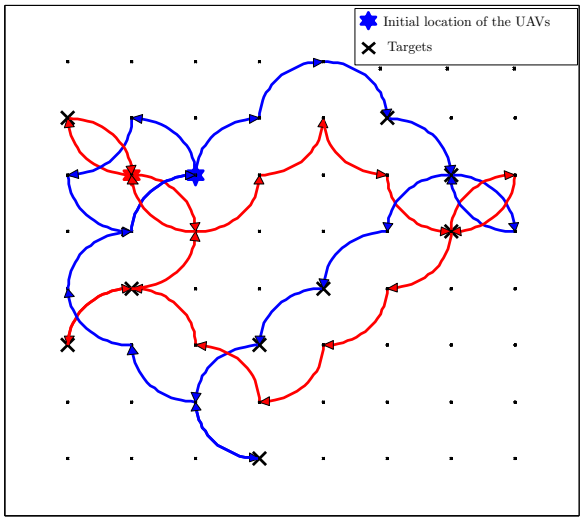


Figure 3.14: An instance with 10 targets solved using algorithm *Approx*

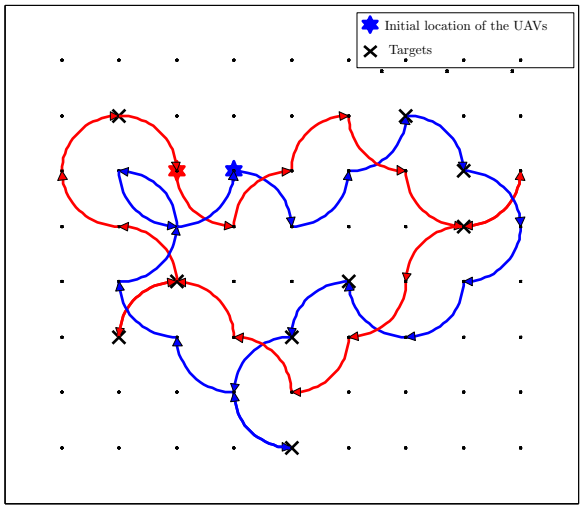


Figure 3.15: An instance with 10 targets solved using algorithm $Approx_{lkh}$

3.5 Conclusion

We considered a problem of routing UAVs with communication constraints in a GPS denied environment. We considered two different architectures for the navi-

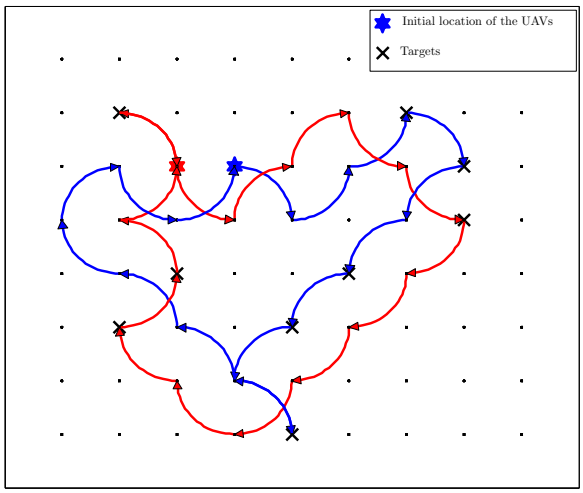


Figure 3.16: Optimal tour of an instance of with 10 targets

gation of the UAVs. We developed a $\frac{9}{2}$-approximation algorithm for the two UAV routing problem, and a transformation method to pose the routing problem as one-in-a-set TSP. These algorithms for the two UAV problem were tested on 50 instances with 10, 20, 30 and 40 targets. The approximation algorithm needed only a fraction of a second and produced solutions within 1.25 times the optimal solution for all of the instances. On the other hand, the transformation method was relatively time consuming but found optimal solutions for most of the instances. Computational results were also presented for the three UAV CCURP using the transformation algorithm; this was tested on four benchmark TSPLIB instances with different communication radii.

