COMBINATORIAL PATH PLANNING FOR A SYSTEM OF MULTIPLE UNMANNED VEHICLES

A Dissertation

by

SAI KRISHNA YADLAPALLI

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

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Mechanical Engineering

COMBINATORIAL PATH PLANNING FOR A SYSTEM OF MULTIPLE UNMANNED VEHICLES

A Dissertation

by

SAI KRISHNA YADLAPALLI

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

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Swaroop Darbha
	K.R. Rajagopal
Committee Members,	Sergiy Butenko
	Bryan Rasmussen
	Sivakumar Rathinam
Head of Department,	Dennis O' Neal

December 2010

Major Subject: Mechanical Engineering

ABSTRACT

Combinatorial Path Planning for a System of Multiple

Unmanned Vehicles. (December 2010)

Sai Yadlapalli, B.Tech., Indian Institute of Technology, Madras;

M.S., Texas A&M University

Co–Chairs of Advisory Committee: Dr. Swaroop Darbha
Dr. K. R. Rajagopal

In this dissertation, the problem of planning the motion of m Unmanned Vehicles (UVs) (or simply vehicles) through n points in a plane is considered. A motion plan for a vehicle is given by the sequence of points and the corresponding angles at which each point must be visited by the vehicle. We require that each vehicle return to the same initial location(depot) at the same heading after visiting the points. The objective of the motion planning problem is to choose at most $q(\leq m)$ UVs and find their motion plans so that all the points are visited and the total cost of the tours of the chosen vehicles is a minimum amongst all the possible choices of vehicles and their tours. This problem is a generalization of the well-known Traveling Salesman Problem (TSP) in many ways: (1) each UV takes the role of salesman (2) motion constraints of the UVs play an important role in determining the cost of travel between any two locations; in fact, the cost of the travel between any two locations depends on direction of travel along with the heading at the origin and destination, and (3) there is an additional combinatorial complexity stemming from the need to partition the points to be visited by each UV and the set of UVs that must be employed by the mission.

In this dissertation, a sub-optimal, two-step approach to motion planning is presented to solve this problem:(1) the combinatorial problem of choosing the vehicles and their associated tours is based on Euclidean distances between points and (2) once the sequence of points to be visited is specified, the heading at each point is determined based on a Dynamic Programming scheme. The solution to the first step is based on a generalization of

Held-Karp's method. We modify the Lagrangian heuristics for finding a close sub-optimal solution.

In the later chapters of the dissertation, we relax the assumption that all vehicles are homogenous. The motivation of heterogenous variant of Multi-depot, Multiple Traveling Salesmen Problem (MDMTSP) derives form applications involving Unmanned Aerial Vehicles (UAVs) or ground robots requiring multiple vehicles with different capabilities to visit a set of locations.

To My Mother and Brother

ACKNOWLEDGMENTS

The following is my heartfelt appreciation to those people, both past and present, who gave me the spirit, knowledge and insight to begin, conduct and complete this dissertation.

I dedicate this dissertation, in full, to my mother, Dr. Bhagya Lakshmi and my dear brother, Mr. Hari Krishna. Their constant guidance and motivation has helped me to re-focus during the testing times encountered while conducting research. Their personal care and contributions right up to completion of this dissertation, shall never be forgotten and I am of deepest gratitude for their efforts. I offer my most humble salutations to "Bhagavan Baba" whose invisible hand has guided me gracefully through out.

Naturally, my greatest appreciation goes to both my advisors, Prof. Swaroop Darbha and Prof. K. R. Rajagopal, for their able guidance, support and motivation. Without their assistance, these pages would not have been written. I consider myself very fortunate for being able to work under their supervision. I shall be profusely indebted to them for providing this glorious opportunity to learn. I take this opportunity to recognize Prof. Swaroop's unwearying patience and extremely helpful nature.

With great pleasure, I extend my gratitude to Dr. Sergiy Butenko, Dr. Bryan Rasmussen and Dr. Sivakumar Rathinam, all of whom, willingly agreed to be on my committee in spite of their busy schedules. I appreciate efforts of Dr. Sivakumar Rathinam and Dr. Waqar Malik for teaching me many new things and for being an amazing source of stimulus.

A very special mention to close friends: Mr. Madhu Miriyala, Mr. Bhanu Prakash Vellanki, Mr. Bhanu Sunku and Mr. Sankar Salvady; all of whom are very special to me and have always extended their help when it mattered. I thank them for the people they are and for being real mates to me. Finally, to all those who directly or indirectly have helped me in the process and are not mentioned here, I beg your pardon. I will wind up saying:

"Many are the great people, my humble salutations to all of them!"

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

CHAPTER I

MOTIVATION AND INTRODUCTON

Unmanned Vehicles (UVs) are currently being used and will be used in the future in a variety of military and civilian applications [1, 2, 3, 4]. In military applications, UVs find a prominent role in surveillance and reconnaissance operations. The main advantage of deploying UVs is that it eliminates the need for human pilots to operate in hostile/hazardous environments. Since there is no need to accommodate a human pilot in a UV, the resulting designs for UVs are much simpler and can result in significantly lower production and operational costs. In civilian applications, UVs are envisioned to be used for border patrolling, fire monitoring, search and rescue operations in the aftermath of hazardous events such as earthquake and fire [5].

UVs carry a limited amount of fuel which must be utilized efficiently. Typical missions require every target (target is a location of interest to the mission) to be visited by some UV in the collection. Since UVs have motion constraints, i.e., they cannot change their heading instantaneously, one must take into account this limitation of UVs in planning their motion for such missions. In some applications, one may not assume that the collection of UVs is homogeneous. Heterogeneity in a collection can arise in two different ways: (1) the UVs are structurally different and (2) the UVs may be structurally identical but functionally different because of the on-board sensors that they may carry. In the case of structural heterogeneity, the cost (typically either the fuel consumed or its proxy- the distance traveled) between any pair of locations is also a function of the employed UV. In the case of functional heterogeneity, one may have additional constraints on the assignment of UVs to targets as some targets may require some specific types of sensors to be

---

The journal model is *IEEE Transactions on Automatic Control.*

serviced/attended to.

Typical missions naturally lead to the following problems and variants: Given a set of m UVs and n targets, and a collective objective function (such as the total cost or the maximum time taken for a mission) to be minimized, the path planning problem for the collection requires answers to the following questions:

- Which subset of UVs must be chosen to accomplish a given mission? It may be possible that some UVs may be held back for contingency purposes. In this case, one must determine the set of UVs that must be deployed.
- How should one choose the subset of targets to be visited by each chosen UV?
- What is the sequence of targets that must be assigned to each UV for visiting? What angle should a UV approach each assigned target in the given sequence?

In this dissertation, we will consider UVs which may be modeled either a Dubins' vehicle or a Reeds-Shepp vehicle. In these two types of vehicles, the inertia of the UV is assumed to be negligible and there is a bound on the heading rate, i.e., the heading of a UV does not change instantaneously. A Reeds-Shepp vehicle differs from a Dubins' vehicle in its ability to reverse the direction of travel. The principal consequences of this assumption are as follows: If a UV has a constant longitudinal speed, V_0, then there is a minimum turning radius given by $\frac{V_0}{\Omega}$, where Ω is the maximum rate of change of the heading angle. Another important consequence of treating a UV as a Dubins' or a Reeds-Shepp vehicle is that the problem of determining the path of minimum length reduces to the determination of a finite number of paths of a specified structure and determining, among these finite number of paths, the path of shortest length [6, 7]. This assumption seems to be reasonable for the purposes of planning the motion and determining the sequence of targets to be visited. When the inertia of a UV is not negligible, the problem of determination of the shortest path, even in the absence of significant disturbances such as wind, from a given origin with a specified heading to a given destination, is a difficult problem in general. The assumption of negligible mass of a UV allows us to focus on the combinatorial aspects of motion planning by simplifying the problem of determination of the cost of traveling between any two targets for every UV.

We will consider the collective objective function to be the total distance traveled by the collection. The rationale for this choice of collective objective function is as follows: As a first approximation, the total distance traveled by the collection is representative of the total fuel consumed by the collection. In applications where the mission time is important, one may compute a lower bound on the optimal mission time as follows: For every convex combination specified by m non-negative numbers that add to 1, one may associate the i^{th} non-negative number representing the scaling factor of the cost of routes for the i^{th} UV. One can specify the modified objective to be the sum of the scaled costs of the routes of UVs in the collection. The problem of minimizing the modified objective function is similar to collective objective function considered in this dissertation. In this case, one may readily obtain a lower bound for the optimal mission time for every convex combination and one may optimize over the convex combinations to determine the best possible lower bound. The central point of this argument is that the collective objective function chosen in this dissertation can be useful for determining the lower bounds for other collective objective functions that one may choose.

The combinatorial problem underlying the problem of planning the motion for a collection of UVs is a generalization of the Traveling Salesman Problem (TSP) and will be referred to as the Multiple Depot, Multiple Traveling Salesmen Problem (MDMTSP). A depot is the starting location for a UV and one may treat every UV as a salesman. The TSP is a NP-hard problem [8, 9, 10] and hence, the MDMTSP is also NP-hard as it is a generalization of the TSP. There are two options to “solve” MDMTSP - the first option is to find an algorithm that determines an approximately optimal or sub-optimal solution with a guaranteed running time and with either a prior or a posterior guarantee on the quality of the solution. A solution is of better quality compared to another if it its cost is closer to the optimal cost. Algorithms that provide a priori guarantee on the quality of solutions are called approximation algorithms and the development of such algorithms is an active topic of research. An excellent overview of such algorithms for some NP-hard problems may be found in the recent book of Vazirani [11]. This approach is useful for two reasons: (1) one may find a sub-optimal solution which may be a starting solution for algorithms that improve this solution, and (2) it is suitable for real-time implementation because of the guarantee of running time and the quality of solution it provides. In practice, the solutions provided by the well-known 2-approx and 1.5-approx algorithms [12] for a single TSP deviate, on an average, from the optimum by no more than 30 %. The a posterior guarantees are useful when improvement heuristics are applied to the solutions obtained by the approximation algorithms.

The second approach is to forego the guarantees on running time, but solve the combinatorial problem to optimality. This is possible because the set of feasible solutions to the problem is finite in some combinatorial problems such as the TSP, and one can find an optimal solution by discarding the sets of feasible solutions that are guaranteed to not contain an optimal solution. This is at the heart of most Branch and Bound (B&B) procedures. Discarding the sets of feasible solutions requires finding a lower/upper bound on the cost of solutions in an efficient manner. The effectiveness of a B&B procedure depends on the tightness of the lower and upper bounds that one has at hand.

The problem of routing UVs considered in this dissertation is significantly more difficult than the counterparts considered in the Operations Research literature for the following reasons:

- Even with the simplified models of UVs, the cost of travel between any two locations can depend on the origin and destination and the heading angles at the origin and destination.

- The total cost of a route for a UV is not only a function of the sequence in which targets are visited, but also on the heading angle at which each target is approached. In essence, there is a coupling between the discrete optimization problem of determining the sequence and the continuous optimization problem of determining the heading angle at each and every target.

- The additional combinatorial complexity of determining the UVs that must be selected for the mission and the partitioning of the targets for assigning them to the UVs makes the problem harder.

The dissertation is organized as follows: In Chapter II, we consider a collection of homogeneous vehicles and provide an algorithm for determining a sub-optimal motion plan, i.e., the sequence of targets to be visited by each UV and the associated heading angles. The results obtained by this algorithm seem promising when the distances between the targets is reasonably large compared to the the minimum turning radius. In other words, this algorithm produces feasible solutions of high quality when the coupling between the discrete and continuous optimization problems is not that strong. In Chapters III and IV, we focus on the development of approximation algorithms for a heterogeneous collection of UVs. In Chapter III, the main focus is on the development of an approximation algorithm for a collection of structurally heterogeneous UVs. In Chapter IV, the main focus is on developing approximation algorithms for a collection of functionally heterogeneous vehicles. We provide a brief description of the contents of Chapters II, III and IV in the following subsections.

A. Generalization of Held-Karp's procedure for determining the lower bound for the Multiple Depot, Multiple Traveling Salesmen Problem (MDMTSP) and for finding sub-optimal solutions

The main content of Chapter II deals with the development of an efficient algorithm for determining a lower bound for the optimal cost of any instance of MDMTSP and to develop a heuristic for finding a sub-optimal solution, whose cost is close to the lower bound. By computing the lower bound efficiently, one can bound the deviation of the cost of any feasible solution from the optimal cost. Such a guarantee of the quality of solution is a posteriori as the quality of the solution may be computed after the solution has been determined.

An efficient scheme for obtaining a tight lower bound for the MDMTSP is useful for two reasons - first, it can be used in a B&B procedure and secondly, it can be used to evaluate the quality of approximate solutions obtained by various heuristics. For this purpose, we will extend the method of Held and Karp [13] for the MDMTSP in this section.

For the purpose of developing a mathematical formulation of the problem, let V be a set of nodes (targets and depots) and let E be the set of roads (edges) connecting the nodes. Let c_{ij} represent the cost of traveling between the i^{th} and j^{th} node. Now, one can define a graph $G = (V, E, c)$, representing the set of targets, T and the network of edges connecting them. The MDMTSP may be posed as follows: Given a set of m UVs starting from distinct nodes (depots), find a tour [1] for each UV in such a way that each node is visited *at least once* by some UV and the total distance traveled by the UVs is a minimum among all possible sets of tours assigned to them. If the triangle inequality holds[2], it can

[1] A tour through a set of vertices $\{i_1, ..., i_k\}$ is the set of $k = 3$ distinct edges $\{(i_1, i_2), (i_2, i_3), ..., (i_{k-1}, i_k), (i_k, i_1)\}$.

[2] if the cost of traveling from a node (or a vertex) i to a node j directly is no costlier than the cost of traveling from a node i to node j through intermediate nodes

be shown that each node is visited exactly once by some UV in the optimal solution.

The binary program considered for MDMTSP is hard to solve and is analogous to the linear, integer programming formulation of Dantzig, Fulkerson and Johnson for the TSP [14]. In this formulation, the number of sub-tour elimination constraints is exponential in the number of targets and depots. A sub-tour elimination constraint disallows a tour among nodes that do not contain a depot. Clearly, such a solution is not feasible because a depot corresponds to a starting location for a UV.

Held and Karp's [13] method uses duality to compute a lower bound. Held-Karp's method considers the formulation of Dantzig, Fulkerson and Johnson [14] and penalizes the degree constraint on the target vertices. The degree of a vertex is twice the number of times a target has been visited. The mathematical problem of Dantzig, Fulkerson and Johnson [14] is an integer linear program where the choice variables indicate what the next node is from a given node, i.e., which edges must be picked. Held and Karp[13] show that the resulting integer program with the penalty variables admits a simple (greedy type) combinatorial algorithm. Hence, for each set of penalty variables, one may compute the optimal penalized cost, which is a lower bound for the optimal cost of TSP. Further, Held and Karp [13] pose the problem of finding the greatest lower bound as that of determining the penalties that maximize this lower bound. This lower bound, referred to as the Held-Karp lower bound, is found to be within 1% of the optimal cost in most instances of TSPLIB. We follow a similar approach for the MDMTSP and the numerical results seem to indicate that the lower bound is equally tight even for the MDMTSP. We also provide a heuristic, which computes a sub-optimal solution that is close to the dual solution and hence, the cost of the sub-optimal solution is close to the lower bound.

B. Motion planning for a collection of structurally heterogeneous vehicles

In Chapter III, the main content of the dissertation deals with the development of motion planning algorithms for a collection of structurally heterogeneous UVs. In this case, the cost of travel from the i^{th} node to the j^{th} node in a graph also depends on the UV deployed and hence, we may represent the cost as c_{ijk} where k is the index of the UV deployed for travel from the i^{th} node to the j^{th} node.

The focus of this chapter is on the development of approximation algorithms. Aiming for approximation algorithms is reasonable in the context of path planning for a collection of Unmanned Aerial Vehicles (UAVs) with motion constraints because the cost of traveling between any two targets for a UAV can depend on several factors including wind disturbances. Hence, it is appropriate to devise approximation algorithms for these planning problems that are relatively inexpensive than devise algorithms that opt for exact solutions. In this sense, the approach adopted in Chapter II is reasonable in decoupling the discrete optimization and continuous optimization by first determining the sequence of targets to be visited by each UV based on the Euclidean distances and then determining the heading angles using a Dynamic Programming technique. To realistically solve this complicated problem in real-time will be difficult and hence, a sub-optimal solution for the discrete problem based on the Euclidean distances may be used for planning the motion of a UAV even in the presence of wind disturbances.

In Chapter III, we introduce a 3-approximation algorithm for the following two depot, heterogeneous TSP (or simply 2-HTSP), when the costs associated with each vehicle satisfy the triangle inequality: Given a set of targets and two heterogeneous UVs that start from distinct depots, find a tour for each vehicle such that each destination is visited exactly once and the total cost of the tours of the vehicles is a minimum. We assume that the headings are specified at each and every target and we relax the motion constraint that a UV travels in the forward direction only, i.e., we will treat each UV to be a Reeds-Shepp vehicle. In this case, the distance between any two nodes is symmetric (i.e., for the k^{th} UV, the cost, c_{ijk}, of travel from the i^{th} node to the j^{th} nodes is the same as the cost, c_{jik}, of travel from the j^{th} node to the i^{th} node. We will further assume that triangle inequality holds for every UV, i.e., for every k and for any three distinct nodes i, j and l, the following inequality holds: $c_{ijk} + c_{jlk} \geq c_{ilk}$.

At the end of Chapter III, we provide generalization of the results of 2-HTSP to collections of more than 2 structurally heterogeneous vehicles. In the process of developing an approximation algorithm for multiple vehicles, we also pose a Heterogeneous, Minimum Cost Spanning Forest (HMSF) problem, a combinatorial problem of independent interest that seems relevant to developing a constant factor approximation algorithm for Multiple depot HTSP.

C. Motion planning for a collection of functionally heterogeneous UVs

In Chapter IV, we consider UVs that are structurally homogeneous but have different capabilities, e.g., they may have different on-board sensors for servicing targets. The cost of travel from i^{th} node to j^{th} node is the same for every UV in the collection. The UVs differ from each other in their sensing capabilities and accordingly, we categorize the targets into three disjoint subsets:

1. **Category I**: Any target in this category may be visited by any UV in the collection.

2. **Category II:** A target in this category may only be visited by a specific UV or a subset of UVs. This arises in a scenario where the technology/equipment to accomplish the desired task on a target is available only to a subset of UVs. Also, if a group of targets form a cluster i.e., they are very close to each other in terms of distance, it might be economical to let one UV perform all the tasks on these group of targets.

3. **Category III:** A target in this category may be unsuitable to be visited by a particular UV or a subset of UVs.

Even though the cost of traveling from one node to another is the same for every UV, these restrictions on the assignment of UVs to target, which we will refer to as assignment constraints, introduce heterogeneity.

In this chapter, the following problem is considered: Given a set of depots (starting locations of UVs) and their corresponding terminals (ending locations of UVs) find a path for each vehicle such that

- the path of each UV starts from its respective depot and ends at the corresponding terminal,
- each target is visited exactly once by some vehicle,
- the assignment constraints are satisfied and,
- the total cost of the paths of all the UVs is a minimum among all possible choices of paths for the UVs.

The above problem is a generalization of the Hamiltonian Path Problem (HPP), which is also NP-hard [8]. An optimal Hamiltonian path is a path that contains each vertex exactly once of minimum total cost. The best approximation algorithm currently available for the HPP was proposed by Hoogeveen [15]. In [15], Hoogeven proposed an approximation algorithm for three variants of single HPP that depend on the choice of the endpoints of the path. Hoogeveen modified the Christofides algorithm, and provided a 3-approx algorithm for the variant of the HPP when at most one endpoint is fixed and proposed a 5-approx algorithm when both endpoints are fixed.

We develop constant factor approximation algorithms based on the work of Hoogeven for the case of multiple UVs.

D. Contributions of the dissertation

The following are the *novel* contributions of this work:

- This dissertation provides a generalization of the Held-Karp lower bound for asymmetric and symmetric MDMTSP; further, it provides sub-optimal motion plans for a collection of homogeneous UVs with bounds on the deviation of the cost of the motion plans from the optimal one. The sub-optimal motion planning algorithm and the computation of the bound is based on the generalization of Held-Karp algorithm for a single TSP and dynamic programming.

- This dissertation identifies a combinatorial problem of independent interest - Minimum cost, Heterogeneous Spanning Forest (MHSF). Although the computational complexity of this problem is not readily apparent, this provides the *first* approximation algorithm for the construction of a suboptimal Heterogeneous Spanning Forest and the associated Multiple Depot, Multiple Heterogeneous Traveling Salesmen Problem.

- Prior to this dissertation, there were no constant factor approximation algorithm for any variant of the heterogeneous, multiple HPP. The contribution on this dissertation is in providing a constant factor approximation algorithm for the variant of HPP considered. In this chapter, a 11/3 approximation algorithm for the multiple depot-terminal HPP with functional heterogeneity constraints is presented. In the special case when the locations of the terminals coincides with their respective depots, the approximation factor of the proposed algorithm reduces to 3.5. This approximation factor of 3.5 also holds true for other variants of the heterogeneous, multiple depot HPP when at most one endpoint is specified for each vehicle.

CHAPTER II

LAGRANGIAN-BASED MOTION PLANNING ALGORITHMS FOR A HOMOGENOUS COLLECTION OF UVS

This chapter * consists of two parts: In the first part, we provide algorithms for planning the motion of a homogeneous collection of UVs whose costs are symmetric, i.e., the cost, c_{ij}, of traveling from a node i to node j is the same as the cost, c_{ji}, of traveling from node j to node i. One obtains such a problem by relaxing the motion constraints of a UV altogether and considering the cost of travel, c_{ij} between the i^{th} and j^{th} nodes to be the Euclidean distance between them. In the second part of this chapter, we relax the assumption of symmetry and this situation corresponds to treating UVs as Dubins' vehicles. We provide a Lagrangian-based algorithm and provide a useful lower bound for this problem through the use of duality. We also provide Lagrangian heuristics to compute a sub-optimal solution from the dual solution. We adopt a two-step approach for the computation of a sub-optimal solution for the UVs in the collection: By considering the cost of travel between a pair of nodes to be the Euclidean distance between the nodes, we develop a partition and assignment of sequence of targets to each UV and we use Dynamic Programming technique to determine the optimal heading angle for each UV at the targets assigned to it.

A. Lagrangian based algorithm for MDMTSP

Motion planning of a collection of unmanned vehicles has significant applications, see [1, 2, 3, 4] and the references therein. The problem of motion planning considered for these applications involves the solution of a combinatorial problem, wherein one must determine

*Part of this chapter was reprinted with permission from "3-Approximation Algorithm for a Two Depot, Heterogenous Traveling Salesman Problem," by Sai Yadlapalli, Sivakumar Rathinam and Swaroop Darbhn, Accepted for Publication in Operation Research Letters, the set of points to be visited by each vehicle and the sequence in which they must be visited before returning to the initial location (depot). Equally important is the consideration of motion constraints of vehicles in the planning.

Underlying the combinatorial problems of motion planning for vehicles is a variant of the Traveling Salesman Problem (TSP), which will be referred to as the Multiple Depot, Multiple Traveling Salesmen Problem (MDMTSP). For a mathematical formulation of the problem, let a graph $G = (V, E, c)$ that represents a network of roads connecting a set of cities (nodes) V and let E be the set of roads (edges) connecting the cities (nodes). Let c_{ij} represent the distance between the i^{th} and j^{th} nodes. The MDMTSP may be posed as follows: Given a set of m salesmen starting from distinct nodes (depots) and a set of distinct nodes that they must collectively visit, find a tour[1] for each salesman in such a way that each node is visited *at least* once by some salesman and the total distance traveled by the salesmen is a minimum among all possible sets of tours assigned to them. If the triangle inequality holds[2], it is easy to see in an optimal solution of the MDMTSP that each node is visited exactly once by some salesman. This problem is an *NP*-hard problem as it is a generalization of the Traveling Salesman Problem (TSP).

In view of this, there are two options to solve TSP - the first is to find a polynomial algorithm that returns an approximate solution whose cost is within a guaranteed factor of the optimal solution which is known *apriori*. The development of such algorithms is an active topic of research; an excellent overview of such algorithms for some *NP*- hard problems is given in the recent book of Vazirani [11]. The second approach is to forego the polynomial running time guarantee but solve the problem exactly. This is possible because the set of feasible solutions to the problem is finite and one can systematically enumerate

[1] A tour through a set of vertices $\{i_1, \ldots, i_k\}$ is the set of $k \geq 3$ distinct edges $\{(i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k), (i_k, i_1)\}$.

[2] A triangle inequality holds for the graph G if for every $i, j, k \in V$, if $c_{ij} + c_{jk} \geq c_{ik}$.

all the feasible solutions and find an optimal one by discarding sets of feasible solutions that are guaranteed to not contain the optimal solution. This is at the heart of most Branch and Bound procedures. Discarding sets of feasible solutions requires finding a lower/upper bound on the cost of solutions in the set in an efficient manner. The effectiveness of a B&B procedure depends on the tightness of the lower and upper bounds that one has at hand.

In a symmetric TSP, the costs of edges (i, j) and (j, i) are the same, i.e., $c_{ij} = c_{ji}$. The symmetric TSP admits constant factor approximation algorithms, notable among them are the $2-$approx algorithm that is based on doubling the Minimum Spanning Tree (MST) of G, the $1.5-$ approximation algorithm of Christofides that is based on the computation of MST of G and a non-bipartite matching of a subgraph of G, and the recent $(1 + \epsilon)$ approximation algorithm of Arora for planar TSP [16]. The development of constant factor approximation algorithms for the Asymmetric TSP (ATSP) is an open problem [11].

