View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Texas A&M Repository

ALGORITHMS FOR ROUTING UNMANNED VEHICLES WITH MOTIONS, RESOURCE, AND COMMUNICATION CONSTRAINTS

A Dissertation

by

KAARTHIK SUNDAR

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Sivakumar Rathinam
Committee Members,	Swaroop Darbha
	Reza Langari
	Sergiy Butenko
Head of Department,	Andreas A. Polycarpou

May 2016

Major Subject: Mechanical Engineering

Copyright 2016 Kaarthik Sundar

ABSTRACT

Multiple small autonomous or unmanned aerial and ground vehicles are being used together with stationary sensing devices for a wide variety of data gathering, monitoring and surveillance applications in military, civilian, and agricultural applications, to name a few. Even though there are several advantages due to the small platforms for these vehicles, they pose a variety of challenges. This dissertation aims to address the following challenges to routing multiple small autonomous aerial or ground vehicles: (i) limited communication capabilities of the stationary sensing devices, (ii) dynamics of the vehicles, (iii) varying sensing capabilities of all the vehicles, and (iv) resource constraints in the form of fuel restrictions on each vehicle. The dissertation formulates four different routing problems for multiple unmanned vehicles, one for each of the aforementioned constraints, as mixed-integer linear programs and develops numerically efficient algorithms based on the branch-and-cut paradigm to compute optimal solutions for practically reasonable size of test instances.

ACKNOWLEDGEMENTS

First and foremost, I owe my deepest gratitude to my advisor Dr. Sivakumar Rathinam, for his constant guidance right from the start of my graduate studies at Texas A&M. The independence he provided me in doing whatever research that I wanted to do has made me the person that I am today. I am fortunate to have been his student and thank him for all the time that he has spent with me trying to mould me into a better researcher, writer, and more importantly, a person.

I am also thankful to Dr. Swaroop Darbha, Dr. Reza Langari and Dr. Sergiy Butenko for serving on my thesis committee. I am also fortunate to have spent time with some of the best teachers in my graduate school. I am particularly grateful to Dr. N. Sivakumar, Dr. Bill Johnson, Dr. Harold Boas, and Dr. Jacob White (Mathematics); Dr. Shankar Bhattacharyya and Dr. P. R. Kumar (Electrical Engineering); Dr. Kiavash Kianfar (Industrial Engineering); Dr. Donald K. Friesen (Computer Science); and Dr. Christopher Menzel (Philosophy).

I am indebted to my friends and colleagues, Shriram, Harsha, Jayavel, Saravanan, Gupta, and Azhagappan, to name a few, without whom this dissertation would not have been possible. In particular, I would like to thank Shriram for spending numerous hours discussing various interesting pathologies and nuances in analysis, linear algebra, dynamics and programming. I also would like to thank Saravanan for the various discussion sessions in implementing decomposition algorithms for large scale combinatorial optimization problems. Special thanks to Jayavel, my room mate, for putting up with me during my tenure as a graduate student and Azhagappan, my neighbor at my lab, for tolerating my chit-chat sessions at his lab, almost every afternoon. Finally, I thank my parents Shri. T. N. Sundar, and Smt. Subha Sundar and my brother Kaushik for supporting me throughout all my studies. Any success of mine would be impossible without their unconditional love and support.

NOMENCLATURE

MDRSP	Multiple Depot Ring-Star Problem
MILP	Mixed-Integer Linear Program
RSP	Ring-Star Problem
MDTSP	Multiple Depot Traveling Salesmen Problem
TSP	Traveling Salesman Problem
SSP	Stable Set Polytope
LP	Linear Program
HMDMTSP	Heterogeneous, Multiple Depot, Multiple Traveling Salesmen Problem
MTSP	Single Depot, Multiple Traveling Salesmen Problem
MDVRP	Multiple Depot, Vehicle Routing Problem
VRP	Vehicle Routing Problem
LKH	Lin-Kernighan-Helsgaun
TSPLIB	Traveling Salesman Problem Library
FCMDVRP	Fuel-Constrained, Multiple Depot, Vehicle Routing Problem
UAV	Unmanned Aerial Vehicle
AFV	Alternate-Fuel Vehicle
MTZ	Miller-Tucker-Zemlin
GMDTSP	Generalized Multiple Depot Traveling Salesmen Problem
GTSP	Generalized Traveling Salesmen Problem
GVRP	Generalized Vehicle Routing Problem
GSEC	Generalized Sub-tour Elimination Constraints
GPEC	Generalized Path Elimination Constraints

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Data gathering and monitoring using autonomous aerial, ground or underwater vehicles has garnered a lot of attention from the scientific community in the last decade (see [86, 24, 14, 15, 74]). These vehicles come with the advantages of being cheap and terrain independent (ground vehicles), can be deployed easily, and can fly at low altitudes (for aerial vehicles), to name a few. Even though these vehicles come with several advantages, they also pose other major challenges to the users in developing cost-efficient plans or routes to perform the assigned mission. Here, cost could imply anything ranging from travel cost to communication cost or sensor battery life. In this thesis, we isolate four different challenges involved in developing cost-efficient plans or routes to accomplish a specific type of data gathering or monitoring mission. As much as possible, we try to make very little assumptions on the type or kind of mission in view of broad applicability. The challenges that we isolate and address separately are as follows:

1. limited communication capabilities of stationary sensing devices when they are used in tandem with autonomous vehicles on a cooperative data gathering mission,

2. routing multiple vehicles with varying sensing capabilities for a monitoring application,

3. resource constraints, in particular, fuel restrictions imposed by the vehicles in a generic data collection or monitoring application, and

4. the constraints imposed by the dynamics of the vehicles in motion planning for a generic data gathering mission with multiple vehicles.

Each of the above challenges has a combinatorial nature and in this thesis, we make that notion clear and formulate each problem as a combinatorial optimization problem. In particular, we formulate each problem as a mixed-integer linear program and develop algorithms to obtain an optimal solution for any given instance of the problem. In the next few sections, we will detail the actual mission, the objectives of the mission, and formally define the problem for each of the aforementioned challenges in the given order.

1.1 Communication capabilities

We consider a cooperative data gather mission using sensing devices together with autonomous vehicles. In particular, we assume that each vehicle is stationed at a distinct location (each location can correspond to a base station) and all the vehicles together have to collect data from a set of sensing devices. A sensing device can communicate its sensed information either to the autonomous vehicles or to its neighbouring sensors. The vehicles have to collect all the sensed information and return to its corresponding location so that the information can be processed further. This approach of using both stationary sensors and autonomous vehicles to collect data is advantageous for several reasons. Firstly, direct communication from the sensed sites to the base station may require a high-power transmitter and may not be suitable for environments with obstructions or non-line-of-sight communications. Simulations/experiments [60, 80] have shown that this type of transmission is also inefficient in terms of energy consumption. Secondly, even if the sensors communicate with the base station through a series of relays (a relay is any device that can receive data from the sensors and transmit it; a sensor can also perform the role of a relay), power consumption may be high as environmental applications require sensing and communicating over thousands of hectares of land. Relays may also have to only depend on battery power for communication as they may be stationed in areas where direct power from the grid is not available. An autonomous vehicle can travel to the monitoring sites and download the sensed data from the sensors, thus reducing the power expended by the sensors in relaying large amounts of data. This process can directly help in increasing the life span of the sensors. Also, by using aerial vehicles to collect data, the sensors are not required to form a connected network and can be spatially distributed depending on the constraints of the application. A natural problem that arises in this context is as follows: "Given the locations of a set of sensors and a set of depots, with one vehicle stationed at each depot, the goal is to (i) find a set of routes, one for each vehicle starting at its depot, visiting a subset of sensors and terminating at its depot, (ii) assign each non-visited sensors to a visited sensor or a depot, and (iii) minimize the sum of the routing costs, *i.e.*, the cost of the routes of the vehicles and the assignments."

We refer to the above problem as the multiple depot ring-star problem (MDRSP). The problem arises naturally in other fields including telecommunications. In a generic telecommunication application, the sensing devices can correspond to terminals or customers in access networks that are connected to switches or multiplexers, and vehicles' routes can correspond to a series of backbone networks that interconnect these multiplexers to its corresponding hub. All the hubs are assumed to be connected via a fixed internal wired network allowing for inter-hub communication. Assuming that this wired network is fixed a priori, the problem of synthesising the backbone network for each hub and the access networks for each multiplexer in an hub reduces to a MDRSP. In chapter 2, this problem is formulated as a mixed-integer linear program on general graphs and algorithms to compute an optimal solution for any instance of the problem is developed. Numerical results are also presented. In the next section, we formally define a problem to address the challenge involved routing multiple heterogeneous vehicles with varying sensing capabilities.

1.2 Routing heterogeneous vehicles

Unmanned aerial vehicles, ground vehicles and underwater vehicles are being used routinely in military applications such as border patrol, reconnaissance, surveillance expeditions. The missions employing these vehicles operate with constraints on time and resource. Often, a heterogeneous fleet of vehicles differing in either structure or function or both is employed for the completion of a mission. This article addresses a commonly encountered routing problem for such missions. We classify the heterogeneity of these vehicles into two categories: *structural* and *functional* heterogeneity. Vehicles are said to be structurally heterogeneous if they differ in design and dynamics. This can lead to differences in fuel consumption, maximum speed at which they can travel, payload capacity, etc. This is a realistic assumption as some structural differences are always present between any pair of vehicles. A collection of vehicles is said to be functionally heterogeneous if not all vehicles may be able to visit a target. Functional heterogeneity results because vehicles may occasionally be equipped with disparate sensors due to the respective payload restrictions. In this case, we partition the set of targets into disjoint subsets: (*i*) targets to be visited by specific vehicles and (*ii*) targets that any of the vehicles can visit. In particular, we define the following problem: "Given a set of targets and a fleet of heterogeneous vehicles located at distinct depots, find a tour for each vehicle that starts and ends at its depot such that each target is visited by at least one vehicle, the vehicle–target constraints are satisfied and the total cost of the tours traveled by all the vehicles in a minimum. " (See Fig. 1.1 for an illustration of a feasible solution to the problem)

We refer to this problem as the heterogeneous multiple depot, multiple vehicle traveling salesmen problem. The related literature, formulation, algorithms and

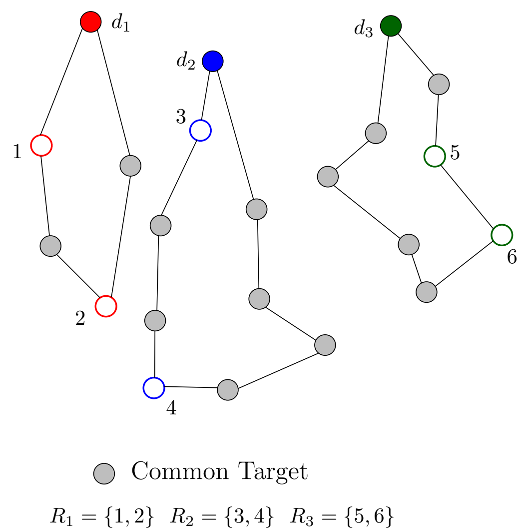


Figure 1.1: Example of a feasible HMDMTSP solution

computational results for the problem is discussed in chapter 3.

1.3 Fuel constraints

Any data gathering mission using multiple vehicles has to account for the vehicles' fuel capacity when planning routes. We define the following problem to address this challenge: "We are given a set of targets, a set of depots and a set of homogeneous vehicles, one for each depot. The depots are also allowed to act as refueling stations. The vehicles are allowed to refuel at any depot, and our objective is to determine a route for each vehicle with a minimum total cost such that each target is visited at least once by some vehicle, and the vehicles never run out fuel as it traverses its route."

We refer to this problem as the fuel-constrained, multiple depot, multiple vehicle routing problem. We note that, for the purpose of addressing the challenge posed by fuel capacity of the vehicles, we assume that the vehicles are homogeneous, unlike the problem defined in Sec. 1.2. The chapter 4 develops four mixed-integer linear programming formulations for the problem and compares them analytically and empirically. We then use the best of the four formulations to obtain an optimal solution to any instance of the problem.

1.4 Vehicle dynamics

Finally, in this section, we formally define a problem to address the challenge of incorporating the dynamics of the vehicles in motion planning for a generic data gathering mission with multiple vehicles. We will assume that the data gathering is performed by a set of homogeneous Reeds-Shepp vehicles [68]. A Reeds-Shepp vehicle is a car that travels with a constant speed, can instantaneously change its direction of motion and has a lower bound on its turn radius. Car-like vehicles are archetypal nonholonomic systems; their motion is restricted to the direction perpendicular to their rear axle and their turn radius is lower bounded due to the mechanical limits on the steering angle. Here, we are given the locations of a set of targets and a set of depots; each depot is equipped with a Reeds-Shepp vehicle and all the vehicles are similar. The objective of the problem is to find a path for each vehicle such that every target is visited by some vehicle, the angle of approach of any vehicle at any target is equal to the angle of departure of the vehicle at that target and the total travel cost for all the vehicles is a minimum. Unlike the problem in Sec. 1.2 where we assumed that the route that any vehicle should take to travel from one target to the other is known a priori, here the route taken by any vehicle to travel from one target to another is a function of the angle of departure and the angle of approach of the vehicle at the corresponding targets. To get around this difficulty, at each target we discretize the angle of approach (departure) *i.e.*, we assume that the angle of approach (departure) of the vehicles in any target is restricted to a discrete set of angles. The basic problem of finding a shortest path for such a vehicle to travel from an oriented initial point (x_i, y_i, α_i) to an oriented final point (x_f, y_f, α_f) was solved by [68] geometrically. We use this result to compute the travel cost for the vehicle to travel from a target i to target j with angle of departure, α_i and angle of arrival, α_j. Now, we restate the discretized version of the problem as follows: "We are given the locations of a set of depots and a set of targets with a Reeds-Sheep vehicle stationed at each depot. We are also given a discrete set of angles for each target. The objective of the problem is to find a set of routes for all the vehicles such that the route for each vehicle starts and ends at its corresponding depot, all the targets are visited by some vehicle, the angle of approach of any vehicle at a target is equal to the angle of departure of the vehicle at that target, and the total cost of travel for all the vehicles is a minimum."

We refer to this problem as the multiple depot one-in-a-set traveling salesmen problem. A generalization of this problem called the generalized multiple depot traveling salesmen problem is presented in chapter 5. We consider a generalization because of its use to wider variety of applications. A mentioned previously, we formulate the problem as a mixed-integer linear program and develop an algorithm to compute an optimal solution to any instance of the problem.

1.5 Organization of the thesis

Each for the four chapters (chapters 2 – 5) is organized as follows: each problem has a concise introduction followed by a detailed literature review. We then introduce notations and formulate the problem. The choice of a particular type of formulation is justified at the appropriate sections. This is followed by either a polyhedral study or as in the case of the fuel-constrained, multiple depot, multiple vehicle routing problem - a theoretical comparison of the various proposed formulations. The details of the branch-and-cut algorithm and extensive computational studies follow. Each chapter is concluded by identifying aspects of the problem that has scope for future work.

2. MULTIPLE DEPOT RING-STAR PROBLEM

In the present chapter, we develop exact algorithms for the MDRSP, a combinatorial optimization problem that arises in optical fibre network design and in applications that collect data using stationary sensing devices and autonomous vehicles. Given the locations of a set of customers and a set of depots, the goal is to (i) find a set of simple cycles such that each cycle (ring) passes through a subset of customers and exactly one depot, (ii) assign each non-visited customer to a visited customer or a depot, and (iii) minimize the sum of the routing costs, *i.e.*, the cost of the cycles and the assignment costs. We present a MILP formulation for the MDRSP and propose valid inequalities to strengthen the linear programming relaxation. Furthermore, we present a polyhedral analysis and derive facet-inducing results for the MDRSP. All these results are then used to develop a branch-and-cut algorithm to obtain optimal solutions to the MDRSP. The performance of the branch-and-cut algorithm is evaluated through extensive computational experiments on several classes of test instances.

2.1 Introduction

The MDRSP is an important combinatorial optimization problem arising in the context of optical fibre network design [3, 40] and in applications pertaining to collecting data using stationary sensing devices and autonomous vehicles [72, 78].

Given the locations of a set of customers (sensors or terminals) and a set of depots (base stations or hubs), (i) find a set of simple cycles such that each cycle (ring) passes through a subset of customers and exactly one depot, (ii) assign each non-visited customer to a visited customer or a depot, and (iii) minimize the sum of the routing costs, *i.e.*, the cost of the cycles and the assignment costs. Fig. 2.1

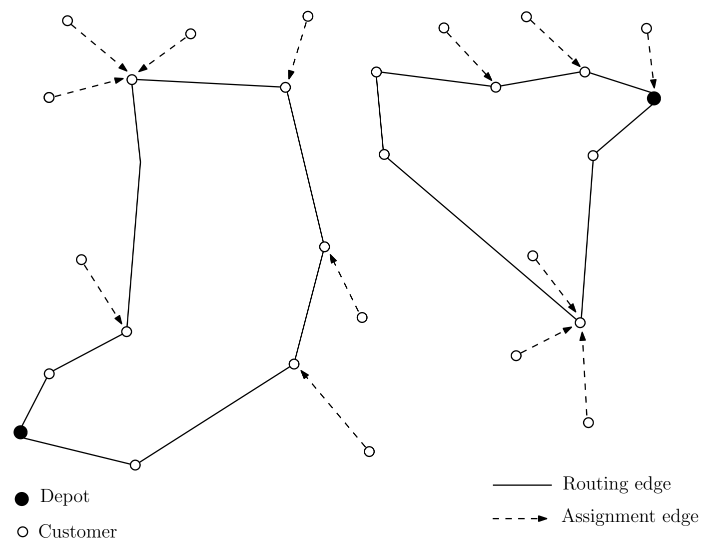


Figure 2.1: Example of a feasible MDRSP solution

shows an example of a feasible solution to the MDRSP. The MDRSP consists of two underlying sub-problems, namely the MDTSP and the assignment problem. The two sub-problems are coupled in the sense that the subset of customers that are present in each cycle is not known a priori. If the assignment costs are very large compared to the routing costs, the MDRSP reduces to the MDTSP [11] and is $\mathcal{NP}$-hard.

This is the first work in the literature that analyzes the facial structure of the MDRSP polytope and derives additional facet-defining inequalities for the polytope of feasible solutions. This chapter develops a MILP formulation using a two-index formulation similar to [40] and also develops non-trivial constraints that eliminate paths between depots for the MDRSP. This work generalizes the results of two related problems namely, the RSP (single depot variant of the MDRSP) in [40] and the MDTSP in [11].

2.2 Related work

The single depot variant of the MDRSP, the RSP, has been well studied in the literature. The RSP was first introduced in the context of communication networks in [57], where the authors develop variable neighborhood tabu-search algorithms to find feasible solutions. In [40, 38], the authors present a polyhedral analysis and branch-and-cut algorithms for computing optimal solutions to the RSP. [41] consider a related problem called the *median-cycle problem* that consists of finding a simple cycle that minimizes the routing cost subject to an upper bound on the total assignment cost. [41] propose integer linear programming models, introduce additional valid inequalities and implement the model in a branch-and-cut framework.