4. CONCLUSIONS AND FUTURE WORK

This thesis is devoted for studying the path planning of UAVs for patrolling missions. Due to certain limitations, the UAVs have motion constraints, communication constraints etc. We addressed the path/route planning of the UAVs with two classes of constraints. The first one is motion constraint induced due to the bounded yaw rate of the fixed wing UAVs. Due to this constraint, the path of a UAV needs to have bounded curvature, and this mandates the UAV to travel along arc-like paths to make turns. Therefore the length of the shortest path between two targets depends on the initial and final heading angles of the UAV. To find the routes with these constraints, apart from the sequence of targets to be visited, one has to find heading angles of the UAVs at each target. The coupling of these two sets of decisions makes it harder to solve this routing problem. We developed a Lagrangian relaxation technique to compute tight lower bounds for the routing problem with one UAV (DTSP), and extended that to the multiple vehicle case, MDMVDTSP.

The second class of constraints considered in this thesis is the finite communication radius of the UAVs. UAVs may be denied access to GPS signals due to hostile environments and may need to rely on communication signals for successful navigation. Since the radius of communication is finite, they would not be able to transmit/receive signals beyond certain range (R). The zone of interest which needs to be patrolled is deployed with unattended ground sensors (UGSs) and the UAVs need to cooperatively travel without losing contact with other UAVs and/or the UGSs. The path planning problem for the UAVs is solved along with the constraints imposed by the limited communication range.

4.1 Contributions of Section 2

- The UAVs with finite yaw rate cannot change their headings instantaneously, and they are modeled as Dubins vehicles. The routing problem (DTSP) has the following complicating constraints: at each target, the incoming heading angle of the UAVs should be the same as the outgoing heading angle.

- We have relaxed the heading angle constraint and penalized the objective whenever they are violated through penalty/dual variables. Then, the problem reduces to a regular ATSP where the cost of an edge between a pair of targets is given by the solution of a variational problem. This variational problem is solved by discretizing the variables and using the Dubins result.

- The resulting ATSP is solved using the LKH heuristic, which gives a tight lower bound to the ATSP, this in turn is a lower bound to the DTSP.

- The lower bounds were made tighter by maximizing with respect to the penalty variables. This was done using a sub-gradient procedure, which updates the penalty variables iteratively and improves the lower bound.

- The lower bound computation using Lagrangian relaxation is extended to the routing problem with multiple UAVs, MDMVDTSP.

- This technique produced significantly better bounds compared to the existing bounds (solution of corresponding Euclidean TSP). The average improvement in the bounds is 30% to 40% compared to the solution of ETSP.

- For certain instances that satisfy a distance criteria, using the convexity property of the Dubins paths, we developed a technique to compute lower bounds for the DTSP/MDMVDTSP.

4.2 Contributions of Section 3

- We have addressed the GPS-denied UAV routing problem with two different architectures involving the UGS and UAVs' localization schemes using the communication signals.

- The first architecture requires two UAVs and it needs the UGS to be placed uniformly as a regular grid; the UAVs navigate using orbiting/circumnavigating controllers on-board. A $\frac{9}{2}$ approximation algorithm is developed for the two UAV CCURP. A transformation method is developed for this problem; this poses the two UAV CCURP as a one-in-a-set TSP. The one-in-a-set TSP is solved by transforming into an asymmetric TSP using a known result. The resulting ATSP is solved using the LKH heuristic and *Concorde* TSP solver. This routing scenario is successfully simulated on AMASE, an AFRL software to study the feasibility of the routing and control algorithms.

- The second architecture requires three UAVs, and this does not need the UGS need to be located in a regular grid. A graph transformation method is developed to pose the three UAV CCURP as a one-in-a-set TSP, which is solved by transforming it into an ATSP.

4.3 Future Work

- Though the lower bounds computed for the DTSP and MDMVDTSP are tighter than the existing bounds, the gaps between the lower bounds and the upper bounds are quite large. One can look at alternate ways of relaxing the motion/heading constraints to compute further tighter lower bounds.

- The incoming and outgoing heading angles may not be equal at each target in the solution of the DTSP with heading angle constraints relaxed. If we

chose appropriate penalty variables, in the solution of the relaxed problem, the incoming and outgoing headings should be reasonably close to each other. Based on this solution, one can develop heuristics to produce near optimal solutions to the DTSP/MDMVDTSP with the quality of the solution readily available from the lower bounds.