B&B algorithm have no polynomial running time guarantees. The B&B scheme proposed by Held and Karp [13] is based on the computation of a "tight" lower bound using $1-$trees. This bound, referred to as the Held-Karp Lower bound (HKLB), is reported to be within $1 - 2\%$ of the optimum on instances in the TSPLIB [17]. Essentially, HKLB is determined by solving the dual program associated with the Dantzig-Fulkerson-Johnson (DFJ) Integer Linear Programming formulation of the TSP [13]. It involves relaxing the degree constraints on all nodes except the first node and retaining the connectivity constraints (which are exponential in the number of nodes and essentially state that if the first node and the edges incident on it are removed from the solution, the resulting graph must be a spanning tree). For every set of penalty variables associated with the relaxed constraints, one can compute the dual in an efficient way through the computation of an associated MST [13, 18, 19]. The maximization of the dual program can then be performed using a sub-gradient method [20]. If triangle inequality holds for the edge costs, HKLB is guaranteed to be at least two-thirds the optimum [21, 22, 23].

There is a wealth of literature dealing with the TSP and the single depot variant of the MDMTSP can be reduced to a TSP [13, 24]. Rao [25] converted the MDMTSP to a TSP when the number of salesmen is restricted to two. It is not clear if the general MDMTSP problem can be converted to a TSP. In light of this shortcoming, the literature on TSP cannot be readily applied to the MDMTSP.

The symmetric version of the MDMTSP admits a $2-$approx algorithm [26, 27] when the edge costs satisfy the triangle inequality. The extension of Held-Karp's approach to finding Hamiltonian paths for multiple vehicles was recently considered in [28]. In this work, the terminal points are not specified, and in principle, it can be converted to a TSP on a directed graph, and one can employ the Held-Karp's approach for directed graphs. Since there is no ready transformation of the MDMTSP to TSP, a generalization of the Held-Karp's method is provided in this chapter and use it to compute primal feasible solution and provide *a posteriori* guarantee of the quality of solution obtained by the proposed method.

The problem of motion planning for a single Dubins' vehicle is considered in [26, 29]. The approach of [26] is to provide an approximate solution that is guaranteed to be within a constant factor of the optimum, while in [29], the authors provide a bead-tiling algorithm which has asymptotic guarantees.

The problem of motion planning of multiple vehicles is considered in [26, 27] with a view towards providing approximate solutions that are guaranteed to be within a certain factor of approximation. The schemes considered make the assumption that the points are well separated, i.e., the distance between points is at least twice the minimum turning radius of the vehicles. This condition is reasonable when the dimension of the sensor footprint is comparable or greater than the turning radius and it enables the separation of the combinatorial problem of finding the set of points to be visited by vehicles and the sequence in which they must be visited from the continuous optimization problem of determining the headings at each point.

Following [26, 27], in this chapter, a two step approach is adopted for solving the MDMTSP when a Dubins vehicle represents a salesman. The combinatorial aspect of the problem can be solved by considering the Euclidean distances between the vertices. To solve the combinatorial problem, a generalization of Held-Karp's method for the MDMTSP is presented. In Section 1, we illustrate the procedure for finding a lower bound and the effectiveness of the lower bound using branch and bound procedure for various cases of the Euclidean MDMTSP. In Section 3, this method will be applied to a motion planning problem for mobile robots. Once the sequence of the vertices to be visited is known for each vehicle, the dynamic programming technique is used to compute the optimal heading for the vehicle at each vertex. Numerical results corroborating the efficacy of the proposed procedures are also included in Section 3.

1. Computation of a lower bound for the MDMTSP

An efficient scheme for obtaining a tight lower bound for the MDMTSP is useful for two reasons - first, it can be used in a B&B procedure and secondly, it can be used to evaluate the quality of approximate solutions obtained by various heuristics. In this section, the method of Held and Karp [13] will be generalized for the MDMTSP. It is known that every combinatorial problem admits multiple integer programming formulations, each reflecting the structure of the problem in a different way. Even the TSP has at least three different formulations: the integer linear programming formulations of Dantzig, Fulkerson and Johnson and that of Miller, Tucker and Zemlin. The former has exponential (in the size of the targets) number of constraints while the latter has only a polynomial number of constraints; a third one is a semi-definite programming formulation recently proposed by Cezik and Iyengar [30]. It is known, in the literature, [31] that the polytope corresponding to Dantzig-Fulkerson-Johnson's formulation [14] is contained in that of the polytope from

Miller-Tucker-Zemlin's formulation [32]. The Cezik and Iyengar's semi-definite programming formulation [30] has not been followed because Goemans [33] pointed out that the connectivity requirements become weak in the semi-definite programming formulation as the size of the vertices in the graph increase; moreover, the bounds obtained by relaxing the integrality constraints asymptotically tend to the bounds obtained by an assignment problem. We adopt a generalization of Dantzig-Fulkerson-Johnson formulation and Held-Karp's procedure for the problem at hand.

For purposes of notation, let the set of depots by D, the set of targets (cities) to be visited by T and the set of vertices (nodes), $V = D \cup T$. The cardinality of the set D is m and that of T is n. The set of edges between the nodes is represented by E. Let the $\delta(X),\ X \subset V$ to denote the set of edges with exactly one end in X and $E(X),\ X \subset V$ to indicate the set of edges with both ends in X. Let $x_e,\ e \in E$ and $y_v,\ v \in D$ to be the binary variables that respectively represent the choice of the edge and the depot in the solution. We will let $c_e,\ e \in E$ to denote the cost of an edge e. In this chapter we assume that the edges satisfy triangle inequality as they represent distances between vertices. This is very crucial in determining that the below binary program produces an optimal solution that corresponds to an optimal solution of the MDMTSP:

$$J = \min \sum_{e \in E} c_e x_e, \tag{2.1}$$

subject to

$$\sum_{e \in \delta(v) \cap \delta(D)} x_e = 2y_v, \;\; v \in D \tag{2.2}$$

$$\sum_{e \in \delta(v)} x_e = 2, \;\; v \in T, \tag{2.3}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \; \forall \, S \subset T, \tag{2.4}$$

$$\sum_{e \in E(T)} x_e + \sum_{y \in D} y_v = n, \tag{2.5}$$

$$\sum_{v \in D} y_v \leq q, \tag{2.6}$$

$$x_e \in \{0, 1\}, \; y_v \in \{0, 1\}. \tag{2.7}$$

Constraints (2.2) and (2.3) represent the degree constraints of the depots and target cities respectively. In particular, if a depot is not chosen, then no edge incident on the depot can be chosen from the solution as constrained by (2.2). The constraint (2.32) indicates that if the depots and the edges incident on the depots were to be removed from the solution, the resulting graph will be acyclic; such constraints were used in the Linear Programming formulation of a Minimum Spanning Tree (MST) by Edmonds [11]. The constraint (2.33) indicates that if p depots were chosen in the solution, then the graph satisfying (2.32) must have exactly p components. The constraint (2.34) indicates that there be at most q depots be chosen in any solution. The discussion below shows that the above binary program represents the MDMTSP problem at hand:

Every feasible solution to MDMTSP requires a choice of at most q vehicles and a tour associated with each vehicle has at least two target cities. Therefore, every feasible solution satisfies the constraints (2.2) through (2.35). Hence, $J \leq MDMTSP^{*}$, the optimal cost of MDMTSP.

Consider an optimal solution to the binary program. Since the degree of every selected depot vertex and the target city vertex is 2, the optimal solution must represent a union of cycles and isolated depots. Clearly, the constraint (2.32) does not admit a cyclic solution amongst the target cities and hence, it must be the case that every cycle of an optimal solution to MDMTSP must contain at least one depot vertex. It cannot have more than one depot vertex; otherwise, using triangle inequality, additional depot vertices can be short cut to produce a solution to MDMTSP with a smaller cost than the optimal solution. Since the optimal solution to the binary program is a feasible solution to MDMTSP, it can be concluded that $J \geq MDMTSP^*$ and hence $J = MDMTSP^*$.

The binary program considered for MDMTSP is hard to solve and is analogous to the DFJ formulation of the TSP. Held-Karp's method dualizes the DFJ formulation by relaxing the degree constraint on the target cities. In the same spirit, one may include the degree constraint on the depots and relax the constraint on the city vertices. Doing so, one gets a relaxed binary linear programming which can be computed as shown in the Lemma 1 that follows and is a lower bound for $MDMTSP^*$.

When the violation of degree constraint (2.3) is penalized, one has a penalty variable, $\pi_v,\ v \in T$. Let Π be the vector of penalty variables, with $\pi_v \equiv 0,\ v \in D$. Such a Π is referred to as an admissible Π. One may then express the Lagrangian as:

$$L(\Pi) := \min \sum_{e \in E} c_e x_e + \sum_{v \in T} \pi_v \Big(\sum_{e \in \delta(v)} x_e - 2 \Big), \tag{2.8}$$

subject to constraints (2.2), (2.32) through (2.35). The objective function may be expressed as:

$$L(\Pi) = \sum_{e=(v,w) \in E} \underbrace{(c_e + \pi_v + \pi_w)}_{c_e(\Pi)} x_e - 2 \sum_{v \in T} \pi_v. \tag{2.9}$$

It is clear that every feasible solution of $MDMTSP$ is at least $L(\Pi)$ and $L(\Pi)$ is a lower bound for $MDMTSP^*$. From (2.36), it is clear that $L(\Pi)$ is concave in Π as it is the minimum of a finite number of linear functions in Π. Consider the following lemma:

Lemma 1. *For any given admissible* Π*, consider the binary program given by objective function in (2.9) and subject to constraints (2.2) through (2.35). This program is solvable in polynomial time and the optimal cost is a concave function of the edge costs,* $c_e,\ e \in E$ *and is a lower bound on* $MDMTSP^*$*.*

Proof: It is sufficient to show that the following program is polynomially solvable for every integral p lying between 1 and q:

$$J_p = \min \sum_{e \in E} c_e(\Pi) x_e, \tag{2.10}$$

subject to

$$\sum_{e \in \delta(v) \cap \delta(D)} x_e = 2y_v,\ v \in D, \tag{2.11}$$

$$\sum_{y \in D} y_v = p, \tag{2.12}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1,\ \forall\ S \subset T, \tag{2.13}$$

$$\sum_{e \in E(T)} x_e = n - p, \tag{2.14}$$

$$x_e \in \{0, 1\},\ \ y_v \in \{0, 1\}. \tag{2.15}$$

The solution to the above program can be found using the following algorithm:

1. Compute the minimum spanning forest, MSF_p^* on T with p components.

2. Determine the two cheapest edges incident on every $v \in D$ and let their total cost be t_v.

3. Sort $t_v,\ v \in D$ and find the cheapest p costs and the set, E_p^* of the corresponding $2p$ edges. Let the total cost of the cheapest p edges is C_p^*.

4. The optimal cost of the binary program is $MSF_p^* + C_p^*$. The corresponding optimal solution can be determined by the set of edges in MSF_p^* and the edges E_p^*.

The polynomial running time guarantee of the algorithm is immediate from the steps (1) through (4). The correctness of the algorithm can be seen by rewriting the binary program given as follows:

$$J_p = \min \sum_{e \in E(T)} c_e x_e + \sum_{e \in \delta(D)} c_e x_e, \tag{2.16}$$

subject to

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \; \forall \; S \subset T, \tag{2.17}$$

$$\sum_{e \in E(T)} x_e = n - p, \tag{2.18}$$

$$\sum_{e \in \delta(v) \cap \delta(D)} x_e = 2y_v, \; v \in D, \tag{2.19}$$

$$\sum_{v \in D} y_v = p, \tag{2.20}$$

$$x_e \in \{0, 1\}, \; y_v \in \{0, 1\}. \tag{2.21}$$

Since the variables in constraints (2.17) through (2.18) and (2.19) through (2.20) are separable (i.e., are not coupled) and the objective function is also separable, the minimization of the objective function can be carried out separately. Clearly, step (1) of the algorithm provides a solution for minimization of the objective function over the variables in constraints (2.17) through (2.18), while steps (2) and (3) solve the minimization of the objective function over the variables in constraints (2.19) through (2.20). It is then easy to see that the objective function, J may be computed as $J = \min_p J_p$. $\square$

For every admissible Π, the optimal solution for MDMTSP does not change when the weight of each edge $e = (v, w)$ is modified as $\tilde{c}_e = c_e + \pi_v + \pi_w$. Further, the cost of the optimal solution changes as:

$$MDMTSP^* + 2\sum_{v\in T}\pi_v.$$

By Lemma 1, it is known that $L(\Pi)$ can be computed in polynomial time and

$$MDMTSP^* + 2\sum_{v\in T}\pi_v \geq L(\Pi).$$

Therefore, one has:

$$MDMTSP^* \geq L(\Pi) - 2\sum_{v\in T}\pi_v. \tag{2.22}$$

Since the above inequality holds for all $\Pi : \pi_v \equiv 0,\ v \in D$, one can then maximize the right hand side of the inequality to get a tighter lower bound.

$$MDMTSP^* \geq \max_{\Pi:\pi_v=0,\ v\in D} \underbrace{L(\Pi) - 2\sum_{v\in T}\pi_v}_{\phi(\Pi)},$$

where $\phi(\Pi)$ is a lower bound to MDMTSP, corresponding to the vector of penalty variables, Π. Let $HKLB = \max_{\Pi:\pi_v=0,\ v\in D} \phi(\Pi)$ (right side of the inequality).

2. Numerical results

With the relaxed constraints on the degree of the targets, the dual solution at each iteration may not be a primal feasible one. The primal feasible solution is computed using the p-spanning forest MSF_p^* generated by the dual algorithm. The procedure of assigning the depots to each component of the MSF_p^* and forming the feasible p-tours through modified Lagrangian heuristics is given below.

Primal feasible Algorithm:

1. For each $v \in D$ and i^{th} component of MSF_p^*, $i \in \{1, 2, \ldots, p\}$, the cost, A_{vi} is

computed to be the total cost of the two cheapest edges in $\delta(v) \cap \delta(S_i)$, where S_i is the set of nodes in i^{th} component of MSF_p^*

2. Assign a depot to every component in MSF_p^* such that the total assignment cost is minimum.

3. Let Z_i be the set of edges in i^{th} component of MSF_p^* and let v_i be the assigned vehicle. Define $F_i := Z_i \cup e_i$, where e_i is the cheapest edge in $\delta(v) \cap \delta(S_i)$.

4. On each F_i, use the Lagrangian heuristics in [34] to modify the relaxed solution into a primal feasible one.

At each iteration k, compute a new set of penalty parameter $[\Pi]^{k+1}$ from $[\Pi]^k$ through an update scheme, so that one can get an improved direction of updating dual cost. Since, by relaxing the constraints on the degrees of the targets a non-smooth dual problem is generated, an non-smooth optimization method is employed. A method that works well in practice for optimization problems of this genre is the sub-gradient method. In each iteration, a new set of penalty parameters are generated. The direction of update is defined through the sub-gradient. The sub-gradient can defined as follows:

$$g_v = \sum_{e \in \delta(v)} x_e - 2, \; \forall v \in T$$

$$g_v = 0, \forall v \in D$$

The new update $[\pi_v]^{k+1}$ is computed as follows:

$$[\Pi]^{k+1} = [\Pi]^k + \beta^k [g]^k \quad \forall v$$

where the size of the step, β at iteration k is computed as

$$\beta^k = \zeta^k \frac{MDMTSP^* - \phi([\Pi]^k)}{||[g]^k||} \tag{2.23}$$

where $\phi([\Pi]^k)$ is the value of $\phi(\Pi)$ at the k^{th} iteration. The above expression (2.51) is commonly referred to as *Polyak rule II*. Since, the optimal solution $MDMTSP^*$ is not known, alternatively use the cost of the best primal solution found so far. A common practice is to start ζ^k with a fixed value and reduce ζ^k by a constant factor after a specified number of iterations or whenever $\phi([\Pi]^k)$ does not increase within specified number of iterations. The iterative procedure can be briefly put as follows:

1. Initial step: k = 0, Initialize $\zeta^k = \zeta_0$.
2. For the computed $[\Pi]^k$, solve the relaxed problem.
3. Use **Primal feasible Algorithm** to generate a primal feasible from the dual solution. Let the cost of best primal feasible solution found so far be $[C^*]^k$
4. Stopping criterion: If $\frac{[C^*]^k - \phi([\Pi]^k)}{\phi([\Pi]^k)} \leq \epsilon$ or $k = N^{max}$, then go to 6.
5. Compute $[\Pi]^{k+1}$ and set $k = k + 1$ and go to 2.
6. Stop the iterative process.

In step 4, $\frac{[C^*]^k - \phi([\Pi]^k)}{\phi([\Pi]^k)}$ is the duality gap and ϵ is the desired duality gap. Since, the primal problem is an integer problem, one may not be able to assure a zero duality gap. One can apply the algorithm presented in the previous section to 20 instances. The maximum number of iterations allowed is chosen to be 50. ζ^k was chosen to start with a value of 0.2 and is reduced by a factor of 2, if the dual does not improve in 10 successive iterations. The value of ϵ for the stopping criterion is chosen to be 10^{-4}. For the first 10 iterations, a primal feasible solution is not computed. The results are shown in the following table. All the depots are allowed to participate in the tour, i.e, $q = |D|$.

In Table I, n refers to the number of targets, m is the number of depots available, C_p^* is the best found cost of the generated primal feasible solution, $\phi([\Pi]^k)^*$ is the best dual cost

Table I. Computational results for various instances of MDMTSP

$\%$ of Duality Gap in k Iterations

n	m	$\phi([\Pi]^k)^*$	k^*_{dual}	C^*	$MDMTSP^*$	$k = 25$	$k = 50$
15	3	2753	24	2753	2753	‡	‡
18	4	2651	49	2662	2653	0.8678	0.4892
19	6	2822	40	2840	2840	0.8228	0.6738
20	4	2430	50	2455	2455	1.4389	1.0409
20	7	1301	11	1301	1301	‡	‡
24	6	2172	46	2247	2217	3.5627	3.4671
25	5	3115	50	3118	3118	0.1032	0.0992
26	5	2583	47	2592	2592	0.5577	0.4047
28	6	3000	50	3052	3000	1.7373	1.7373
30	3	2599	40	2681	2659	4.1973	3.1605
33	6	3173	50	3173	3173	0.1144	0.0113
35	3	3313	41	3400	3330	2.8272	2.6455
37	3	3483	50	3484	3484	0.1677	0.0453
40	5	3459	47	3462	3462	0.1419	0.1051
40	7	3488	49	3587	3518	4.4790	2.2483
40	8	3776	46	3834	3814	2.5343	1.5862
44	5	3369	47	3464	3409	3.8615	2.8482
44	6	3031	40	3068	3031	1.3789	1.2578
45	5	3681	50	3696	3696	0.5226	0.4399
45	10	3409	43	3452	3410	1.6487	1.2555

‡ indicates that the stopping criterion is met before reaching that step

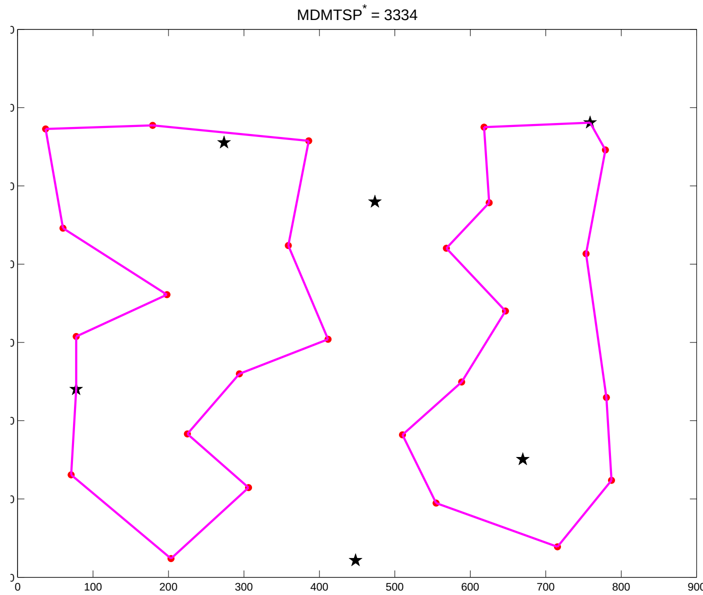

Fig. 1. The optimal solution of MDMTSP generated through YALMIP.

computed, k^*_{dual} is the iteration at which the best dual occurred and $MDMTSP^*$ refers to the optimal cost for that instance. For computing MDMTSP*, the integer program is solved using GLPK (GNU Linear Programming Kit). The code is written in Matlab and YALMIP [35] is used to formulate the problem and also provides the interface to GLPK.

An illustration of this method using $m = 6$, $n = 25$ is shown in Fig. 1. The dual solution at the end of 50 iterations can be seen in Fig. 2. The black stars correspond to the vehicles. There are two tours in the approximation, i.e., $p = 2$. These are shown in Fig. 3 using solid lines. The total cost of the 2 tours is 3334 units and is guaranteed to be less than twice the optimal cost (2 $MDMTSP^*$). $MDMTSP^*$ computed by solving the integer programming through Yalmip gives a cost of 3334 units, which is same as the primal cost at the end of 50 iterations in this instance. The cost of the dual solution generated is 3332 units and the duality gap is 0.06%.

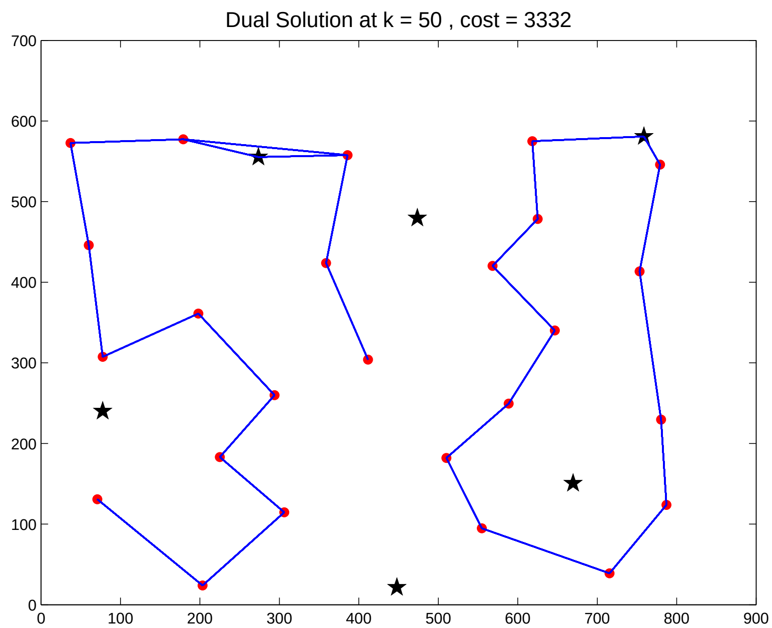

Fig. 2. Dual Solution generated at $k = 50$

3. Determination of optimal heading angles for visiting targets in a given sequence

If the sequence of targets to be visited is specified a priori for a Dubins' vehicle, the problem of determination of optimal heading angles at each target can be solved through Dynamic Programming (DP). A method for the determination of the sequence of targets to be visited, by each vehicle, based on Euclidean distances between targets was presented in the previous section. A sub-optimal solution is settled upon in order to decouple the continuous optimization problem of determining optimal heading angles at each vertex in T from the combinatorial problem. The sub-optimal solution is obtained by carrying over the solution to the combinatorial problem from the previous section and require that the Dubins' vehicle visit the specified vertices in the specified order.