Several authors have also considered graph structures (other than a cycle) such as a path or a tree [42]. [51] address a related single-depot problem called the *Steiner ring-star problem*; it consists of finding a minimum cost ring-star in the presence of Steiner vertices. This problem arises frequently in the context of digital data service network design where the objective is to connect terminals to concentrators using point-to-point links (star topology) and to then interconnect the concentrators through a ring structure. The authors develop a branch-and-cut algorithm to solve the problem to optimality. A tabu search algorithm was also proposed for the *Steiner ring-star problem* in [85].

The capacitated version of the RSP is also well studied in the literature. Heuristics and exact algorithms based on a branch-and-cut approach are available for a *capacitated multiple ring-star problem* [4]. Heuristics and lower bounds were presented for a capacitated variant of the MDRSP in [3]. A branch-and-cut algorithm to solve the capacitated variant of the MDRSP to optimality was presented in [33]. [33] also developed a meta-heuristic to obtain feasible solutions. The computational results in [33] indicate that their meta-heuristic outperforms the heuristic proposed by [3] for most of the instances considered.

Another closely related variant of the MDTSP and hence of MDRSP is the hamiltonian p-median problem [28]. This problem seeks p disjoint cycles which cover all the nodes with minimum cost. One of the main differences between the hamiltonian p-median problem and the MDTSP is that in the hamiltonian p-median problem one seeks exactly p cycles and each cycle need not necessarily contain a depot, which is not the case for the MDTSP or the MDRSP.

2.3 Problem description

Let $D := \{r_1, r_2, \ldots, r_n\}$ denote the set of depots. Let T represent the set of customers. Consider a mixed graph $G = (V, E \cup A)$ where $V = D \cup T$, E is a set of undirected edges joining any two distinct vertices in V, and A is a set of directed arcs that includes self-loops (*i.e.,* $A = \{[i, j] : i, j \in V\}$). Edges in E refer to the undirected routing edges, and the arcs in A refer to the directed assignment edges. For each edge $(i, j) = e \in E$, we associate a non-negative routing cost $c_e = c_{ij}$, and for each arc $[i, j] \in A$, we associate a non-negative assignment cost d_{ij}. Given a subset $E' \subset E$, let $\mathcal{V}(E')$ denote the set of vertices incident to at least one edge in E'. Note that we allow the degenerate case where a cycle can only consist of depot and a customer. A ring-star R is denoted by $(V, E' \cup A')$ where $E' \subset E$ is a simple cycle (ring) containing exactly one depot from D, and $A' \subseteq A$ is a set of assignment edges (star) between a subset of $T \setminus \mathcal{V}(E')$ and the vertices of $\mathcal{V}(E')$. We say that a customer i is assigned to a ring-star R if it is either visited by the simple cycle (*i.e.,* $i \in \mathcal{V}(E')$) or it is connected to a vertex present in a cycle using an assignment edge (*i.e.,* a vertex j exists such that $[i, j] \in A'$). The cost of the ring star R is the sum of the routing cost of edges in E' and the communication cost of the arcs in A'.

The objective of the MDRSP is to design at most n ring-stars so that each customer is assigned to exactly one ring-star and the sum of the cost of all the ring-stars is minimal.

2.4 Mathematical formulation

This section presents a mathematical formulation for the MDRSP inspired by the models for the standard routing problems [11, 40, 81].

We propose a two-index formulation for the MDRSP. We associate to each feasible solution $\mathcal{F}$, a vector $\mathbf{x} \in \mathbb{R}^{|E|}$ (a real vector indexed by the elements of E) such that the value of the component x_e associated with edge e is the number of times e appears in the feasible solution $\mathcal{F}$. Note that for some edges $e \in E$, $x_e \in \{0, 1, 2\}$. If e connects two vertices i and j, then (i, j) and e will be used interchangeably to denote the same edge. Similarly, associated with $\mathcal{F}$, is a vector $\mathbf{y} \in \mathbb{R}^{|A|}$, *i.e.,* a real vector indexed by the elements of A. The value of the component y_{ij} associated with a directed arc $[i, j] \in A$ is equal to 1 if the customer i is assigned to customer j and 0 otherwise. Furthermore, we require that a customer i be present in a cycle if it is assigned to itself, *i.e.,* $y_{ii} = 1$.

For any $S \subset V$, we define $\gamma(S) = \{(i, j) \in E : i, j \in S\}$ and $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$. If $S = \{i\}$, we simply write $\delta(i)$ instead of $\delta(\{i\})$. Finally, for any $\hat{E} \subseteq E$, we define $x(\hat{E}) = \sum_{(i,j)\in\hat{E}} x_{ij}$, and for any disjoint subsets $A, B \subseteq V$, $x(A : B) = \sum_{i\in A, j\in B} x_{ij}$. Using the above notations, the MDRSP is formulated as a mixed integer linear program as follows:

$$\text{Minimize} \quad \sum_{e\in E} c_e x_e + \sum_{[i,j]\in A} d_{ij} y_{ij} \tag{2.1}$$

subject to

$$x(\delta(i)) = 2y_{ii} \quad \forall i \in T, \tag{2.2}$$

$$\sum_{j\in V} y_{ij} = 1 \quad \forall i \in T, \tag{2.3}$$

$$x(\delta(S)) \geq 2\sum_{j\in S} y_{ij} \quad \forall S \subseteq T, i \in S, \tag{2.4}$$

$$x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 2(y_{jj} + y_{kk})$$

$$\forall j,k \in T; j \neq k; D' \subset D, \tag{2.5}$$

$$x(D' : \{j\}) + 2x(\gamma(\bar{S})) + x(\{k\} : D \setminus D') \leq \sum_{v\in\bar{S}} 2\, y_{vv} - \sum_{b\in S} y_{ab}$$

$$\forall a \in S; j,k \in T; S \subseteq T \setminus \{j,k\}, S \neq \emptyset; \bar{S} = S \cup \{j,k\}; D' \subset D, \tag{2.6}$$

$$y_{ii} = 1 \quad \forall i \in D, \tag{2.7}$$

$$y_{ij} = 0 \quad \forall i \in D;\ j \in T, \tag{2.8}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E;\ i,j \in T, \tag{2.9}$$

$$x_{ij} \in \{0,1,2\} \quad \forall (i,j) \in E;\ i \in D;\ j \in T, \tag{2.10}$$

$$y_{ij} \in \{0,1\} \quad \forall [i,j] \in A. \tag{2.11}$$

In the above formulation, the constraints in (2.2) ensure the number of undirected (routing) edges incident on any vertex $i \in T$ is equal to 2 if and only if target i is assigned to itself ($y_{ii} = 1$). The constraints in (2.3) enforce the condition that a vertex $i \in T$ is either in a cycle ($y_{ii} = 1$) or assigned to a vertex j in a cycle (*i.e.*, $y_{ij} = 1$ for some $j \in V$, $j \neq i$). The constraints in (2.4) are the connectivity or sub-tour elimination constraints. They ensure a feasible solution has no sub-tours of any subset of customers in T. The constraints in (2.5) and (2.6) are the path elimination constraints. They do not allow for any cycle in a feasible solution to consist of more than one depot. The validity of these constraints is discussed in subsection 2.4.1. Constraints in Eq. (2.7) and (2.8) are the assignment constraints for the depots. Finally, the constraints (2.9)-(2.11) are the integrality restrictions on the $\mathbf{x}$ and $\mathbf{y}$ vectors.

2.4.1 Path elimination constraints

To the best of our knowledge, the first version of any kind of path elimination constraints was developed for the location routing problem [44]. These constraints were facet-inducing for the version of location routing problem considered in [44]. [44] first develop a path elimination constraint from first principles for paths of length 3 (length refers to number of edges in the path) such that it is a facet and extend that approach to develop tight path elimination constraints for paths of length at least 4. Ever since, this approach has been used successfully for developing tight path elimination constraints for a variety of problems [11, 10]. The second approach that is taken in the literature for developing path elimination constraints is to consider a single constraint to eliminate all paths. This is achieved by a single constraint as follows: for any path $P = \{(d_1, t_2), (t_2, t_3), \ldots, (t_{p-1}, d_2)\}$ that starts at depot d_1 and terminates at depot d_2, $x(P) \leq |P|-1$ eliminates P [22, 28]). This type of constraints will remove paths of any length starting and terminating at distinct depots. Usually this inequality is not used as is, and it is lifted to higher dimensions to make the constraint tighter. For the MDRSP, we chose the former approach because it was more suited for proving the inequality is facet-inducing.

Any path that originates from a depot and visits exactly two customers before terminating at another depot is removed by the constraint in (2.5). The validity of the constraint (2.5) can be easily verified [44]. Any other path $d_1, t_1, \cdots, t_p, d_2$, where $d_1, d_2 \in D$, $t_1, \cdots, t_p \in T$ and $p \geq 3$, violates inequality (2.6) with $D' = \{d_1\}$, $S = \{t_2, \cdots, t_{p-1}\}$, $j = t_1$, $k = t_p$ and $a = t_r$ where $2 \leq r \leq p-1$. We now state and prove a result concerning inequality (2.4) that will aid in the verifying the validity of the constraint in Eq. (2.6).

Lemma 2.1. *The connectivity constraints in Eq.* (2.4) *is equivalent to* $x(\gamma(S)) \leq \sum_{v \in S} y_{vv} - \sum_{j \in S} y_{ij}$ *for all* $i \in S$, $S \subseteq T$.

Proof. Consider a set S with $\emptyset \neq S \subseteq T$. Then, for any feasible solution to the MDRSP, we have the following equality,

$$\begin{aligned} \sum_{v \in S} x(\delta(v)) &= 2x(\gamma(S)) + x(\delta(S)) \\ \sum_{v \in S} 2y_{vv} &= 2x(\gamma(S)) + x(\delta(S)) \qquad (2.12) \\ \sum_{v \in S} 2y_{vv} &\geq 2x(\gamma(S)) + 2\sum_{j \in S} y_{ij} \quad \forall i \in S \text{ (from Eq.(2.4))} \\ x(\gamma(S)) &\leq \sum_{v \in S} y_{vv} - \sum_{j \in S} y_{ij} \quad \forall i \in S \text{ (from Eq.(2.2))} \qquad (2.13) \end{aligned}$$

Hence proved. □

The above lemma states that inequalities (2.4) and (2.13) are equivalent and any feasible solution to the MDRSP satisfies both these constraints. We use this equivalence to prove the validity of (2.6) for the MDRSP in the following proposition.

Proposition 2.1. *Any feasible solution to the MDRSP is not eliminated by the path elimination constraint in* (2.6).

Proof. Using the lemma 2.1, we first reduce the constraint in (2.6) to

$$x(D' : \{j\}) + 2\,x(\{j\} : S) + 2\,x(\{k\} : S) + x(\{k\} : D \setminus D') + 2x_{jk} \leq 2\,(y_{jj} + y_{kk}) + \sum_{b \in S} y_{ab} + \left(x(\delta(S)) - 2 \sum_{b \in S} y_{ab} \right). \quad (2.14)$$

Any feasible solution to the MDRSP will satisfy the sub-tour elimination constraints in Eq. (2.4). Hence, any feasible solution to the MDRSP will either satisfy $x(\delta(S)) = 2 \sum_{b \in S} y_{ab}$ or $x(\delta(S)) > 2 \sum_{b \in S} y_{ab}$.

Case: $x(\delta(S)) = 2 \sum_{b \in S} y_{ab}$

Consider any feasible solution $\mathcal{F}$ that satisfies $x(\delta(S)) = 2 \sum_{b \in S} y_{ab}$. Then, either $\sum_{b \in S} y_{ab} = 1$ or $\sum_{b \in S} y_{ab} = 0$.

If $\sum_{\mathbf{b} \in \mathbf{S}} \mathbf{y_{ab}} = \mathbf{0}$ in the feasible solution, the inequality in (2.14) reduces to

$$x(D' : \{j\}) + x(\{k\} : D \setminus D') + 2x_{jk} \leq 2\,(y_{jj} + y_{kk})$$

which is trivially satisfied by the solution.

If $\sum_{\mathbf{b} \in \mathbf{S}} \mathbf{y_{ab}} = \mathbf{1}$ in the solution, the inequality in (2.14) reduces to

$$x(D' : \{j\}) + 2x(\{j\} : S) + 2x(\{k\} : S) + x(\{k\} : D \setminus D') + 2x_{jk} \leq 2\,(y_{jj} + y_{kk}) + 1. \quad (2.15)$$

The proof that the feasible solution satisfies the above equation is as follows:

1. Let $y_{jj} = 0$. In this subcase, the degree constraints indicate that $x(D' : \{j\}) = x(\{j\} : S) = x_{jk} = 0$. Hence, the constraint (2.15) reduces to $2x(\{k\} : S) + x(\{k\} : D \setminus D') \leq 2y_{kk} + 1$, which is satisfied by the feasible solution (since $x(\delta(S)) = 2$). A similar argument holds for the subcase when $y_{kk} = 0$.

2. For $y_{jj} = y_{kk} = 1$, the right-hand-side (RHS) of constraint (2.15) takes the value 5. It is not difficult to observe that for any feasible solution with $x(\delta(S)) = 2$, the maximum value that the left-hand-side (LHS) of the constraint (2.15) can take is also 5.

Case: $x(\delta(S)) > 2\sum_{b \in S} y_{ab}$

Consider any feasible solution $\mathcal{F}$ that satisfies $x(\delta(S)) > 2\sum_{b \in S} y_{ab}$.

1. Consider the subcase where $y_{jj} = 0$. Then, the constraint reduces to $2x(\{k\} : S) + x(\{k\} : D \setminus D') \leq 2y_{kk} + \sum_{b \in S} y_{ab} + (x(\delta(S)) - 2\sum_{b \in S} y_{ab})$. This constraint is trivially satisfied by $\mathcal{F}$ when $y_{kk} = 0$. When $y_{kk} = 1$, observe that the minimum value of the RHS and the maximum value of the LHS of the constraint are both 4 (since $\mathcal{F}$ has $x(\delta(S)) > 2\sum_{b \in S} y_{ab}$, the minimum value of $x(\delta(S)) - 2\sum_{b \in S} y_{ab}$ is 2). Hence, $\mathcal{F}$ satisfies Eq. (2.14) when $y_{jj} = 0$. A similar argument holds for the subcase when $y_{kk} = 0$.

2. Consider the subcase where $y_{jj} = y_{kk} = 1$. First, we observe that the minimum value taken by the RHS of the constraint is 6. Hence, we need only to look at the instances when the LHS of the constraint takes a value greater than 6. This occurs when $x(\{j\} : S) = x(\{k\} : S) = 2$ and the LHS of the constraint would take a value 8. In such a case, $x(\delta(S)) \geq 6$, for else $\mathcal{F}$ would not be feasible. Then, the RHS of the constraint would take a minimum value of 9.

Hence, any feasible solution to the the MDRSP is not eliminated by the path elimination constraint in (2.6). □

We note that our formulation allows for a feasible solution with paths connecting two depots and visiting exactly one customer. In the literature, such paths are referred to as 2-paths. As the formulation allows for two copies of an edge between a depot and a target, 2-paths will be eliminated since there always exists an optimal solution which does not contain any 2-path. In the following subsection, we shall strengthen the linear programming relaxation of the model (2.2)-(2.11) by the introduction of additional valid inequalities.

2.4.2 Additional valid inequalities

In this section, we develop four classes of valid inequalities for the MDRSP. Consider the constraints in Eq. (2.4). For any $S = \{i, j\}$ where $i, j \in T$ and $i \neq j$, the equation reduces to $x(\delta(i)) + x(\delta(j)) - 2x_{ij} \geq 2y_{ii} + 2y_{ij}$. Further simplification using the degree constraints yields

$$x_{ij} \leq y_{jj} - y_{ij}. \tag{2.16}$$

Another set of useful constraints similar to (2.16) is given by

$$x_{ij} \leq 2y_{jj} \quad \text{for all } i \in D, j \in T. \tag{2.17}$$

Inequalities valid for a TSP polytope are also valid for the MDRSP. We particularly examine the *2-matching inequalities* available for the TSP polytope [29]. Specifically, we consider the following inequality:

$$x(\gamma(H)) + x(\mathcal{T}) \leq \sum_{i \in H} y_{ii} + \frac{|\mathcal{T}| - 1}{2} \tag{2.18}$$

for all $H \subseteq T$ and $\mathcal{T} \subset \delta(H)$. Here H is called the handle, and $\mathcal{T}$ the teeth. H and $\mathcal{T}$ satisfy the following conditions:

- the edges in the teeth are not incident to any depots in the set D,
- no two edges in the teeth are incident on the same customer,
- $|\mathcal{T}| \geq 3$ and odd.

The 2-matching inequality is also valid for the RSP [40] and hence, they are also valid for the MDRSP. The constraints in Eq. (2.18) are also equivalent to the *blossom's inequality* for the 2-matching problem and a special case of the *comb inequality* for the symmetric TSP [2]. Eq. (2.18) is a comb inequality where the cardinality of every tooth is two and both the handle and the teeth contain only vertices from set T. The next set of valid inequalities is derived using the valid inequalities for the SSP. In any feasible solution to the MDRSP, for any triplet of vertices $i, j, k \in T$, the assignments y_{ij} and y_{ik} are incompatible when $j \neq k$. The stable set problem associated with these incompatible assignments is a relaxation of the MDRSP polytope. A similar observation was made for the RSP in [40]. This property leads to the following odd-hole inequalities for the MDRSP:

$$y_{ij} + y_{jk} + y_{ki} \leq 1 \quad \text{for all } i, j, k \in T \text{ and } i \neq j \neq k. \tag{2.19}$$

$$x(\delta(S)) \geq 2(y_{ij} + y_{jk} + y_{ki}) \quad \text{for all } i, j, k \in T,\ i \neq j \neq k \text{ and } S \subseteq T \text{ such that } i, j, k \in S. \tag{2.20}$$

Eq. (2.20) is the valid inequality obtained from the two previously mentioned relaxations of the MDRSP, *i.e.,* the SSP and TSP relaxations.

We will next develop a few valid inequalities that are specific to the MDRSP. In particular, we will examine a special type of 2-matching inequality with multiple depots. We will call these inequalities *depot-2-matching* inequalities. Consider the following inequality:

$$x(\gamma(H)) + x(\mathcal{T}) \leq \sum_{i \in H} y_{ii} + |\mathcal{T}| \tag{2.21}$$

for all $H \subseteq T$ and $\mathcal{T} \subset \delta(H)$; H is the handle, and $\mathcal{T}$ is the teeth. H and $\mathcal{T}$ satisfy the following conditions:

- every edge in the teeth must be incident on a depot,
- no two edges in the teeth are incident on the same depot,
- number of edges is $\mathcal{T}$ is greater than equal to one, and
- there exists at least one customer and one depot outside the handle and teeth.