- To find the optimal solutions to the CCURP, the one-in-a-set TSP is transformed into ATSP, which in turn is transformed into a symmetric TSP and solved using *Concorde* solver. After the two transformations, the problem size may increase and become too large to be solved in reasonable time. In the three UAV CCURP, this problem would be aggravated if there are too many targets within a circle of radius equal to the communication range of the UAVs . This may lead to a huge number of possible configurations and the size of the resulting one-in-a-set TSP may be too large. Therefore, one can pose the CCURP as a mixed-integer linear program similar to the multiple salesmen problem. Additional constraints could be added to maintain the communication feasibility, which is necessary for the localization of the UAVs. The resulting formulation could be efficiently solved with branch and cut algorithms, and this may find optimal solutions for larger instances of the CCURP. This formulation could be extended to solve the routing problem involving a swarm of vehicles, which need to maintain communication links between themselves and/or with a remotely located ground station.

REFERENCES

[1] C. Schumacher, P. R. Chandler, and S. R. Rasmussen, "Task allocation for wide area search munitions," in *Proc. IEEE American Control Conference*, vol. 3, 2002, pp. 1917–1922.

[2] D. Gross, S. Rasmussen, P. Chandler, and G. Feitshans, "Cooperative operations in urban terrain (counter)," in *Proc. ISOP Defense and Security Symposium*, 2006, pp. 62 490G–62 490G.

[3] A. Girard, A. Howell, and J. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proc. IEEE Conference on Decision and Control(CDC)*, vol. 1, 2004, pp. 620–625.

[4] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: Video surveillance of highway traffic," in *Proc. of the ACM International Workshop on Video Surveillance & Sensor Networks*, 2004, pp. 131–135.

[5] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[6] D. W. Murphy and J. Cycon, "Applications for mini vtol uav for law enforcement," in *Proc. ISOP Enabling Technologies for Law Enforcement and Security*, 1999, pp. 35–43.

[7] D. Jea, A. Somasundara, and M. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in *Distributed Com-*

puting in Sensor Systems. Springer, Berlin Heidelberg, 2005, pp. 244–257.

[8] G. Anastasi, M. Conti, and M. Di Francesco, "Data collection in sensor networks with data mules: An integrated simulation analysis," in *Proc. IEEE Symposium on Computers and Communications*, 2008, pp. 1096–1102.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, July 1957.

[10] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[11] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[12] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley, New York, 1985, vol. 3.

[13] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, no. 1, pp. 60–100, 1991.

[14] G. Gutin and A. P. Punnen, *Travleing Salesman Problem and its Variations*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[15] D. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem - A Computational Study*. Princeton University press, Princeton, 2006.

[16] G. Reinelt, *The Traveling Salesman Problem - Computational Solutions for TSP Applications*. Lecture notes in Computer Science, No. 840, Springer-Verlag, Berlin Heidelberg, 1994.

[17] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," *Operations Research*, vol. 18, no. 6, pp. 1138–1162, 1970.

[18] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[19] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a multiple depot, multiple traveling salesman problem," in *Proc. IEEE American Control Conference*, 2009, pp. 2636–2641.

[20] X. Goaoc, H. Kim, and S. Lazard, "Bounded-curvature shortest paths through a sequence of points," INRIA, Tech. Rep., 2010.

[21] G. Warwick, "Lightsquared tests confirm GPS jamming," Aviation Week, July 2011.

[22] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

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

[24] 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.

[25] 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.

[26] W. Malik, S. Rathinam, and S. Darbha, "An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem," *Operations Research Letters*, vol. 35, no. 6, pp. 747–753, 2007.

[27] S. Manyam, S. Rathinam, and S. Darbha, "Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2013, pp. 2378–2383.

[28] W. Malik, S. Rathinam, S. Darbha, and D. Jeffcoat, "Combinatorial motion planning of multiple vehicle systems," in *Proc. IEEE Conference on Decision and Control*, 2006, pp. 5299 – 5304.

[29] K. Savla, E. Frazzoli, and F. Bullo, "On the point to point and traveling salesman problem for dubins' vehicle," in *Proc. IEEE American Control Conference*, 2005, pp. 786–791.

[30] K. Helsgaun, "An effective implementation of the lin-kernighan traveling salesman heuristic," *European Journal of Operations Research*, vol. 126, no. 1, pp. 106 – 130, 2000.

[31] Z. Tang and U. Ozguner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.

[32] H. Chitsaz and S. LaValle, "Time-optimal paths for a dubins airplane," in *Proc. IEEE Conference on Decision and Control*, 2007, pp. 2379–2384.

[33] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution manoeuvres," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 110–120, 2001.