Let the sequence of targets to be visited by a chosen vehicle $v \in D$ be $\{1, 2, \dots, k\}$

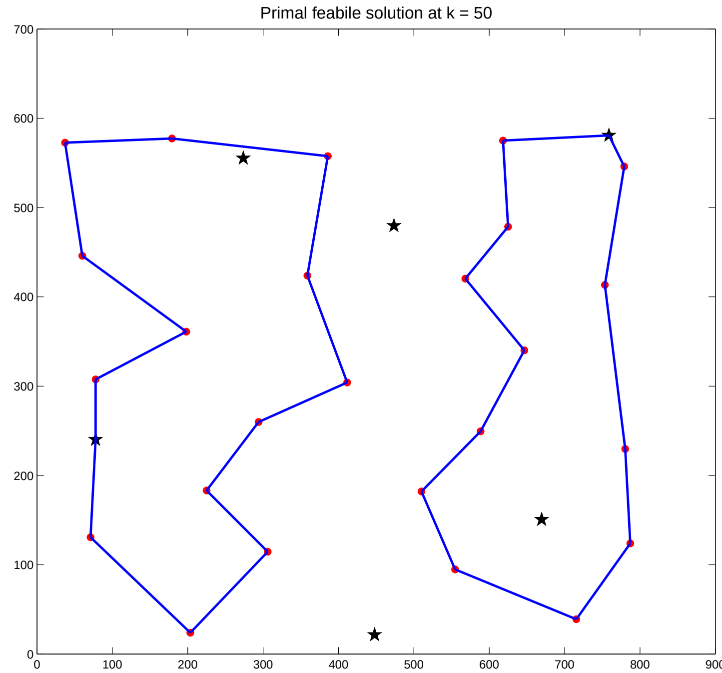

Fig. 3. Tours generated through primal heuristics at k = 50

[3]. Let Θ_i be the set of heading angles allowed for the Dubins' vehicle at the i^{th} target. Let θ_0 be the heading angle of the v at its starting location. If Θ_i contains only one element, it implies that the heading is specified at the i^{th} target. For the sake of converting a tour to a path, let a fictitious $(k+1)^{st}$ target be located at the same place as v is initially located and has the same heading angle as the initial heading angle of v. Let $d_{ij}(\theta_i, \theta_j)$ denote the shortest path from the i^{th} target to the j^{th} target. Such a distance can be computed efficiently using the result of Dubins [6]. Assume that θ_0 is known and thus the problem

[3]The primal solution generates a tour for each chosen vehicle. Since, there is no sense of directionality associated with the tour generated, $\{k, k-1, \ldots, 2, 1\}$ is also a valid sequence. The dynamic programming is detailed for the sequence $\{1, 2, \ldots, k\}$. However, the same process needs to be repeated for the reverse sequence and the minimum of the two costs provides the best approximate tour for the chosen vehicle with motion constraints.

can be posed as:

$$d^* = \min_{\theta_i \in \Theta_i,\, i=1,\ldots,n+1} \sum_{i=0}^{n} d_{i,i+1}(\theta_i, \theta_{i+1}). \tag{2.24}$$

To obtain the DP recursion equations:

$$d^*_{i,i+1}(\theta_i, \theta_{i+1}) := d_{i,i+1}(\theta_i, \theta_{i+1}),\ i = 0, \ldots, k, \tag{2.25}$$

and for $l \geq 2$ and $0 \leq i \leq k+1-l$,

$$d^*_{i,i+l}(\theta_i, \theta_{i+l}) \quad := \quad \min_{\theta_{i+l-1} \in \Theta_{i+l-1}} d^*_{i,i+l-1}(\theta_i, \theta_{i+l-1}) \ + \ d^*_{i+l-1,i+l}(\theta_{i+l-1}, \theta_{i+l}). \tag{2.26}$$

Using this recursion, one can compute $d^* = d^*_{0,k+1}(\theta_0, \theta_0)$ and the corresponding optimal arguments $\theta^*_1, \theta^*_2, \ldots, \theta^*_k$.

a. Implementation

If the set Θ_i is an interval, one may partition it with N heading angles, $\theta_i^k,\ k = 1, \ldots, N$, in the partition. Assume that N is at least the size of any discrete set, (say Θ_p), one may have. One can now construct a graph with at most $O(nN)$ vertices with each vertex corresponding to a target that may be visited at a discretized value of Θ_i. One can now construct edges between vertices as follows: There are edges only between vertices that correspond to targets that must be successively visited. For this reason, let $d_{i,j}^{k,l}$ denote the distance $d_{i,j}(\theta_i^k, \theta_j^l)$. The source node corresponds to the location of v with its specified initial heading and the terminal node again corresponds to the same. The problem of determining the optimal heading angles is posed as determining the shortest path from the source to the terminal. Since edges exist only between targets that must be successively visited, the shortest distance will not be zero, as there is no edge between the source and the terminal. Using (2.26), one can show that the time complexity of the algorithm is $O(nN^2)$.

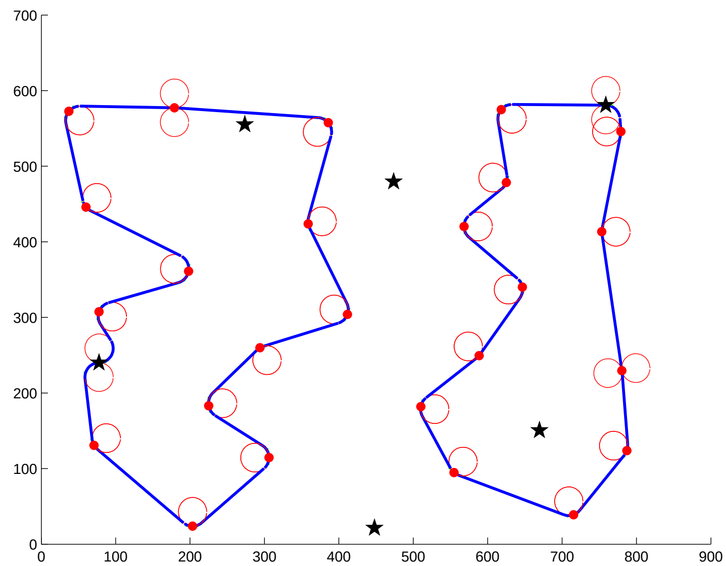

Fig. 4. Tours for each vehicle imposing motion constraints

b. Numerical results

Consider the example in Section 1. Fig. 3 chooses 2 vehicles and constructs a tour for each of them, and hence the sequence of targets to be visited by each used vehicle is specified. Taking into consideration, the motion constraints given in [6] for a Dubins' vehicle and compute sub-optimal heading angles at each target through the Dynamic Programming (DP) procedure detailed above. Assume that the initial (and final) heading of each vehicle to be $0°$. The sub-optimal tours generated is shown in Figure 4. The total cost of the sub-optimal is 3549.7 units. It can be easily calculated *a posteriori* that the sub-optimal Dubins' cost is within $\frac{3549.7}{3332} = 1.0653$ of the optimal Dubins' cost.

B. Extension to asymmetrical variant of the problem

The problem of motion planning involves the solution of a combinatorial problem, wherein one must determine the set of targets to be visited by each vehicle and the sequence in which they must be visited before returning to its initial location (depot). Equally important is the consideration of motion constraints of the vehicles in the planning. In this section, a combinatorial motion planning problem involving a homogeneous collection of vehicles where the motion of each vehicle satisfies a non-holonomic constraint is addressed. The non-holonomic constraint considered is that the yaw rate of the vehicle at any time is upper bounded by a constant. Hence, if the vehicle is traveling at constant speed, this constraint is equivalent to a lower bound on the turning radius of the vehicle. The combinatorial motion planning problem (CMP) is as following:

Given a set of m vehicles and n targets on a plane, the heading angles of each target and the initial heading angles of each vehicle, the CMP is to

- choose at most $p(\leq m)$ vehicles,
- assign a set of targets for each chosen vehicle such that each target is visited exactly once,
- find a feasible path (i.e. a path that satisfies the yaw rate constraints) for each chosen vehicle such that the vehicle starts at its initial position, visits its assigned set of targets at their respective heading angles in a specified sequence and returns to its initial position.

The goal is to minimize the sum of the distances traveled by all the chosen vehicles.

The problem of finding the minimum distance path the vehicle must take between any two positions on a plane subject to the constraints on the yaw rate has been solved by Dubins [6]. Hence, the CMP can be posed as a multiple depot Asymmetric Traveling

Salesman Problem (ATSP). This problem is a generalization of the single TSP and is NP-Hard. The difficulty of this CMP is due to the following reasons:

1. The vehicle-target assignment is not given.

2. Given the vehicle-target assignment, finding the optimal sequence for each vehicle is again a single depot ATSP which is hard. Several approximation algorithms and heuristics that work well for the single symmetric TSP does not work well for single depot ATSP [36].

Reference [37] provides an extensive review of the solution procedures for the multiple Traveling Salesman Problem. As previously mentioned in the introduction, CMP is NP-Hard. Unlike the symmetric counterparts that have constant factor approximation[4] algorithms [26, 27], the best approximation algorithms available *even* for a single depot ATSP have approximation ratios scale in the order of $\log(n)$ [38, 39]. One way to address a CMP is to convert to the CMP into a single ATSP and use the algorithms available for ATSP to solve CMP. But this is currently available only for $m = 2$ [25]. For a general m, Laporte gives a transformation of CMP to a constrained assignment problem. As mentioned in [40], [37], it is an incomplete transformation due to the presence of non assignment constraints.

Branch and Bound methods can be used to solve CMP [31]. In general the effectiveness of a B&B procedure depends on the tightness of the lower and upper bounds that one has at hand. In this chapter, tight lower bounds for CMP are generated using Lagrangian Relaxation. This generalizes the results by Held-Karp [13] available for the single TSP for the CMP.

It will be assumed that the heading of each target is known. This allows one to view CMP purely as a combinatorial problem using Dubins [6] result. The CMP without this

[4]A polynomial algorithm that returns an approximate solution whose cost is within a guaranteed factor of the optimal solution.

assumption has also received significant attention in the literature [26, 29, 41, 42]. Though motion constraints are an integral part of all these variants of the CMP, it is hard to envision good algorithms or heuristics for the same that does not exploit the combinatorial structure of the problem.

1. Asymmetrical variant: Problem formulation

Let D represent the set of depots (initial locations of vehicles), T represent the set of targets and let $V = D \cup T$. The cardinality of D is m and that of T is n. The set of all the edges connecting any two vertices in V is represented by E. An arc $e = (x, y)$ is considered to be directed from x to y. y is called the head and x is called the tail of the arc. Let c_e be the cost of arc e. Basically, c_e is the length of the Dubins path from vertex x to vertex y. Note that the costs, c_e, satisfy triangle inequality. Let $\delta(A)$ to indicate the set of edges with their tails in A, $\Delta(A)$ to indicate the set of edges with their heads in A and $E(X), X \subset V$ to indicate the set of edges with both their heads and tails in X. Let $x_e,\ e \in E$ and $y_v,\ v \in D$ to be the binary variables that respectively represent the choice of the edge and the depot in the solution. The **integer program** for the CMP is formulated as follows:

$$CMP^* \quad = \quad \min \sum_{e \in E} c_e x_e, \tag{2.27}$$

subject to

$$\sum_{e\in\delta(v)\bigcap\Delta(T)} x_e = y_v, v \in D \tag{2.28}$$

$$\sum_{e\in\Delta(v)\bigcap\delta(T)} x_e = y_v, v \in D \tag{2.29}$$

$$\sum_{e\in\delta(v)} x_e = 1, v \in T. \tag{2.30}$$

$$\sum_{e\in\Delta(v)} x_e = 1, v \in T. \tag{2.31}$$

$$\sum_{e\in E(S)} x_e \leq |S|-1, \forall S \subset T, \tag{2.32}$$

$$\sum_{e\in E(T)} x_e + \sum_{v\in D} y_v = n, \tag{2.33}$$

$$\sum_{v\in D} y_v \leq q, \tag{2.34}$$

$$x_e \in \{0,1\},\ y_v \in \{0,1\}. \tag{2.35}$$

Constraints (2.28) and (2.29) represent the out-degree and the in-degree constraints on the depots respectively. In particular, if a depot is not chosen, then no edge incident on the depot(incoming or outgoing) can be chosen from the solution as stated by (2.28) and (2.29). Constraints (2.31) and (2.30) require the in-degree and out-degree of each target equal to one. The constraint (2.32) eliminates the presence of any cycles among the target vertices. Constraint (2.33) indicates that if p depots were chosen in the solution, then the graph $(T, E(T))$ must have exactly p components. Constraint (2.34) requires that any feasible solution must choose at most q depots. As one can notice, Objective function and constraints (2.322.332.34) are also present in symmetric part of the problem as they hold in this variant too.

Proposition 1. *The integer program for the CMP is* valid *(i.e. the optimal solution of the integer program is an optimal solution to the CMP) if the costs, c_e, satisfy triangle inequality.*

Proof. Every feasible solution to the CMP satisfies the constraints (2.28) through (2.35). Now, consider an optimal solution to the integer program. Since the in-degree and the out-degree of every selected depot vertex and the target vertex is 1, the optimal solution must represent a union of cycles and isolated depots. Clearly, the constraint (2.32) does not admit a cyclic solution amongst the target cities and hence, it must be the case that every cycle of an optimal solution to CMP must contain at least one depot vertex. It cannot have more than one depot vertex; otherwise, using triangle inequality, additional depot vertices can be short cut to produce a solution to CMP with a smaller cost than the optimal solution. Since the optimal solution to the binary program is a feasible solution to CMP, the integer program formulated for the CMP is correct. □

2. A Lagrangian relaxation of the CMP

In this section, w tight lower bounds are obtained for the integer program stated in the previous section. In later sections, the results in this section are used to develop a heuristic for the CMP. The method here (Lagrangian Relaxation) follows the approach by Held and Karp who used it for solving the symmetric TSP [13]. The basic idea in Lagrangian Relaxation is to first identify the constraints that make the integer program difficult to solve. Then, remove these complicating constraints and penalize them in the objective whenever they are violated. A Lagrangian Relaxation of the integer program for CMP is:

$$\begin{aligned} L(\Pi, \Psi) := \min \sum_{e \in E} c_e x_e + \sum_{v \in T} \pi_v \Big(\sum_{e \in \delta(v)} x_e - 1 \Big) + \\ \sum_{v \in T} \psi_v \Big(\sum_{e \in \Delta(v)} x_e - 1 \Big) \end{aligned} \tag{2.36}$$

subject to

$$
\begin{aligned}
\sum_{e\in\delta(v)\bigcap\Delta(T)} x_e &= y_v, v \in D \\
\sum_{e\in\Delta(v)\bigcap\delta(T)} x_e &= y_v, v \in D \\
\sum_{e\in E(S)} x_e &\leq |S| - 1, \forall S \subset T, \\
\sum_{e\in E(T)} x_e + \sum_{v\in D} y_v &= n, \\
\sum_{v\in D} y_v &\leq q, \\
x_e &\in \{0,1\},\ y_v \in \{0,1\}.
\end{aligned}
$$

where, π_v (ψ_v) is the penalty variable when the out-degree (in-degree) constraint of a target vertex v is violated and Π (Ψ) indicates the vector of penalty variables π_v (ψ_v). The following lemma shows that $L(\Pi, \Psi)$ can be computed using a polynomial time algorithm. Hence, for any given Π and Ψ, computing $L(\Pi, \Psi)$ would yield a lower bound for CMP^*.

Lemma 2. *For any given* Π*,*Ψ*, the Lagrangian Relaxation in (2.36) is solvable in polynomial time.*

Proof. It is sufficient to show that the following program is polynomially solvable for every integer p lying between 1 and q:

$$
\begin{aligned}
J_p(\Pi, \Psi) := \min \sum_{e\in E} c_e x_e + \sum_{v\in T} \pi_v \Big(\sum_{e\in\delta(v)} x_e - 1\Big) + \\
\sum_{v\in T} \psi_v \Big(\sum_{e\in\Delta(v)} x_e - 1\Big),
\end{aligned}
$$

subject to

$$\begin{aligned}
\sum_{e\in\delta(v)\bigcap\Delta(T)} x_e &= y_v, v\in D && (2.37)\\
\sum_{e\in\Delta(v)\bigcap\delta(T)} x_e &= y_v, v\in D && (2.38)\\
\sum_{v\in D} y_v &= p, && (2.39)\\
\sum_{e\in E(S)} x_e &\le |S|-1, \forall S\subset T, && (2.40)\\
\sum_{e\in E(T)} x_e &= n-p, && (2.41)\\
x_e &\in \{0,1\},\ y_v\in\{0,1\}. && (2.42)
\end{aligned}$$

Observe that the variables in constraints (2.37,2.38,2.39), $\{y_v : v\in D\}, \{x_e : e\in \delta(D)\bigcup\Delta(D)\}$, and the variables in constraints (2.40,2.41), $\{x_e : e\in E(T)\}$, are not coupled. Hence the Lagrangian Relaxation can be decoupled into two problems and can be solved separately as follows:

Problem I:

$$\begin{aligned}
J_p^1(\Pi,\Psi) := \min \sum_{e\in E(T)} c_e x_e + \sum_{v\in T}\pi_v \sum_{e\in\delta(v)\bigcap E(T)} x_e +\\
\sum_{v\in T}\psi_v \sum_{e\in\Delta(v)\bigcap E(T)} x_e,
\end{aligned}$$

subject to

$$\begin{aligned}
\sum_{e\in E(S)} x_e &\le |S|-1, \forall S\subset T,\\
\sum_{e\in E(T)} x_e &= n-p,\\
x_e &\in \{0,1\},\ y_v\in\{0,1\}.
\end{aligned}$$

Problem II:

$$\begin{aligned} J_p^2(\Pi, \Psi) := \min \sum_{e \in E \setminus E(T)} c_e x_e + \sum_{v \in T} \pi_v \Big(\sum_{e \in \delta(v) \bigcap \Delta(D)} x_e - 1 \Big) + \\ \sum_{v \in T} \psi_v \Big(\sum_{e \in \Delta(v) \bigcap \delta(D)} x_e - 1 \Big), \end{aligned}$$

subject to

$$\begin{aligned} \sum_{e \in \delta(v) \bigcap \Delta(T)} x_e &= y_v, v \in D \\ \sum_{e \in \Delta(v) \bigcap \delta(T)} x_e &= y_v, v \in D \\ \sum_{v \in D} y_v &= p, \\ x_e &\in \{0, 1\}, \ y_v \in \{0, 1\}. \end{aligned}$$

Problem I involves computing a minimum cost, p-component, directed spanning forest ($DMSF_p^*$) that can be solved using a polynomial time algorithm given in the appendix. The solution to **problem II** can be found using the following steps:

1. Let the modified cost of each edge e in $\delta(T)$ ($\Delta(T)$) be $c_e + \pi_{v:e \in \delta(v)}$ ($c_e + \psi_{v:e \in \Delta(v)}$). Determine the cheapest incoming edge and outgoing edge incident on every $v \in D$. Let their total cost be t_v.

2. Sort $t_v, \ v \in D$. The optimal solution, E_p^*, is the set of $2p$ edges corresponding to the p cheapest costs.

The optimal cost of the Lagrangian Relaxation, $L(\Pi, \Psi)$, can be computed as $L(\Pi, \Psi) = \min_p (J_p^1(\Pi, \Psi) + J_p^2(\Pi, \Psi))$. $\square$

Now, since for every Π, Ψ, $CMP^* \geq L(\Pi, \Psi)$, one can conclude that

$$CMP^* \geq \max_{\Pi,\Psi} L(\Pi,\Psi). \tag{2.43}$$

$\max_{\Pi,\Psi} L(\Pi,\Psi)$ is the Lagrangian Dual of the integer program for CMP. Note that $L(\Pi,\Psi)$ is a concave function of Π and Ψ. Details on how to solve this Lagrangian Dual are given in the following section.

3. Computing a constrained, directed spanning forest

Add a root vertex r and join r to each of the vertices in T with a zero cost edge. Now, the problem of finding the minimum cost, p-component directed spanning forest can be posed as a problem of finding the minimum cost, directed spanning tree with a degree constraint on the root vertex as follows:

$$\min \sum_{e \in E(T \bigcup \{r\})} c_e x_e, \tag{2.44}$$

subject to

$$\begin{aligned}
\sum_{e \in \delta(\{r\})} x_e &= p && (2.45)\\
\sum_{e \in E(S)} x_e &\leq |S| - 1,\ \forall\, S \subset T \bigcup \{r\}, && (2.46)\\
\sum_{e \in E(T)} x_e &= n - p, && (2.47)\\
x_e &\in \{0,1\}. && (2.48)\\
& && (2.49)
\end{aligned}$$

Removing the zero cost edges from the optimal solution to the above problem would yield the desired minimum cost forest. Consider the following Lagrangian relaxation of the above problem:

$$L(z) = \min_x \sum_{e \in E(T \bigcup \{r\})} c_e x_e + z(\sum_{e \in \delta(\{r\})} x_e - p) \qquad (2.50)$$

subject to

$$\begin{aligned}
\sum_{e \in E(S)} x_e &\leq |S| - 1, \; \forall \, S \subset T \bigcup \{r\}, \\
\sum_{e \in E(T)} x_e &= n - p, \\
x_e &\in \{0, 1\}.
\end{aligned}$$

Let (z^*) solve the Lagrangian dual $\max_z L(z)$. If x^* is the unique optimal solution that solves the minimization problem in $L(z^*)$, then using the results in [43],[27] it can conclude that x^* also satisfies the complicating constraint. Perturb the cost of the edges so that, in practice, one needs to find a unique optimal solution x^*. So, the algorithm can be used to find the degree constrained spanning tree is as follows:

a. Directed spanning forest algorithm

1. Perturb the cost of each edge c_e to $\widetilde{c}_e = c_e + u_e$, where $\{u_e : e \in E(T \bigcup \{r\})\}$ represent independent, uniform random variables chosen in the interval[5] $[0, \frac{1}{2(n+1)}]$.

2. Solve the Lagrangian dual problem (2.50) corresponding to cost $\widetilde{c}$. The solution to the Lagrangian dual problem is the desired optimal solution to problem (2.44) with probability one.

[5]Assume c_e for all $e \in E(T \bigcup \{r\})$ are integers. If c_e are rational numbers one can always multiply them by appropriate constants to make them integers.

The following part of the section gives a simple proof as to why the Lagrangian dual problem must have a unique optimal solution with probability one. Specifically, Proposition 2 states why there should be a unique feasible solution and proposition 3 shows why the unique feasible solution is also optimal.

Let x_1 and x_2 be any two feasible solutions that satisfy the constraints in 2.47 and 2.48. Let $cost(x, c) = \sum_{e \in E(T \bigcup \{r\})} c_e x_e$.

Proposition 2. *Let* $\boldsymbol{P}(cost(x_1, c+u) = cost(x_2, c+u))$ *indicate the probability that the solutions* x_1 *and* x_2 *have the same cost. Then,* $\boldsymbol{P}(cost(x_1, c+u) = cost(x_2, c+u)) = 0$.

Let S_c^* be the set of all the optimal solutions that solve the minimization problem in (2.44) corresponding to the cost function c_e.

Proposition 3. *For all* $e \in E(T \bigcup \{r\})$*, let* a_e *be any constant in the interval* $[0, \frac{1}{2(n+1)}]$*. Then* $S^*_{(c+a)} \subseteq S^*_c$.

Proof. Consider a solution $x^1 \notin S_c^*$ and any $x^* \in S_c^*$. Since all c_e are integers, $cost(x^1, c) - cost(x^*, c) \geq 1$. If all the edges corresponding to x^* are perturbed from c_e to $c_e + a_e$, then $cost(x^*, c+a) \leq cost(x^*, c) + \frac{n}{2(n+1)} < cost(x^*, c) + 1$. Hence $cost(x^1, c+a) > cost(x^*, c+a)$. Therefore, $S^*_{(c+a)} \subseteq S^*_c$. □

Table II presents the convergence results of this randomized algorithm for computing the minimum cost, directed spanning forest. In Table II, n refers to the number of targets, p^* refers to the desired number of components and i^* is the number of iterations required to compute the optimal directed tree.

4. Primal feasible algorithm for CMP

To generate a feasible solution, the p-directed spanning forest $DMSF_p^*$ resulting through the Lagrangian relaxation is used. The primal algorithm that assigns the depots to each component of the $DMSF_p^*$ and forms the feasible p-directed tours is given below:

1. For each $v \in D$ and i^{th} component of $DMSF_p^*$, $i \in \{1, 2, \ldots, p\}$, compute the cost, A_{vi} to be the total cost of the cheapest edge in $\delta(v) \cap \Delta(S_i)$ and the cheapest edge in $\Delta(v) \cap \delta(S_i)$, where S_i is the set of nodes in i^{th} component of $DMSF_p^*$.

2. Let v_i be the depot assigned to component corresponding to set of nodes, S_i. Define $V_i = S_i \cup v_i$. Assign a depot to every component in $DMSF_p^*$ such that the total assignment cost $\min_{v_i} \sum_i A_{v_i i}$ is minimum.

3. The problem of finding a directed, feasible tour with nodes in V_i is transformed to a problem of finding feasible tour with symmetric costs by doubling the nodes in V_i as described in ([36]). The transformation can be simply put as follows: Each node n is replaced by a pair of nodes n^+, n^- and the define the costs as follows: Let $n_1, n_2 \in S_i$ then $\tilde{c}_i({n_1}^+, {n_2}^-) = c(n_1, n_2)$ and $\tilde{c}_i({n_2}^+, {n_1}^-) = c(n_2, n_1)$. We also set $\tilde{c}_i({n_1}^-, {n_1}^+) = -M$ and all the other costs in $\tilde{c}_i$ to be $+M$, where M is a sufficiently large positive number such that all the arcs whose costs are $+M$ are excluded from all the feasible tours and all the arcs with $-M$ are included in any feasible tour.

4. Now for each modified cost matrix c_i and the node set S_i, the Lagrangian heuristic in [34] is used to get a primal feasible tour.

5. Experimental results

In this section, the implementation details and the overall algorithm accompanied with the simulation results are presented. To calculate the best lower bound discussed in section 2, $\max\limits_{\Pi,\Psi} L(\Pi, \Psi)$ is computed using a gradient ascent algorithm. Let $[\Pi]^k$ and $[\Psi]^k$ indicate the values of Π and Ψ at the k^{th} iteration respectively. At each iteration k, compute a new set of penalty parameters ,$[\Pi]^{k+1}$, $[\Psi]^{k+1}$, from $[\Pi]^k$, $[\Psi]^k$ respectively through an update scheme where the direction of update is defined through the sub-gradient. The sub-gradient as follows:

$$\begin{aligned} gi_v &= \sum_{e \in \Delta(v)} x_e - 1, \forall v \in T \\ go_v &= \sum_{e \in \delta(v)} x_e - 1, \forall v \in T \\ gi_v &= 0, \forall v \in D \\ go_v &= 0, \forall v \in D \end{aligned}$$

Let $g = [gi \quad go]$ be the vector of all the sub-gradients stacked together. The new update $[\pi_v]^{k+1}$ is computed as follows:

$$[\Pi]^{k+1} = [\Pi]^k + \beta^k [go]^k \quad \forall v$$

where the size of the step, β at iteration k is computed as

$$\beta^k = \zeta^k \frac{MDMTSP^* - \phi([\Pi, \Psi]^k)}{||[g]^k||} \tag{2.51}$$

$[\Psi]^{k+1}$ can be computed in the similar fashion as $[\Pi]^{k+1}$. The above expression (2.51) is commonly referred to as *Polyak rule II*. Since, the optimal solution CMP^* is not known, alternatively one can use the cost of the best primal solution found so far. A common practice is to start ζ^k with a fixed value and reduce ζ^k by a constant factor after a specified number of iterations or whenever $\phi([\Pi]^k, [\Psi]^k)$ does not increase within specified number of iterations. The iterative procedure can be briefly put as follows:

1. Initial step: k = 0, Initialize $\zeta^k = \zeta_0$.

2. For the computed $[\Pi]^k$ and $[\Psi]^k$, solve the Lagrangian relaxation $L([\Pi]^k, [\Psi]^k)$.

3. Use the **Primal feasible Algorithm** to generate a primal feasible solution from the dual solution. Let the cost of the best primal feasible solution found so far be $[C^*]^k$.