Proposition 2.2. *The depot-2-matching inequality in Eq.* (2.21) *is valid for any feasible solution to the MDRSP.*

Proof. For any $H \subseteq T$ and $\mathcal{T} \subset \delta(H)$ satisfying the conditions stated previously, we have the following equality:

$$2x(\gamma(H)) + x(\delta(H)) = \sum_{v \in H} x(\delta(v))$$
$$\Rightarrow \; 2x(\gamma(H)) + x(\mathcal{T}) + x(\delta(H) \setminus \mathcal{T}) = 2\sum_{v \in H} y_{vv} \quad \text{(from Eq. (2.2))}.$$

We also have $x(\mathcal{T}) \leq 2|\mathcal{T}|$ for the set $\mathcal{T}$ (since the edges in the teeth are incident on the depots). Adding this inequality to the above equality, we obtain,

$$\begin{aligned} 2x(\gamma(H)) + 2x(\mathcal{T}) + x(\delta(H) \setminus \mathcal{T}) &\leq 2\sum_{v \in H} y_{vv} + 2|\mathcal{T}| \\ \Rightarrow\ 2x(\gamma(H)) + 2x(\mathcal{T}) &\leq 2\sum_{v \in H} y_{vv} + 2|\mathcal{T}|. \end{aligned}$$

The last inequality follows because $x(\delta(H) \setminus \mathcal{T}) \geq 0$. Further simplification yields

$$x(\gamma(H)) + x(\mathcal{T}) \leq \sum_{v \in H} y_{vv} + |\mathcal{T}|.$$

Hence the *2-depot-matching* inequality is valid for the MDRSP. □

Observe that the depot-2-matching inequality also allows for the number of edges in the teeth to be even and that a 2-depot-matching inequality with more that two edges in the teeth can also eliminate depot-depot paths.

In the following section, we develop some polyhedral results and facet-inducing properties for the valid inequalities discussed thus far.

2.5 Polyhedral analysis

We will show the polyhedral results for the MDRSP while leveraging on the results already known for a MDTSP. MDTSP is a special case of the MDRSP when each customer must be visited by one of the vehicles. Let P denote the polytope that represents the convex hull of feasible solutions to the MDRSP (*i.e.,* satisfies (2.2)-(2.11)) and Q denote the corresponding MDTSP polytope [11].

If u denotes the number of customers, we observe that there are u equalities in (2.2), u equalities in (2.3), n equalities in (2.7) and nu equalities in (2.8). Therefore, the system (2.2), (2.3), (2.7) and (2.8) has $2u + n + nu$ equalities. We also note that this system of equality constraints are linearly independent.

The number of x_e variables in the formulation is $\binom{u}{2} + nu$ ($\binom{u}{2}$ is the number of edges between customers and nu is the number of edges between depots and customers). Similarly, the number of y_{ij} variables in the formulation is $u^2 + n + 2nu$ (u^2 is the number of customer to customer arcs, n is the number of arcs that assigns a depot to itself and $2nu$ is the number of arcs that assigns a depot to a customer and vice versa). Let m denote the total number of variables used in the problem formulation *i.e.*, $m = \binom{u}{2} + u^2 + n + 3nu$.

Let $\chi_{(\mathbf{x},\mathbf{y})} \in \mathbb{R}^m$ denote the incidence vector of a solution $(\mathbf{x}, \mathbf{y})$ to the MDRSP in the graph G. Now we have,

$$\begin{aligned} P &:= \text{conv}\{\chi_{(\mathbf{x},\mathbf{y})} : (\mathbf{x}, \mathbf{y}) \text{ is a feasible MDRSP solution}\} && (2.22) \\ Q &:= \{(\mathbf{x}, \mathbf{y}) \in P : y_{ii} = 1 \text{ for all } i \in T\} && (2.23) \end{aligned}$$

The dimension of the polytope Q was shown to be $\binom{u}{2} + u(n-1)$ [11]. Let $F \subseteq T$ denote a subset of customers. To relate P and Q, we define an intermediate polytope $P(F)$ as follows:

$$P(F) := \{(\mathbf{x}, \mathbf{y}) \in P : y_{ii} = 1 \text{ for all } i \notin F\}. \tag{2.24}$$

We observe that, $P(\emptyset) = Q$ and $P(T) = P$. Also, for any $(\alpha, \beta) \in \mathbb{R}^m$ and $\gamma \in \mathbb{R}$, define the hyperplane

$$\mathcal{H}(\alpha, \beta, \gamma) := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^m : \alpha\mathbf{x} + \beta\mathbf{y} = \gamma\} \tag{2.25}$$

Lemma 2.2. *Let $v_1, \cdots, v_u$ be an ordering of the customers in the set T and $F_k = \{v_1, \cdots, v_k\}$ for all $k \in \{1, \cdots, u\}$. If for each $k = 1, \ldots, u$ and each $v_l \in V \setminus \{v_k\}$,* *there exists a feasible solution to the MDRSP, such that*

1. $y_{v_jv_j} = 1$ *for all* $j > k$ *i.e., every customer in the set* $T \setminus F_k$ *is in some cycle,*

2. $y_{v_jv_j} + \sum_{r \in D} y_{v_jr} = 1$ *for all* $j < k$ *i.e., every vertex in the set* F_k *must be either in a cycle or assigned to a depot,*

3. $y_{v_kv_l=1}$ *i.e., the vertex* v_k *must be assigned to the vertex* v_l*, and*

4. $\alpha\mathbf{x} + \beta\mathbf{y} = \gamma$,

then, $\dim(P \cap \mathcal{H}(\alpha, \beta, \gamma)) \geq \dim(Q \cap \mathcal{H}(\alpha, \beta, \gamma)) + u(u + n - 1)$.

Proof. We prove by induction on $|F_k|$ that $\dim(P(F_k) \cap \mathcal{H}(\alpha, \beta, \gamma)) \geq \dim(Q \cap \mathcal{H}(\alpha, \beta, \gamma)) + |F_k|(u + n - 1)$. This in turn proves the lemma because when $F_k = T$, we have $P(F_k) = P$ and $|F_k| = u$. Let $N_k := \dim(Q \cap \mathcal{H}(\alpha, \beta, \gamma)) + |F_k|(u + n - 1)$. The base case for induction holds since $k = 0$ implies $F_k = \emptyset$ and $P(F_k) = Q$. Now, suppose $k > 0$. Then by induction hypothesis, we have $\dim(P(F_{k-1}) \cap \mathcal{H}(\alpha, \beta, \gamma)) \geq N_{k-1}$. Hence, there are at least $N_{k-1} + 1$ affine independent points in the polytope $P(F_{k-1}) \cap \mathcal{H}(\alpha, \beta, \gamma)$. All these affine independent points satisfy $y_{v_kv_k} = 1$ (since $v_k \notin F_{k-1}$). From the definition of $P(F)$ in Eq. (2.24), we have $P(F_k) \cap \mathcal{H}(\alpha, \beta, \gamma) \supset P(F_{k-1}) \cap \mathcal{H}(\alpha, \beta, \gamma)$. Therefore, these $N_{k-1} + 1$ affine independent points satisfying $y_{v_kv_k} = 1$ lie in $P(F_k) \cap \mathcal{H}(\alpha, \beta, \gamma)$. The assumptions of the lemma provide for additional $(u + n - 1)$ affine independent points in $P(F_k) \cap \mathcal{H}(\alpha, \beta, \gamma)$ that satisfy $y_{v_kv_k} = 0$. Therefore, $\dim(P(F_k) \cap \mathcal{H}(\alpha, \beta, \gamma)) \geq N_{k-1} + (u + n - 1) = \dim(Q \cap \mathcal{H}(\alpha, \beta, \gamma)) + |F_k|(u + n - 1)$. Hence proved. $\square$

The Lemma 2.2's hypothesis provides a family of feasible solutions to the MDRSP that are guaranteed to be linearly independent. The dimension of the MDRSP polytope P is computed in the following corollary of Lemma 2.2.

Corollary 2.1. $\dim(P) = \binom{u}{2} + u^2 + 2u(n-1)$.

Proof. The number of variables used in formulation of MDRSP is $\binom{u}{2} + u^2 + n + 3nu$ and all the solutions of the MDRSP satisfy the $2u + n + nu$ linearly independent equality constraints in the system (2.2, 2.3, 2.7, 2.8). Hence, $\dim(P) \leq \binom{u}{2} + u^2 + n + 3nu - (2u + n + nu) = \binom{u}{2} + u^2 + 2u(n-1)$. Also, we have,

$$
\begin{aligned}
\dim(P) &= \dim(P \cap \mathcal{H}(0,0,0)) \\
&\geq \dim(Q \cap \mathcal{H}(0,0,0)) + u(u+n-1) \qquad \text{(using Lemma 2.2)} \\
&= \dim(Q) + u^2 + u(n-1) \\
&= \binom{u}{2} + u(n-1) + u^2 + u(n-1) \\
&= \binom{u}{2} + u^2 + 2u(n-1)
\end{aligned}
$$

Hence, $\dim(P) = \binom{u}{2} + u^2 + 2u(n-1)$. □

An important consequence of Lemma 2.2 is that any valid inequality $\alpha\mathbf{x} + \beta\mathbf{y} \leq \gamma$ that is facet-inducing to the MDTSP polytope Q and satisfying the conditions (1)–(4) of the lemma is valid and facet-inducing to the MDRSP polytope P. This observation will be used in all of the subsequent results concerning the polyhedral analysis of P.

Proposition 2.3. *If* $|T| \geq 4$, *the inequality* $x_e \geq 0$ *is facet-inducing for* P *for every* $e \in E$.

Proof. For any ordering of the customers in T, it is trivial to construct feasible solutions satisfying the conditions *1–4* of Lemma 2.2 ($x_e = 0$ is the hyperplane here) for a fixed $e = (i,j) \in E$. To construct such feasible solutions satisfying the assumptions of the Lemma, we require the condition $|T| \geq 4$ (in Fig. 2.2, when $|T| < 4$ in Prop. (2.3), a feasible solution to the MDRSP with customers 2,3 in the

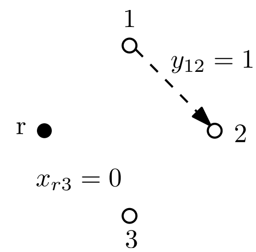


Figure 2.2: Counter-example for the case when $|T| < 4$ in Prop. 2.3

cycle associated with depot r such that $x_{r3} = 0$ and $y_{12} = 1$ cannot be constructed). The proposition follows by noting that $x_e = 0$ is a facet to the MDTSP polytope Q if $|T| \geq 4$ (see [11]). $\square$

Remark. We also note that the inequality $x_{ij} \leq 1$ for all $(i, j) \in E$ and $i, j \in T$ is not facet-inducing for P since it is dominated by the constraint in Eq. (2.16). Similarly, the inequality $x_{ij} \leq 2$ for all $(i, j) \in E$, $i \in D$ and $j \in T$ is not facet-inducing for the polytope P as it is dominated by the corresponding constraint in Eq. (2.17).

Proposition 2.4. *The sub-tour elimination constraint given by Eq.* (2.4)*, i.e.,* $x(\delta(S)) \geq 2\sum_{j \in S} y_{ij}$ *is facet-inducing for the MDRSP polytope for each* $S \subseteq T$*,* $i \in S$*,* $|S| \geq 2$*.*

Proof. Consider any ordering of the customers in set T such that the customer $i \in T$ is in the last position of the ordering. We will prove the proposition by constructing feasible solutions satisfying assumptions of Lemma 2.2 ($x(\delta(S)) = 2\sum_{j \in S} y_{ij}$ is the hyperplane here) for the considered ordering.

Choose an arbitrary customer $k \in T \setminus \{i\}$. Given k, we construct $(u + n - 1)$ feasible solutions satisfying the assumptions of the Lemma 2.2 as follows: construct a cycle spanning all the customers in $T \setminus \{k\}$ and some depot r with exactly 2 edges in $\delta(S)$ and customer k assigned to any vertex in the set $V \setminus \{k\}$ (illustrated in

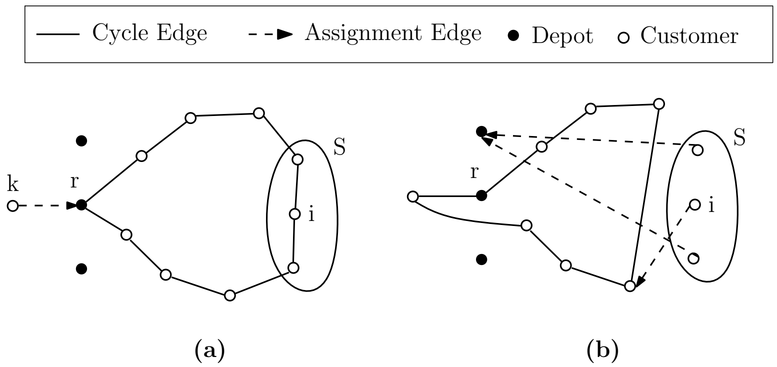


Figure 2.3: Feasible solutions described in Prop. 2.4

Fig. 2.3-(a)). The cardinality of the set $V \setminus \{k\}$ is $(u + n - 1)$ and hence we obtain $(u + n - 1)$ feasible solutions satisfying the assumptions of the Lemma.

We now detail the procedure for constructing another $(u+n-1)$ feasible solutions for the last customer $i \in T$. Construct a cycle spanning depot r and all the customers in $S \setminus \{i\}$ with exactly two edges in $\delta(S)$ while assigning i to any vertex in $S \setminus \{i\}$. This provides for $|S|-1$ feasible solutions that satisfy the assumptions of the Lemma 2.2. Another set of $|V \setminus S|$ feasible solutions is obtained as follows: construct a cycle spanning the depot r and the vertex set $T \setminus S$, and assign the customers in $S \setminus \{i\}$ to one of the depots and the customer i to any vertex in the set $V \setminus S$ (illustrated in Fig. 2.3-(b)). This final set of feasible solutions ensure $x(\delta(S)) = 0$ and $2\sum_{j \in S} y_{ij} = 0$. The proposition then follows because $x(\delta(S)) \geq 2\sum_{j \in S} y_{ij}$ reduces to a facet-inducing inequality $x(\delta(S)) \geq 2$ for the polytope Q of the MDTSP (see [11]). □

Remark. The Prop. 2.4 does not hold for $|S| = 1$, since the degree constraint in Eq. (2.2) dominates the corresponding constraint with $|S| = 1$. Similarly, when $i \notin S$,

[40] showed that Prop. 2.4 is not valid for the RSP because of the inequality

$$x(\delta(S \cup \{i\})) = x(\delta(S)) + x(\delta(i)) - 2\sum_{j\in S} x_{ij} \geq 2\sum_{j\in S\cup i} y_{ij} = 2(y_{ii} + \sum_{j\in S} y_{ij}).$$

The above inequality implies $x(\delta(S)) \geq 2\sum_{j\in S}(y_{ij} + x_{ij})$ which dominates the corresponding constraint in Eq. (2.4) when $i \notin S$. The same argument holds for the MDRSP.

Proposition 2.5. *The constraint given by Eq.* (2.5), $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 2(y_{jj} + y_{kk})$, *is facet-inducing for the MDRSP polytope* P *for every* $j, k \in T$, $j \neq k$, $|T| \geq 2$, $D' \subset D$, *and* $D' \neq \emptyset$.

Proof. We shall again use Lemma 2.2 to prove the proposition. Given $j, k \in T$ and $D' \subset D$, consider any ordering of the vertices in T where j and k appear in the last two positions. We also assume $r_1 \in D'$ and $r_2 \in D \setminus D'$. We claim there is a feasible solution for every vertex $i \in T$ and for each vertex $v \in V \setminus \{i\}$ that satisfy the assumptions *1–3* of Lemma 2.2 and the equation $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') = 2(y_{jj} + y_{kk})$. This claim combined with the known result that $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 4$ is facet-inducing for the MDTSP polytope Q (see [11]) proves the proposition. We shall now prove our claim.

For any arbitrary customer $i \in T \setminus \{j, k\}$, consider the following solutions to the MDRSP: a cycle spanning the depot r_1 and all the customers in $T \setminus \{i\}$ such that the customer j is adjacent to the depot r_1 and customer k with the customer i assigned to any vertex in the set $V \setminus \{i\}$. Each of these solutions is feasible to the MDRSP and satisfy the equation $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') = 2(y_{jj} + y_{kk}) = 4$ (since $x(D' : \{j\}) = 1$ and $x_{jk} = 1$). For the customer j, consider feasible solutions where j is assigned to a vertex in $V \setminus \{j\}$, the vertex k is the lone vertex spanned the cycle associated with depot r_2 and all the customers in $T \setminus \{j,k\}$ are spanned by cycle associated with depot r_1. These solutions satisfy the equation $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') = 2(y_{jj} + y_{kk}) = 2$ (since $x(\{k\} : D \setminus D') = 2$). A similar construction can also be done for the vertex k. Therefore the claim, and as a result, the proposition is true. $\square$

Proposition 2.6. *The constraint given by Eq.* (2.6), $x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\}) + x(\{k\} : D \setminus D') \leq \sum_{v \in S \cup \{j,k\}} 2\, y_{vv} - \sum_{b \in S} y_{ab}$, *is facet-inducing for the MDRSP polytope P for every $j,k \in T$, $j \neq k$, $S \subseteq T \setminus \{j,k\}$, $D' \subset D$, $D' \neq \emptyset$, and $a \in S$.*

Proof. Consider any ordering of the customers in T such that the j,k, and a appear (in that order) in the last three positions in the ordering. We assume $r_1 \in D'$ and $r_2 \in D \setminus D'$. We claim there exists a feasible MDRSP solution for every vertex $i \in T$ and for each vertex $v \in V \setminus \{i\}$ that satisfy the assumptions *1–3* of Lemma 2.2 and the equation $x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\}) + x(\{k\} : D \setminus D') = \sum_{v \in S \cup \{j,k\}} 2\, y_{vv} - \sum_{b \in S} y_{ab}$. This claim together with the known result that $x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\}) + x(\{k\} : D \setminus D') \leq 2|S| + 3$ is facet-inducing for the MDTSP polytope Q (see [11]) proves the proposition. We shall now prove our claim.

Choose an arbitrary customer $i \in T \setminus \{j,k,a\}$. Given i, we now construct $(u+n-1)$ feasible MDRSP solution satisfying the assumptions of the Lemma 2.2 as follows: construct a cycle spanning the customers $j, t \in S \setminus \{i\}, k, t \in T \setminus (S \cup \{j,k,i\})$ in that order and depot r_1, with the customer i assigned to any vertex in $V \setminus \{i\}$ (illustrated in Fig. 2.4-(a)). For all the above $(u+n-1)$ solutions, the LHS and the RHS of the constraint (2.6) takes the value $2|S \setminus \{i\}| + 3$ *i.e.*, the feasible solutions satisfy the constraint (2.6) at equality.