[34] S. M. LaValle, *Planning algorithms*. Cambridge University Press, New York, 2006.

[35] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 50, no. 12 supplement, pp. 1861–1871, 2004.

[36] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*. Wiley-Interscience Publication, New York, 1988.

[37] 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.

[38] A. Hashemi, Y. Cao, D. W. Casbeer, and G. Yin, "Unmanned aerial vehicle circumnavigation using noisy range-based measurements without global positioning system information," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, pp. 31 009–31 019, 2015.

[39] Y. Cao, J. Muse, D. Casbeer, and D. Kingston, "Circumnavigation of an unknown target using uavs with range and range rate measurements," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2013, pp. 3617–3622.

[40] I. Shames, S. Dasgupta, B. Fidan, and B. Anderson, "Circumnavigation using distance measurements under slow drift," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 889–903, April 2012.

[41] T. H. Summers, M. R. Akella, and M. J. Mears, "Coordinated standoff tracking of moving targets: control laws and information architectures," *Journal of*

Guidance, Control, and Dynamics, vol. 32, no. 1, pp. 56–69, 2009.

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

[43] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[44] R. Sharma, R. Beard, C. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, April 2012.

[45] R. Jonker and T. Volgenant, "Transforming asymmetric into symmetric traveling salesman problems," *Operations Research Letters*, vol. 2, no. 4, pp. 161–163, 1983.

[46] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Concorde tsp solver," *See: http://www. tsp. gatech. edu/concorde. html*, 2005.

[47] G. Reinelt, "TSPLIB: A traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

APPENDIX A

TRANSFORMATION OF ONE-IN-A-SET TSP INTO AN ATSP

Transforming general one-in-a-set TSP instance into an instance with mutually exclusive node-sets

We present a transformation method to transform the one-in-a-set set instance with intersecting node-sets into an instance with mutually exclusive node-sets, which is also referred to as canonical one-in-a-set TSP. Then it is transformed into a regular asymmetric TSP. Here we present an overview of the transformation, but one can refer to [37] for detailed description and proofs. Let the one-in-a-set TSP instance be defined on a graph $P0$ constructed in section 3.2.2. Let the set of nodes of $P0$ be $V0$ and edges be $E0$. Each node $i \in V0$ 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 $|\mathcal{M}_i|$ be the size of this set. An instance of a problem is shown in Figure A.1(a). 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 P_1 with the same set of nodes ($V_1 = V0$) as of $P0$, but different edges. Add only the edges in $P0$ which enters atleast one new node-set, and the cost of these edges are same as the cost of the edges in $P0$.

- *Stage 2:* Construct graph P_2 with same set of nodes ($V_2 = V_1$) and edges as in P_1. Let c_{ij} be the cost of the edge $(i, j) \in E0$. Define the cost of each edge in P_2 as $c_{ij} + m\alpha$, where α is a positive number, strictly greater than the sum of the cost of all edges in P_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 a