4. Stopping criterion: If $[\epsilon]^k \leq \epsilon^*$ or $k = N^{max}$, go to 6.

5. Compute $[\Pi]^{k+1}$,$[\Psi]^{k+1}$ and set $k = k + 1$ and go to 2.

6. Stop the iterative process.

where $[\epsilon]^k$ is the duality gap at iteration k and is defined as $\frac{[C^*]^k - \phi([\Pi]^k, [\Psi]^{k+1})}{\phi([\Pi]^k, [\Psi]^{k+1})}$. ϵ^* is the desired duality gap. The maximum number of iterations allowed is chosen to be 50. ζ^k was chosen to start with a value of 0.5 and is reduced by a factor of 2, if the dual does not improve in 3 successive iterations. The value of ϵ for the stopping criterion is chosen to be 10^{-4}. In the simulations, depots are allowed to participate in the tour, i.e, $q = |D|$.

In Table III and IV, n refers to the number of targets, m is the number of depots available, $[Cprimal^*]^k$ is the cost of the best primal found at iteration k. In Table III and IV, the dual gap at iterations $k = 25$ and $k = 50$ is reported respectively. CMP^* refers to the optimal cost for that instance. CMP^* is computed using the GNU Linear Programming Kit (GLPK). The code is written in Matlab and YALMIP [35] is used to formulate the problem and also provides the interface to GLPK. In Figure 5 the convergence of the dual gap with the number of iterations is shown for few random instances. The sizes of the instances are as indicated.

In Figure 6 the optimal solution generated by YALMIP for a random instance with 18 cities and 6 vehicles is shown . The red dots denote the location of all the cities. The black stars show the depot for each vehicle. All the points are randomly generated on a region of dimensions $5km$ x $5km$. Each vehicle is assumed to behave like a Dubin's car. The turning radius of each vehicle is considered to be $100m$. The heading angles at all cities and depot are assumed to be known apriori. In Figure 7 the dual solution generated through the procedure detailed above is shown. In Figure 8 the solution generated through the primal feasible algorithm is shown. The extension of Lagrangian method for Asymmetric problem is published in [44] and similar approach is extended to a problem with precedence

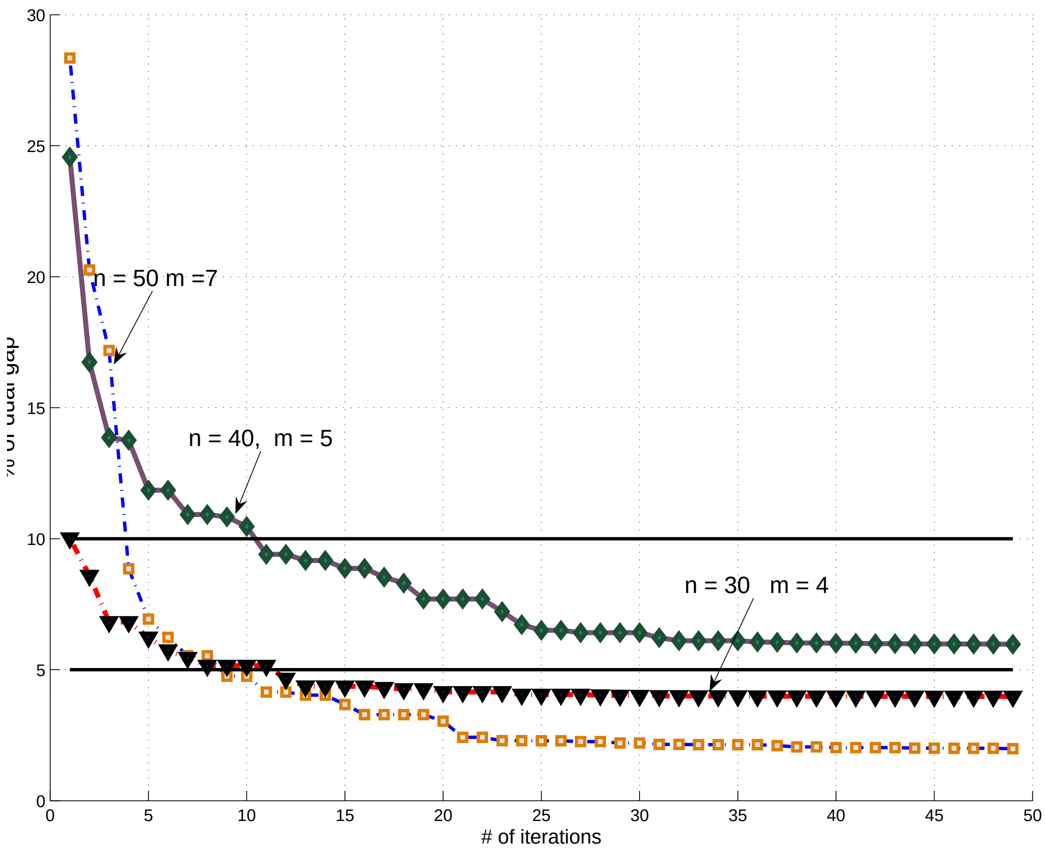

Fig. 5. Convergence of dual gap for random instances

constraints, where there are additional timing constraints based on which target is visited first [45]

C. Conclusions

1. We formulate CMP as an integer program with $(n+m)^2+m$ variables (one variable for each edge joining any two vertices and one variable for each depot). This formulation exploits the fact that the Dubins' distances satisfy triangle inequality.

2. We provide an approach for assigning the sequence of targets to a UV by solving a

Lagrangian dual [46] of the formulated integer program. This step involves finding a minimum cost directed spanning tree with a degree constraint. The problem is solved by penalizing this degree constraint if violated and using the approach given in [43].

3. Given a set of targets, we provide a Lagrangian heuristic to find the sequence of targets each UV must visit. The Lagrangian heuristic modifies the dual solution and constructs a sub-optimal solution to the combinatorial (partitioning and sequencing) problem..

4. The Lagrangian dual of the integer program also gives a tight lower bound for the integer program. This lower bound was used in the Branch and Bound solver to find the optimal solution to the integer program.

5. Numerical results are provided which compare the cost of the solution produced by the algorithm given in this chapter with the optimal cost of the integer program.

Table II. # of iterations for computing $DMSF^*$

n	p^*	i^*
16	3	9
21	6	15
27	3	21
30	7	20
32	6	12
33	5	20
34	6	13
35	5	2
38	4	8
40	8	13
41	2	15
42	6	14
44	7	16
45	3	13
47	7	8
48	5	8
49	8	8
50	3	8
57	7	14
66	4	2

Table III. Duality gap ($\%$) for various instances at 25^{th} iteration.

n	m	CMP^*	$\phi([\Pi]^k, [\Psi]^k)_{k=25}$	$[Cprimal^*]^k_{k=25}$	$[\epsilon]^k_{k=25}$
14	3	1624.8	1566.6	1624.8	3.7163
19	4	2142.9	2142.2	2142.9	0.030342
22	7	2076.9	2041.8	2204.5	7.968
24	3	2638.1	2637.9	2638.1	0.0068235
24	7	2352.7	2294.8	2370.9	3.3161
26	3	2833.2	2833.2	2916.8	2.9493
26	6	2678.9	2598.9	2706.7	4.1476
28	5	2824.5	2728.5	2824.6	3.5187
30	4	2872.1	2759.2	2944.6	6.7193
31	4	3333.9	3268	3459.6	5.8622
32	3	2898.2	2786.1	2940.6	5.545
36	3	3271.1	3149.7	3386.6	7.5216
38	4	3497.9	3479.1	3497.9	0.54181
40	7	3061	2992.4	3061	2.2918
45	2	3724.6	3685.5	3748.4	1.7061
48	5	3722.1	3681.9	3723.7	1.1342
50	5	3242.4	3216	3346.2	4.0475

Table IV. Duality gap $(\%)$ for various instances at 50^{th} iteration.

n	m	CMP^*	$\phi([\Pi]^k, [\Psi]^k)_{k=50}$	$[Cprimal^*]^k_{k=50}$	$[\epsilon]^k_{k=50}$
14	3	1624.8	1568.5	1624.8	3.5959
19	4	2142.9	2142.9	2142.9	0.00046666
22	7	2076.9	2049.7	2204.5	7.5535
24	3	2638.1	2638.1	2638.1	0.00075812
24	7	2352.7	2312.2	2370.9	2.5418
26	3	2833.2	2833.2	2916.8	2.9493
26	6	2678.9	2611.6	2706.7	3.6387
28	5	2824.5	2740.2	2824.6	3.0794
30	4	2872.1	2778.7	2944.6	5.9711
31	4	3333.9	3268	3459.6	5.8622
32	3	2898.2	2786.1	2940.6	5.545
36	3	3271.1	3167.3	3386.6	6.9235
38	4	3497.9	3480	3497.9	0.51408
40	7	3061	3001.4	3061	1.9854
45	2	3724.6	3692.6	3748.4	1.5119
48	5	3722.1	3688	3723.7	0.96692
50	5	3242.4	3218.3	3346.2	3.9754

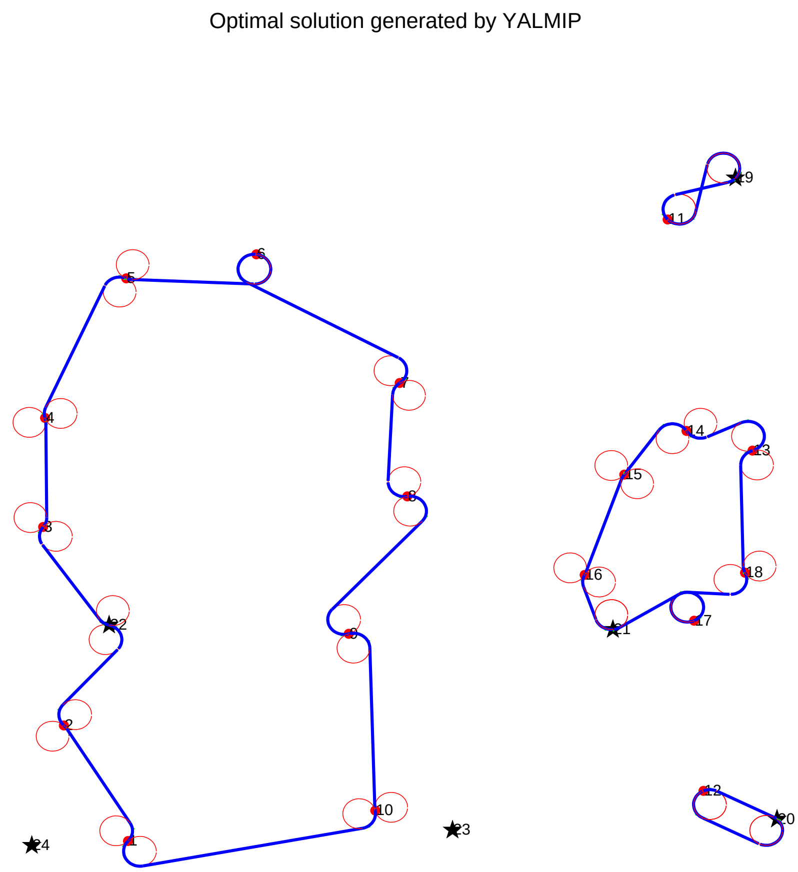

Fig. 6. Optimal solution for a random instance

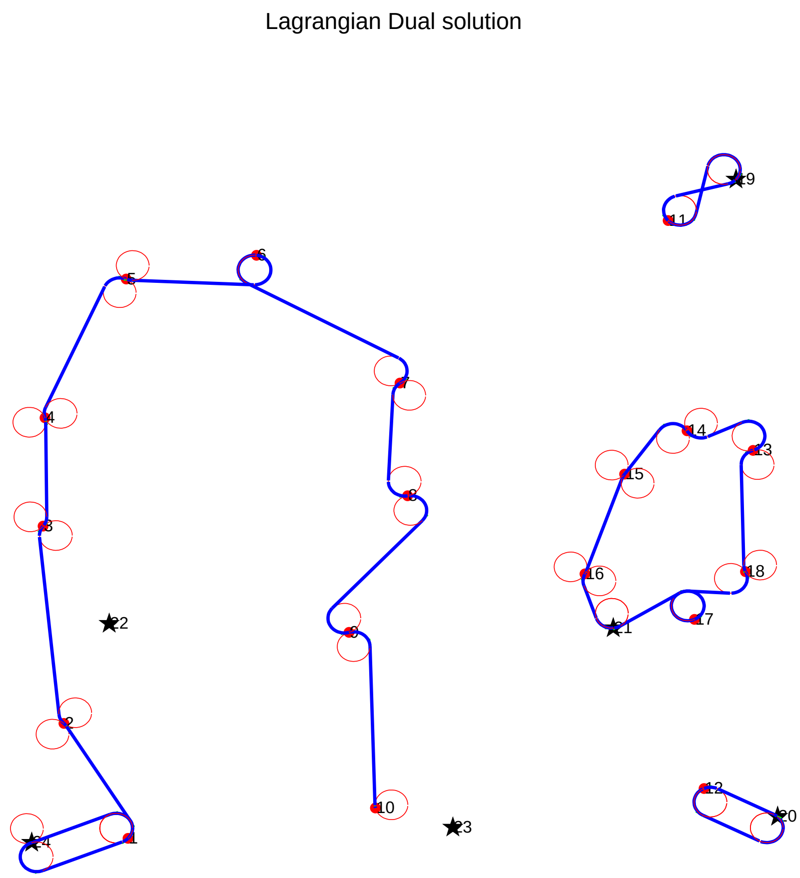

Fig. 7. Dual solution for a random instance

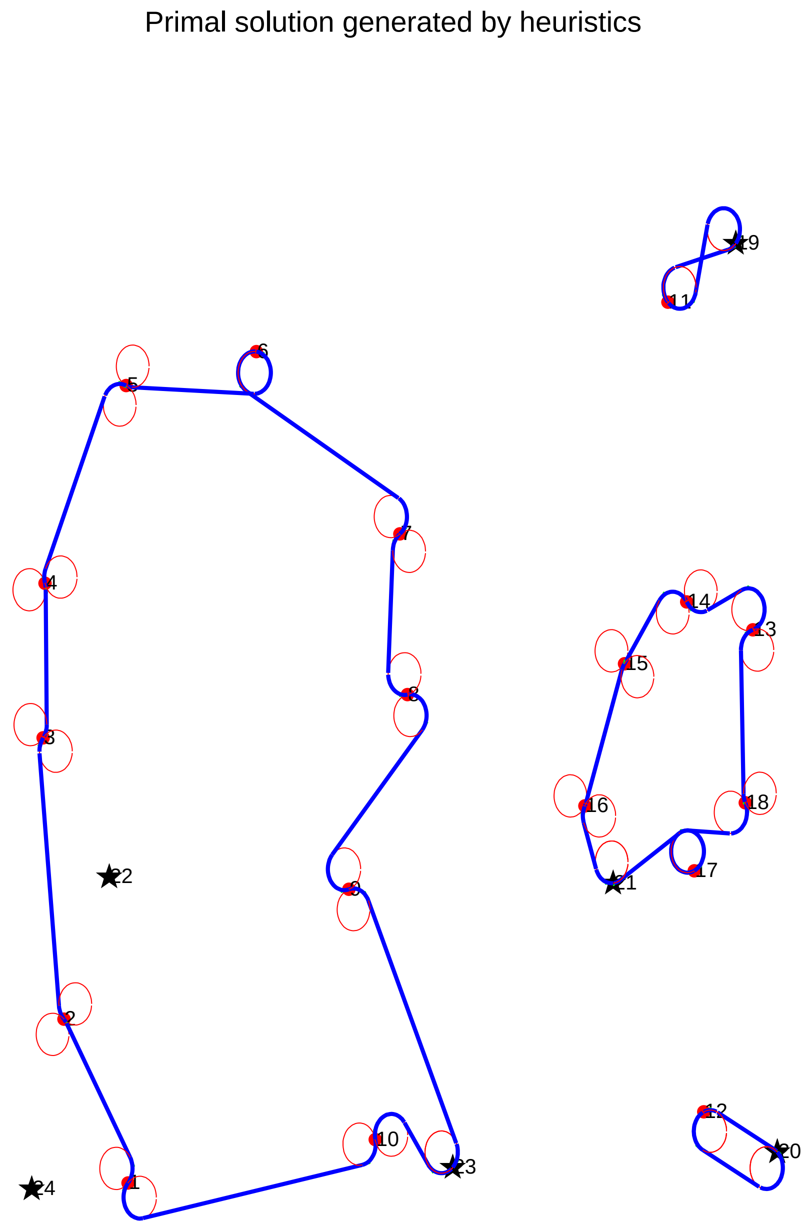

Fig. 8. Primal solution for a random instance

CHAPTER III

AN APPROXIMATION ALGORITHM FOR A TWO DEPOT, HETEROGENEOUS TRAVELING SALESMAN PROBLEM

A. Introduction

In the previous chapter, we considered a homogeneous collection of UVs. It is conceivable that UVs with different capabilities may be required to operate collectively in a mission and their motion plans need to be determined. In this case, the cost of traveling between any two locations may also depend on the direction of travel and the UV deployed. In this chapter *, we consider the motion planning of structurally heterogeneous collection of UVs. We begin with a simplified problem when the collection consists of only two heterogeneous UVs. Specifically, the routing problem we address is a 2-depot, Heterogeneous Traveling Salesman Problem (2-HTSP) which is stated as follows: Given a set of nodes (or destinations) and two heterogeneous vehicles that start from distinct depots, find a tour for each vehicle such that each destination is visited exactly once and the total cost of the tours of the vehicles is a minimum.

In this chapter, two types of heterogeneity for both the vehicles are considered, i.e., *structural* heterogeneity and *functional* heterogeneity. If the vehicles are structurally different, the cost of traveling between two destinations not only depends on the position of the destinations but also on the vehicle. In the case of functional heterogeneity, the vehicles are identical structurally but there may be additional vehicle-destination constraints that must be met. In this case, the destinations may be partitioned into three disjoint subsets: a subset

*Part of this chapter was reprinted with permission from "3-Approximation Algorithm for a Two Depot, Heterogeneous Traveling Salesman Problem," by Sai Yadlapalli, Sivakumar Rathinam and Swaroop Darbhn, Accepted for Publication in Operation Research Letters, Copyright © 2010 by Elsevier http://www.elsevier.com/

of destinations the first vehicle must visit, a subset of destinations the second vehicle must visit and a set of common destinations that either of the two vehicles can visit.

There are several applications ([47],[48],[49],[50]) where routing problems such as the 2-HTSP could arise. In UV applications, it is possible that the vehicles have different constraints on their maximum speeds depending on the vehicle type. Even if we ignore the constraints on the turning radius of the vehicles when the destinations are reasonably far apart, the cost of traveling between any two destinations is still dependent on the type of the vehicle. Also, the UVs can carry different sensors, and therefore, there may be additional constraints that require a subset of destinations must be visited by a specific UV.

The 2-HTSP is a generalization of the single Traveling Salesman Problem and is NP-Hard [36]. Therefore, we are interested in developing approximation algorithms for the 2-HTSP. An $\alpha-$approximation algorithm [11] is an algorithm that

- has a polynomial-time running time, and
- returns a solution whose cost is within α times the optimal cost.

It is assumed that the cost of traveling from an origin to a destination directly for each vehicle is no more expensive than the cost of traveling from the same origin to the destination through an intermediate location. When the costs satisfy the triangle inequality, they are said to satisfy the above property. It is currently known that there cannot exist a constant factor approximation algorithm for a single Traveling Salesman Problem if the triangle inequality is not satisfied unless $P = NP$.

Aiming for approximation algorithms is reasonable in the context of path planning of unmanned aerial vehicles with motion constraints because the cost of traveling between any two targets for an unmanned aerial vehicle can depend on several factors including wind disturbances. Hence, it is appropriate to devise approximation algorithms for these planning problems that are relatively inexpensive than devise algorithms that opt for exact solutions. In this chapter, a 3-approximation algorithm is introduced for the 2-HTSP when the costs associated with each vehicle satisfy the triangle inequality.

1. Literature review

The 2-HTSP is related to a well known class of problems that has received significant attention in the area of combinatorial optimization. These problems include the Traveling Salesman Problem (TSP), the Hamiltonian Path Problem (HPP) and their generalizations [36, 11, 51, 12]. As this article deals with constant factor approximation algorithms, henceforth, we assume that, for every vehicle the costs satisfy the triangle inequality. The symmetric TSP has two well known approximation algorithms - the $2-$approximation algorithm obtained by doubling the minimum spanning tree (MST) and the $1.5-$approximation algorithm of Christofides obtained through the construction of MST and a weighted non-bipartite matching of nodes of MST with odd degree [52].

There are $2-$approximation algorithms for variants of the homogeneous, multiple TSP and HPP in [53],[26]. Also, Rathinam et al. in [54] have developed $1.5-$approximation algorithm for two variants of a 2 depot, Hamiltonian Path Problem. Currently, there are no approximation algorithms for any heterogeneous, multiple TSP known in the literature. In this article, we present the first 3-approximation algorithm for the 2-HTSP when the costs satisfy the triangle inequality.

The 2-HTSP problem is formulated as an integer program with assignment, degree and connectivity constraints on a multi-graph. Given any two destinations, we construct this multi-graph by adding an edge joining the two destinations for each vehicle. The cost assigned to an edge would then be equal to the distance required by the corresponding vehicle to travel that edge. The basic idea for the approximation algorithm is as follows: We first relax all the binary decision variables and solve the resulting linear program to find the subset of destinations each vehicle must visit. Once the partitioning problem is solved, Christofides algorithm [52] is used on the partitions to get a tour for each vehicle. Even though the number of cut constraints in this linear program grow exponentially with the number of destinations, the relaxed linear program can be shown to be solvable in polynomial time using the Ellipsoid method[55]. Using the result that the cost of the feasible solution produced by the Christofides algorithm is at most $\frac{3}{2}$ times the cost of the Held-Karp relaxation of the single TSP [23] and the parsimonious property of the Held-Karp relaxation[56], one can show that the proposed algorithm has an approximation ratio of 3. A key part of our approximation algorithm is in the way we formulate the 2-HTSP and relax the constraints. The formulation is presented in the following section.

B. Problem formulation

Let $T = \{1, \ldots, n\}$ be the set of vertices that denote all the destinations and $V = \{d_1, d_2\}$ be the set of vertices that correspond to the initial depots of the vehicles. For each depot vertex d_i, we also introduce a copy of the depot vertex called the terminal vertex, d_i', that exactly coincides with the location of the depot vertex. Each vehicle after visiting its share of destinations will visit its corresponding terminal before returning to its depot. Our integer programming formulation includes a terminal vertex for each vehicle in order to allow for each vehicle to visit exactly one destination if needed (this will be further discussed in **Remark 1** later.).