Now, we construct $2(u+n-1)$ feasible solutions for the customer j and k respectively. We will construct the solutions for j and the same procedure can be

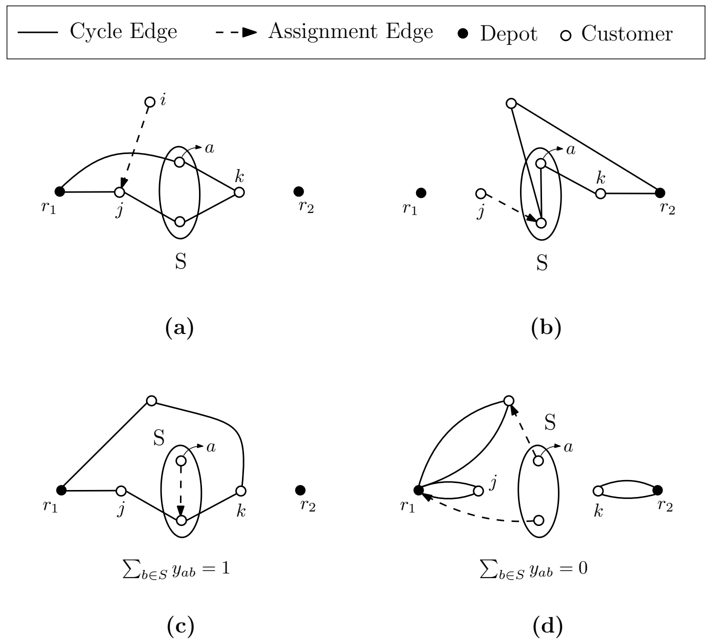


Figure 2.4: Feasible solutions described in Prop. 2.6

followed to construct solutions for the customer k. For the customer j, construct a cycle spanning the customers $k, t \in S$, and $t \in T \setminus (S \cup \{j, k\}$ in that order and the depot r_2, with j assigned to any vertex in $V \setminus \{j\}$ (illustrated in Fig. 2.4-(b)). A similar procedure for constructing feasible MDRSP solutions for the customer k yields another $(u + n - 1)$ solutions.

We finally detail the procedure to construct the $(u + n - 1)$ feasible MDRSP solutions for the last customer in the ordering, a. Construct a cycle spanning $r_1, j, t \in S \setminus \{a\}, k$ and $t \in T \setminus (S \cup \{i, j\})$ in that order with the customer a assigned to one of the customers in $S \setminus \{a\}$. This provides for $|S| - 1$ feasible solutions that satisfy the assumptions of the Lemma 2.2 (see Fig. 2.4-(c)). The remaining set of $|V \setminus S|$ feasible solutions is obtained as follows: construct two cycles one with the vertices j and r_1 and the other with k and r_2 (*i.e.*, $x_{jr_1} = x_{kr_2} = 2$.), assign all the customers in $T \setminus \{j, k, a\}$ to r_1 and the customer a to any vertex in $V \setminus S$. This set of feasible solutions have $x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\}) + x(\{k\} : D \setminus D') = 4$ and $\sum_{v \in S \cup \{j,k\}} 2\, y_{vv} - \sum_{b \in S} y_{ab} = 4$ (see Fig. 2.4-(d)). Now, the proposition follows because Eq. (2.6) reduces to a facet-defining inequality $x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\}) + x(\{k\} : D \setminus D') \leq 2|S| + 3$ for the polytope Q. □

Proposition 2.7. *The 2-matching inequality in Eq.* (2.18) *for all* $H \subseteq T$ *and* $\mathcal{T} \subset \delta(H)$*, satisfying the conditions:*

1. *the edges in the teeth are not incident to any depots in the set* D*,*

2. *no two edges in the teeth are incident on the same customer,*

3. $|\mathcal{T}| \geq 3$ *and odd.*

is facet-inducing for the MDRSP polytope P *when* $|T| \geq 6$*.*

Proof. The proof proceeds by constructing feasible solutions that satisfy conditions *1–3* of the Lemma 2.2 and the hyperplane $x(\gamma(H)) + x(\mathcal{T}) = \sum_{i \in H} y_{ii} + \frac{|\mathcal{T}|-1}{2}$. For a fixed H and $\mathcal{T}$ satisfying the conditions stated in the proposition, and for each $k \in T$, it is straightforward to construct a cycle spanning some depot $r \in D$ and all the customers in $T \setminus \{k\}$ such that $x(\gamma(H)) + x(\mathcal{T}) = \sum_{i \in H} y_{ii} + \frac{|\mathcal{T}|-1}{2}$ (refer to Fig. 2.5). Each of these cycles can be converted to a feasible solution by the addition of an assignment from customer k to a vertex in the set $V \setminus \{k\}$. The figures show portions of the cycle when $k \in T$ is in the handle and teeth respectively. In Fig. 2.5–(a), the vertex k is in the handle H and in Fig. 2.5–(b), k is in a tooth. We also note that the valid inequality $x(\gamma(H)) + x(\mathcal{T}) \leq \sum_{i \in H} y_{ii} + \frac{|\mathcal{T}|-1}{2}$ reduces to $x(\gamma(H)) + x(\mathcal{T}) \leq |H| + \frac{|\mathcal{T}|-1}{2}$ for a MDTSP. The proposition follows since the hyperplane defined by $x(\gamma(H)) + x(\mathcal{T}) \leq |H| + \frac{|\mathcal{T}|-1}{2}$ is a facet for the MDTSP polytope Q when $|T| \geq 6$ (see [11]). □

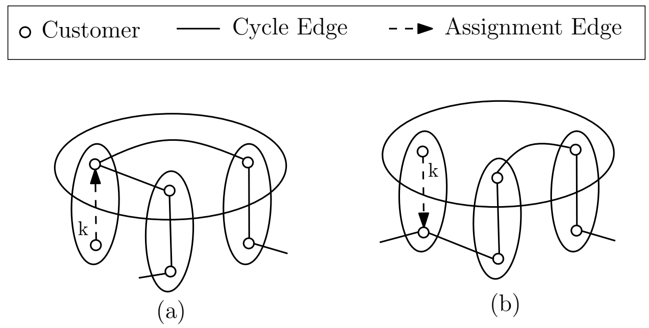


Figure 2.5: Feasible solutions described in Prop. 2.7

2.6 Separation algorithms

In this section, we discuss the algorithms that are used to find violated families of constraints described in Sec. 2.4. We denote by $G^* = (V^*, E^*)$ the *support graph* associated with a given fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$ *i.e.,* $V^* := \{i \in V : y^*_{ii} > 0\}$ and $E^* := \{(i,j) \in E : x^*_{ij} > 0\}$. We also define $A^* := \{[i,j] \in A : y^*_{ij} > 0\}$.

2.6.1 Separation of sub-tour elimination constraints

As shown previously, the inequalities in Eq. (2.4) reduce to Eq. (2.16) when $|S| = 2$. The violation of the inequality in Eq. (2.16) can be verified by examining the inequality for every pair of customers in the set T. Next, we examine the connected components in G^*. Each connected component C such that $D \cap C = \emptyset$ generates a violated sub-tour elimination constraint for $S = C$ and for each $i \in S$. If a connected component C has $D \cap C \neq \emptyset$, the following procedure is used to find the largest violated sub-tour elimination constraint in $x(\delta(S)) \geq 2\sum_{j\in S} y_{ij}$. For any $S \subseteq T$, given any $i \in S$, we can rewrite the constraint in Eq. (2.4) as

$$x(\delta(S)) + 2\sum_{j\notin S} y_{ij} \geq 2 \qquad \forall S \subseteq T, i \in S. \tag{2.26}$$

Given a connected component C such that $D \cap C \neq \emptyset$, $i \in C \cap T$, and a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the most violated constraint (2.26) can be obtained by computing a minimum s-t cut on a capacitated undirected graph $\bar{G} = (\bar{V}, \bar{E})$, with $\bar{V} = (V^* \cap T) \cup \{s\}$. The vertex s denotes the source vertex and is formed by contracting all the depots into a single vertex. The vertex t denotes the sink vertex and $t = i$. The edge set $\bar{E} = E^* \cup \{(s, j) : j \in V^* \cap T\}$. Every edge (s, j) where $j \in (V^* \cap T) \setminus \{i\}$ is assigned a capacity $\sum_{d \in D} x^*_{dj}$. The edge (i, j) where $j \in \bar{V} \setminus \{i\}$ is assigned a capacity equal to $x^*_{ij} + 2y^*_{ij}$ and any remaining edge e is assigned a capacity x^*_e. We now compute the minimum s-t cut $(S, \bar{V} \setminus S)$ with $t \in \bar{V} \setminus S$. The vertex set $S' = \bar{V} \setminus S$ defines the most violated inequality if the capacity of the cut is strictly less than two. A similar separation procedure is also used to separate the sub-tour elimination constraints in [40, 4].

2.6.2 Separation of path elimination constraints

We first discuss the procedure used to separate violated constraints in Eq. (2.5). Consider every pair of targets $j, k \in T \cap V^*$. We rewrite the constraint in (2.5) as $x(D' : \{j\}) + x(\{k\} : D \setminus D') \leq 2(y_{kk} + y_{jj}) - 3x_{jk}$. Given j, k and fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the RHS of the above inequality is a constant and is equal to $2(y^*_{kk} + y^*_{jj}) - 3x^*_{jk}$. We observe that the LHS of the inequality, $x^*(D' : \{j\}) + x^*(\{k\} : D \setminus D')$, is maximum when $D' = \{d \in D : x^*_{jd} \geq x^*_{kd}\}$. Furthermore, when $D' = \emptyset$, no path constraint in Eq. (2.5) is violated for the given pair of vertices j and k. With $D' = \{d \in D : x^*_{jd} \geq x^*_{kd}\}$, if $x^*(D' : \{j\}) + x^*(\{k\} : D \setminus D')$ is strictly greater than $2(y^*_{kk} + y^*_{jj}) - 3x^*_{jk}$, the path constraint in Eq. (2.5) is violated for the pair of vertices j, k and the subset of depots D'.

We now discuss the separation procedure for the the constraint in Eq. (2.6). We note that this path constraint is determined by a pair of vertices $j, k \in T$, a subset of vertices $S \subseteq T \setminus \{j,k\}$, a vertex $a \in S$ and a subset of depots $D' \subset D$. In what follows we develop a procedure that is applied to every pair of clients $\{j,k\}$. It is obvious that (2.6) will never be violated if j and k belong to different connected components of the support graph G^*; hence, we only consider pairs of those $\{j,k\}$ belonging to the same connected component in G^*. We denote $\bar{S} = S \cup \{j,k\}$. Using this notation, we reformulate the constraint in Eq. (2.6) to Eq. (2.27), whose violation can be deduced using a minimum s-t cut algorithm. The reduction is shown below:

$$x(D' : \{j\}) + x(\{k\} : D \setminus D') + 2x(\gamma(\bar{S})) \leq \sum_{v \in \bar{S}} 2\, y_{vv} - \sum_{b \in S} y_{ab},$$

$$x(D' : \{j\}) + x(\{k\} : D \setminus D') \leq x(\delta(\bar{S})) - \sum_{b \in S} y_{ab},$$

$$x(D' : \{j\}) + x(\{k\} : D \setminus D') + 1 \leq x(\delta(\bar{S})) + \sum_{b \notin \bar{S}} y_{ab} + y_{aj} + y_{ak},$$

$$x(D' : \{j\}) + x(\{k\} : D \setminus D') + 1 - y_{aj} - y_{ak} \leq x(\delta(\bar{S})) + \sum_{b \notin \bar{S}} y_{ab}. \qquad (2.27)$$

The second inequality follows by applying Eq. (2.12) in Lemma 2.1 to the set $\bar{S}$. Eq. (2.27) is an equivalent representation of the path constraint in Eq. (2.6). Now, given a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the pair $\{j,k\}$ in the same connected component C, and an $a \in (C \setminus \{j,k\}) \cap T$, the LHS of (2.27) attains a maximum value for $D' = \{d \in D : x^*_{jd} \geq x^*_{kd}\}$ (when $D' = \emptyset$, the corresponding path constraint (2.6) is not violated). Let $\mathcal{L} = x^*(D' : \{j\}) + x^*(\{k\} : D \setminus D') + 1 - y^*_{aj} - y^*_{ak}$. Now, the most violated constraint (2.6) can be found by computing a minimum s-t cut on a capacitated undirected graph $\bar{G} = (\bar{V}, \bar{E})$ with $\bar{V} = V^* \cup \{s,t\}$. The vertex s denotes the source. The vertex t denotes the sink and is formed by contracting all the depots to a single vertex. We add edges with very large capacity from the source vertex s to vertices j, k and a. Every edge (i,a) where $i \in V^* \setminus \{a\}$ is assigned a capacity

$x^*_{ai} + y^*_{ai}$ and any remaining edge e is assigned a capacity x^*_e. The minimum s-t cut (S', T') on $\bar{G}$ would have $j, k, a, s \in S'$ and $t, r \in T'$ for every $r \in D$. The pair j, k, the vertex set $S = S' \setminus \{s\}$ and the vertex $a \in S$ defines the most violated inequality if the capacity of the cut is strictly less than $\mathcal{L}$.

2.6.3 Separation of 2-matching and depot-2-matching constraints

We discuss exact and heuristic separation procedures for the 2-matching constraints. Using a construction similar to the one proposed by [66] for the b-matching problem, the separation problem for 2-matching inequalities can be transformed into a minimum capacity odd cut problem; hence this separation problem is exactly solvable in polynomial time. This procedure is computationally intensive, and so we use the following simple heuristic proposed by [23]. We consider an undirected graph $\bar{G} = (\bar{V}, \bar{E})$ with $\bar{V} = V^* \cap T$ and $\bar{E} = \{e : 0 < x^*_e < 1\}$. Then, we consider each connected component H of $\bar{G}$ as a handle of a possibly violated 2-matching inequality whose two-node teeth correspond to the edges $e \in \delta(H)$ with $x^*_e = 1$. We reject the inequality if the number of teeth is even. The procedure can be implemented in $O(|\bar{V}| + |\bar{E}|)$ time. A similar procedure is used for separating the depot-2-matching constraints. In this case, we consider two-node teeth corresponding to edges incident on the depots *i.e.,* $e \in \delta(H)$ with $x^*_e = 1$ and $e = (t, d)$, where $t \in T \cap H$ and $d \in D$. This procedure also eliminates paths between the depots.

2.6.4 Separation of odd-hole and clique inequalities

For the constraints in Eq. (2.19) and Eq. (2.20), we use the separation procedures discussed in [40]. The inequalities in Eq. (2.19) can be separated by a complete enumeration of $i, j, k \in T$ such that $y^*_{ij} > 0$, $y^*_{jk} > 0$ and $y^*_{ki} > 0$. Similarly, for each $i, j, k \in T$ such that $y^*_{ij} > 0$, $y^*_{jk} > 0$ and $y^*_{ki} > 0$, a min-cut separating D from $\{i, j, k\}$ in G^* would detect the most violated constraint in Eq. (2.20), if any.

2.7 Branch-and-cut algorithm

In this section, we describe important implementation details of the branch-and-cut algorithm for the MDRSP. The algorithm is implemented within a CPLEX 12.6.1 framework using the CPLEX callback functions [34]. The callback functions in CPLEX enable the user to completely customize the branch-and-cut algorithm embedded into CPLEX including, the choice of node to explore in the enumeration tree, the choice of branching variable, the separation and the addition of user-defined cutting planes and the application of heuristic methods.

The lower bound at the root node of the enumeration tree is computed by solving the LP relaxation of the formulation in Sec. 2.4 that is further strengthened using the cutting planes described in Sec. 2.4.2. The initial linear program consisted of all constraints in (2.1)-(2.11) and (2.17), except (2.4), (2.5) and (2.6). Several numerical experiments indicated that the inequalities in Eq. (2.19) and Eq. (2.20) were not computationally helpful for the branch-and-cut procedure, and so they were not used in the final implementation of the algorithm. For a given LP solution, we identify violated inequalities using the separation procedures in the following order: (i) sub-tour elimination constraints in Eq. (2.16), (ii) sub-tour elimination constraints in Eq. (2.4) (iii) path elimination constraints in Eq. (2.5) and Eq. (2.6) (iv) 2-matching and depot-2-matching constraints in Eq. (2.18) and (2.21), respectively. Furthermore, we disabled the separation of all the cuts embedded into the CPLEX framework because enabling these cuts increased the average computation time for the instances. Once the new cuts generated using these separation procedures were added to the linear program, the tighter linear program was resolved. This procedure was iterated until either of the following conditions was satisfied: (i) no violated constraints could be generated by the separation procedures, (ii) the current lower bound of the enumeration tree was greater or equal to the current upper bound. If no constraints are generated in the separation phase, we create sub-problems by branching on a fractional variable. First, we select a fractional y_{ii} variable, based on the *strong branching* rule [1]. If all these variables are integer, then we select a fractional x_e variable using the same rule. As for the node-selection rule, we used the best-first policy for all our computations, *i.e.,* select the sub-problem with the lowest objective value.

2.7.1 Heuristics

We discuss a greedy algorithm called *LP-heuristic,* that aides in speeding up the convergence of the branch-and-cut algorithm. The *LP-heuristic* constructs a feasible solution from a given fractional LP solution. It is used only at the root node of the enumeration tree, once in every three iterations. *LP-heuristic* is based on a transformation method [63]. Given $\mathbf{y}^*$, the vector of fractional LP assignment values, the heuristic greedily assigns every customer in the set T to some vertex in the set V. We call this procedure the greedy assignment procedure; a pseudo-code of the algorithm is shown in Fig 2.6. Once we have the assignment, we can compute the set of vertices that are spanned by some cycle (the set of vertices that are assigned to itself). We then solve the MDTSP on these vertices and D. A heuristic based on the transformation method [63] and LKH heuristic [31] is used to solve the MDTSP.

2.8 Computational results

In this section, we discuss the computational results of the branch-and-cut algorithm. The algorithm was implemented in C++ (GCC version 4.6.3), using the elements of Standard Template Library (STL) and CPLEX 12.4 framework. As mentioned in Sec. 2.7, the internal CPLEX cut generation was disabled and, CPLEX was used only to manage the enumeration tree. All the simulations were performed on

Procedure - *Greedy Assignment*
Input: $\mathbf{y}^*$;
Output: assignments σ, set P of vertices that are spanned by some cycle;
comment: initialization
for each $i \in T$ **do** $\sigma(i) := -1$;
$\bar{T} := T$; **comment:** customers to be assigned
$P := V$; **comment:** vertices that are spanned by some cycle
comment: customer assignment
while $\bar{T} \neq \emptyset$ **do**
  Select a customer $i \in \bar{T}$ randomly; $\bar{T} = \bar{T} \setminus \{i\}$;
  $\sigma(i) = \text{argmax}\{y^*_{ik} : k \in P\}$;
  if $\sigma(i) \neq i$ **then** $P = P \setminus \{i\}$;
endwhile

Figure 2.6: The greedy assignment procedure

a Dell Precision T5500 workstation (Intel Xeon E5630 processor @2.53 GHz, 12 GB RAM). The computation times reported were expressed in seconds and we imposed a time limit of 7200 seconds for each run of the algorithm. The performance of the algorithm was tested on different classes of test instances, all generated using the traveling salesman problem library [69].