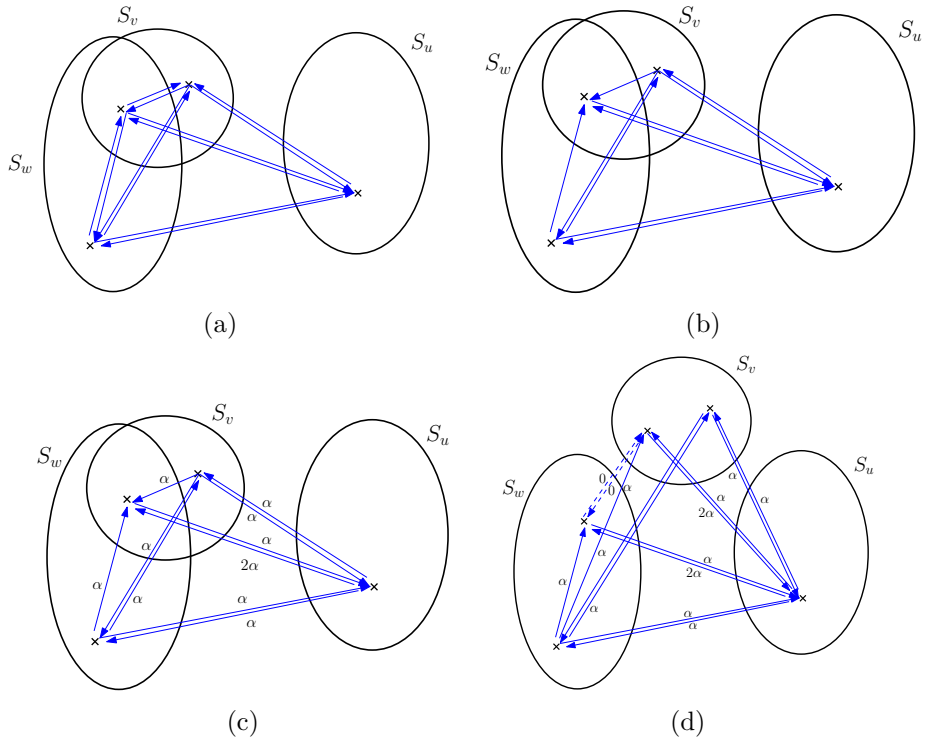


Figure A.1: Transformation of one-in-a-set TSP

$|\mathcal{M}_i|$ 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 P_2. Add edges between the replicas of each node with zero cost and refer to this new graph as $P_3 = (V_3, E_3)$.

- *Stage 4:* In each node-set, remove the edges between nodes belonging to the same set, and refer to this graph as P.

After this transformation through the four stages, the problem is posed as one-in-a-set TSP on graph P with mutually exclusive node-sets. This can be transformed into a regular asymmetric TSP as explained in the next section.

Transforming canonical one-in-a-set TSP into ATSP

We present a method to transform the one-in-a-set TSP to a regular ATSP using the result from [37]. In one-in-a-set TSP, the salesman needs to start from an initial location, visit one of the cities in each set and return to the starting location, such that the total distance traveled is the minimum. Here we explain the transformation in the context of the UAV routing problem. A target in CCURP corresponds to a set and a configuration corresponds to a vertex in one-in-a-set TSP. 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 vertices 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 Figure A.2. 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 to be the 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.

Let us 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

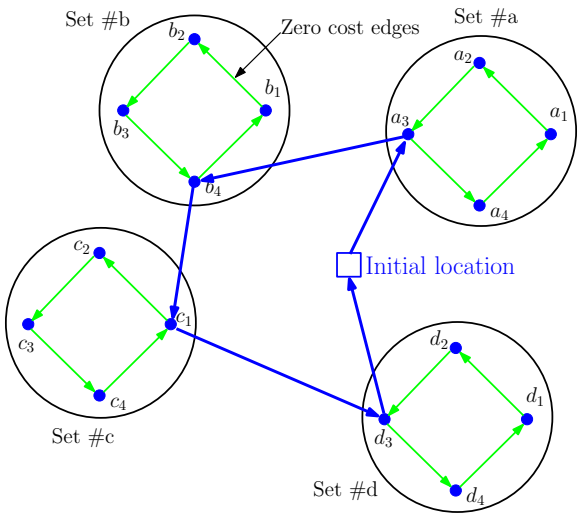


Figure A.2: Transformation of one-in-a-set TSP to ATSP: zero cost edges added.

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 from 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

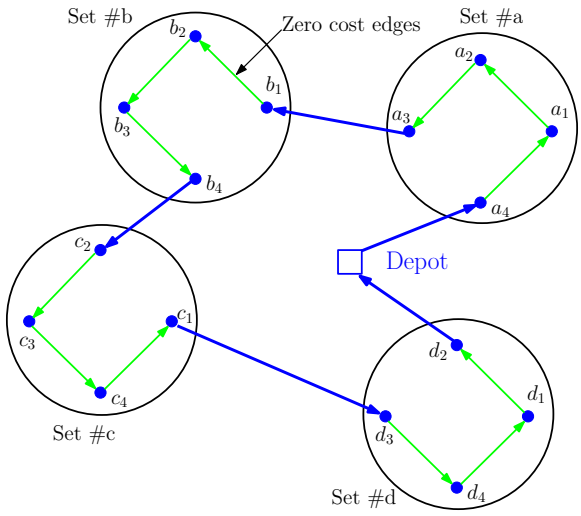


Figure A.3: Transformation of one-in-a-set TSP to ATSP: vehicle entering a set at one vertex exits from the successive vertex.

P', then in the optimal solution of the ATSP, there will be only n edges connecting vertices from different sets.