Let $\mathbb{V}^i = \{d_i, d_i'\} \bigcup T$ denote the set of all the vertices corresponding to the i^{th} vehicle. Let E^i stand for the set of all the edges joining any two vertices in $\mathbb{V}^i$. For any $S \subset \mathbb{V}^i$, let $\delta_i(S)$ denote the set of all the edges $e \in E^i$ that has one end point in S and one end point in $\mathbb{V}^i \setminus S$. Each edge $e \in E^i$ has a cost $C_e^i \in \mathbb{Q}^+$ associated with it where $\mathbb{Q}^+$ is the set of all positive rational numbers. Assume that all the costs satisfy the triangle inequality. Let R_1 and R_2 be the set of vertices that must be visited by the first and the second vehicle respectively. Note that $R_1 \bigcap R_2 = \emptyset$ and each destination in $T \setminus (R_1 \bigcup R_2)$ can be visited by either the first or the second vehicle. Let x_e (y_e) denote the binary variables that decide whether edge e is present in the routes of the first (second) vehicle. An edge e is present in the tour of the first vehicle if $x_e = 1$ and is not present otherwise. y_e is defined similarly. Let ϕ_i denote the binary decision variable that is equal to 1 if destination i is visited by the first vehicle and is equal to 0 otherwise. Similarly, let η_i denote the binary decision variable that is equal to 1 if destination i is visited by the second vehicle and is equal to 0 otherwise. The following is the **integer programming formulation** of the 2-HTSP:

$$C_{opt} = \min_{x,y,\phi,\eta} \sum_{e \in E^1} x_e C_e^1 + \sum_{e \in E^2} y_e C_e^2 \tag{3.1}$$

$$\phi_i = 1, \text{ for all } i \in R_1, \tag{3.2}$$

$$\eta_i = 1, \text{ for all } i \in R_2, \tag{3.3}$$

$$\phi_i + \eta_i = 1, \text{ for all } i \in T \setminus \{R_1 \bigcup R_2\}, \tag{3.4}$$

$$\sum_{e \in \delta_1(\{i\})} x_e = 2\phi_i, \forall i \in T, \tag{3.5a}$$

$$\sum_{e \in \delta_1(\{u\})} x_e \geq 2\phi_i, u \in \{d_1, d_1'\}, \forall i \in T, \tag{3.5b}$$

$$\sum_{e \in \delta_1(\{u\})} x_e \leq 2, u \in \{d_1, d_1'\}, \tag{3.5c}$$

For all $i \in T$,

$$\sum_{e \in \delta_1(S)} x_e \geq 2\phi_i, \forall S \subset \mathbb{V}^1, \text{ such that } i \in S, \ |S \bigcap \{d_1, d_1'\}| \leq 1, \tag{3.5d}$$

$$x_e \in \{0,1\} \ \forall e \in E^1, \tag{3.5e}$$

$$\phi_i \in \{0,1\} \ \forall i \in T. \tag{3.5f}$$

$$\sum_{e \in \delta_2(\{i\})} y_e = 2\eta_i, \forall i \in T, \tag{3.6a}$$

$$\sum_{e \in \delta_2(\{u\})} y_e \geq 2\eta_i, u \in \{d_2, d_2'\}, \forall i \in T, \tag{3.6b}$$

$$\sum_{e \in \delta_2(\{u\})} y_e \leq 2, u \in \{d_2, d_2'\}, \tag{3.6c}$$

For all $i \in T$,

$$\sum_{e \in \delta_2(S)} y_e \geq 2\eta_i, \forall S \subset \mathbb{V}^2, \text{ such that } i \in S, \ |S \bigcap \{d_2, d_2'\}| \leq 1, \tag{3.6d}$$

$$y_e \in \{0,1\} \ \forall e \in E^2, \tag{3.6e}$$

$$\eta_i \in \{0,1\} \ \forall i \in T. \tag{3.6f}$$

The constraints in (3.2) and (3.3) state that the destinations in R_1 and R_2 must be respectively visited by the first and the second vehicle. The assignment constraints in (3.4) require that a destination in $T \setminus \{R_1 \bigcup R_2\}$ can be visited either by the first vehicle or the second vehicle but not both. The degree constraints in (3.5a, 3.6a) together indicate that the number of edges incident on each destination vertex must be equal to 2. The degree constraints in (3.5b,3.6b) specify that the number of edges incident on a depot/terminal must be at least equal to 2 if the vehicle corresponding to the depot/terminal is visiting at least one destination. The degree constraints in (3.5c,3.6c) state that the number of edges incident on both the depots and the terminals can at most be equal to 2. If a destination is visited by a vehicle, the cut constraints in (3.5d, 3.6d) enforce a requirement that there must be at least two edge disjoint paths from the destination to the depot/terminal corresponding to the vehicle visiting that destination. These cut constraints in combination with the degree constraints also eliminate the presence of any cycles among the destination vertices.

Remark 1: The terminal vertices d_1', d_2' were added to the problem to essentially allow for a vehicle to visit exactly one destination if needed. For example, by adding these terminal vertices, one could allow a tour for the first vehicle to be of the form $\{d_1, u, d_1', d_1\}$ where u is a vertex denoting a destination. Then, the first vehicle visits the destination u, and then the terminal d_1' before returning to its depot. However, adding these terminal vertices could also result in a solution where the optimal tour for the i^{th} vehicle is of the form $\{d_i, v_{i1}, \cdots, v_{il_i}, d_i', v_{il_{i+1}}, \cdots, v_{ik_i}, d_i\}$ where $v_{ij} \in T$ for $j = 1, \cdots, k_i$. In this case, the depot and its corresponding terminal vertex are not adjacent vertices in the tour. However, this is not an issue in this article as it is assumed all the costs associated with every vehicle satisfy the triangle inequality. Therefore, one can always shortcut the edges in the optimal solution to obtain tours so that each vehicle returns to its depot immediately after visiting its corresponding terminal.

Remark 2: The cut constraints in (3.5d,3.6d) can also be written equivalently as given below. The reason for formulating the constraints as stated in (3.5d,3.6d) is to simplify the proofs of the approximation algorithm discussed in section D.

$$\begin{aligned}\sum_{e\in\delta_1(S)} x_e &\geq 2\max_{i\in S}\phi_i, \forall S\subset \mathrm{V}^1 \text{ such that } |S\bigcap T|>0,\ |S\bigcap\{d_1,d_1'\}|\leq 1,\\ \sum_{e\in\delta_2(S)} y_e &\geq 2\max_{i\in S}\eta_i, \forall S\subset \mathrm{V}^2 \text{ such that } |S\bigcap T|>0,\ |S\bigcap\{d_2,d_2'\}|\leq 1.\end{aligned}$$

Remark 3: Using the max-flow min-cut theorem [57],[11], the cut constraints in (3.5d,3.6d) can also be formulated using flow constraints. Therefore, ϕ_i and η_i can be interpreted as the amount of flow shipped from the first and second depot respectively to the i^{th} destination.

The following Linear Programming (LP) relaxation of the 2-HTSP plays a crucial role in the development of the algorithm.

$$C^*_{lp} = \min_{x,y,\phi,\eta} \sum_{e\in E^1} x_e C^1_e + \sum_{e\in E^2} y_e C^2_e \tag{3.7}$$

$$\phi_i \geq 1, \text{ for all } i\in R_1, \tag{3.8}$$

$$\eta_i \geq 1, \text{ for all } i\in R_2, \tag{3.9}$$

$$\phi_i+\eta_i \geq 1, \text{ for all } i\in T\setminus\{R_1\bigcup R_2\}, \tag{3.10}$$

$$\sum_{e\in\delta_1(\{u\})} x_e \geq 2\phi_i, u\in\{d_1,d_1'\}, \tag{3.11a}$$

For all $i\in T$,

$$\sum_{e\in\delta_1(S)} x_e \geq 2\phi_i, \forall S\subset \mathrm{V}^1, \text{ such that } i\in S,\ |S\bigcap\{d_1,d_1'\}|\leq 1, \tag{3.11b}$$

$$0 \leq x_e \leq 1\ \forall e\in E^1, \tag{3.11c}$$

$$\phi_i \geq 0\ \forall i\in T. \tag{3.11d}$$

$$\sum_{e\in\delta_2(\{u\})} y_e \geq 2\eta_i, u\in\{d_2,d_2'\}, \tag{3.12a}$$

For all $i\in T$,

$$\sum_{e\in\delta_2(S)} y_e \geq 2\eta_i, \forall S\subset \mathrm{V}^2, \text{ such that } i\in S,\ |S\bigcap\{d_2,d_2'\}|\leq 1, \tag{3.12b}$$

$$0 \leq y_e \leq 1\ \forall e\in E^2, \tag{3.12c}$$

$$\eta_i \geq 0\ \forall i\in T. \tag{3.12d}$$

C. Approximation algorithm for the 2-HTSP

The following is the proposed algorithm *Approx* for the 2-HTSP:

1. Solve the Linear Programming relaxation formulated in equations (3.7-3.12) using the Ellipsoid method [55]. Let an optimal solution to this relaxation be denoted by $(x^*, y^*, \phi^*, \eta^*)$. We will later show that this relaxation is solvable in polynomial time.

2. ϕ_i^* (η_i^*) essentially denotes the optimal fraction of the flow shipped to the i^{th} destination using the first vehicle (second vehicle). Assign each destination to the vehicle that ships its largest fraction. Break ties arbitrarily. This step essentially partitions the destinations into two groups. Let $\mathcal{U}_1 = \{i : i \in T, \psi_i \geq \eta_i\}$ correspond to those destinations which are assigned to the first vehicle, and $\mathcal{U}_2 = T \setminus \mathcal{U}_1$ be the set of destinations assigned to the second vehicle.

3. For the i^{th} vehicle, if $\mathcal{U}_i$ is not empty, apply the Christofides algorithm to find a tour that visits all the vertices in $\mathcal{U}_i \bigcup \{d_i, d_i'\}$.

Clearly, the tours produced by the above algorithm is a feasible solution for the integer program formulated in equations (3.1-3.6f). The following theorem is the main result of this paper:

Theorem 1. *Algorithm Approx is a polynomial time algorithm for the 2-HTSP with an approximation ratio of 3.*

D. Proof of the 3-approximation ratio of Approx

In the following lemma, we first show that *Approx* is a polynomial time algorithm.

Lemma 3. *Approx is a polynomial time algorithm.*

Proof. The main steps in *Approx* involve solving a linear program defined by equations (3.7-3.12) and using the Christofides algorithm. If there are n destinations, it is known that the number of steps required for the Christofides algorithm is of $O(n^3)$. Therefore, step (3) of the algorithm *Approx* requires $O(|\mathcal{U}_1|^3) + O(|\mathcal{U}_2|^3) \approx O(n^3)$ steps. Shortly, it will be shown that the linear program (3.7-3.12) is solvable in polynomial time using the Ellipsoid method [55]. In [55], Grötschel, Lovász and Schrijver showed that the polynomial solvability of a linear program is equivalent to the polynomial solvability of the following separation problem using the Ellipsoid method:

Let $\mathcal{P}$ denote the polytope defined by all the constraints of the linear program in (3.8-3.12). Given $x_e\ \forall e \in E^1, y_e\ \forall e \in E^2$, and $\phi_i, \eta_i\ \forall i \in T$, decide whether the given solution is in $\mathcal{P}$ and if not, find a violated constraint.

The cut constraints defined by equations (3.11b,3.12b) are the only set of constraints that grow exponentially with the number of destinations. Therefore, the separation problem is solvable in polynomial time if a separation algorithm can be developed for these cut constraints. For each destination $i \in T$, the cut constraints defined in (3.11b) are as follows:

$$\sum_{e \in \delta_1(S)} x_e \geq 2\phi_i, \forall S \subset \mathbb{V}^1 \text{ such that } i \in S \text{ and } |S \bigcap \{d_1, d_1'\}| \leq 1. \qquad (3.13)$$

Applying max-flow, min-cut theorem [57], the above cut constraints imply that there must at least be a flow of $2\phi_i$ from vertex i to both the depot d_1 and the terminal d_1'. Therefore, given a destination vertex $i \in T$, $x_e\ \forall e \in E^1$ and ϕ_i, one can use the max-flow algorithm to decide whether the given solution is feasible for the constraints in (3.13) or find a cut that violates these constraints in polynomial time. By repeating this argument for each of the destination vertices, we can conclude that a polynomial time separation algorithm is available to handle the constraints defined in (3.11b). By using similar arguments, one can also develop a separation algorithm for the constraints in (3.12b). Therefore, there is a polynomial time algorithm for the separation problem. Hence, the linear program defined in equations (3.11,3.12) is solvable in polynomial time using the Ellipsoidal method [55]. $\square$

In the remaining part of this discussion, it will be shown that the approximation ratio of *Approx* is 3. Let the tour produced for the i^{th} vehicle by *Approx* be denoted by $TOUR_i$. Let the cost of these tours be denoted by $C(TOUR_1)$ and $C(TOUR_2)$ respectively. For a single TSP, Shmoys and Williamson [23] have shown that the cost of the solution produced by the Christofides algorithm is at most a factor of $\frac{3}{2}$ away from the optimal cost of the Held-Karp relaxation of the single TSP. Using this result, one can deduce that $C(TOUR_1) \leq \frac{3}{2}C^1_{hk}$ where C^1_{hk} denotes the optimal cost of the Held-Karp's relaxation for the first vehicle visiting all the vertices in $\mathcal{U}_1 \bigcup \{d_1, d'_1\}$. Similarly, it follows that $C(TOUR_2) \leq \frac{3}{2}C^2_{hk}$ where C^2_{hk} denotes the optimal cost of the Held-Karp's relaxation for the second vehicle visiting all the vertices in $\mathcal{U}_2 \bigcup \{d_2, d'_2\}$. The relaxation costs C^1_{hk} and C^2_{hk} are essentially defined as follows:

$$
\begin{aligned}
C^1_{hk} &= \min_x \sum_{e \in E^1} x_e C^1_e \\
\sum_{e \in \delta_1(S)} x_e &\geq 2, \forall S \subset \mathcal{U}_1 \bigcup \{d_1, d'_1\}, \\
\sum_{e \in \delta_1(\{i\})} x_e &= 2, \forall i \in \mathcal{U}_1 \bigcup \{d_1, d'_1\}, \\
\sum_{e \in \delta_1(\{i\})} x_e &= 0, \forall i \in \mathcal{U}_2, \\
x_e &\geq 0 \; \forall e \in E^1.
\end{aligned}
\qquad
\begin{aligned}
C^2_{hk} &= \min_y \sum_{e \in E^2} y_e C^2_e \\
\sum_{e \in \delta_2(S)} y_e &\geq 2, \forall S \subset \mathcal{U}_2 \bigcup \{d_2, d'_2\}, \\
\sum_{e \in \delta_2(\{i\})} y_e &= 2, \forall i \in \mathcal{U}_2 \bigcup \{d_2, d'_2\}, \\
\sum_{e \in \delta_2(\{i\})} y_e &= 0, \forall i \in \mathcal{U}_1, \\
y_e &\geq 0 \; \forall e \in E^2.
\end{aligned}
$$

As all the costs satisfy the triangle inequality, Goemans and Bertsimas [56] have shown that the optimal relaxation cost will not change if one were to remove all the degree constraints in the above Held-Karp relaxation. In [56], Goemans and Bertsimas proved this property for a more general survivable network design problem. This property is essentially called the parsimonious property of a network design problem. That is,

$$\begin{aligned} C_{hk}^1 &= \min_x \sum_{e \in E^1} x_e C_e^1 \qquad (3.14) \\ \sum_{e \in \delta_1(S)} x_e &\geq 2, \forall S \subset \mathcal{U}_1 \bigcup \{d_1, d_1'\}, \\ x_e &\geq 0 \; \forall e \in E^1. \end{aligned}$$

$$\begin{aligned} C_{hk}^2 &= \min_y \sum_{e \in E^2} y_e C_e^2 \qquad (3.15) \\ \sum_{e \in \delta_2(S)} y_e &\geq 2, \forall S \subset \mathcal{U}_2 \bigcup \{d_2, d_2'\}, \\ y_e &\geq 0 \; \forall e \in E^2. \end{aligned}$$

The sum of the optimal cost of the Held-Karp relaxations, $C_{hk}^1 + C_{hk}^2$, can now be upper bounded by two times the optimal cost, C_{lp}^*, of the LP relaxation (3.7-3.12) of the 2-HTSP. To prove this, consider any optimal solution $(x^*, y^*, \phi^*, \eta^*)$ to the LP in (3.7-3.12). One can construct a solution, $\widehat{x}$, for the Held-Karp relaxation in (3.14) by choosing $\widehat{x} = 2x^*$. To prove that $\widehat{x}$ is feasible solution for (3.14), note that, for any $S \subset \mathcal{U}_1 \bigcup \{d_1, d_1'\}, |S \bigcap \{d_1, d_1'\}| = 2$,

$$\begin{aligned} \sum_{e \in \delta_1(S)} \widehat{x}_e &= 2 \sum_{e \in \delta_1(S)} x_e^*, \\ &= 2 \sum_{e \in \delta_1(\mathbb{V}^1 \setminus S)} x_e^*, \\ &\geq 4\phi_i^*, \text{ for all } i \in \mathbb{V}^1 \setminus S, \quad \text{(from constraint } 3.11b) \\ &\geq 4\phi_i^*, \text{ for all } i \in \mathcal{U}_1 \setminus S, \\ &\geq 2. \end{aligned}$$

Similarly, for any $S \subset \mathcal{U}_1 \bigcup \{d_1, d_1'\}, |S \bigcap \mathcal{U}_1| \geq 1, |S \bigcap \{d_1, d_1'\}| \leq 1$,

$$\begin{aligned}\sum_{e \in \delta_1(S)} \widehat{x}_e &= 2 \sum_{e \in \delta_1(S)} x_e^*, \\ &\geq 4\phi_i^*, \text{ for all } i \in S \bigcap \mathcal{U}_1, \quad \text{(from constraint } 3.11b) \\ &\geq 2.\end{aligned}$$

Also, for $u = d_1$ or $u = d_1'$,

$$\begin{aligned}\sum_{e \in \delta_1(u)} \widehat{x}_e &= 2 \sum_{e \in \delta_1(u)} x_e^*, \\ &\geq 4\phi_i^*, \text{ for all } i \in \mathcal{U}_1, \quad \text{(from constraint } 3.11a) \\ &\geq 2.\end{aligned}$$

Therefore, $\widehat{x}$ is a feasible solution for (3.14). In the same way, one can also show that $\widehat{y} = 2y^*$ is also a feasible solution for the Help-Karp relaxation defined in (3.15). Therefore, $C_{hk}^1 + C_{hk}^2 \leq 2\sum_{e \in E^1} x_e^* C_e^1 + 2\sum_{e \in E^2} y_e^* C_e^2 = 2C_{lp}^*$. Putting together all the results:

$$\begin{aligned}C(TOUR_1) + C(TOUR_2) &\leq \frac{3}{2}(C_{hk}^1 + C_{hk}^2), \\ &\leq 3C_{lp}^*, \\ &\leq 3C_{opt}.\end{aligned}$$

E. Extension to other problems

1. The related min-max problem

The above approach can also be extended to obtain a 3-approximation algorithm for a 2 depot, Heterogeneous TSP where the objective is to minimize the maximum cost traveled by either of the vehicles. To see this, consider the following min-max problem:

$$C_{opt}^{max*} = \min_{x,y,\phi,\eta} \max\{\sum_{e \in E^1} x_e C_e^1, \sum_{e \in E^2} y_e C_e^2\} \tag{3.16}$$

subject to the constraints defined in (3.5a-3.6f). The above min-max problem can also be restated as:

$$C_{opt}^{max*} = \min_{t,x,y,\phi,\eta} t \tag{3.17}$$

$$\begin{aligned} t &\geq \sum_{e \in E^1} x_e C_e^1, \\ t &\geq \sum_{e \in E^2} y_e C_e^2, \end{aligned} \tag{3.18}$$

and the constraints in (3.5a-3.6f). Therefore, a LP relaxation of this min-max problem will have an objective defined in (3.17) subject to constraints in (3.18,3.8-3.12). The approximation algorithm for the min-max problem also follows the same approach as Algorithm *Approx* in section C: 1) Solve the LP relaxation of the min-max problem; 2) Assign any destination i to the first vehicle if $\phi_i \geq \eta_i$; 3) For each vehicle, use the Christofides algorithm to obtain a tour to visit its set of destinations. Let C_{lp}^{max*} be the optimal cost of the LP relaxation of the min-max problem. Using the same notations and similar arguments as in the previous section, the following can be arrived at:

$$\begin{aligned} \max(C(TOUR_1), C(TOUR_2)) &\leq \frac{3}{2} \max(C_{hk}^1, C_{hk}^2), \\ &\leq 3C_{lp}^{max*}, \\ &\leq 3C_{opt}^{max*}. \end{aligned}$$

2. Generalization of 2-depot heterogeneous problem

In general, the approach given in this chapter can be extended to obtain a $\frac{3m}{2}$-approximation algorithm for variants of a m-depot, Heterogeneous Traveling Salesman Problem.

When there are more than 2 vehicles, the vehicle-destination constraints that are present due to the functional heterogeneity can be posed in different ways. For example, one can specify that a vehicle must visit a subset of targets or that a set of vehicles must not visit a subset of targets. The $\frac{3m}{2}$-approximation algorithm that is obtained by extending the approach in this paper, takes into consideration both these specifications. The 3-Approximation algorithm for 2-HTSP presented above in this chapter is accepted for publication in [58].

F. 2-component heterogeneous minimum spanning forest

In this section, we pose a 2-component Heterogeneous, Minimum Cost Spanning Forest (HMSF) problem, a combinatorial problem that is relevant to the tour problem discussed above. The homogeneous case of Multiple TSP admits a 2-approximation algorithm [53]. That 2-approximation algorithm relies on doubling the edges of Minimum Spanning Forest (MSF) and short-cutting the resulting edges to form a feasible tour. Each component of MSF contains exactly one depot and a partition of targets that are to be *visited* by the vehicle starting at that depot. MSF can be constructed in polynomial time and the procedure is detailed in [53]. In the same spirit, we are interested in combinatorial formulation of Heterogeneous MSF (HMSF). The computational complexity of HMSF is not clear. In the rest of section, we present our formulation for 2-HMSF and a 4-approximation algorithm for the same. The approximation algorithm presented in this chapter for 2-HMSF has been published in [59]

The problem of 2-HMSF is as follows: Construct two *disjoint* trees rooted at d_1 and

d_2, so that all the targets are spanned and the total cost of constructing the two trees is minimum. The cost of the tree rooted at d_1 is computed with the edge costs associated with the first vehicle while the cost of tree rooted at d_2 is computed with the edge costs associated with the second vehicle. In this section, we pose the flow based formulation for $2 - HMSF$ in detail.

Let p_{ij}^k denote the flow of k^{th} commodity originating from the first depot and flowing from node i to node j. Let q_{ij}^k be the corresponding flow from the second depot through the directed edge (i, j) to the k^{th} target. Though both the flows, p_{ij}^k, q_{ij}^k, can flow through (i, j), they are constrained in amount by the capacity of the arc (i, j). Let f_{ij} denote whether arc (i, j) is used by the first vehicle in its tour and similarly let g_{ij} denote whether arc (i, j) is used by the second vehicle. It should be noted that the directionality of arc is important here. The following capacity constraints naturally arise:

$$0 \le p_{ij}^u \le f_{ij} \quad \forall i, j \in T \cup d_1, \tag{3.19}$$

$$0 \le q_{ij}^u \le g_{ij} \quad \forall i, j \in T \cup d_2. \tag{3.20}$$

Consider an edge $e \in E$. Let (i, j) be endpoints of e. Let x_e and y_e represent the variables which decide whether edge e is present in routes of first vehicle and second vehicle respectively. Edge e is present in the tour ($x_e = 1$) of the first vehicle if either there is a directed arc from i to j ($f_{ij} = 1$) or there is a directed arc from j to i ($f_{ji} = 1$). These conditions can be stated as follows:

$$f_{ij} + f_{ji} = x_e \quad \forall e \in E, \tag{3.21}$$

$$g_{ij} + g_{ji} = y_e \quad \forall e \in E. \tag{3.22}$$

A shipment of the u^{th} commodity shipped from either of the depots can only be delivered to the u^{th} target. Let ψ_u be the quantity of the u^{th} commodity shipped to the u^{th} target from the first depot and let η_u be the corresponding quantity shipped from the second depot. The following are the flow balance equations for flows p and q respectively:

$$\sum_{j \in T} p_{ij}^k - p_{ji}^k = \begin{cases} \psi_k & \forall k \in T \text{ and } i = d_1, \\ 0 & \forall i, k \in T \text{ and } i \neq k, \\ -\psi_k & \forall i, k \in T \text{ and } i = k. \end{cases} \tag{3.23}$$

$$\sum_{j \in T} q_{ij}^k - q_{ji}^k = \begin{cases} \eta_k & \forall k \in T \text{ and } i = d_2, \\ 0 & \forall i, k \in T \text{ and } i \neq k, \\ -\eta_k & \forall i, k \in T \text{ and } i = k. \end{cases} \tag{3.24}$$

Since atleast one unit of commodity is to be shipped for each $u \in T$, we have the following relation:

$$\psi_u + \eta_u \geq 1, \forall u \in T \tag{3.25}$$

The $2 - HMSF$ may thus be posed as the following integer program:

$$HMSF^* = \min \sum_{e \in E} c_e x_e + d_e y_e. \tag{3.26}$$

subject to capacity constraints [3.19, 3.20], flow balance constraints [3.23, 3.24], directed constraints [3.21, 3.22], coupling constraint [3.25] and the following restriction on the domain of the variables:

$$x_e, y_e, f_{ij}, g_{ij} \in Z^+ \quad p_{ij}^k, q_{ij}^k, \psi_k, \eta_k \in \Re^+ \tag{3.27}$$

where Z^+ is the set of all positive integers.

The complexity of $2 - HMSF$ is not clear. However, we provide a 4-*approx* algorithm for the $2 - HMSF$ through the following algorithm.

HMSF Algorithm

1. Relax the integrality constraints in the above IP for the 2-HMSF and solve it. The relaxed program (call it LP^*) can be solved in polynomial time as the number of variables and constraints only scale polynomially with the size of V.

2. Find the optimal fractional quantities of each commodity shipped from both the depots. Partition the targets into two disjoint groups according to which depot ships the maximum amount of commodity to the target. if both depots ship equal amount of commodity to a particular target, it does not matter to which group it belongs to. Let $\mathcal{X} = \{k| \quad \psi_k \geq \frac{1}{2}\}$. $\mathcal{X}$ corresponds to those targets who have received maximum shipment of their commodity from d_1. Let $\mathcal{Y}$ be the rest of targets.

3. Find a tree spanning the targets $\mathcal{X}$ and the depot d_1 of minimum cost. The minimum cost spanning tree (MST) is computed according to the cost of edges associated with the vehicle starting at depot d_1. Similarly find a minimum-cost tree spanning the targets $\mathcal{Y}$ and the depot d_2. Clearly, this is a feasible solution to the above laid integer program. We show in the following theorem that the feasible solution constructed is within *four times* the cost of the relaxed linear program and hence, is less than $4HMSF^*$.