Instance generation: We generated two classes of test instances (I and II) having the same underlying graph, but with a different assignment cost structure (similar to [4, 40]). For each of the two classes and for each value of $|T| \in \{29, 51, 76, 101\}$, we generated 12 MDRSP instances using four TSPLIB instances [69] namely, *bays29, eil51, eil76* and *eil101.* We performed a computational study on these instances with $|D| \in \{3, 4, 5\}$. The depot locations were randomly generated. The routing costs and assignment costs were generated as follows:

Class I: The routing and assignment cost for a pair of vertices i, j is equal to the Euclidean distance l_{ij} between the two vertices.

Class II: For each pair of vertices i, j, the routing cost $c_{ij} = \alpha l_{ij}$ and the assignment cost $d_{ij} = (10 - \alpha)l_{ij}$ where $\alpha \in \{3, 5, 7, 9\}$. We refer to α as the scale factor.

Tables 2.1–2.3 summarize the computational behavior of the branch-and-cut algorithm on the two classes of instances. The column headings are defined as follows:

Name: instance name (for Classes I and II);

$|D|$**:** number of depots (for Classes I and II);

α**:** scale factor (for Class II);

%-LB: percentage LB/opt, where LB is the objective value of the LP relaxation computed at the root node of the enumeration tree (for Classes I and II);

%-LB0: percentage LB/opt, where LB is the objective value of the LP relaxation computed at the root node of the enumeration tree without adding the additional valid inequalities for the MDRSP (for Class II);

Pair: number of constraints (2.16) generated (for Classes I and II);

SEC: number of constraints (2.4) with $|S| > 2$ generated (for Classes I and II);

2mat: number of constraints (2.18) generated (for Classes I and II);

PEC: number of constraints (2.5) and (2.6) generated (for Classes I and II);

Nodes: total number of nodes examined in the enumeration tree (for Classes I and II);

Time: total computation time in seconds (for Classes I and II).

%Ring: total percentage of customers in present in the ring for the optimal MDRSP solution (for Class II)

Name	$\lvert D \rvert$	%-LB	Pair	SEC	2mat	PEC	Nodes	Time
bays29	3	94.81	133	2939	17	618	119	4.52
bays29	4	99.30	46	676	8	1107	21	5.46
bays29	5	100.00	42	374	1	282	0	0.69
eil51	3	100.00	76	739	5	24	0	1.59
eil51	4	100.00	74	1182	6	83	0	6.76
eil51	5	100.00	78	1251	2	614	0	10.18
eil76	3	99.83	129	2615	23	1519	44	105.19
eil76	4	99.74	130	2483	10	2835	34	39.04
eil76	5	99.54	148	3738	70	7182	353	260.42
eil101	3	99.93	176	5441	8	1328	5	261.57
eil101	4	99.92	178	4551	9	1954	4	252.69
eil101	5	99.96	174	4118	8	3135	3	277.35
Averages		99.42	121.88	2508.92	13.92	1723.42	48.85	102.12

Table 2.1: Computational results for Class I instances

Table 2.2: Computational results for Class II instances (*bays29* and *eil51*)

Name	$\|D\|$	α	%-LB	%-LB0	Pair	SEC	2mat	PEC	Nodes	Time	%-Ring
bays29	3	3	98.08	97.45	4	276	22	447	35	4.29	100.00
bays29	4	3	98.08	97.45	2	191	21	473	45	4.48	100.00
bays29	5	3	98.31	97.91	8	293	25	572	128	4.3	100.00
bays29	3	5	98.99	98.99	126	2939	17	618	119	4.59	65.52
bays29	4	5	99.30	98.99	46	676	8	1116	21	5.52	65.52
bays29	5	5	100.00	99.93	42	374	1	301	0	0.72	65.52
bays29	3	7	99.80	99.75	258	309	0	469	0	3.97	41.38
bays29	4	7	99.77	99.72	221	323	0	291	0	2.16	41.38
bays29	5	7	97.32	97.27	194	512	2	2156	352	38.22	27.59
bays29	3	9	100.00	99.84	257	51	0	0	0	0.09	0.00
bays29	4	9	100.00	99.88	219	6	0	0	0	0.06	0.00
bays29	5	9	100.00	99.90	177	6	0	0	0	0.06	0.00
eil51	3	3	98.95	98.79	13	2280	187	7452	577	266.88	100.00
eil51	4	3	98.73	98.70	18	1754	161	12715	881	451.58	100.00
eil51	5	3	98.69	98.33	8	1442	119	9160	1274	387.43	100.00
eil51	3	5	100.00	99.75	76	739	5	24	0	1.59	66.67
eil51	4	5	100.00	99.76	74	1182	6	83	0	6.75	64.71
eil51	5	5	99.97	99.97	78	1251	2	614	1	10.19	54.90
eil51	3	7	99.90	99.87	424	796	0	2028	4	123.11	29.41
eil51	4	7	97.91	97.91	394	2762	5	19480	8527	2567.99	25.49
eil51	5	7	97.85	97.48	387	3692	19	19089	7841	2680.55	23.53
eil51	3	9	100.00	99.97	402	20	0	0	0	0.24	0.00
eil51	4	9	100.00	100.00	355	20	0	0	0	0.25	0.00
eil51	5	9	100.00	100.00	353	20	0	0	0	0.25	0.00

Table 2.3: Computational results for Class II instances (*eil76* and *eil101*)

Name	$\|D\|$	α	%-LB	%-LB0	Pair	SEC	2mat	PEC	Nodes	Time	%-Ring
eil76	3	3	99.79	99.52	12	680	50	1075	41	38.13	100.00
eil76	4	3	99.75	99.26	21	929	43	2322	74	44.1	100.00
eil76	5	3	99.68	99.26	10	851	19	1284	35	22.19	100.00
eil76	3	5	99.83	99.72	129	2615	23	1519	44	105.3	73.68
eil76	4	5	99.74	99.73	131	2407	11	2835	35	39.19	73.68
eil76	5	5	99.56	99.56	139	2899	45	6075	317	195.58	71.05
eil76	3	7	99.38	99.52	1410	1564	0	3096	6	236.43	38.16
eil76	4	7	99.22	99.19	1391	1592	0	1999	13	280.31	32.89
eil76	5	7	98.75	98.64	977	7627	84	13098	1637	1928.61	30.26
eil76	3	9	99.83	99.98	1417	1105	0	2335	3	223.09	5.26
eil76	4	9	100.00	100.00	1388	1016	0	1787	0	139.85	5.26
eil76	5	9	100.00	99.49	943	450	0	407	0	3.88	0.00
eil101	3	3	99.46	99.24	12	680	50	1075	41	211.37	100.00
eil101	4	3	99.42	99.17	21	929	43	2322	74	131.63	100.00
eil101	5	3	99.62	99.37	10	851	19	1284	35	103.39	100.00
eil101	3	5	99.93	99.83	129	2615	23	1519	44	258.51	72.28
eil101	4	5	99.92	99.74	131	2407	11	2835	35	253.16	74.26
eil101	5	5	99.96	99.88	139	2899	45	6075	317	273.2	71.29
eil101	3	7	99.78	99.78	1410	1564	0	3096	6	724.76	35.64
eil101	4	7	99.90	99.88	1391	1592	0	1999	13	562.4	34.65
eil101	5	7	99.55	99.55	977	7627	84	13098	1637	2938.66	34.65
eil101	3	9	100.00	99.57	1417	1105	0	2335	3	449.83	6.93
eil101	4	9	100.00	99.50	1388	1016	0	1787	0	266.98	6.93
eil101	5	9	100.00	99.29	943	450	0	407	0	199.8	6.93
Averages:		3	99.05	98.71	11.58	929.67	63.25	3348.42	270.00	139.15	100.00
		5	99.77	99.66	103.33	1916.92	16.42	1967.83	77.75	96.19	68.26
		7	99.26	99.04	786.17	2496.67	16.17	6658.25	1669.67	1007.26	32.92
		9	99.99	99.78	771.58	438.75	0.00	754.83	0.50	107.03	2.61

The results tabulated in Tables 2.1–2.3 indicate that the proposed branch-and-cut algorithm can solve instances involving up to 101 customers with modest computation times. All the instances were solved by the branch-and-cut algorithm within an hour. For a scale factor value of 3, we observe that the MDTSP solution is the optimal solution to the MDRSP. As the scale factor value is increased, this is clearly not the case because the percentage of customers present in the cycles decreases considerably. Furthermore, we observe that the Class II instances are more difficult, on an average, especially for a scale factor equal to 7. For the scale factor value of 7, the average percentage of customers present in the cycle in the optimal solution is 68%. These are the instances that take the maximum average computation time of 1007 seconds. Hence, the difficult instances tend to be those with relatively few assignment edges in the optimal solution. This is in contrast to the RSP [40], where the difficult instances tend to be those where the optimal cycle consists of about 20% of the customers. This major variation in the trade-off between the cycle costs and the assignment costs is due to the presence of the path elimination constraints in the MDRSP and the inherent challenges involved in solving multiple depot variants. The %-LB column in both the tables indicate that the lower bound obtained at the root node of the enumeration tree is very tight, typically within 0.5% of the optimum. The %-LB0 column in the Tables 2.2 and 2.3 is the ratio of the lower bound obtained at the root node of the enumeration tree to the optimal solution; here the lower bound is obtained by not using any of the additional valid inequalities developed for the MDRSP. This average %-LB0 is observed to be within 1.2% of the optimal solution for all the instances in Class II. Hence, we conclude that proposed mixed-integer linear programming formulation for the MDRSP is by itself very tight. But a numerically observed advantage of the *depot-2-matching* inequalities was that for the instances where the number of violated depot-2-matching inequalities were large, the number of path-elimination constraints added to the enumeration tree was reduced leading to an overall reduction in the computation time. This is because these inequalities can themselves eliminate depot to depot paths. Overall, we were able to solve all the 60 test instances within an hour, with the largest instance involving 101 customers and 5 depots.

2.9 Conclusion

In this chapter, we have presented an exact algorithm for the MDRSP, a problem that arises in designing an optical fiber network in telecommunications and allocating resources in monitoring applications. A mixed integer linear programming formulation including several classes of valid inequalities was proposed and a complete polyhedral analysis with facet-inducing results was investigated together with a branch-and-cut algorithm. The algorithm was tested on a wide class of benchmark instances from a standard library. The largest solved instance involved 101 vertices. Future work can be directed towards development of branch-and-cut approaches accompanied with a polyhedral study to solve capacitated versions of the problem.

3. HETEROGENEOUS, MULTIPLE DEPOT, MULTIPLE TRAVELING SALESMAN PROBLEM*

In this chapter, we formally define the HMDMTSP and present an exact algorithm based on the branch-and-cut paradigm to compute an optimal solution to the problem. Unmanned aerial vehicles are being used in several monitoring applications to collect data from a set of targets. These vehicles are heterogeneous in the sense that they can differ either in their motion constraints or sensing capabilities. Furthermore, not all vehicles may be able to visit a given target because vehicles may occasionally be equipped with disparate sensors due to the respective payload restrictions. This chapter addresses a problem where a group of heterogeneous vehicles located at distinct depots visit a set of targets. The targets are partitioned into disjoint subsets: targets to be visited by specific vehicles and targets that any of the vehicles can visit. The objective is to find an optimal tour for each vehicle starting at its respective depot such that each target is visited at least once by some vehicle, the vehicle–target constraints are satisfied and the sum of the costs of the tours for all the vehicles is minimized. We formulate the problem as a MILP and develop a branch-and-cut algorithm to compute an optimal solution to the problem. Computational results show that optimal solutions for problems involving 100 targets and 5 vehicles can be obtained within 300 seconds on average, further corroborating the effectiveness of the proposed approach. This chapter is published as a conference article in [76].

---

*Reprinted with permission from "An exact algorithm for a heterogeneous multiple depot, multiple traveling salesman problem" by Kaarthik Sundar and Sivakumar Rathinam. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 366371. Copyright [2015] by IEEE.

3.1 Introduction

The HMDMTSP is a generalization of the MDTSP which is known to be $\mathcal{NP}$-Hard [11]. We formulate the HMDMTSP as a MILP and develop a branch-and-cut algorithm to compute optimal solutions for the same. The reminder of the chapter is organized as follows. In Sec. 3.2, we discuss the relevant literature. In Sec. 3.3, we formulate the HMDMTSP as a MILP and present additional valid inequalities to strengthen the linear programming relaxation. A branch-and-cut algorithm based on the formulation for the HMDMTSP is described in Sec. 3.4, and Sec. 3.5 presents computational results on several classes of test instances.

3.2 Related work

The single vehicle variant of the HMDMTSP is the TSP. Over the past two decades, several methods including exact algorithms, heuristics, and approximation algorithms have been developed to address the TSP [50]. The HMDMTSP reduces to the MDTSP when all the vehicles are homogeneous. [11] present an exact algorithm to solve the MDMTSP. Another variant of the MDMTSP that has received considerable attention in the literature is the MTSP. In the MTSP, there are m homogeneous vehicles that have to visit a set of customers from a single depot, and every vehicle must at least visit one target. For a homogeneous MTSP and its variations, [37] present some integer linear programming formulations. [8] reviews the applications, exact and heuristic solution procedures and transformations of MTSP to the TSP. A branch-and-bound-based method for large-scale MTSP may be found in [25].

The HMDMTSP can also be considered as a special case of MDVRP. The MDVRP consists of finding a set of routes based on a set of given depots to serve the demand of a set of customers with multiple homogeneous vehicles of limited capacity. [49] study variants of this problem with asymmetric costs and propose branch-and-bound algorithms to optimally solve the problem. More recently, [6] have developed an exact solution framework to solve different vehicle routing problems that can be applied to the MDVRP as well. [79] introduced and developed a column generation heuristic for the VRP using an heterogeneous fleet. [79] assumed the fleet of vehicles to be structurally heterogeneous. Since then, a wide range of heuristics, exact algorithms and approximation algorithms have been developed for routing problems with structurally heterogeneous fleet of vehicles. To our knowledge, there is no exact algorithm available in the literature to solve any variant of heterogeneous VRPs. [5] give an overview of approaches to solve heterogeneous VRPs. In particular, they classify the variants described in the literature, review the lower bounds and the heuristics and compare the performance of the different algorithms on benchmark instances. Routing problems with functionally heterogeneous vehicles have also been addressed in the vehicle routing literature. They are often referred to as site-dependent vehicle routing problems. The site-dependent vehicle routing problem generalizes the classical VRP in order to represent the compatibility relationship between customer sites and vehicle types. In this problem, we have a functionally heterogeneous fleet of vehicles with vehicle–target constraints. A variety of heuristics based on local search methods, tabu search etc. are available in the literature for solving the site dependent VRP and some of its variants [58, 12].

[17] present an approximation algorithm for the 2-depot heterogeneous hamiltonian path problem. This is the first paper that considers both functional and structural heterogeneous vehicles. Apart from [17], we are not aware of any literature that addresses multiple depot routing problem with a functional and structural heterogeneous fleet of vehicles and develops exact algorithms for the same. The main focus of this chapter is the development of an exact algorithm based on branch-and-cut method [61, 40] for the HMDMTSP. We also present a computational study for

the algorithm in order to evaluate its performance.

3.3 Mathematical formulation

Let T denote the set of targets. We have a heterogeneous fleet of n vehicles initially stationed at a distinct depot. Let $D = \{d_1, d_2, \ldots, d_n\}$ represent the set of depots. Consider an undirected graph $G = (V, E)$, where $V = T \cup D$ and E is a set of edges joining any two vertices[1] in V. We assume G does not have any self-loops. Let the cost of traversing an edge $(i, j) = e \in E$ for a vehicle $v \in \{1, \ldots, n\}$ be c_e^v. We will assume that for each vehicle v, the costs satisfy triangle inequality, *i.e.*, for every $i, j, k \in V$, $e_1 := (i, j), e_2 := (j, k)$ and $e_3 := (i, k)$, $c_{e_1}^v + c_{e_2}^v \geq c_{e_3}^v$. Furthermore, we also assume that there are vehicle–target constraints where each vehicle v is required to visit a subset of targets $R_v \subseteq T$ with $\cap_i R_i = \emptyset$. We refer to these targets as functional heterogeneous targets. Note that the sets $R_1, \ldots, R_n$ are specified a priori and only a common target present in $T \setminus (\cup_i R_i)$ can be visited by any vehicle.

We now present a mathematical formulation for the HMDMTSP, inspired by the models for the standard routing problems [81, 50]. For each vehicle $v \in \{1, \ldots, n\}$, we associate with each edge e a variable x_e^v, whose value is the number of times e appears in a the feasible solution. Note that for some edges $e \in E$, $x_e^v \in \{0, 1, 2\}$ *i.e.*, we permit the degenerate case where a tour for vehicle v can consist of just its depot and a target. If e connects two vertices i and j, then (i, j) and e will be used interchangeably to denote the same edge. We also remark that for a vehicle v, we do not have any decision variables x_e^v for edges connecting depot $d_{v'}$ such that $v \neq v'$. Similarly, for each vehicle $v \in \{1, \ldots, n\}$, we associate with each target $i \in T$ a binary variable y_i^v, which takes a value 1 when the target i is visited by vehicle k and 0 otherwise.

[1]We remark that an edge between any pair of depots is not present in the edge set E.

For any $S \subset V$, we define $\delta(S) = \{(i,j) \in E : i \in S,\, j \notin S\}$ and $\gamma(S) = \{(i,j) \in E : i,j \in S\}$. If $S = \{i\}$, we simply write $\delta(i)$ instead of $\delta(\{i\})$. Finally, for any $\bar{E} \subseteq E$, we define $x^k(\bar{E}) = \sum_{(i,j)\in\bar{E}} x^k_{ij}$. Using the above notations, the HMDMTSP is formulated as an integer linear program as follows:

$$\min \sum_{k=1}^{n} \sum_{e\in E} c^k_e x^k_e \quad \text{subject to:} \tag{3.1}$$

$$x^k(\delta(i)) = 2y^k_i \quad \forall i \in T, k \in \{1,\dots,n\}, \tag{3.2}$$

$$x^k(\delta(S)) \geq 2y^k_i \quad \forall i \in S, S \subseteq T, k \in \{1,\dots,n\}, \tag{3.3}$$

$$\sum_{k=1}^{n} y^k_i = 1 \quad \forall i \in T, \tag{3.4}$$

$$y^k_i = 1 \quad \forall k \in \{1,\dots,n\}, i \in R_k, \tag{3.5}$$

$$x^k_e \in \{0,1,2\} \quad \forall e \in \{(d_k, j) : j \in T\},\ k \in \{1,\dots,n\}, \tag{3.6}$$

$$x^k_e \in \{0,1\} \quad \forall e \in \{(i,j) : i \in T, j \in T\},\ k \in \{1,\dots,n\}, \tag{3.7}$$

$$y^k_i \in \{0,1\} \quad \forall i \in T, k \in \{1,\dots,n\}. \tag{3.8}$$

In the above formulation, the constraints in Eq. (3.2) ensure the number of edges of vehicle k, incident on a target $i \in T$ is equal to 2 if and only if target i is visited by the vehicle k. The constraints in Eq. (3.4) ensure that each target $i \in T$ is visited by some vehicle. The constraints in Eq. (3.3) are the connectivity or sub-tour elimination constraints. They ensure a feasible solution has no sub-tours of any subset of targets in T. The constraints in Eq. (3.5) are the vehicle–target assignment constraints for the functional heterogeneous targets. Constraints in Eq. (3.6), (3.7) and (3.8) are the integrality restrictions on the decision variables. If the integrality restrictions in constraints (3.6), (3.7) and (3.8) are relaxed, then we call that model a linear programming relaxation. In the following subsection, we shall strengthen the linear programming relaxation of the model (3.1)–(3.8) by introducing additional valid inequalities.