Theorem 4. *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.*

Proof. We will prove this by contradiction. Let indices $i = a...z$ represent each set in P'. Let $C_{a_1a_2}, C_{a_1a_3}$.... be the costs of the edges before adding M. M is chosen such that it is greater than the sum of all the edges in P'.

$$M \geq \sum_{(i,j)\in P'} C_{ij}. \tag{A.1}$$

Let $\overline{C}_{ij}$ be the cost of the edges after adding the constant M, $\overline{C}_{ij} = C_{ij} + M$, $\forall (i,j) \in P'$. Let us say there are more than n edges connecting vertices from different sets in the optimal solution of the ATSP. Therefore, at least for one of the sets, there

should be two edges connecting the vertices in the set with the vertices from other sets. Let us assume that set $\#q$ has three edges connecting the vertices in it with the vertices from other sets as illustrated in Figure A.4(a). We can remove two of those edges and construct a new tour by connecting all the vertices in set $\#q$ with the zero cost edges. A section of the original and new tour is illustrated in Figure A.4(b) and we call this section a 'sub-tour'. The edges connecting the set of vertices $\{q_1, q_2, q_3, q_4, r_1, r_2, r_3, r_4, t\}$ are different in the assumed tour and the modified tour, and all other edges are same in both the tours. Let C_s and C'_s be the cost of the edges in the sub-tour of the assumed optimal solution and modified solution. Ignoring the zero cost edges, C_s and C'_s can be written as summations of the costs of edges as following:

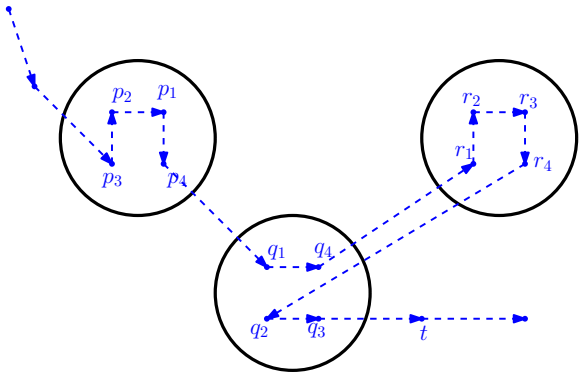
$$C_s = \overline{C}_{q_4r_1} + \overline{C}_{r_4q_2} + \overline{C}_{q_3t} \tag{A.2}$$

$$C'_s = \overline{C}_{q_4r_1} + \overline{C}_{r_4t} \tag{A.3}$$

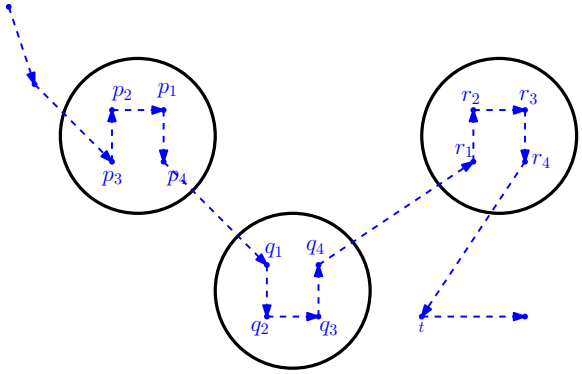
Since M is the sum of the costs of all the edges in P, we have the following inequality:

$$\begin{aligned}
& C_{q_4r_1} + C_{r_4t} \leq M \\
\Rightarrow \quad & \overline{C}_{q_4r_1} + \overline{C}_{r_4t} \leq 3M \\
\Rightarrow \quad & \overline{C}_{q_4r_1} + \overline{C}_{r_4t} \leq (M + C_{q_4r_1}) + (M + C_{r_4q_2}) + (M + C_{q_3t}) \\
\Rightarrow \quad & \overline{C}_{q_4r_1} + \overline{C}_{r_4t} \leq \overline{C}_{q_4r_1} + \overline{C}_{r_4q_2} + \overline{C}_{q_3t} \\
\Rightarrow \quad & C'_s \leq C_s
\end{aligned} \tag{A.4}$$

The cost of the modified tour is less than the assumed optimal solution and hence the assumption is wrong. Therefore the optimal solution of the ATSP in P' cannot have more than n edges connecting vertices from different sets. □



(a) Sub-tour in the assumed optimal solution



(b) Modified sub-tour

Figure A.4: Sub-tour in the optimal solution of the ATSP