Theorem 2. *HMSF Algorithm is 4-approx.*

Proof: Solution of LP^* produces optimal quantities of commodities shipped from each depot. Let the optimal cost of the solution be C_{LP^*}. Let ψ^*, η^* be the optimal quantities of u^{th} commodity shipped from d_1 and d_2 respectively. We formulate a new linear program LP_1 by replacing the coupling constraint [3.25] with the following constraints.

$$\psi_k \geq \psi_k^* \quad \forall k \in T, \tag{3.28}$$

$$\eta_k \geq \eta_k^* \quad \forall k \in T. \tag{3.29}$$

Let C_{LP_1} be the optimal cost of this linear program. We shall prove shortly that $C_{LP^*} = C_{LP_1}$. Any feasible solution of LP_1 is also a feasible solution of LP^*. But over the feasible solutions of LP^* (includes the feasible solutions of LP_1), ψ_k^*, η_k^* are the optimal quantities to be shipped (cost is C_{LP}^*), which is also feasible solution of LP_1. Hence,

$$C_{LP_1} = C_{LP^*}. \tag{3.30}$$

Now lets construct another linear program LP_2, replacing the constraints [3.28, 3.29] with the following constraints:

$$\psi_k \geq \frac{1}{2} \quad \forall k \in \mathcal{X}, \tag{3.31}$$

$$\eta_k \geq \frac{1}{2} \quad \forall k \in \mathcal{Y}. \tag{3.32}$$

Consider any feasible set of commodities, ψ_1, η_1 for the LP_1 problem.

$$\begin{aligned} \psi_1 &\geq \psi_k^* \quad \forall k \in T \quad \text{by the feasibility} \\ &\geq \frac{1}{2} \quad \forall k \in \mathcal{X} \quad \text{by the definition of } \mathcal{X} \end{aligned}$$

So every feasible solution of LP_1 is also a feasible solution of LP_2. Hence,

$$C_{LP_1} \geq C_{LP_2}. \tag{3.33}$$

We now construct another linear program LP_3 by replacing constraints [3.31,3.32] with the following constraints:

$$\psi_k \geq 1 \quad \forall k \in \mathcal{X}, \tag{3.34}$$

$$\eta_k \geq 1 \quad \forall k \in \mathcal{Y}. \tag{3.35}$$

Essentially, we are just doubling the commodity requirement. Everything else remains the same. We shortly prove that $C_{LP_2} = \frac{1}{2} C_{LP_3}$.

Let $\mathcal{Y}_2 = \{x_2, y_2, f_2, g_2, p_2, q_2, \psi_k^2, \eta_k^2 \quad \forall k\}$ be the optimal solution of LP_2. Now consider $\mathcal{Y}_3 = 2\mathcal{Y}_2$. Clearly, $\mathcal{Y}_3$ is a feasible solution of LP_3, as there no restrictions on the domain of variables. It should be noted that since the cost function is linear in x, y, the cost of feasible soluton of LP_3 is twice the cost of feasible solution of LP_2. Let $C_{LP_3(\mathcal{Y}_3)}, C_{LP_3}$ be the cost of LP_3 corresponding to Y_3 and optimal cost of LP_3 respectively. The following is trivially true:

$$C_{LP_2} = \frac{1}{2}C_{LP_3(\mathcal{Y}_3)} \geq \frac{1}{2}C_{LP_3}$$

When the same procedure is reversed, it follows immediately that, $C_{LP_3} \geq 2C_{LP_2}$. Hence

$$C_{LP_2} = \frac{1}{2}C_{LP_3}. \tag{3.36}$$

It should be noted that constraints in LP_3 are decoupled into following two sets of variables: x, p, f, ψ_k and y, q, g, η_k. This implies that the objective function can be separately minimized. We shall denote the $LP(\mathcal{X})$ as the linear program which is minimized over x, p, f, ψ_k and $LP(\mathcal{Y})$ as the linear program which is minimized over y, q, g, η_k. It should be noted T is partitioned into $\mathcal{X}$ and $\mathcal{Y}$ as defined earlier. Hence,

$$C_{LP_3} = C_{LP(\mathcal{X})} + C_{LP(\mathcal{Y})}. \tag{3.37}$$

1. Steiner tree (ST) problem

For establishing the result, we relate $LP(\mathcal{X})$ to a well-known problem in optimization literature [11], known as the Steiner tree problem. Given an undirected graph $G = (V, E)$ with edge costs and subset of nodes, $R \subset V$, the ST problem is to find minimum weight tree spanning all the nodes in R. The resulting tree may or may not have the optional nodes (i.e, nodes in $V \backslash R$). The optional nodes are often referred to as the Steiner nodes.

Lets consider the following integer programming formulation of Steiner Tree with terminal nodes $\mathcal{R}$ (ST_{cut}):

$$Z_{cut}(R) = \min \sum_{e \in E} c_e x_e,$$

subject to

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for} \quad S \subset V, S \cap R \neq \phi, R \backslash S \neq \phi \tag{3.38}$$

$$x_e \in \{0, 1\} \quad \text{for} e \in E \tag{3.39}$$

We relate the above integer formulation with an equivalent multi-commodity flow formulation of the Steiner tree. The multi-commodity formulation relies on using shipping multiple commodities from a depot instead of a single commodity. Each commodity being shipped has a specific target at which its delivered. Lets choose a node $d \in R$, call it a depot. We formulate a flow based formulation of the Steiner tree problem and show that it is similar to the cut-based formulation (ST_{cut}). The idea is to ship atleast one unit of commodity corresponding to each node from the depot (a chosen terminal node). It is necessarily that all the terminal nodes receive their commodities. The optional ones if needed, will receive their commodity too. Consider the following formulation (ST_{mcflow}). As explained earlier let p_{ij}^k denote the k^{th} commodity passing from node i to node j.

$$Z_{mcflow} = \min \sum_{e \in E} c_e x_e,$$

subject to

$$0 \leq p_{ij}^u \leq f_{ij} \quad \forall i, j \in V, u \in V \backslash \{d\} \tag{3.40}$$

$$f_{ij} + f_{ji} = x_e \quad \forall e \in E, \tag{3.41}$$

$$\sum_j p_{ij}^k - p_{ji}^k = \begin{cases} \psi_k & \forall k \in V \backslash \{d\}, i = d, j \in V \\ 0 & \forall i, k \in V \backslash \{d\}, i \neq k, j \in V \\ -\psi_k & \forall i, k \in V \backslash \{d\}, i = k, j \in V \end{cases} \tag{3.42}$$

$$\psi_k \geq 1 \forall \quad k \in R \backslash \{d\} \tag{3.43}$$

$$\psi \geq 0, f \geq 0 \text{ and } x_e \in Z^+ \text{ for all arcs } e \in E \tag{3.44}$$

We establish shortly that equations [3.40] - [3.43] and equation [3.38] are equivalent.

Consider any set $S \subset V$ such that $S \cap R \neq \phi$ and $R \backslash S \neq \phi$. Without loss of generality lets assume that the depot d, belongs to set S [1] Clearly, S contains atleast one node in R and not all of them.

The flow model (ST_{mcflow}) implies a flow of unit commodity from the depot node in S to all the terminal nodes. This implies that all the terminal nodes (there is atleast one of them) in $\bar{S}$ receive their shipment from depot. By virtue of capacity constraint [3.40], such a flow is feasible only if for every cut $\delta(S)$, that separates the depot in S from the other terminal nodes in $\bar{S}$ contains atleast one edge(capacity). Hence, if x_e is a feasible solution to ST_{mcflow}, i.e., constraints [3.40] - [3.43] of the flow formulation are met, x_e is a feasible solution for cut formulation.

From the max-flow min-cut theorem, $\sum_{e \in \delta(S)} x_e \geq 1$ with chosen S, if and only if the graph has a feasible flow of atleast one unit from the depot ($d \in S$) to terminal node in $\bar{(S)}$. Hence, a feasible solution to the cut formulation, is also a feasible solution to the flow formulation.

If one choses, $R = \mathcal{X} \cup d_1$, it is easy to see that $LP(\mathcal{X})$ formulated earlier is indeed LP relaxation of $Z_{mcflow}(R)$. Since, we proved that the cut formulation Z_{cut} and the multi-commodity flow formulation Z_{mcflow}, are equivalent, we conclude that the optimal value of $LP(\mathcal{X})$ is same as the optimal value of the LP relaxation of $Z_{cut}(R)$. In the arguments that follow, we use the cut formulation. Similarly if one choses $R' = \mathcal{Y} \cup d_2$, it follows that the optimal value of the $LP(\mathcal{Y})$ is same as optimal value of LP relaxation of $Z_{cut}(R')$. Let us call is $Z^*_{LP}(R')$. Hence we have the following:

$$C_{LP(\mathcal{Y})} = Z^*_{LP}(R') \tag{3.45}$$

where, $R' = \mathcal{Y} \cup d_2$. $\mathcal{Y}$ represents the subset of targets whose commodities must be

[1]If it does not belong to S, the rest of the argument holds by considering the S'.

shipped by depot 2. To establish the result in the paper, we seek an important result concerning the LP relaxation of the Steiner tree problem, which is well-known in the literature.

The cut formulation Z_{cut} is a special case of the the following generalized problem.

$$Z = \min \sum_{e \in E} c_e x_e,$$

subject to

$$\sum_{e \in \delta(S)} x_e \geq f(S) \quad \text{for} \quad S \subset V \tag{3.46}$$

$$x_e \in \{0, 1\} \quad \text{for} e \in E \tag{3.47}$$

It is obvious that we obtain the Z_{cut} formulation of Steiner problem by taking $f(S) = 1$ whenever $S \subset V, S \cap R \neq \phi, R \backslash S \neq \phi$.

We note here that the above generalized problem when $f(S)$ satisfies certain properties is proven in [56] to have a 2- approximation algorithm. If the edge costs satisfy triangle inequality, i.e., for any three vertices u, v, w, $c(u, v) \leq c(u, w) + c(v, w)$, then output of the algorithm is the minimum spanning tree on the terminal nodes (R). It is proven that the cost of $MST(R)$ is within twice the cost of LP relaxation and hence within twice the cost of the optimal integral solution.

Let Z^*_{LP} be the cost of optimal solution to the LP relaxation of the Integer program Z_{cut}. We know that $Z^*_{LP} \leq Z$. Let R be the set of terminal nodes in V. The following is a theorem from [56].

Theorem 3. *The algorithm produces a set of edges $\mathcal{F}'$ whose incidence vector of edges if feasible for integer program, and such that*

$$\sum_{e \in \mathcal{F}'} c_e \leq (2 - \frac{2}{|R|}) Z^*_{LP}(|R|) \leq (2 - \frac{2}{|R|}) Z.$$

Let $C_{MST}(R)$ be the cost of minimum spanning tree on R. Hence, we have the following result as a straight forward deduction from [56].

$$C_{MST}(R) \leq 2Z^*_{LP}(R) \leq 2Z_{cut}(R). \tag{3.48}$$

For more details on the algorithm and the proof for the approximation factor, the readers are referred to [[56], [11]]. We now shall establish the main result of this work.

From the earlier inequalities [3.30, 3.33, 3.36, 3.37], we have the following:

$$C_{LP^*} \geq \frac{1}{2}(C_{LP(\mathcal{X})} + C_{LP(\mathcal{Y})})$$

But my virtue of [3.45] we have the following:

$$C_{LP^*} \geq \frac{1}{2}(Z^*_{LP}(\mathcal{X} \cup d_1) + Z^*_{LP}(\mathcal{Y} \cup d_2)) \tag{3.49}$$

where C_{LP^*} is the cost of LP relaxation of HMSF problem and$Z^*_{LP}(\mathcal{X} \cup d_1)$ represents the cost of LP relaxation of Steiner problem with essential nodes as $\mathcal{X} \cup d_1$.

Using equations [3.48, 3.49] and appropriate substitution the following can be deduced:

$$4HMSF^* \geq 4C_{LP^*} \geq \underbrace{(C_{MST}(\mathcal{X} \cup d_1) + C_{MST}(\mathcal{Y} \cup d_2))}_{C_{feasible}} \tag{3.50}$$

The set of edges from $MST(\mathcal{X} \cup d_1) \cup MST(\mathcal{Y} \cup d_2)$ are obviously a feasible solution to the integer program *HMSF*, since $\mathcal{X}$, $\mathcal{Y}$ is *some* partition of the target set. Therefore, $C_{feasible} \geq HMSF^*$. Hence, we established that ***HMSF*** is 4-approx.

G. Conclusions

In this chapter, we considered the motion planning of structurally heterogeneous collection of UVs. We begin with a simplified problem when the collection consists of only two heterogeneous UVs. Specifically, the routing problem we addressed is 2-depot, Heterogeneous Traveling Salesman Problem (2-HTSP). The $\frac{3}{2}$ approximation algorithm presented in this chapter for 2-HTSP is novel and its extensions to related variants is also novel. The formulation of 2-HMSF is novel and the 4-approximation algorithm formulated for this problem is first of it's kind.

CHAPTER IV

APPROXIMATION ALGORITHMS FOR VARIANTS OF A HETEROGENOUS MULTIPLE DEPOT HAMILTONIAN PATH PROBLEM

A. Introduction

In the previous chapter, two types of heterogeneity for the vehicles are considered, i.e., *structural* heterogeneity and *functional* heterogeneity. In this chapter *, we assume that the fleet considered for motion planning is *structurally* homogeneous but differ from each other in terms of *sensing* capabilities (functional heterogeneity).

Specifically, there are m UVs that must collectively visit n targets. It is assumed that the vehicles are identical dynamically and hence, the cost of traveling from any target A to any other target B with identical headings is the same for every UV in the collection. The UVs differ from each other in their sensing capabilities and accordingly, we categorize the targets into three disjoint subsets:

1. **Category I**: Subset of targets which can be visited by any UV.

2. **Category II**: Subset of targets that can be visited only by a specific UV or a subset of UVs. This arises in a scenario where the technology/equipment to accomplish the desired task on a target is available only to a subset of UVs. Also, if a group of targets form a cluster $i.e.$, they are very close to each other in terms of distance, it might be economical to let one UV perform all the tasks on these group of targets.

3. **Category III**: Subset of targets that are unsuitable to be visited by a particular UV or a subset of UVs.

---

*Part of this chapter was reprinted with permission from "3-Approximation Algorithm for a Two Depot, Heterogeneous Traveling Salesman Problem," by Sai Yadlapalli, Sivakumar Rathinam and Swaroop Darbhn, Accepted for Publication in Operation Research Letters, Copyright © 2010 by Elsevier http://www.elsevier.com/

In this chapter, the following problem is considered: Given a set of depots (starting locations of UVs) and their corresponding terminals (ending locations of UVs) find a path for each vehicle such that

- the path of each UV starts from its respective depot and ends at the corresponding terminal,
- each target is visited exactly once by some vehicle,
- the assignment constraints are satisfied and,
- the total cost of the paths of all the UVs is a minimum.

There are several applications ([47],[49],[50],[26], [60],[54]) where the above routing problem arise. This problem is a generalization of the Hamiltonian Path Problem (HPP) and its closely related Traveling Salesman Problem (TSP) and is NP-Hard. The generalizations of the HPP and TSP have received significant attention in the field of Combinatorial Optimization ([36],[12],[11],[15]). Because the problem is NP-hard, one may not expect to find an optimal solution with a running time guarantee that is polynomial in the size of the problem. The focus in the chapter is on arriving at approximation algorithms, which are polynomial time algorithms but relax the requirement of optimality; however, they provide bounds on the deviation of the cost of the suboptimal solution from the optimal cost without ever computing the optimal cost. An $\alpha-$approximation algorithm [11] is an algorithm that

- has a polynomial-time running time, and
- returns a solution whose cost is within α times the optimal cost.

The cost of traveling from an origin to a target directly for each vehicle is assumed to be no more expensive than the cost of traveling from the same origin to the target through an intermediate location. The costs satisfy the triangle inequality if they satisfy the above property. It is known that there cannot exist a constant factor approximation algorithm for a HPP or a TSP if the triangle inequality is not satisfied unless $P = NP$. For this reason, this property holds for the cost associated with travel for every UV.

1. Literature review

There are a few approximation algorithms that are available for the variants of the TSP and the HPP. The symmetric TSP has two well known approximation algorithms - the 2 approximation algorithm obtained by doubling the minimum spanning tree (MST) and the 1.5 approximation algorithm of Christofides obtained through the construction of MST and a minimum perfect matching of vertices of MST with odd degree [52].

The best approximation algorithm currently available for the single HPP (a path that contains each vertex exactly once of minimum total cost) was proposed by Hoogeveen [15]. In [15], he proposed an approximation algorithm for three variants of single HPP that depend on the choice of the endpoints of the path. Hoogeveen modified the Christofides algorithm, and provided a $\frac{3}{2}-$approximation algorithm for the variant of the HPP problem when at most one endpoint is fixed and proposed a $\frac{5}{3}-$approximation algorithm when both endpoints are fixed.

Rathinam et al. have provided $2-$approximation algorithms for variants of the homogeneous, multiple TSP and HPP in ([26],[53],[60]). A $\frac{3}{2}-$approximation algorithm was also developed for two variants of a 2-depot Hamiltonian path problem in [54] when the costs are symmetric and satisfy the triangle inequality.

Until now, there is no constant factor approximation algorithm for any variant of the heterogeneous, multiple HPP. The contribution here, is in providing a first constant factor approximation algorithm for the variant of HPP considered. In this chapter, a $\frac{11}{3}$-approximation algorithm for the multiple depot-terminal HPP with functional heterogeneity constraints is presented. In the special case when the locations of the terminals coincides with their respective depots, the approximation factor of the proposed algorithm reduces to 3.5. This approximation factor of 3.5 also holds true for other variants of the heterogeneous, multiple depot HPP when at most one endpoint is specified for each vehicle.

B. Problem formulation

Let the set of vertices D and T represent all the distinct depots and terminals respectively. Let $|D|$ = $|T|$. Assume that there is an UV initially located at each of the depots. For every depot, $d_i \in$ D, let there exist exactly one terminal vertex denoted by $t_i \in T$. We require that each UV starting at its depot end its path at its corresponding (fixed) terminal. Let $p := |D|$ denote the total number of depots.

First, consider all the targets belonging to categories in I and II. It is assumed that all the targets are distinct, $i.e.$, there are no two targets present at the same location. Let the set of targets which can be only visited by the i^{th} UV that starts at $d_i \in D$ be represented by A_i. Let us define A = $A_1 \cup A_2 ... \cup A_p$. It is also *assumed* that all the A_i's are disjoint, i.e., $A_1 \cap A_2 ... \cap A_p = \phi$. Let the common set of targets which can be reached by all UVs be F.

Define a graph (V, E) with $V = D \bigcup T \bigcup A \bigcup F$ denoting the set of all the vertices and $E := V \times V$ denoting the set of all the edges joining any two vertices in V. Let $c(V_i, V_j)$ or simply c_{ij} represent the cost of traveling from vertex V_i to vertex V_j for all $V_i, V_j \in V$. We further assume that the costs are positive, symmetric and satisfy the triangle inequality, i.e., for all $V_i, V_j, V_k \in V$ and $i \neq j \neq k$, $C_{ij} + C_{jk} \geq C_{ik}$. The symmetry of costs may not hold true for all UVs in general; however, by relaxing motion constraints, one can obtain symmetry in the cost of travel between any two targets. This is especially so when the constraint associated with forward travel in a Dubins' vehicle is relaxed, one gets a Reed-

Shepp vehicle and the costs are symmetric. While such a relaxation may not solve the original problem, it serves two purposes: firstly, it provides a lower bound for the optimal solution, and secondly, if the distances between targets is sufficiently large compared to the turning radius as in the case of Dubins' vehicle, the asymmetry in the cost is not so significant compared to the Euclidean distance between the targets. In such circumstances, the proposed approximation algorithms provide "adequate" feasible solutions.

A path for a UV is a sequence of vertices visited by the vehicle. The first vertex is called the *start vertex* and the last vertex in the sequence is called the *end vertex*. A path with no repeated vertices is called a **simple path**. In this work, we refer simple paths as simply paths. However, it should be noted that since the costs satisfy triangle inequality, it is always possible to shortcut a repeated vertex and obtain another path of lower cost spanning (or visiting) all the vertices.

A path traveled by the i^{th} UV is an ordered set, $PATH_i$, and can be represented by the form $\{d_i, V_{i_1}, ..., V_{i_r}, t_i\}$, where $V_{i_l} \in A \bigcup F$, $l = 1, ..., r$ corresponds to the r distinct targets being visited in that sequence by the i^{th} UV. These set of targets being visited by the i^{th} UV must include the set A_i (which can be only visited by i^{th} UV and subset (could be empty) of common targets, F. The cost of traveling $PATH_i$ is defined as $C(PATH_i) = c_{d_i i_1} + \sum_{j=1}^{j=r-1} c_{i_k i_{k+1}} + c_{i_r t_i}$. The Combinatorial Motion planning Problem (**CMP**) addressed in this article is to find a $PATH_i$ for the i^{th} UV $(i = 1, \cdots, p)$ such that each target is visited exactly once, all the assignment constraints are satisfied and the total cost defined by $\sum_{i=1}^{i=p} C(PATH_i)$ is minimized.

In section D, we also address an generalized variant of the **CMP** where the targets fall under categories II and III. Let the set of targets which cannot be visited by the i^{th} UV be denoted by N_i. We also *assume* that all the N_i's are disjoint, i.e., $N_1 \cap N_2 ... \cap N_p = \phi$. As defined before, let the set of targets which must be visited only by the i^{th} UV be A_i. The Generalized Combinatorial Motion planning Problem (**GCMP**) addressed in this work is to find a $PATH_i$ for the i^{th} UV $(i = 1, \cdots, p)$ such that

- each target is visited exactly once,
- each target in A_i is visited by the i^{th} UV for all $i = 1, \cdots, p$
- no target in N_i is visited by the i^{th} UV for all $i = 1, \cdots, p$, and,
- the total cost defined by $\sum_{i=1}^{i=p} C(PATH_i)$ is minimized.

C. Approximation algorithm for the **CMP**

Here, we present an algorithm, $Approx_{cmp}$, which constructs a feasible solution to the **CMP**. We later prove that this algorithm produces a solution with an approximation factor of $\frac{11}{3}$. $Approx_{cmp}$ is as follows:

1. For each $i \in 1, \cdots, p$, do the following:

   - Consider the subset of vertices $S_i = \{d_i\} \cup A_i \cup \{t_i\}\ \forall i = \{1.....p\}$, where d_i and t_i are the depot and terminal vertices corresponding to the i^{th} UV. Compute a feasible depot-terminal path, HPP_i, that starts from d_i and ends at t_i using the Hoogeveen's algorithm [15]. Let E_{HPP_i} be the set of all the edges present in HPP_i.

   Let $E_{HPP} = \bigcup_{i=1}^{p} E_{HPP_i}$. Let the total cost of these paths be denoted by $C_{HPP} = \sum_{i=1}^{i=p} C_{(HPP_i)}$.

2. In this step we distribute the common targets, F, among all the UVs. After the distribution, we will construct a tour for each UV that starts at its depot and visits its assigned set of common targets. The algorithm for distributing the common targets among the UVs is as follows:

Consider the set $M = D \cup F$. Assign zero costs to all the edges among the depots. For the rest of edges retain the costs assigned earlier. Now, construct a Minimum Spanning Tree (MST) on M with the assigned costs using Kruskal's algorithm. Truncate all the zero cost edges (among depots) in the resultant MST. This results in a forest with exactly p connected components. Each of the connected component has exactly one depot in it. (This follows from the fact that, during each iteration, the Kruskal's algorithm adds a (non-used) cheapest, cost edge to the solution such that no cycle is formed among all the added edges in the solution. Therefore, there are exactly $|p-1|$ zero cost edges joining the p depots in the solution.) Let E_F be the set of the remaining edges after removing all the zero cost edges from the MST. E_F corresponds to a forest with p trees where each tree contains one depot. Also, let E_{F_i} be the set of edges present in the i^{th} tree.

3. Double the edges of E_{F_i}. Since E_{F_i} is a tree, doubling the edges of E_{F_i} would result in a connected, Eulerian graph. Therefore, one can find an tour (T_{F_i}) by short-cutting the edges in the Eulerian tour. The cost of this tour must be at most twice the cost of the edges in E_{F_i} since the costs satisfy the triangle inequality.

4. Consider the set of edges denoted in $T_{F_i} \cup HPP_i$. By construction, there are *exactly* three edges incident on d_i where *one* belongs to the path HPP_i and *two* belong to the tour, T_{F_i}. By short-cutting an edge from T_{F_i} and an edge that belongs to HPP_i one can form a path P_i that starts from depot d_i, ends at terminal t_i and visits all the targets in A_i and F_i. This short cutting procedure is illustrated later in an example in section E. Let $P = \cup_{i=1}^{i=p} P_i$. Since P is a collection of edge-disjoint simple paths and satisfies all the constraints, P is a feasible solution to **CMP**.

The following theorem establishes the approximation ratio of the above algorithm.

Theorem 4. *The approximation factor of $Approx_{cmp}$ is $\frac{11}{3}$.*

Proof. First, we will prove that the running time of $Approx_{cmp}$ is a polynomial function of the number of targets and depots. The number of steps required by $Approx_{cmp}$ is dominated by the computations in steps 1 and 2 of the algorithm. Step 1 of $Approx_{cmp}$ uses the Hoogeveen algorithm which requires $O(m^3)$ steps where m is the total number of targets. Step 2 of $Approx_{cmp}$ uses the Kruskal's algorithm which requires $O((m+p)^2 log(m+p))$ steps to compute. Therefore, the running time of $Approx_{cmp}$ is a polynomial function of the number of targets and depots.

Now, we will prove the guarantee on the quality of the solutions. Let $\mathcal{OPT}$ denote an optimal solution to the **CMP** and let $C_{\mathcal{OPT}}$ denote its corresponding cost. Let the optimal path corresponding to the UV at depot d_i in $\mathcal{OPT}$ be $\mathcal{OPT}_i$.