3.3.1 Additional valid inequalities

In this section, we develop two classes of valid inequalities for the HMDMTSP. Consider the constraints in Eq. (3.3). For any vehicle $k \in \{1, \ldots, n\}$ and $S = \{i, j\}$ where $i, j \in T$, Eq. (3.3) reduces to $x^k(\delta(i)) + x^k(\delta(j)) - 2x^k_{ij} \geq 2y^k_i$ and $x^k(\delta(i)) + x^k(\delta(j)) - 2x^k_{ij} \geq 2y^k_j$. Further simplification using Eq. (3.2) yields

$$x^k_{ij} \leq y^k_j \text{ and } x^k_{ij} \leq y^k_i. \tag{3.9}$$

The inequalities that are valid for a MDTSP are also valid for the HMDMTSP. We particularly examine the *2-matching inequalities* available for the MDTSP, TSP, and MDRSP [11, 50]. Specifically, we consider the following inequality for every vehicle k:

$$x^k(\gamma(H)) + x^k(\mathcal{T}) \leq \sum_{i \in H} y^k_i + \frac{|\mathcal{T}| - 1}{2} \tag{3.10}$$

for all $H \subseteq T$ and $\mathcal{T} \subset \delta(H)$. Here H is called the handle, and $\mathcal{T}$ the teeth. H and $\mathcal{T}$ satisfy the following conditions:

- the edges in the teeth are not incident to any depots in the set D,
- no two edges in the teeth are incident on the same target,
- $|\mathcal{T}| \geq 3$ and odd.

The proof of validity of the above inequality is given by the following proposition:

Proposition 3.1. *The 2-matching inequality in Eq.* (3.10) *is valid for any feasible solution to the HMDMTSP.*

Proof. See 2.2 $\square$

3.4 Branch-and-cut algorithm

We now outline the main components of our branch-and-cut algorithm to compute optimal solutions for the HMDMTSP. Let $\bar{\tau}$ denote the optimal solution to the problem.

Step 1 (Initialization). Set the iteration count $t \leftarrow 1$ and the initial upper bound $\bar{\alpha}$ on the optimal objective as $+\infty$. The initial linear sub-problem is then defined as

$$\min \sum_{k=1}^{n} \sum_{e \in E} c_e^k x_e^k \quad \text{subject to:}$$

$$\begin{aligned}
x^k(\delta(i)) &= 2y_i^k \quad \forall i \in T, k \in \{1, \ldots, n\}, \\
\sum_{k=1}^{n} y_i^k &= 1 \quad \forall i \in T, \\
y_i^k &= 1 \quad \forall k \in \{1, \ldots, n\}, i \in R_k, \\
x_e^k &\geq 0 \quad \forall k \in \{1, \ldots, n\}, e \in E \text{ and} \\
y_i^k &\geq 0 \quad \forall i \in T, k \in \{1, \ldots, n\}.
\end{aligned}$$

The initial sub-problem is solved and inserted in a list $\mathcal{L}$.

Step 2 (Termination check and sub-problem selection). If the list $\mathcal{L}$ is empty, then stop. Otherwise, select a sub-problem from the list with the lowest objective value. This choice of sub-problem is called best-first policy [61].

Step 3 (Sub-problem solution). Set $t \leftarrow t+1$. Let α be the solution objective value.

If $\alpha \geq \bar{\alpha}$, then go to STEP 2. Otherwise, if the solution is feasible for the HMDMTSP, set $\bar{\alpha} \leftarrow \alpha$, update $\bar{\tau}$ and go to STEP 2.

STEP 4 (LP-rounding heuristic). If the solution is fractional, $\bar{\alpha} = +\infty$ and t is a multiple of 3, apply the following heuristic: Given a fractional solution $(\mathbf{x}, \mathbf{y})$, we partition the set T into n subsets, one for each vehicle. We assign target $i \in T \setminus (\cup_k R_k)$ to a vehicle k that has the maximum y_i^k value in the fractional solution. The targets in the set R_k are assigned to vehicle k. We now have n disjoint subsets of the set T. We then solve a traveling salesman problem for each vehicle k on its partition and its depot d_k, using the LKH heuristic [31]. Let us denote the resulting feasible solution by τ^* and let α^* be the objective value of the solution τ^*. If $\alpha^* \leq \bar{\alpha}$, set $\bar{\alpha} \leftarrow \alpha^*$ and update $\bar{\tau}$ with τ^*.

STEP 5 (Constraint separation and generation). Introduce violated sub-tour elimination constraints (3.3), connectivity constraints (3.9) and 2-matching constraints (3.10). If no constraints can be generated using the current fractional solution, then go to STEP 6, else go to STEP 3.

STEP 6 (Branching.) Create two sub-problems by branching on a fractional y_i^k or x_e^k variable. First, select a fractional y_i^k variable, based on the *strong branching* rule [1]. If all these variables are integer, then select a fractional x_e^k variable using the same rule. Then insert both the sub-problems in the list $\mathcal{L}$ and go to STEP 2.

In the following paragraphs we detail the separation algorithms used to generate violated constraints in STEP 5. For every vehicle k, we denote by $G_k^* = (V_k^*, E_k^*)$ the *support graph* associated with a given fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$ *i.e.*, $V_k^* := \{i \in$

$T : y_i^{k*} > 0\} \cup \{d_k\}$ and $E_k^* := \{e \in E : x_e^{k*} > 0\}$.

Separation of constraints (3.3) *and* (3.9)

As shown previously Sec. 3.3.1, the inequalities in Eq. (3.3) reduce to Eq. (3.9) when $|S| = 2$. For every vehicle k, the violation of the inequality in Eq. (3.9) can be verified by examining the inequality for every pair of targets in the set V_k^*. Next, we examine the connected components in G_k^*. Each connected component C that does not contain the depot d_k generates a violated sub-tour elimination constraint for $S = C$ and for each $i \in S$. If a connected component C contains the depot d_k the following procedure is used to find the largest violated sub-tour elimination constraint in $x^k(\delta(S)) \geq 2y_i^k$. Given a connected component C that contains a depot d_k, $i \in C \setminus \{d_k\}$, and a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the most violated constraint of the form $x^k(\delta(S)) \geq 2y_i^k$ can be obtained by computing a minimum $s - t$ cut on a capacitated undirected graph $\bar{G}_k = (\bar{V}_k, \bar{E}_k)$, with $\bar{V}_k = V_k^*$. The vertex s denotes the source vertex and $s = d_k$. The vertex t denotes the sink vertex and $t = i$. The edge set $\bar{E}_k = E_k^*$. Every edge $e \in \bar{E}_k$ is assigned a capacity x_e^{k*}. We now compute the minimum $s - t$ cut $(S, \bar{V}_k \setminus S)$ with $t \in \bar{V}_k \setminus S$. The vertex set $S' = \bar{V}_k \setminus S$ defines the most violated inequality if the capacity of the cut is strictly less than $2y_i^{k*}$. Clearly, the targets i with y_i^{k*} need not be considered. This algorithm can be repeated for every vehicle to generate violated sub-tour elimination constraints.

Separation of 2-matching constraints (3.10)

We use a separation procedure similar to the one used for the MDRSP to separate out the 2-matching constraints. We consider each connected component H of G_k^* as a handle of a possibly violated 2-matching inequality whose two-node teeth correspond to the edges $e \in \delta(H)$ with $x_e^{k*} = 1$. We reject the inequality if the number of teeth is even. The procedure can be implemented in $O(|V_k^*|+|E_k^*|)$ time and can be repeated for each vehicle k.

3.5 Computational results

In this section, we discuss the computational results of the branch-and-cut algorithm. The algorithm was implemented in C++ (GCC version 4.6.3), using the elements of Standard Template Library (STL) and CPLEX 12.4 framework. The internal CPLEX cut generation was disabled and hence, CPLEX was used only to manage the enumeration tree. All the simulations were performed on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @2.53 GHz, 12 GB RAM). The computation times reported are expressed in seconds, and we imposed a time limit of 500 seconds for each run of the algorithm. The performance of the algorithm was tested on instances generated using TSPLIB [69].

3.5.1 Instance generation

We generated 36 HMDMTSP instances using four TSPLIB instances [69] namely, $bays29$, $eil51$, $eil76$ and $eil101$. These instances have $|T| = 29, 51, 76$ and 101 respectively. We performed a computational study on these instances with the number of vehicles $n \in \{3, 4, 5\}$. The depot locations for the vehicles were randomly generated. For a given instance, we had the same cardinality for all the functional heterogeneous target sets R_i. The cardinality of each R_i was chosen from the set $\{1, 3, 5\}$. Hence, for each TSPLIB instance we generated 9 HMDMTSP instances with all possible combinations of n and $|R_i|$ which resulted in a total for 36 instances. The travel cost of each edge for all the vehicles was generated according to the following procedure: for each edge $e = (i, j)$ the cost of traversing the edge e for vehicle $k \in \{1, \ldots, n\}$ was chosen to be $c_e^k = 0.1 \times L_e(2k - 1)$, where L_e is the euclidean distance be-

n	$\|R\|$	%-LB	Nodes	Time
3	1	100.00	1	0.32
3	3	99.99	2	0.42
3	5	99.98	2	0.21
4	1	100.00	1	0.72
4	3	100.00	2	0.99
4	5	100.00	1	0.11
5	1	100.00	1	0.84
5	3	100.00	2	0.42
5	5	100.00	1	0.05

Table 3.1: Computational results for the instance $bays29$

tween the two vertices. Tables 3.1–3.4 summarize the computational behaviour of the branch-and-cut algorithm for all the 36 instances. The column headings are defined as follows:

n**:** number of vehicles;

$|R|$**:** number of functional heterogeneous targets per vehicle;

%-LB: percentage LB/opt, where LB is the objective value of the linear programming relaxation computed at the root node of the enumeration tree and opt is the cost of the optimal solution to the instance;

Nodes: total number of nodes examined in the enumeration tree;

Time: time taken to compute the optimal solution in seconds.

The results show that the proposed branch-and-cut algorithm can solve instances involving up to 101 targets with modest computation times. The %-LB column in both the tables indicate that the lower bound obtained at the root node of the enumeration tree is very tight, typically within 0.5% of the optimum. Hence the proposed integer linear programming formulation for the HMDMTSP is by itself very tight. The maximum computation time over all the 36 instances was 309.04 seconds. Overall, we were able to solve all the 36 TSPLIB based instances, with

n	$\|R\|$	%-LB	Nodes	Time
3	1	99.74	6	1.22
3	3	99.94	3	3.63
3	5	99.86	7	2.59
4	1	99.94	3	1.95
4	3	99.90	6	5.34
4	5	100.00	1	3.67
5	1	99.94	6	4.75
5	3	99.93	7	9.28
5	5	99.99	2	2.97

Table 3.2: Computational results for the instance $eil51$

n	$\|R\|$	%-LB	Nodes	Time
3	1	99.61	12	48.44
3	3	99.50	6	30.47
3	5	99.89	4	10.32
4	1	99.81	11	50.39
4	3	99.52	46	50.02
4	5	99.93	4	18.6
5	1	99.86	8	48.52
5	3	99.82	7	36.18
5	5	99.96	2	99.97

Table 3.3: Computational results for the instance $eil76$

n	$\|R\|$	%-LB	Nodes	Time
3	1	99.92	4	11.18
3	3	100.00	1	31.44
3	5	99.56	135	168.91
4	1	99.97	8	97.69
4	3	99.88	104	229.17
4	5	99.90	23	108.19
5	1	99.85	16	56.06
5	3	100.00	1	222.56
5	5	99.91	58	309.04

Table 3.4: Computational results for the instance $eil101$

the largest instance involving 101 targets, 5 vehicles and 5 functional heterogeneous targets per vehicle.

3.6 Conclusion

In this chapter, we have presented an exact algorithm for the HMDMTSP that arises in the context of monitoring a set of targets and collect relevant data. An integer linear programming formulation including two classes of valid inequalities was proposed. A customized branch-and-cut algorithm was also developed using the proposed formulation. The algorithm was tested on a wide class of benchmark instances from a standard library. The largest solved instance involved 101 targets. Future work can be directed towards development of branch-and-cut approaches accompanied with a polyhedral study to solve the problem with asymmetric costs.

4. FUEL-CONSTRAINED, MULTIPLE DEPOT, VEHICLE ROUTING PROBLEM

In this chapter, we consider a multiple depot, multiple vehicle routing problem with fuel constraints. We are given a set of targets, a set of depots and a set of homogeneous vehicles, one for each depot. The depots are also allowed to act as refueling stations. The vehicles are allowed to refuel at any depot, and our objective is to determine a route for each vehicle with a minimum total cost such that each target is visited at least once by some vehicle, and the vehicles never run out fuel as it traverses its route. We refer this problem as FCMDVRP. This paper presents four new mixed integer linear programming formulations to compute an optimal solution for the problem. Extensive computational results for a large set of instances are also presented.

4.1 Introduction

We extend the classic MDVRP to include fuel constraints for the vehicles. We are given sets of targets, a set of depots, and a set of vehicles, with each vehicle initially stationed at a distinct depot. The depots also perform the role of refueling stations, and it is reasonable to assume that whenever a vehicle visits a depot, it refuels to its full capacity. The objective of FCMDVRP is to determine a route for each vehicle starting and ending at its corresponding depot such that (i) each target is visited at least once by some vehicle, (ii) no vehicle runs out of fuel as it traverses its path, and (iii) the total cost of the routes for the vehicles is minimized. Some of the applications for the FCMDVRP are path-planning for UAVs [73, 75, 52], routing for electric vehicles based on the locations of recharging stations [70, 32], and routing for green vehicles [18]. Some of these application domains are elaborated on the following sections.

4.1.1 Path-planning for UAVs

Small UAVs are being used routinely in military applications such as border patrol, reconnaissance, and surveillance expeditions, and civilian applications like remote sensing, traffic monitoring, and weather and hurricane monitoring [24, 15, 86]. Even though there are several advantages due to small platforms for UAVs, there are resource constraints due to their size and limited payload. It may not be possible for a small UAV to complete a surveillance mission before refueling at one of the depots due to the fuel constraints. For example, consider a typical surveillance mission involving multiple vehicles monitoring a set of targets. To complete this mission, the vehicles might have to start at their respective depot, visit a subset of targets and reach one of the depots for refueling before starting a new route for the rest of the targets. This can be modeled as a FCMDVRP with the depots acting as refueling stations.

4.1.2 Routing problem for green and electric vehicles

Green vehicle routing problem is a variant of the VRP and was introduced by [18] to account for the challenges associated with operating a fleet of AFVs. The US transportation sector accounts for 28% of national greenhouse gas emissions [83]. Several efforts over many decades focusing towards the introduction of cleaner fuels (e.g. ultra low sulphur diesel) and efficient engine technologies have lead to reduced emissions and greater mileage per gallon of fuel used. Government organizations, municipalities, and private companies are converting their fleet of vehicles to AFVs either voluntarily to alleviate the environmental impact of fossil based fuels or to meet environmental regulations. For instance, FedEx, in its overseas operations, employs AFVs that run on biodiesel, liquid natural gas, or compressed natural gas.

A major challenge that hinders the increase in usage of AFVs is the number of alternate-fuel stations available for refueling. The FCMDVRP is a natural problem that arises in this application. An algorithm to compute an optimal solution to the FCMDVRP would generate low cost routes for the vehicles, while respecting their fuel constraints.

Increasing concerns about climate changes and rising green house gas emissions drive the research in sustainable and energy efficient mobility. One such example is the introduction of electrically-powered vehicles. One of the main operational challenges for electric vehicles in transport applications is their limited range and the availability of recharging stations. The number of electric stations in the US is a mere 9,571 with a total of 24,631 charging outlets [82]. Fig. 4.1 shows a map with the locations of the electric stations in Texas, USA; observe that the distribution of the electric stations is very sparse except in the four major cities Dallas, Houston, Austin, and San Antonio. Successful adoption of electric vehicles will strongly depend on the methods to alleviate the range and recharging limitations. If we consider the range and the recharging stations for the electric vehicles as analogues to the fuel capacity and refueling stations of vehicles that run on fossil-based or alternate fuels respectively, then the problem of electric vehicle routing subject to the range constraints and limited availability of electric stations can be modeled as an FCMDVRP. Clearly, any feasible solution to the FCMDVRP can be used to implement a feasible route for an electric vehicle.

4.2 Related work

The FCMDVRP is $\mathcal{NP}$-hard because it contains the VRP as a special case. The existing literature on the FCMDVRP is quite scarce. The multiple depot, single vehicle variant of the FCMDVRP was first introduced in [39]. When the travel costs

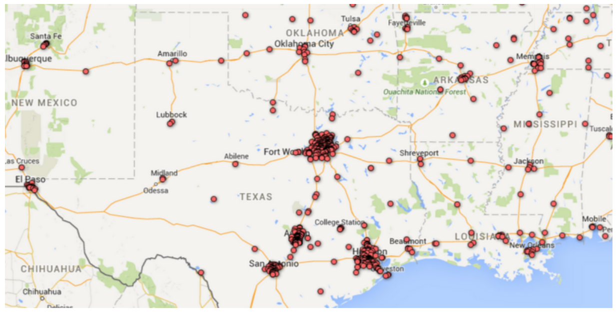


Figure 4.1: Electric station locations in Texas, USA

are symmetric and satisfy the triangle inequality, [39] provide an approximation algorithm for this variant. They assume that the minimum fuel required to travel from any target to its nearest depot is at most equal to $F\alpha/2$ units, where α is a constant in the interval $[0, 1)$ and F is the fuel capacity of the vehicle. This is a reasonable assumption as, in any case, one cannot have a feasible tour if there is a target that cannot be visited from any of the depots. Using these assumptions, [39] present a $(3(1+\alpha))/(2(1-\alpha))$ approximation algorithm for the problem. [73] formulate this multiple depot single vehicle variant as a MILP and present k-opt based exchange heuristics to obtain feasible solutions within 7% of the optimal, on an average. Later, [75] extend the approximation algorithm in [39] to the asymmetric case and also present heuristics to solve the asymmetric version of this variant. Furthermore, variable neighborhood search heuristics for FCMDVRP with heterogeneous vehicles, *i.e.,* vehicles with different fuel capacities, are presented in [52]. More recently, an approximation algorithm and heuristics are developed for the FCMDVRP in [56].