We will now bound the costs of all the HPP's found in step 1 of $Approx_{cmp}$. Consider the Single Depot-Terminal HPP restricted to the set $S_i = \{d_i\} \cup A_i \cup \{t_i\}$. Let HPP_i^* be an optimal solution to this problem. Note that the HPP_i found in step 1 of $Approx_{cmp}$ is a feasible solution to the single Depot-Terminal HPP on S_i. Also note that the path $\mathcal{OPT}_i$ visits each target in S_i in addition to some common targets from F. Since the costs satisfy the triangle inequality, by short-cutting all the common vertices in $\mathcal{OPT}_i$ that do not belong to S_i, one can easily conclude that:

$$C_{\mathcal{OPT}_\rangle} \geq C_{HPP_i^*} \geq \frac{3}{5} C_{HPP_i}. \tag{4.1}$$

The latter part of the above inequality follows from Hoogeveen's result for Single Depot-Terminal HPP. Summing the above result for all the vehicles, we get,

$$\frac{5}{3} C_{\mathcal{OPT}} \geq C_{HPP}. \tag{4.2}$$

In the following discussion, we will bound the costs of all the tours found in steps 2 and 3 of $Approx_{cmp}$. Note that the optimal path $\mathcal{OPT}_i$ visits some common vertices from F in addition to visiting each vertex in A_i. By short-cutting all the vertices in $t_i \cup A_i$ from $\mathcal{OPT}_i$, one can obtain a tree that spans the depot vertex d_i and all the common vertices in $\mathcal{OPT}_i$. Let the set of edges spanning this tree be $E_F^{\mathcal{OPT}_i}$. Let $E_F^{\mathcal{OPT}} = \cup_{i=1}^{i=p} E_F^{\mathcal{OPT}_i}$. The set of edges in $E_F^{\mathcal{OPT}}$ corresponds to a $p-$component forest that consists of a depot in each tree and spans all the common vertices in F. Since the costs satisfy the triangle inequality, it follows that

$$C_{\mathcal{OPT}} \geq C(E_F^{\mathcal{OPT}}) \geq C(E_F), \tag{4.3}$$

where $C(E_F)$ is the sum of the cost of edges in E_F (found in step 2 of $Approx_{cmp}$). From inequalities (4.2) and (4.3), we obtain:

$$\frac{11}{3} C_{\mathcal{OPT}} \geq C_{HPP} + 2C(E_F) \geq C_{HPP} + C(T_F). \tag{4.4}$$

In the above equation $C(T_F)$ is the total cost of the tours obtained by doubling the trees and short-cutting. From step 4 of $Approx_{cmp}$, we can deduce that

$$C_{HPP} + C(T_F) \geq C_P. \tag{4.5}$$

By combining Equations (4.4) and (4.5)

$$\frac{11}{3} C_{\mathcal{OPT}} \geq C_P \geq C_{\mathcal{OPT}}. \tag{4.6}$$

Hence proved. □

Remark 1. *The approximation factor of* $Approx_{cmp}$ *can be improved for the special case of the* ***CMP*** *when each location of each terminal coincides with its respective depot. In this case, instead of using Hoogeveen's [15] algorithm in step 1 of* $Approx_{cmp}$*, one can use the*

Christofides [52] algorithm for finding a path for each vehicle that starts and ends at its depot. Since the approximation factor of the Christofides algorithm is 1.5, the approximation factor of $Approx_{cmp}$ *for this special case reduces to* $2 + 1.5 = 3.5$*.*

Remark 2. *It is also easy to see that the* $Approx_{cmp}$ *can be easily extended to the variant of the* ***CMP*** *when the final vertex of each path is not specified. In this variant, instead of using the* $\frac{5}{3}-$*approximation algorithm by Hoogeveen in step 1 of* $Approx_{cmp}$*, one can use the 1.5-approximation algorithm by Hoogeveen [15] where the terminal vertex is not specified for a path. Therefore, the approximation factor of* $Approx_{cmp}$ *for this variant would be equal to* 3.5*.*

In Remark 1, the special case in which location of each terminal coincides with its respective depot, becomes a corresponding TOUR problem (CTP). More details to the claim made in Remark 1 are provided here:

Proof for Remark 1

1. Consider the subsets of vertices $Z_i = d_i \cup A_i \; \forall i = \{1.....p\}$. Compute a tour on each of Z_i using Christofides algorithm [52].

2. As in Step 2 of $Approx_{cmp}$, distribute common targets F among the depots. It should result in a p-component Minimum Spanning Forest (E_F).

3. Double all the edges in the Minimum Spanning Forest to obtain an Eulerian graph. Find an Eulerian tour in each component of the Eulerian graph and shortcut the edges to obtain a tour for each vehicle [11].

4. Combine the tours in Step 1 and Step 3 of this algorithm. By construction, each depot will have *four* edges adjacent on it. By using triangle inequality, one can shortcut these four edges such that only *two* edges are incident on the depot resulting in a feasible solution to the tour problem.

Let $Z = \cup_{i=1}^{i=p} Z_i$. Let us take a look at the optimal solution to the p-component tour problem on Z. This solution should be union of optimal solutions to tour problem on each of Z_i. The above statement holds true since all the targets in Z_i has to be visited by vehicle starting from d_i, and hence should be in the same tour as d_i. Let C_Z be the cost of optimal p-component tour problem and $C_{\mathcal{TOUR}}$ be the optimal cost of CTP. Let C_f be the cost of feasible solution found in Step 1 of the above algorithm. By virtue of Christofides algorithm, we have the following:

$$C_Z \leq C_f \leq \frac{3}{2} C_Z \leq \frac{3}{2} C_{\mathcal{TOUR}} \tag{4.7}$$

Let C_{TOUR_F} be the cost of tour formed by doubling p-component Minimum Spanning Forest (cost is C_{E_F}) and short-cutting it. The following holds true as well:

$$C_{TOUR_F} \leq 2C_{E_F} \leq 2C_{\mathcal{TOUR}} \tag{4.8}$$

Let $C_{feasible}$ be the cost of feasible solution to CTP formed in Step 4. By using Equation [4.7,4.9], it follows that:

$$C_{feasible} \leq C_f + C_{TOUR_F} \leq C_f + 2C_{E_F} \leq 3.5 C_{\mathcal{TOUR}} \tag{4.9}$$

Hence proved.

D. Approximation algorithm for the **GCMP**

In this section, a *generalized* version of the problem (**GCMP**) is considered when the possible set of targets fall under categories II and III. This problem admits a $(2p + \frac{5}{3})$-approximation algorithm. The following are the main steps of the approximation algorithm for the **GCMP**:

1. For $i = \{1.....p\}$, do the following:

- Consider the subset of target vertices denoted by N_i. Note that each target in N_i must not be visited by the i^{th} vehicle. Therefore, using the spanning tree algorithm discussed in steps 2 and 3 of $Approx_{cmp}$, find a tour for all the vehicles at depots $d\backslash\{d_i\}$ such that each target in N_i is visited by any vehicle other than the vehicle at the i^{th} depot.

- Find a depot-terminal path that starts at d_i, visits the set of targets in A_i and terminates at t_i.

2. For each vehicle,

   - add the edges in its corresponding tours and the path obtained during the previous step. If any depot vertex d_i is visited more than once, one can always shortcut the edges so that a path can be obtained for the vehicle that starts at d_i, visits all the targets in A_i and the tours, and ends at t_i.

Using a similar proof technique as in Theorem 1, it can be shown that the approximation factor of the above algorithm is $2p + \frac{5}{3}$.

E. Illustration of the algorithm $\text{Approx}_{\text{cmp}}$ with an example

An instance of the depots, targets and terminals in shown in Figure 9. The blue colored star denotes depot 1 and the black colored star denotes depot 2. The square shaped vertices are terminals and are colored respectively. The red colored vertices denote the common targets that can be visited by any one of the vehicles. Step 1 of $Approx_{cmp}$ is illustrated in figure 10. In this step, a feasible solution to single depot-terminal HPP is constructed using Hoogeveen's algorithm on each of vertices constituting a depot, its corresponding terminal and the vertices that *must be visited* by the corresponding vehicle.

The distribution of the common targets among the depots is shown in figure 11 (Step

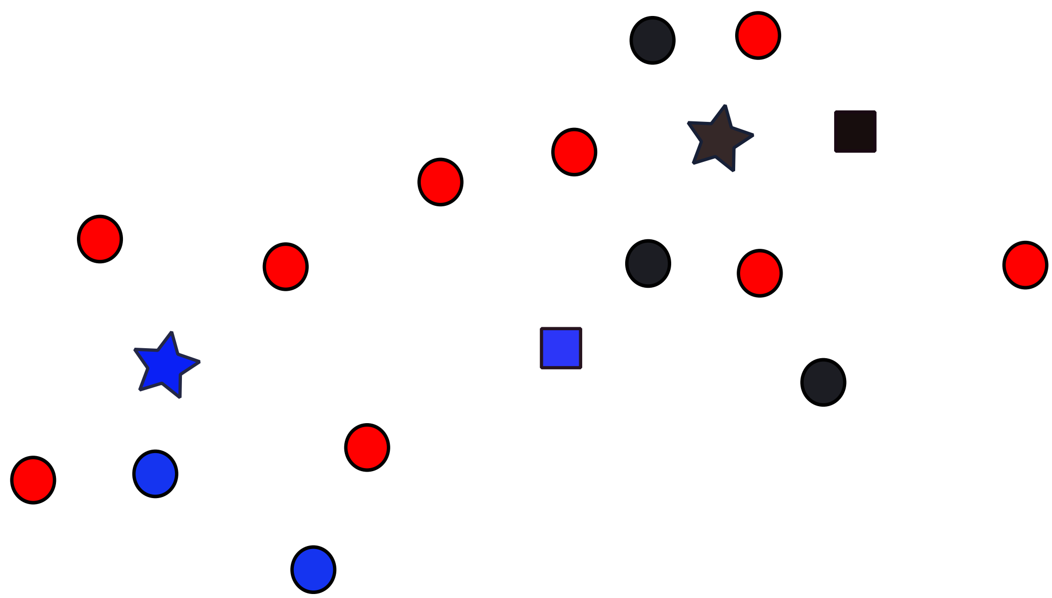

Fig. 9. An instance of depots, terminals and targets.

2 of $Approx_{cmp}$). The resulting graph consists of two components with a depot in each of them. By doubling and short-cutting the resulting graph, we obtain a tour for each vehicle as shown in Figure 12. Figure 13 shows the graph resulting from adding all the tours and paths constructed thus far.

Figure 14 shows the short-cutting process. The *dashes* are the edges being shortcut. One can notice that these two edges are incident on the depot where one of them belongs to the tour and the other to the path incident on the depot. The paths obtained after short-cutting these edges is shown in Figure 15. The final solution has two components and each component is a path starting from a depot and ending at its respective terminal.

1. Illustration of remark 1

The algorithm for the corresponding tour problem (CTP) is implemented on a set of vertices shown in figure 16. Using Christofides algorithm, the tours of each depot visiting its specific targets are constructed in figure 17. In figure 18, 2-component MSF (Minimum

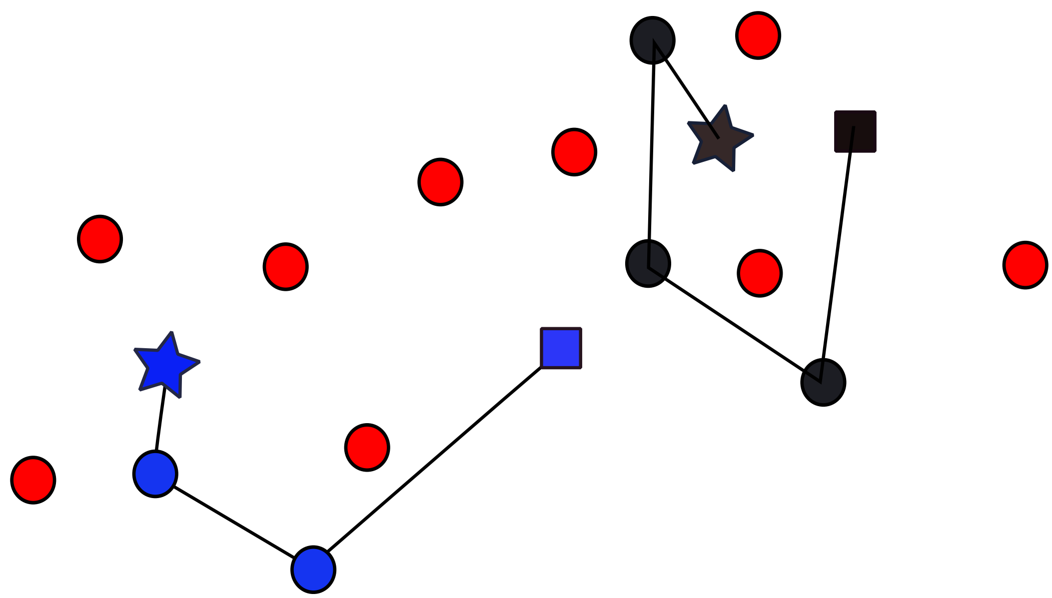

Fig. 10. Feasible Hamiltonian Path on depot, terminal and essential vertices using Hoogeveen's algorithm

Spanning Forest) is constructed by using a greedy algorithm which distributes the common targets among both the depots. The edges of this forest can be doubled and shortcut to form a tour. The resulting tour is shown in figure 19. This tour is not unique as short-cutting can be done in several ways. Figure 20 shows the combined solution of both the tours. And finally, by short-cutting an edge from both the tours incident on the depot, one can obtain the feasible solution to CTP as shown in Figure 21.

F. Conclusions

The focus of this chapter is on the first approximation algorithm for a variant of a Multiple Depot-Terminal Hamiltonian Path Problem when the costs are symmetric and satisfy the triangle inequality. A variant of the problem is considered where each vehicle starting from its depot should end its path at a terminal corresponding to the depot. The vehicles considered in this problem are dynamically identical. However, the complexity is enhanced

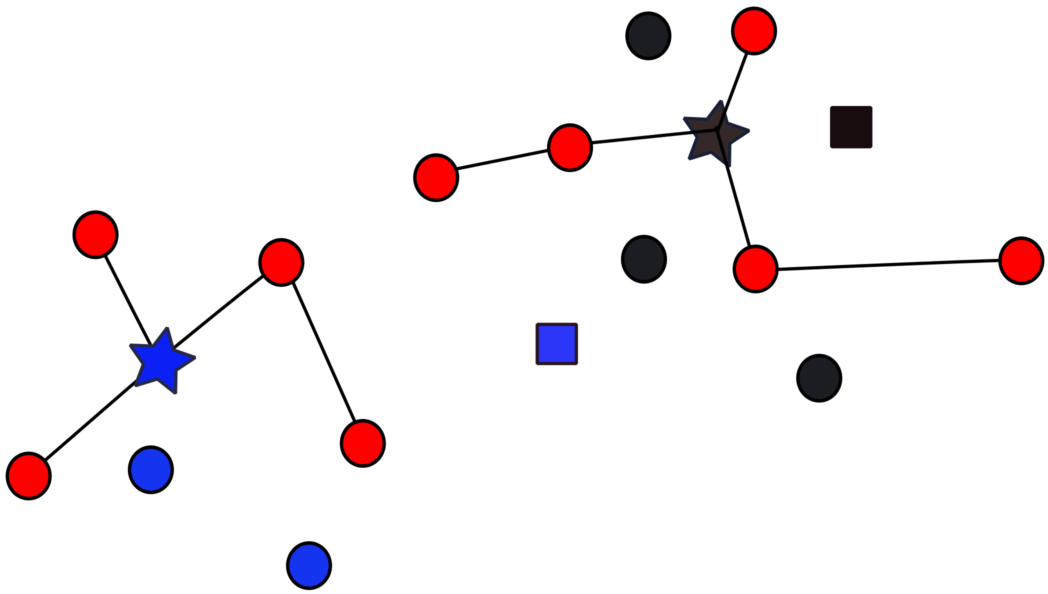

Fig. 11. Common targets are distributed among the UVs at the depots.

by including the possibility that their capabilities or equipment available onboard could be different. Currently, the proposed algorithm for the generalized version of the problem (**GCMP**) has an approximation factor of $2p + \frac{5}{3}$ where p is the number of vehicles. This approximation factor is essentially dependent on the number of vehicles used.

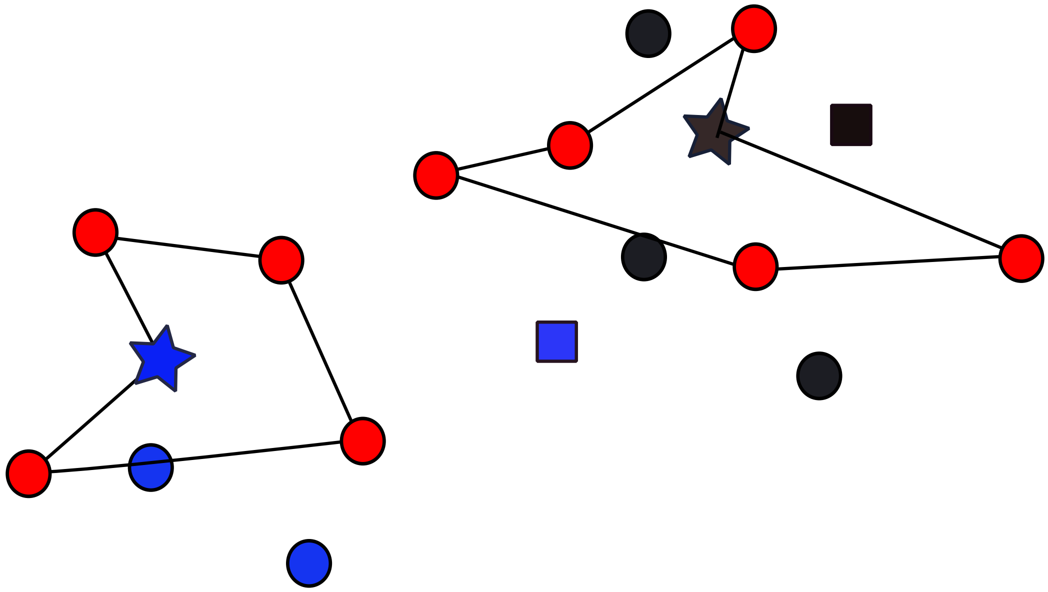

Fig. 12. Doubling the edges and short-cutting to tours

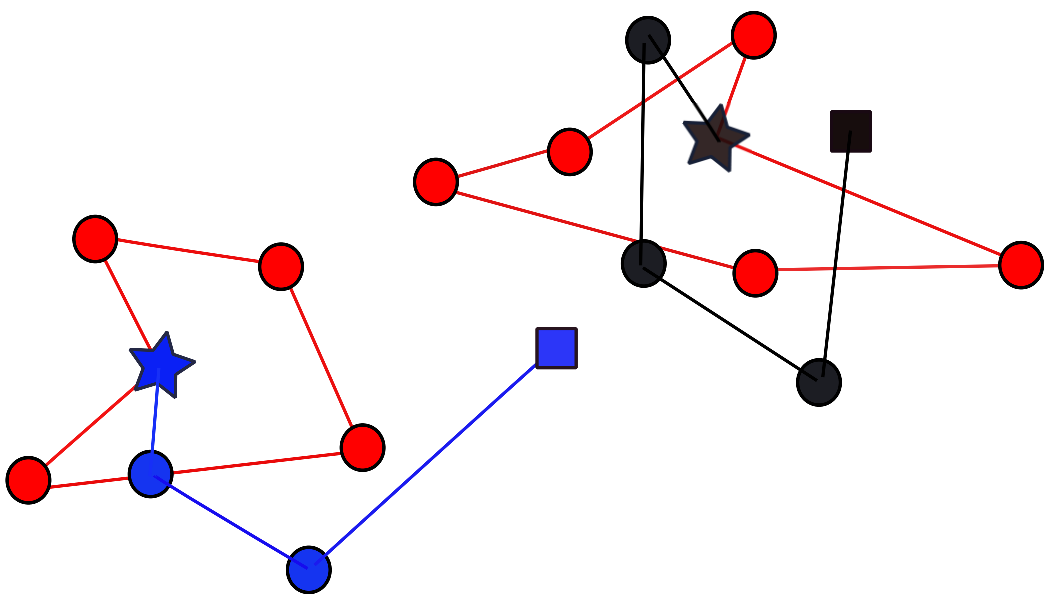

Fig. 13. Combining edges from path and tours.

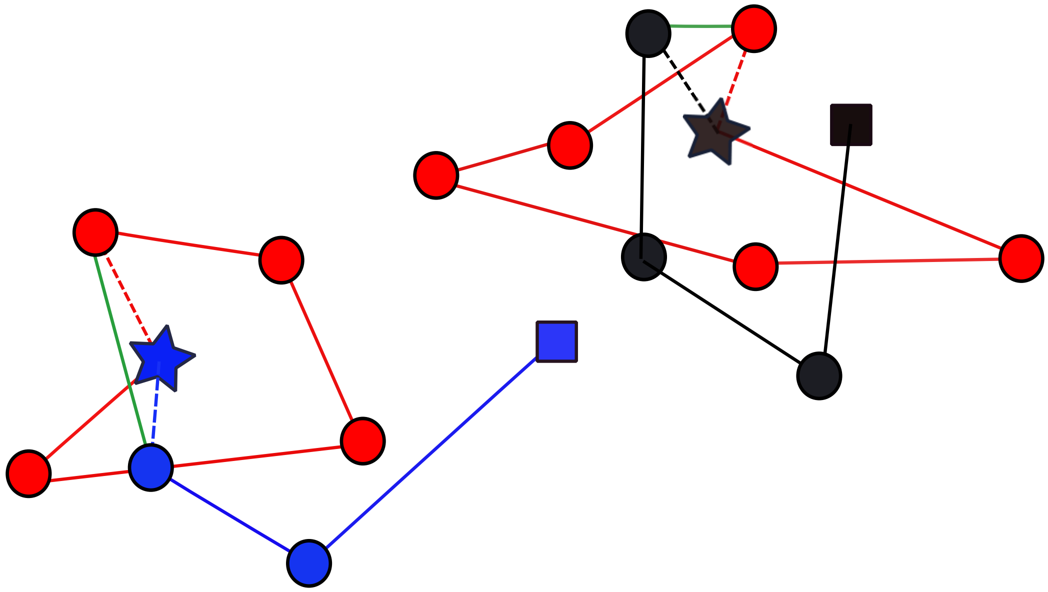

Fig. 14. Short-cutting edges adjacent to each depot to create feasible solution.

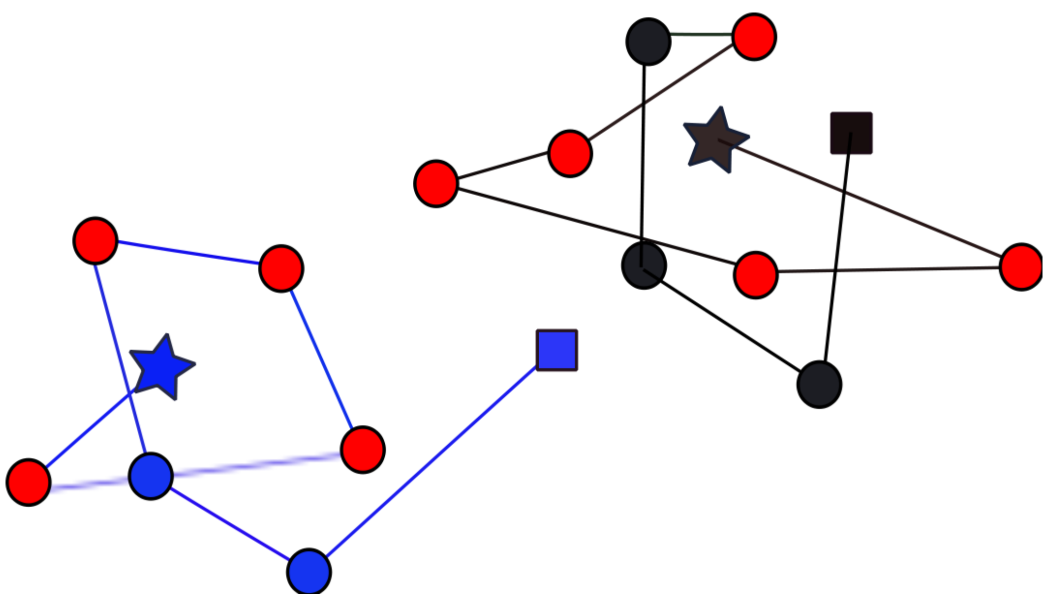

Fig. 15. Feasible solution obtained using the $Approx_{cmp}$.

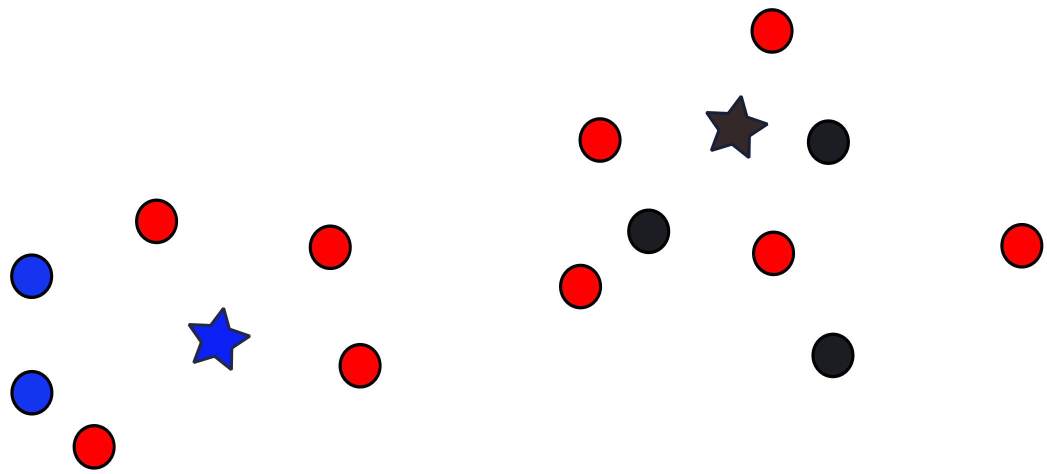

Fig. 16. Target and Depots distribution for CTP

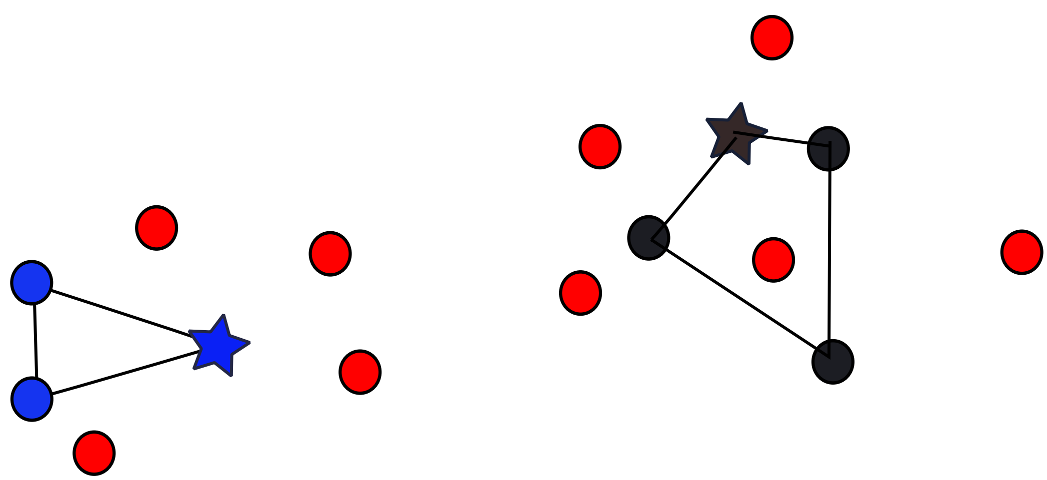

Fig. 17. Using Christofides algorithm on each depot’s specific targets vertices

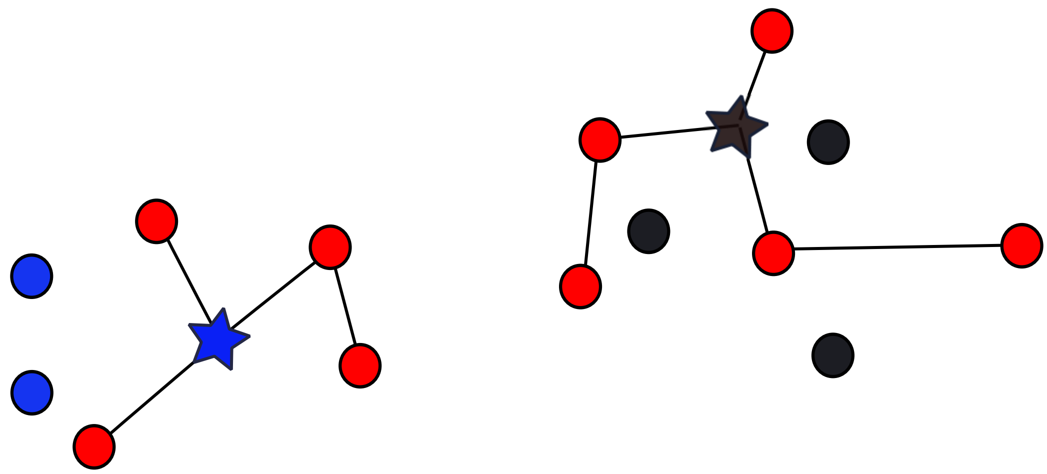

Fig. 18. Construction of 2-component Minimum Spanning Forest

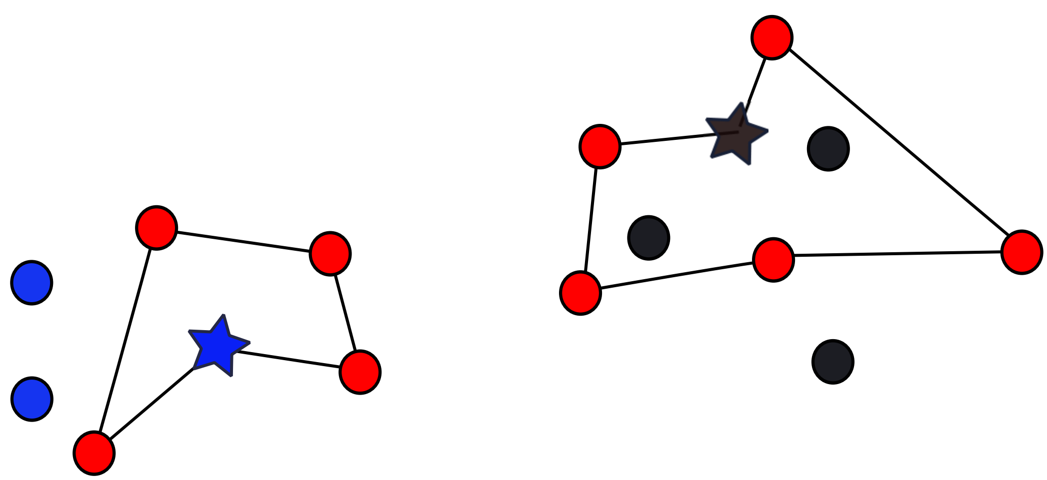

Fig. 19. Construction of tour by doubling each component and short-cutting

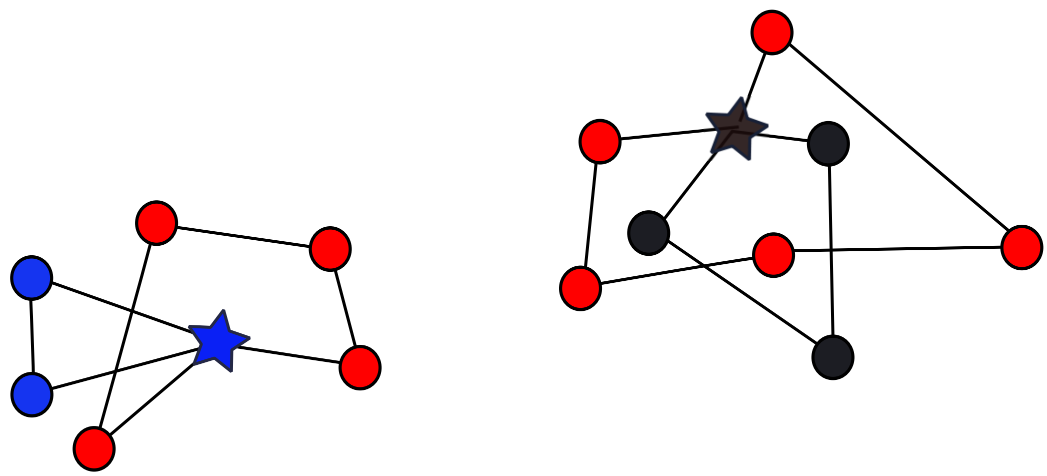

Fig. 20. Combing both the tours constructed

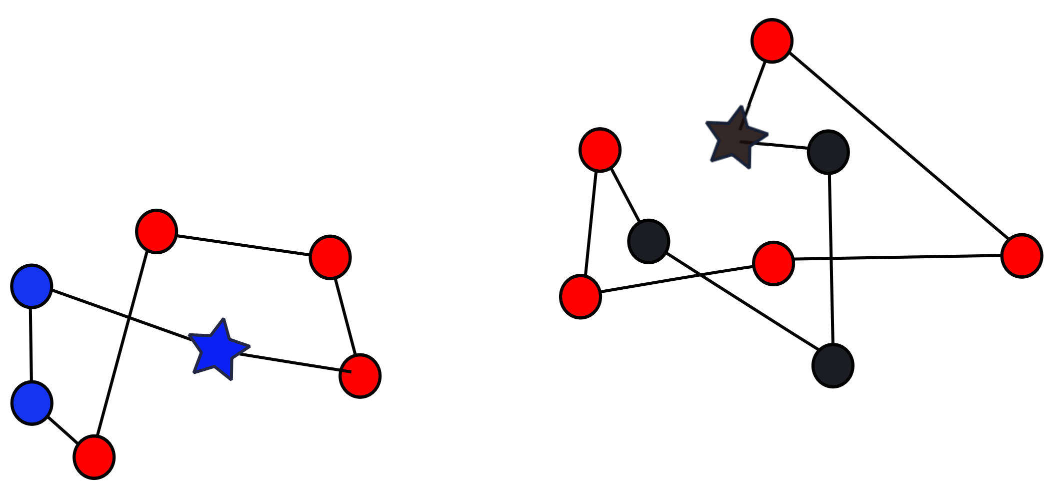

Fig. 21. Construction of feasible solution by short-cutting one edge from both tours incident on a depot

CHAPTER V

CONCLUSIONS AND FUTURE WORK

This thesis, as a whole, was devoted to the study of motion planning problem of m (possibly heterogeneous) UVs through n points in a plane. In this thesis, we assumed a simple model of the UV, with the only constraint being that the rate of change of heading angle is bounded. As a consequence, a UV traveling at a constant speed has a minimum turning radius.

In Chapter II, we considered a collection of homogeneous vehicles and provided an algorithm for determining a sub-optimal motion plan, i.e., the sequence of targets to be visited by each UV and the associated heading angles. The results obtained by this algorithm were promising when the distances between the targets is reasonably large compared to the the minimum turning radius. In other words, this algorithm produced feasible solutions of reasonably high quality when the coupling between the discrete and continuous optimization problems is not that strong.

In Chapters III and IV, we focussed on the development of approximation algorithms for a heterogeneous collection of UVs. In Chapter III, the main focus was on the development of an approximation algorithm for a collection of structurally heterogeneous UVs. In Chapter IV, the main focus was on developing approximation algorithms for a collection of functionally heterogeneous vehicles. In Chapter IV, a *path-type* problem was considered, where a vehicle starting from a depot, after visiting the assigned targets, can ends its path at a corresponding *terminal* or a target. In the sections to follow, the main contributions of each chapter will be highlighted.

A. Contribution of Chapter II

In Chapter I, the problem of motion planning of UVs with motion constraints was considered which was referred to as MDMTSP. The motion constraints were modeled as that Dubins' like vehicle or Reeds Schepp vehicle. In the first part of the chapter, an assumption was made that the targets were well separated, i.e., the distance between targets is at least twice the minimum turning radius of the vehicles. This condition is reasonable when the dimension of the sensor footprint is comparable or greater than the turning radius and it enables the separation of the combinatorial problem of finding the set of points to be visited by vehicles and the sequence in which they must be visited from the continuous optimization problem of determining the headings at each point.

A 2-step approach was adopted to solve MDMTSP when a Dubins' vehicle represents a salesman. The combinatorial aspect of the problem was solved by considering the Euclidean distances between the vertices. To solve the combinatorial problem, a generalization of Held-Karp's method for the MDMTSP was provided. Further, a procedure was shown for finding a lower bound. We presented numerical results to corroborate the effectiveness of the lower bound for various cases of the Euclidean MDMTSP. Once the sequence of the vertices to be visited was determined for each vehicle, the dynamic programming technique was then used to compute the optimal heading for the vehicle at each vertex.

Combinatorial Optimization problems admit different integer programming formulations. Given that there is a duality gap in the integer programming problems, it matters significantly which formulation is considered and which constraints in the formulation are penalized. Herein lies the novelty of the work we have proposed.

1. The Binary Program formulated for this problem generalizes the Dantzig-Fulkerson-Johnson formulation for MDMTSP.

2. The generalization of Held-Karp's procedure for this problem.

3. In application of Lagrangian relaxation, the novelty of the work conducted in this chapter lies in the choice of constraints to be penalized. Moreover, the choice of the formulation and the constraints to be penalized in the formulation plays a critical role in reducing this gap.

4. The application of this combinatorial optimization technique and the use of dynamic programming to determine the optimal heading angles for the Dubins' vehicles.

B. Contribution of Chapter III

In Chapter III, an important problem of 2-Depot Heterogeneous TSP(2-HTSP) was addressed. 2-HTSP can be briefly stated as follows: Given a set of destinations and two heterogeneous vehicles that start from distinct depots, find a tour for each vehicle such that each destination is visited exactly once and the total cost of the tours of the vehicles is a minimum.

In Chapter III, we considered the vehicles available to be *structurally* heterogeneous. If the vehicles are *structurally* different, the cost of traveling between two destinations not only depends on the position of the destinations but also on the vehicle. The 2-HTSP is a generalization of the Single Traveling Salesman Problem (Single TSP) and is NP-Hard [36]. The following are the *contributions* of Chapter III:

1. A 3-approximation algorithm was introduced for the 2-HTSP when the costs associated with each vehicle satisfy the triangle inequality. This is the best approximation algorithm in literature for 2-HTSP to this point.

2. In general, the approach given in this chapter can be extended to obtain a $\frac{3m}{2}$-approximation algorithm for variants of a m-depot, Heterogeneous Traveling Salesman Problem(HTSP).

3. The approach adopted in the chapter can also be extended to obtain a 3-approximation algorithm for a 2 depot, Heterogeneous TSP where the objective is to minimize the maximum cost traveled by either of the vehicles (min-max 2-HTSP). The min-max problem admits the same approximation factor as 2-HTSP.

4. Relaxing the degree constraints of all the destinations in the formulation of the 2-HTSP would yield a corresponding heterogeneous spanning forest problem (2-HMSF) with two connected components. Therefore, an other way to develop an approximation algorithm for the 2-HTSP is to find the optimal heterogeneous spanning forest, and double and shortcut the edges in this forest to obtain tours for the vehicles. However, it is not yet clear if 2-HMSF is in the class of P or NP-Hard. However, a 4-approximation algorithm is presented in this thesis for the 2- heterogeneous spanning forest. This is *first* approximation algorithm for the construction of a suboptimal Heterogeneous Spanning Forest

Hence, future research directions could include using similar approaches for finding better algorithms for HMSF and 2-HTSP.

C. Contribution of Chapter IV

In Chapter IV, we considered UVs that are structurally homogeneous but have different capabilities. The cost of travel from i^{th} node to j^{th} node is the same for every UV in the collection. However, the fleet of UVs employed for the task differ from each other in terms of sensing abilities, thus introducing heterogeneity. We coined the word *functional* heterogeneity to represent this distinction amongst UVs. The following are the *contributions* of Chapter IV:

In Chapter IV, $\frac{11}{3}$ approximation algorithm was presented for a variant of Multiple Depot Heterogeneous Hamiltonian Path problem. It was assumed that the vehicles are identical dynamically and hence, the cost of traveling from any target A to any other target B with identical headings is the same for every UV in the collection. The UVs differ from each other in their sensing capabilities.

1. Up to the point of this thesis, there was no constant factor approximation algorithm for any variant of the heterogeneous, multiple HPP. The contribution here, is in providing a first constant factor approximation algorithm for the variant of HPP considered.

2. In this chapter, a $\frac{11}{3}$-approximation algorithm for the multiple depot-terminal HPP with functional heterogeneity constraints is presented. In the special case when the locations of the terminals coincides with their respective depots, the approximation factor of the proposed algorithm reduces to 3.5.

3. This approximation factor of 3.5 also holds true for other variants of the heterogeneous, multiple depot HPP when at most one endpoint is specified for each vehicle.

4. The proposed algorithm for the generalized version of the problem (**GCMP**) has an approximation factor of $2p+\frac{5}{3}$ where p is the number of vehicles. This approximation factor is essentially dependent on the number of vehicles used.

Future work could be directed at exploring the possibility of *constant* factor approximation algorithms for the generalized problem.

REFERENCES

[1] C. Schumacher, P. Chandler, and S. J. Rasmussen, "Task allocation for wide area search munitions," in *Proc. American Control Conference*, Anchorage, AK, May 2002, pp. 1917–1922.

[2] D. Gross, S. Rasmussen, P. Chandler, and G. Feitshans, "Cooperative operations in urban terrain (COUNTER)," in *Proc. SPIE Defense Security Symposium*, vol. 6249, Orlando, FL, Apr. 2006, p. 62490G.

[3] C. T. Cunningham and R. S. Roberts, "An adaptive path planning algorithm for cooperating unmanned air vehicles," in *Proc. 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp. 3981–3986.

[4] C. Alexopoulos and P. M. Griffin, "Path planning for a mobile robot," *IEEE Transactions on Systems, Man and Cybernautics*, vol. 22, p. 318322, Mar. 1992.

[5] S. Zak, "Survey of uav applications in civil markets," in *Proc. 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia, Jun. 2001, p. 11.

[6] 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, pp. 497–516, July 1957.

[7] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards." *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[8] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman & Co., 1990.

[9] T. H. Cormen, C. Stein, R. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: The MIT Press, 2001.

[10] R. M. Karp, "Complexity of computation," in *SIAM-AMS Proc.*, vol. 7, 1974.

[11] V. V. Vazirani, *Approximation Algorithms*. Berlin: Springer, July 2001.

[12] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*, ser. Lecture Notes in Computer Science (LNCS). Berlin, Heidelberg: Springer-Verlag, 1994, vol. 840.

[13] M. Held and R. Karp, "The Traveling-Salesman Problem and Minimum Spanning Trees," *Operations Research*, vol. 18, no. 6, pp. 1138–1162, 1970.

[14] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.

[15] J. A. Hoogeveen, "Analysis of christofides' heuristic: Some paths are more difficult than cycles," *Operations Research Letters*, vol. 10, no. 5, pp. 291 – 295, 1991.

[16] S. Arora, "Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems," *Journal of the ACM*, vol. 45, pp. 753–782, September 1998.

[17] D. S. Johnson and L. A. McGeoch, "Experimental Analysis of Heuristics for the STSP," in *Local Search in Combinatorial Optimization*. New York: Wiley & Sons, 2001.

[18] T. H. Cormen, C. E. Lieserson, and R. L. Rivest, *Introduction to Algorithms*. Newyork, NY: McGraw Hill, 1990.

[19] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization; Algorithms and Complexity*, 2nd ed. Mineola, NY: Dover Publications, 1998.

[20] M. Held and R. Karp, “The traveling-salesman problem and minimum spanning trees: Part II,” *Mathematical Programming*, vol. 1, pp. 6–25, 1971.

[21] L. A. Wolsey, “Heuristic analysis, linear programming and branch and bound,” in *Combinatorial Optimization II*, ser. Mathematical Programming Studies. Berlin: Springer, 1980, vol. 13, pp. 121–134.

[22] M. X. Goemans and D. J. Bertsimas, “Survivable networks, linear programming relaxations and the parsimonious property,” *Mathematical Programming*, vol. 60, pp. 145–166, 1993.

[23] D. P. Williamson, “Analysis of the held-karp heuristic for the traveling salesman problem,” Master’s thesis, Massachusetts Institute of Technology, Cambridge, Jun. 1990.

[24] M. Bellmore and S. Hong, “Transformation of multisalesman problem to the standard traveling salesman problem,” *Journal of the ACM*, vol. 21, pp. 500–504, July 1974.

[25] M. R. Rao, “Technical Note–A Note on the Multiple Traveling Salesmen Problem,” *Operations Research*, vol. 28, no. 3-Part-I, pp. 628–632, 1980.

[26] S. Rathinam, R. Sengupta, and S. Darbha, “A resource allocation algorithm for multi-vehicle systems with nonholonomic constraints,” *Automation Science and Engineering, IEEE Transactions on*, vol. 4, no. 1, pp. 98 –104, Jan. 2007.

[27] W. A. 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.

[28] S. Rathinam and R. Sengupta, "Lower and upper bounds for a symmetric multiple depot, multiple travelling salesman problem," Institute of Transportation Studies, Research Report UCB-ITS-RR-2006-2, 2006.

[29] K. Savla, E. Frazzoli, and F. Bullo, "On the point-to-point and traveling salesperson problems for Dubins' vehicle," in *Proc. American Control Conference*, Jun. 2005, pp. 786 – 791 vol. 2.

[30] M. T. Cezik and G. Iyengar, "Semidefinite relaxations for the traveling salesman problem," Computation Optimization Research Center, Technical Report TR-2001-031, 2001.

[31] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY: Wiley-Interscience, 1999.

[32] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM*, vol. 7, pp. 326–329, October 1960.

[33] M. X. Goemans and F. Rendl, "Combinatorial optimization," in *Handbook of Semidefinite Programming: Theory, Algorithms and Application*, R. S. Henry Wolkowicz and L. Vandenberghe, Eds. New York, NY: Springer, 2000.

[34] S. Timsjö, "An application of lagrangian relaxation to the traveling salesman problem," Department of Mathematics and Physics, Malardalen University, Vasteras, Sweden, Technical Report IMa- TOM-1999-02, 1999.

[35] J. Lofberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, Sep. 2004, pp. 284 –289.

[36] G. Gutin and A. P. Punnen, Eds., *The Traveling Salesman Problem and Its Variations (Combinatorial Optimization)*. New York, NY: Springer, May 2007.

[37] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209 – 219, 2006.

[38] M. Bläser, "A new approximation algorithm for the asymmetric tsp with triangle inequality," in *Proc. 14th annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '03. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003, pp. 638–645.

[39] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko, "Approximation algorithms for asymmetric tsp by decomposing directed regular multigraphs," *Journal of the ACM*, vol. 52, pp. 602–626, July 2005.

[40] Y. GuoXing, "Transformation of multidepot multisalesmen problem to the standard travelling salesman problem," *European Journal of Operational Research*, vol. 81, no. 3, pp. 557–560, March 1995.

[41] J. L. Ny and E. Feron, "An approximation algorithm for the curvature constrained traveling salesman problem," presented at the Proc. 43rd Annual Allerton Conference on Communications, Control and Computing, Allerton, IL, 2005.

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

[43] R. Ravi and M. Goemans, "The constrained minimum spanning tree problem," in *Algorithm Theory SWAT'96*, ser. Lecture Notes in Computer Science, R. Karlsson and A. Lingas, Eds. Heidelberg: Springer, 1996, vol. 1097, pp. 66–75.

[44] S. Yadlapalli, W. Malik, S. Darbha, and M. Pachter, “A lagrangian-based algorithm for a multiple depot, multiple travelling salesmen problem,” in *American Control Conference, 2007. ACC ’07*, Jul. 2007, pp. 4027 –4032.

[45] S. Yadlapalli, W. A. Malik, S. Darbha, and S. Rathinam, “A lagrangian-based algorithm for a combinatorial motion planning problem,” in *Advances in Cooperative Control and Optimization*, ser. Lecture Notes in Control and Information Sciences, P. Pardalos, R. Murphey, D. Grundel, and M. Hirsch, Eds. Heidelberg : Springer, 2007, vol. 369, pp. 373–387.

[46] M. L. Fisher, “The lagrangian relaxation method for solving integer programming problems,” *Management Science*, vol. 50, pp. 1861–1871, December 2004.

[47] P. Chandler and M. Pachter, “Research issues in autonomous control of tactical UAVs,” in *American Control Conference, 1998. Proc. 1998*, vol. 1, Philadelphia, PA, Jun. 1998, pp. 394 –398 vol.1.

[48] P. Chandler, M. Pachter, D. Swaroop, J. Fowler, J. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, “Complexity in UAV cooperative control,” in *American Control Conference, 2002. Proc. 2002*, vol. 3, Anchorage, AK, 2002, pp. 1831 – 1836 vol.3.

[49] D. K. M. G. Lagoudakis, E. Markakis, P. Keskinocak, A. Kleywegt, and S. Koeni, “Auction-based multi-robot routing,” in *Proc. of Robotics: Science and Systems*, Cambridge, MA, June 2005, pp. 343–350.

[50] J. Tisdale, A. Ryan, M. Zennaro, X. Xiao, D. Caveney, S. Rathinam, J. K. Hedrick, and R. Sengupta, “The Software Architecture of the Berkeley UAV Platform,” in *Control Applications, 2006. CCA ’06. IEEE International Conference on*, Munich, Germany, 2006, pp. 1420–1425.

[51] E. Lawler, *Combinatorial Optimization:networks and matroids*. Mineola, NY: Dover Publications, March 2001.

[52] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, Technical Report 388, 1976.

[53] W. A. 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.

[54] S.Rathinam and R. Sengupta, "3/2-approximation algorithm for two variants of a 2-depot hamiltonian path problem," *Operations Reseach Letters*, vol. 38, no. 1, pp. 63–68, 2010.

[55] M. Grtschel, L. Lovsz, and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, vol. 1, pp. 169–197, 1981.

[56] M. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, vol. 24, pp. 296–317, 1992.

[57] J. L.R. Ford and D. Fulkerson, *Flows and Networks*, ser. Lecture Notes in Computer Science (LNCS). Princeton, NJ: Princeton Univ Press, 1962.

[58] S. Yadlapalli, S. Rathinam, and S. Darbha, "3-approximation algorithm for a two depot, heterogeneous traveling salesman problem," *Optimization Letters*, pp. 1–12, 2010.

[59] ——, "An approximation algorithm for a 2-depot, heterogeneous vehicle routing problem," in *Proc. American Control Conference, 2009. ACC '09.*, Jun. 2009, pp. 1730 –1735.

[60] S. Rathinam, R. Sengupta, and S. Darbha, “A resource allocation algorithm for multi-vehicle systems with nonholonomic constraints,” *Automation Science and Engineering, IEEE Transactions on*, vol. 4, no. 1, pp. 98 –104, Jan. 2007.

VITA

Sai Krishna Yadlapalli was born in Andhra Pradesh, India. He received his Bachelor of Technology in mechanical engineering from Indian Institute of Technology (IIT), Madras in June, 2002. He graduated with a masters degree in mechanical engineering from Texas A & M University in 2005 and earned his Ph.D. in mechanical engineering in December 2010. He may be contacted through Prof. Swaroop Darbha at the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123.