Variants of the classic VRP that are closely related to the FCMDVRP include the distance constrained VRP [47, 53, 35, 36, 59], the orienteering problem [21, 84], and the capacitated version of the arc-routing problem [26, 67]. The distance-constrained

VRP is a special case of the FCMDVRP with a single vehicle and single depot that can be considered as a fuel station. The FCMDVRP is also quite different and more general compared to orienteering problem where one is interested in maximizing the number of targets visited by the vehicle subject to its fuel constraints. Lastly, the arc routing problem is a single depot VRP given a set of intermediate facilities, and the vehicle has to cover a subset of edges along which targets are present. The vehicle is required to collect goods from the targets as it traverses the given set of edges and unloads the goods at the intermediate facilities. The goal of this problem is to find a tour of minimum length that starts and ends at the depot such that the vehicle visits the given subset of edges, and the total amount of goods carried by the vehicle does not exceed the capacity of the vehicle along the tour. One of the key differences between the arc routing problem and the FCMDVRP is that there is no requirement that any subset of edges must be visited in the FCMDVRP.

The aim of this paper is to introduce and compare four different formulations for the FCMDVRP and present branch-and-cut algorithms for the formulations. The first two formulations are arc-based, and the rest are node-based formulations that use the MTZ constraints [55]. The major contributions of this paper are as follows: (1) present four new formulations for the FCMDVRP, (2) compare the formulations both analytically and empirically, and (3) through extensive computational experiments, show that instances with maximum of 40 targets are within the computational reach of a branch-and-cut algorithm based on the best of the four formulations.

The rest of the paper is organized as follows. Sec. 4.3 states the formal definition of the problem and introduces notations. In Sec. 4.4, we develop the four mixed integer linear programming formulations. The first two formulations are arc-based and the rest are node-based formulations *i.e.,* decision variables for enforcing the fuel constraints are introduced for each edge and each target for the arc-based and the node-based formulations, respectively. The linear programming relaxations of the formulations are analytically compared in this section. In Sec. 4.5, we present the computational results followed by conclusions and possible extensions.

4.3 Problem definition

Let T denote the set of targets $\{t_1, \ldots, t_n\}$. Let D denote the set of depots or refueling stations $\{d_1, \ldots, d_k\}$; each depot d_k is equipped with a vehicle v_k. The FCMDVRP is defined on a directed graph $G = (V, E)$ where $V = T \cup D$ and E is the set of edges joining any two vertices in V. We assume that G does not contain any self-loops. Each edge $(i, j) \in E$ is associated with a non-negative cost c_{ij} required to travel from vertex i to vertex j and f_{ij}, the fuel spent by traveling from i to j. It is assumed that the cost of traveling from vertex i to vertex j is directly proportional to the fuel spent in traversing the edge (i, j) *i.e.*, $c_{ij} = K \cdot f_{ij}$ (c_{ij} and c_{ji} may be different, but for the purpose of this paper, we assume $c_{ij} = c_{ji}$). It is also assumed that travel costs satisfy the triangle inequality *i.e.*, for every $i, j, k \in V$, $c_{ij} + c_{jk} \geq c_{ik}$. Furthermore, let F denote the fuel capacity of all the vehicles. The FCMDVRP consists of finding a route for each vehicle such that the vehicle v_k starts and ends its route at its depot d_k, each target is visited at least once by some vehicle, the fuel required by any vehicle to travel any segment of the route which joins two consecutive depots in the route must be at most equal to F, and the sum of the cost of all the edges present in the routes is a minimum.

4.4 Mathematical formulations

This section presents four formulations for the FCMDVRP. The first two formulations are arc based, and the remaining formulations are node based. The arc based and edge based formulations have additional decision variables for each edge and vertex respectively, to impose the fuel constraints. For any given formulation

$\mathcal{F}$, let $\mathcal{F}^L$ denote its linear programming relaxation obtained by allowing the integer variables to take continuous values within the lower and upper integer bounds, and opt($\mathcal{F}$) denote the cost of its optimal solution.

4.4.1 Arc-based formulations

We first present an arc based formulation $\mathcal{F}_1$ for the FCMDVRP, inspired by the models for standard routing problems [81, 36]. Each edge $(i, j) \in E$ is associated with a variable x_{ij}, which equals 1 if the edge (i, j) is traversed by the vehicle, and 0 otherwise. Also, associated with each edge (i, j) is a flow variable z_{ij} which denotes the total fuel consumed by any vehicle as it starts from a depot to the vertex j, when the predecessor of j is i. Using the above variables, the formulation $\mathcal{F}_1$ is given as follows:

$$(\mathcal{F}_1) \quad \text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

subject to:

$$\sum_{i\in V} x_{di} = \sum_{i\in V} x_{id} \quad \forall\, d \in D, \tag{4.1}$$

$$\sum_{i\in V} x_{ij} = 1 \text{ and } \sum_{i\in V} x_{ji} = 1 \quad \forall\, j \in T, \tag{4.2}$$

$$\sum_{j\in V} z_{ij} - \sum_{j\in V} z_{ji} = \sum_{j\in V} f_{ij}x_{ij} \quad \forall\, i \in T, \tag{4.3}$$

$$0 \leq z_{ij} \leq F x_{ij} \quad \forall\, (i, j) \in E, \tag{4.4}$$

$$z_{di} = f_{di}x_{di} \quad \forall\, i \in T,\, d \in D, \text{ and} \tag{4.5}$$

$$x_{ij} \in \{0, 1\} \quad \forall\, (i, j) \in E. \tag{4.6}$$

In the above formulation the Eqs. (4.1) – (4.2) impose the degree constraints on the depots and the targets. The constraints in Eqs. (4.3) are the connectivity constraints; they eliminate sub tours of the targets. Eqs. (4.4) and (4.5) together impose $0 \leq z_{ij} \leq F$ and they ensure that the fuel consumed by the vehicle to travel up to a depot does not exceed the fuel capacity F. Finally, the constraints in Eqs. (4.6) impose the binary restrictions on the variables.

Next, we present another arc-based formulation $\mathcal{F}_2$ which is a strengthened version of $\mathcal{F}_1$. To strengthen the formulation $\mathcal{F}_1$, we use a well-known general principle, called *lifting*.

The following proposition is a modified version of the Proposition 1 presented in [36] for the distance constrained vehicle routing problem; it strengthens the bounds given by the constraints in (4.4).

Proposition 4.1. *The inequalities in* (4.4) *can be strengthened as follows:*

$$z_{ij} \leq (F - t_j)x_{ij} \quad \forall j \in T,\ (i,j) \in E, \tag{4.7}$$

$$z_{id} \leq F x_{id} \quad \forall i \in T \text{ and } d \in D, \tag{4.8}$$

$$z_{ij} \geq (s_i + f_{ij})x_{ij} \quad \forall i \in T,\ (i,j) \in E, \tag{4.9}$$

where, $t_i = \min_{d \in D} f_{id}$ *and* $s_i = \min_{d \in D} f_{di}$*.*

Proof. When j is a depot, the constraints in (4.8) and (4.4) coincide. We now discuss the case when both i and j are targets. When $x_{ij} = 1$, any vehicle that traverses this edge (i,j) consumes at least $(s_i + f_{ij})$ amount of fuel. As a result, the constraint in (4.9) strengthens the lower bound of z_{ij} in (4.4). Similarly, the total fuel consumed by any vehicle that traverses the edge (i,j) cannot be greater that $(F - t_j)$, where t_j is the minimum amount of fuel required by any vehicle to reach a depot from target j. Therefore, the constraint in (4.7) strengthens the upper bound of z_{ij} in (4.4). $\square$

Hence, the second arc-based formulation is as follows:

$$(\mathcal{F}_2) \quad \text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

subject to: (4.1) – (4.3), (4.5) – (4.6), and (4.7) – (4.9).

Corollary 4.1. $\mathrm{opt}(\mathcal{F}_2^L) \geq \mathrm{opt}(\mathcal{F}_1^L)$. □

4.4.2 Node-based formulations

In this section, we present a node-based formulation for the FCMDVRP based on the models for the distance constrained VRP [16, 35]. For the node based formulation, apart from the binary variable x_{ij} for each edge $(i,j) \in E$, we have an auxiliary variable u_i for each vertex i, that indicates the amount of fuel spent by a vehicle when it reaches the vertex i. We assume $u_d = 0$ as the vehicles are refueled to their capacity when they reach a depot. In addition, we will also use the following two parameters: $t_i = \min_{d\in D} f_{id}$ and $s_i = \min_{d\in D} f_{di}$ for every vertex $i \in V$. For any $d \in D$, $t_d = 0$ and $s_d = 0$. Using the above notations, the formulation $\mathcal{F}_3$ is given as follows:

$$(\mathcal{F}_3) \quad \text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

subject to: (4.1), (4.2), and (4.6),

$$u_i - u_j + Mx_{ij} \leq M - f_{ij} \quad \forall i \in V, j \in T, \tag{4.10}$$

$$u_i \geq s_i + \sum_{d\in D}(f_{di} - s_i)x_{di} \quad \forall i \in T, \text{ and} \tag{4.11}$$

$$u_i \leq F - t_i - \sum_{d\in D}(f_{id} - t_i)x_{id} \quad \forall i \in T. \tag{4.12}$$

The constraints in Eq. (4.10) serve both as sub-tour elimination and fuel constraints. It eliminates sub tours of the targets and ensures any route that starts and ends at a depot consumes at most F amount of fuel. This can be easily observed by aggregating the constraints for any sub tour of the targets and for any route starting and ending at a depot [16]. The value of M in the constraint is given by $M = \max_{(i,j)\in E}\{F - s_j - t_i + f_{ij}\}$. The constraints in Eqs. (4.11) and (4.12) specify the upper and lower bounds on u_i, for every vertex i. The following proposition strengthens the fuel constraints and the bounds on u_i.

Proposition 4.2. *The inequalities in* (4.10), (4.11), *and* (4.12) *can be strengthened as follows:*

$$u_i - u_j + Mx_{ij} + (M - f_{ij} - f_{ji})x_{ji} \leq M - f_{ij} \quad \forall i, j \in T, \tag{4.13}$$

$$u_i \geq \sum_{j\in V}(s_j + f_{ji})x_{ji} \quad \forall i \in T, \tag{4.14}$$

$$u_i \leq F - \sum_{j\in V}(t_j + f_{ij})x_{ij} \quad \forall i \in T, \textit{ and} \tag{4.15}$$

$$u_i \leq F - t_i - \sum_{d\in D}(F - t_i - f_{di})x_{di} \quad \forall i \in T, \tag{4.16}$$

where, $x_{ii} = 0$ *and* $x_{ij} = 0$ *whenever* $s_i + f_{ij} + t_j > F$.

Proof. The constraint in Eq. (4.13) can be obtained by lifting the variable x_{ji} in Eq. (4.10). We compute the value of the coefficient α that makes the following constraint valid:

$$u_i - u_j + Mx_{ij} + \alpha x_{ji} \leq M - f_{ij}.$$

The equation is valid when $x_{ji} = 0$, as it reduces to (4.10). When $x_{ji} = 1$, we have $x_{ij} = 0$ and $u_j + f_{ji} = u_i$. Hence, the best value of α that makes the equation valid is given by $M - f_{ij} - f_{ji}$.

Similarly, Eq. (4.14) can be obtained by lifting every x_{ji} variable for $j \in T$ in any order. We will illustrate the lifting procedure for one of the x_{ji} variables. This involves computing the coefficient α that makes the following constraint valid:

$$u_i \geq s_i + \sum_{d \in D} (f_{di} - s_i) x_{di} + \alpha x_{ji}.$$

The above equation is valid when $x_{ji} = 0$, and when $x_{ji} = 1$, we have $x_{di} = 0$ and $\alpha \leq u_i - s_i$. The best value of α that does not remove any feasible solution is hence given by $s_j + f_{ji} - s_i$. Similarly, the coefficients of the other x_{ji} variables can be computed. The resulting constraint is given by

$$u_i \geq s_i + \sum_{j \in V} (s_j + f_{ji} - s_i) x_{ji} \quad \forall i \in V.$$

In the above equation, $s_j = 0$ for $j \in D$. The above equation reduces to Eq. (4.14) due to the degree constraints in (4.2). The constraints in Eq. (4.15) are similarly obtained from (4.12) by lifting the x_{ij} variable for every $j \in T$. The proof is omitted as it is similar to the previous ones in the proposition. The constraints in Eq. (4.16) are valid bounding constraints for the FCMDVRP when the target i is the first target that is visited by any vehicle as it leaves the depot. In this case, the Eq. (4.12) reduces to $u_i \leq F - t_i$. We further strengthen this constraint by lifting the variable x_{di} for every $d \in D$. The lifting coefficient α for x_{di} takes the value $-(F - t_i - f_{di})$ and the resulting constraint is given by Eq. (4.16). $\square$

Hence, the second node-based formulation is as follows:

$$(\mathcal{F}_4) \quad \text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}x_{ij}$$

subject to: (4.1), (4.2), (4.6), and (4.13) – (4.16).

Corollary 4.2. $\mathrm{opt}(\mathcal{F}_4^L) \geq \mathrm{opt}(\mathcal{F}_3^L)$. □

4.5 Computational results

In this section, we discuss the computational performance of the four formulations presented in the previous section. The mixed integer linear programs were implemented in Java, using the traditional branch-and-cut framework of CPLEX version 12.4. All the simulations were performed on a Dell Precision T5500 workstation (Intel Xeon E5630 processor @2.53 GHz, 12 GB RAM). The computation times reported are expressed in seconds, and we imposed a time limit of 3,600 seconds for each run of the algorithm. The performance of the algorithm was tested with randomly generated test instances.

4.5.1 Instance generation

The problem instances were randomly generated in a square grid of size [100,100] with 5 fixed depot locations. The number of targets varies from 10 to 40 in steps of five, while their locations were uniformly distributed in the square grid; for each $|T| \in \{10, 15, 20, 25, 30, 25, 40\}$, we generated five random instances. Each depot contains a vehicle. The travel costs and the fuel consumed to travel between any pair of vertices are assumed to be directly proportional to the Euclidean distances between the pair. For each of these problems, we generate four possible fuel capacities F as a function of the the distance to the farthest target from any depot, λ.

The fuel capacity F of the vehicles gets the values 2.25λ, 2.5λ, 2.75λ and 3λ. In total, we generated 140 instances, and ran the branch-and-cut algorithm for all the formulations.

Tables 4.1 and 4.2, and Fig. 4.2–4.3 summarize the computational behavior of the algorithms for all the 140 instances. The following nomenclature is used throughout the rest of the paper:

#: instance number;

opt($\mathcal{F}_i^L$): linear programming relaxation solution for formulation i;

n: instance size *i.e.*, number of targets in the instance;

%-LB: percentage LB/opt, where LB is the objective value of the linear programming relaxation computed at the root node of the branch and bound tree and opt is the cost of the optimal solution to the instance;

total: total number of test instances of a given size;

succ: number of instances for which optimal solutions were computed within a time limit of 3,600 seconds.

Table 4.1 compares the cost of the LP relaxations of the four formulations presented in Sec. 4.4 for the 40 target instances. The results in table 4.1 provide an empirical comparison of the formulations presented in 4.4; the observed behavior is expected because the formulations $\mathcal{F}_2$ and $\mathcal{F}_4$ are strengthened versions of $\mathcal{F}_1$ and $\mathcal{F}_3$, respectively (see corollaries 4.1 and 4.2). As for the LP relaxations of formulations $\mathcal{F}_2$ and $\mathcal{F}_4$, it is difficult to conclude that one is better than the other since $\mathcal{F}_4$ produces better relaxation values than $\mathcal{F}_2$ only for 60% of the instances. Hence, the rest of the computational results compares the formulations $\mathcal{F}_2$ and $\mathcal{F}_4$.

#	opt($\mathcal{F}_1^L$)	opt($\mathcal{F}_2^L$)	opt($\mathcal{F}_3^L$)	opt($\mathcal{F}_4^L$)
1	496.42	509.24	426.17	518.00
2	487.31	496.39	426.17	518.00
3	480.55	487.40	426.17	518.00
4	475.23	480.33	426.17	518.00
5	444.35	458.01	389.08	434.00
6	435.45	445.70	389.08	434.00
7	428.44	436.47	389.08	434.00
8	423.06	429.97	389.08	434.00
9	396.10	403.96	367.11	452.00
10	392.87	398.72	367.11	452.00
11	390.42	394.66	367.11	452.00
12	388.40	391.85	367.11	452.00
13	481.22	493.64	427.04	461.00
14	469.76	479.81	427.04	461.00
15	461.16	469.20	427.04	461.00
16	454.80	461.47	427.04	461.00
17	503.19	516.58	461.07	523.00
18	494.98	504.84	461.07	523.00
19	489.64	496.31	461.07	523.00
20	485.92	489.99	461.07	523.00

Table 4.1: Cost of the LP relaxation for the 40 target instances

Table 4.2 shows the number of instances of different sizes solved to optimality by the formulations $\mathcal{F}_2$ and $\mathcal{F}_4$ within the time limit of 3600 seconds. The plot in Fig. 4.2 shows the average time taken by the two formulations to compute the optimal solution. The table 4.2 and Fig. 4.2 indicate that the arc-based formulation $\mathcal{F}_2$ outperforms the node-based formulation $\mathcal{F}_4$ for the larger instances. For the smaller sized instances, it is difficult to differentiate between the two formulations. The plot in Fig. 4.3 shows the percentage LB/opt for both the formulations (LB is the objective value of the linear programming relaxation computed at the root node of the branch and bound tree and opt is the cost of the optimal solution to the instance; for the instances not solved to optimality, opt represents the cost of the best feasible solution obtained at the end of 3,600 seconds). We observe that the %LB is consistently better for formulation $\mathcal{F}_2$. This plot also provides empirical

		$\mathcal{F}_2$	$\mathcal{F}_4$
n	total	succ	succ
10	20	20	20
15	20	20	20
20	20	20	20
25	20	20	14
30	20	20	5
35	20	20	15
40	20	19	1

Table 4.2: Comparison of formulations $\mathcal{F}_2$ and $\mathcal{F}_4$

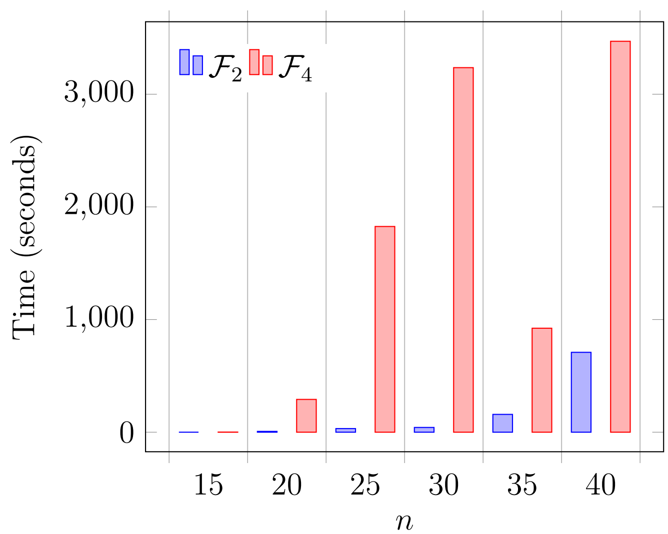


Figure 4.2: Average computation time

evidence to the claim that the arc based formulation $\mathcal{F}_2$ outperforms the node based formulation $\mathcal{F}_4$.

4.6 Conclusion

In this chapter, we have presented four different MILP formulations for the multiple depot fuel constrained multiple vehicle routing problem. The problem arises frequently in the context of path planning for UAVs, green vehicle routing and rout-

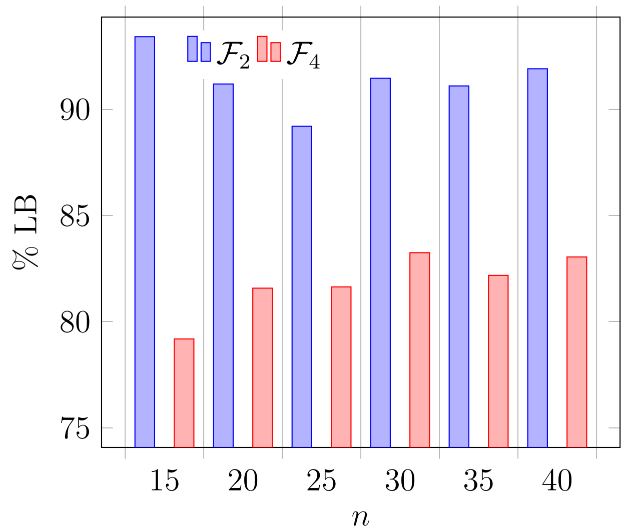


Figure 4.3: Average % LB

ing electric vehicles. The formulations have been compared both analytically and empirically, and it is observed that a strengthened arc-based formulation ($\mathcal{F}_2$) performs better in terms of computing optimal solutions to the problem. Computational experiments on a large number of test instances corroborate this observation. Future work can be directed towards developing similar MILP formulations and branch-and-cut algorithms to solve a heterogeneous variant of the problem *i.e.,* with vehicles having different fuel capacities.

5. GENERALIZED MULTIPLE DEPOT TRAVELING SALESMEN PROBLEM*

In this chapter, we present the GMDTSP is a variant of the MDTSP, where each salesman starts at a distinct depot, the targets are partitioned into clusters and at least one target in each cluster is visited by some salesman. The GMDTSP is an $\mathcal{NP}$-hard problem as it generalizes the MDTSP and has practical applications in design of ring networks, vehicle routing, flexible manufacturing scheduling and postal routing. We present an integer programming formulation for the GMDTSP and valid inequalities to strengthen the linear programming relaxation. Furthermore, we present a polyhedral analysis of the convex hull of feasible solutions to the GMDTSP and derive facet-defining inequalities that strengthen the linear programming relaxation of the GMDTSP. All these results are then used to develop a branch-and-cut algorithm to obtain optimal solutions to the problem. The performance of the algorithm is evaluated through extensive computational experiments on several benchmark instances.

5.1 Introduction

The GMDTSP is an important combinatorial optimization problem that has several practical applications including but not limited to maritime transportation, health-care logistics, survivable telecommunication network design [9], material flow system design, postbox collection [46], and routing unmanned vehicles [54, 64]. The GMDTSP is formally defined as follows: let $D := \{d_1, \ldots, d_k\}$ denote the set of de-

*Reprinted with permission from "Generalized multiple depot traveling salesmen problem - polyhedral study and exact algorithm" by Kaarthik Sundar and Sivakumar Rathinam. *Computers & Operations Research*, 70:39 55, Copyright [2016] by Elsevier Ltd.

pots and T, the set of targets. We are given a complete undirected graph $G = (V, E)$ with vertex set $V := T \cup D$ and edge set $E := \{(i, j) : i \in V, j \in T\}$. In addition, a proper partition $C_1, \ldots, C_m$ of T is given; these partitions are called *clusters*. For each edge $(i, j) = e \in E$, we associate a non-negative cost $c_e = c_{ij}$. The GMDTSP consists of determining a set of at most k simple cycles such that each cycle starts an ends at a distinct depot, at least one target from each cluster is visited by some cycle and the total cost of the set of cycles is a minimum. The GMDTSP reduces to a MDTSP [11] when every cluster is a singleton set. The GMDTSP involves two related decisions:

1. choosing a subset of targets $S \subseteq T$, such that $|S \cap C_h| \geq 1$ for $h = 1, \ldots, m$;

2. solving a MDTSP on the subgraph of G induced by $S \cup D$.

The GMDTSP can be considered either as a generalization of the MDTSP in [11] where the targets are partitioned into clusters and at least one target in each cluster has to be visited by some salesman or as a multiple salesmen variant of the symmetric GTSP [20, 23]. [11] and [20] present a polyhedral study of the MDTSP and GTSP polytope respectively, and develop a branch-and-cut algorithm to compute optimal solutions for the respective problem.

This is the first work in the literature that analyzes the facial structure and derives additional valid and facet-defining inequalities for the convex hull of feasible solutions to the GMDTSP. This chapter presents a MILP formulation and develops a branch-and-cut algorithm to solve the problem to optimality. This work generalizes the results of the two aforementioned problems namely the MDTSP [11] and the GTSP [20].

5.1.1 Related work

A special case of the GMDTSP with one salesman, the symmetric GTSP, was first introduced by [43] and [71] in relation to record balancing problems arising in computer design and to the routing of clients through agencies providing various services respectively. Since then, the GTSP has attracted considerable attention in the literature as several variants of the classical traveling salesman problem can be modeled as a GTSP [46, 19, 63, 54]. [62] developed a procedure to transform a GTSP to an asymmetric traveling salesman problem and [48] investigated the asymmetric counterpart of the GTSP. Despite most of the aforementioned applications of the GTSP [46] extending naturally to their multiple depot variant, there are no exact algorithms in the literature to address the GMDTSP.

A related generalization of the GMDTSP can be found in the VRP literature. VRPs are capacitated counterparts for the TSPs where the vehicles have a limited capacity and each target is associated with a demand that has to be met by the vehicle visiting that target. The multiple VRPs can be classified based on whether the vehicles start from a single depot or from multiple depots. The GVRP is a capacitated version of the GMDTSP with all the vehicles starting from a single depot. [9] present four formulations for the GVRP, compare the linear relaxation solutions for them, and develop a branch-and-cut to optimally solve the problem. [45] models the GVRP as a location-routing problem. On the contrary, [27] develop an algorithm to transform the GVRP into a capacitated arc routing problem, which therefore enables one to utilize the available algorithms for the latter to solve the former. In a more recent paper, [7] study a special case of the GVRP derived from a waste collection application where each cluster contains at most two vertices. The authors describe a number of heuristic solution procedures, including two constructive heuristics, a local search method and an ant colony heuristic to solve several practical instances. To our knowledge, there are no algorithms in the literature to compute optimal solutions to the generalized multiple depot vehicle routing problem or the GMDTSP.

The objective of this paper is to develop an integer programming formulation for the GMDTSP, study the facial structure of the GMDTSP polytope and develop a branch-and-cut algorithm to solve the problem to optimality. The rest of the paper is organized as follows: in Sec. 5.2 we introduce notation and present the integer programming formulation. In Sec. 5.3, the facial structure of the GMDTSP polytope is studied and its relation to the MDTSP polytope [11] is established. We also introduce a general theorem that allows one to lift any facet of the MDTSP polytope into a facet of the GMDTSP polytope. We further use this result to develop several classes of facet-defining inequalities for the GMDTSP. In the subsequent sections, the formulation is used to develop a branch-and-cut algorithm to obtain optimal solutions. The performance of the algorithm is evaluated through extensive computational experiments on 116 benchmark instances from the GTSP library [30].

5.2 Problem formulation

We now present a mathematical formulation for the GMDTSP inspired by models in [11] and [20]. We propose a two-index formulation for the GMDTSP. We associate to each feasible solution $\mathcal{F}$, a vector $\mathbf{x} \in \mathbb{R}^{|E|}$ (a real vector indexed by the elements of E) such that the value of the component x_e associated with edge e is the number of times e appears in the feasible solution $\mathcal{F}$. Note that for some edges $e \in E$, $x_e \in \{0, 1, 2\}$ *i.e,* we allow the degenerate case where a cycle can only consist of a depot and a target. If e connects two vertices i and j, then (i, j) and e will be used interchangeably to denote the same edge. Similarly, associated to $\mathcal{F}$, is also a vector $\mathbf{y} \in \mathbb{R}^{|T|}$, *i.e.,* a real vector indexed by the elements of T. The value of the component y_i associated with a target $i \in T$ is equal to one if the target i is visited by a cycle and zero otherwise.

For any $S \subset V$, we define $\gamma(S) = \{(i,j) \in E : i,j \in S\}$ and $\delta(S) = \{(i,j) \in E : i \in S,\ j \notin S\}$. If $S = \{i\}$, we simply write $\delta(i)$ instead of $\delta(\{i\})$. We also denote by $C_{h(v)}$ the cluster containing the target v and define $W := \{v \in T : |C_{h(v)}| = 1\}$. Finally, for any $\hat{E} \subseteq E$, we define $x(\bar{E}) = \sum_{(i,j)\in\bar{E}} x_{ij}$, and for any disjoint subsets $A, B \subseteq V$, $(A : B) = \{(i,j) \in E : i \in A, j \in B\}$ and $x(A : B) = \sum_{e\in(A:B)} x_{ij}$. Using the above notations, the GMDTSP is formulated as a mixed integer linear program as follows:

$$\text{Minimize} \quad \sum_{e\in E} c_e x_e \tag{5.1}$$

subject to

$$x(\delta(i)) = 2y_i \quad \forall i \in T, \tag{5.2}$$

$$\sum_{i\in C_h} y_i \geq 1 \quad \forall h \in \{1,\dots,m\}, \tag{5.3}$$

$$x(\delta(S)) \geq 2y_i \quad \forall S \subseteq T, i \in S, \tag{5.4}$$

$$x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 2(y_j + y_k) \;\; \forall j,k \in T; D' \subset D, \tag{5.5}$$

$$x(D' : \{j\}) + 2x(\gamma(\bar{S})) + x(\{k\} : D \setminus D') \leq \sum_{v\in\bar{S}} 2\,y_v - y_i$$

$$\forall i \in S; j,k \in T; S \subseteq T \setminus \{j,k\}, S \neq \emptyset; \bar{S} = S \cup \{j,k\}; D' \subset D, \tag{5.6}$$

$$x_e \in \{0,1\} \quad \forall e \in \gamma(T), \tag{5.7}$$

$$x_e \in \{0,1,2\} \quad \forall e \in (D : T), \tag{5.8}$$

$$y_i \in \{0,1\} \quad \forall i \in T. \tag{5.9}$$

In the above formulation, the constraints in (5.2) ensure the number edges incident on any vertex $i \in T$ is equal to 2 if and only if target i is visited by a cycle ($y_i = 1$). The constraints in (5.3) force at least one target in each cluster to be visited. The constraints in (5.4) are the connectivity or sub-tour elimination constraints. They ensure a feasible solution has no sub-tours of any subset of customers in T. The constraints in (5.5) and (5.6) are the path elimination constraints. They do not allow for any cycle in a feasible solution to consist of more than one depot. The validity of these constraints is discussed in the subsection 5.2.1. Finally, the constraints (5.7)-(5.9) are the integrality restrictions on the $\mathbf{x}$ and $\mathbf{y}$ vectors.

5.2.1 Path elimination constraints

The first version of the path elimination constraints was developed in the context of location routing problems [44]. [10] and [11] use similar path elimination constraints for the location routing and the multiple depot traveling salesmen problems. The version of path elimination constraints presented in this chapter is adapted from 2.4.1. Any path that originates from a depot and visits exactly two customers before terminating at another depot is removed by the constraint in (5.5). The validity of the constraint (5.5) can be easily verified as in [44]. Any other path $d_1, t_1, \cdots, t_p, d_2$, where $d_1, d_2 \in D$, $t_1, \cdots, t_p \in T$ and $p \geq 3$, violates inequality (5.6) with $D' = \{d_1\}$, $S = \{t_2, \cdots, t_{p-1}\}$, $j = t_1$, $k = t_p$ and $i = t_r$ where $2 \leq r \leq p-1$. The proof of validity of the constraint in Eq. (5.6) is discussed as a part of the polyhedral analysis of the polytope of feasible solutions to the GMDTSP in the next section (see proposition 5.5).

We note that our formulation allows for a feasible solution with paths connecting two depots and visiting exactly one customer. We refer to such paths as 2-paths. As the formulation allows for two copies of an edge between a depot and a target, 2-paths can be eliminated and therefore there always exists an optimal solution which does not contain any 2-path. In the following subsection, we prove polyhedral results and derive classes of facet-defining inequalities for the model in (5.2)-(5.9).

5.3 Polyhedral analysis

In this section we analyse the facial structure of the GMDTSP polytope while leveraging the results already known for the MDTSP.

If the number of targets $|T| = n$ and the number of depots $|D| = k$, then the number of x_e variables is $|E| = \binom{n}{2} + nk$ ($\binom{n}{2}$ is the number of edges between the targets and nk is the number of edges between targets and depots). Also the number of y_i variables is $|T| = n$ and hence, the total number of variables used in the problem formulation is $|E| + |T| = \binom{n}{2} + nk + n$. Let P and Q denote the GMDTSP and MDTSP as follows:

$$P := \text{conv}\{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{|E|+|T|} : (\mathbf{x}, \mathbf{y}) \text{ is a feasible GMDTSP solution}\}, \tag{5.10}$$

$$Q := \{(\mathbf{x}, \mathbf{y}) \in P : y_v = 1 \text{ for all } v \in T\}. \tag{5.11}$$

The dimension of the polytope Q was shown to be $\binom{n}{2} + n(k-1)$ in [11]. To relate the polytopes P and Q, we define an intermediate polytope $P(F)$ as follows:

$$P(F) := \{(\mathbf{x}, \mathbf{y}) \in P : y_v = 1 \text{ for all } v \in F\}, \tag{5.12}$$

where $\emptyset \subseteq F \subseteq T$. Observe that $P(\emptyset) = P$ and $P(T) = Q$. Now, we determine the dimension of the polytope $P(F)$. The number of variables in the equation system for $P(F)$ is $|E|+|T| = \binom{n}{2}+nk+n$. The system also includes $|T| = n$ linear independent equations in (5.2) and variable fixing equations given by

$$y_v = 1 \text{ for all } v \in F \cup W$$

where, W is the set of targets that lie in clusters that are singletons (defined in Sec. 5.2). The following lemma gives the dimension of $P(F)$.

Lemma 5.1. *For all* $F \subseteq T$, $\dim(P(F)) = \binom{n}{2} + nk - |F \cup W|$.

Proof. Since the equation system for $P(F)$ has $\binom{n}{2} + nk + n$ variables and $n + |F \cup W|$ linear independent equality constraints, the $\dim(P(F)) \leq \binom{n}{2} + nk - |F \cup W|$. We claim that $P(F)$ contains $\binom{n}{2} + nk - |F \cup W| + 1$ affine independent points. The claim proves $\dim(P(F)) \geq \binom{n}{2} + nk - |F \cup W|$. Hence, the lemma follows. We prove the claim by induction on the cardinality of the set F.

For the base case, we have $F = T$ and $P(T) = Q$ where Q is the the MDTSP polytope. Since $\dim(Q) = \binom{n}{2} + nk - n$ [11], there are $\binom{n}{2} + nk - n + 1$ affine independent points in Q. Assume that the claim holds for a set F_i with $|F_i| = i$ and $i > 0$, and consider a subset of targets F_{i-1} such that $|F_{i-1}| = i - 1$. Let v be any target not in F_{i-1}, and define $F_i := F_{i-1} \cup \{v\}$. The induction hypothesis provides $\binom{n}{2} + nk - |F_i \cup W| + 1$ affine independent points belonging to $P(F_i)$ and hence, to $P(F_{i-1})$ (since $P(F_i) \subseteq P(F_{i-1})$). If $v \in W$, then $|F_{i-1} \cup W| = |F_i \cup W|$ and we are done. Otherwise, $|F_{i-1} \cup W| = |F_i \cup W| - 1$ and we need an additional point on the polytope $P(F_{i-1})$ that is affine independent with the rest of the $\mathcal{L} = \binom{n}{2} + nk - |F_i \cup W| + 1$ points. All these $\mathcal{L}$ points satisfy the equation $y_v = 1$. An additional point that is affine independent with the $\mathcal{L}$ points always exists and is given by any feasible MDTSP solution in the subgraph induced by the set of vertices $(T \cup D) \setminus \{v\}$ because, any feasible MDTSP solution on the set of vertices $(T \cup D) \setminus \{v\}$ satisfies $y_v = 0$. □

Corollary 5.1. $\dim(P) = \binom{n}{2} + nk - |W|$.

Lemma 5.1 indicates that for any given subset $F \subseteq T$ and $v \in F$, either $\dim(P(F \setminus \{v\})) = \dim(P(F))$ (if $v \in W$) or $\dim(P(F \setminus \{v\})) = \dim(P(F)) + 1$ (when $v \notin W$)

$i.e.$, the dimension of the polytope $P(F)$ increases by at most one unit when a target is removed from F. Hence, we can lift any facet-defining valid inequality for $P(F)$ to be facet-defining for $P(F \setminus \{v\})$. In the ensuing proposition, we introduce a result based on the sequential lifting for zero-one programs [65] which we will use to lift facets of Q into facets of P. The proposition generalizes a similar result in [20] used to lift facets of the travelling salesman problem to facets of GTSP.

Proposition 5.1. *Suppose that for any $F \subseteq T$ and $u \in F$,*

$$\sum_{e \in E} \alpha_e x_e + \sum_{v \in T} \beta_v (1 - y_v) \geq \eta$$

is any facet-defining inequality for $P(F)$. Then the lifted inequality

$$\sum_{e \in E} \alpha_e x_e + \sum_{v \in T \setminus \{u\}} \beta_v (1 - y_v) + \bar{\beta}_u (1 - y_u) \geq \eta$$

is valid and facet-defining for $P(F \setminus \{u\})$, where $\bar{\beta}_u$ takes an arbitrary value when $u \in W$ and

$$\bar{\beta}_u = \eta - \min \left\{ \sum_{e \in E} \alpha_e x_e + \sum_{v \in T \setminus \{u\}} \beta_v (1 - y_v) : (\mathbf{x}, \mathbf{y}) \in P(F \setminus \{u\}), y_u = 0 \right\}$$

when $u \notin W$. Note that the statement can be trivially modified to deal with "$\leq$" inequalities.

Proof. The proof follows from the sequential lifting theorem in [65]. □

Proposition 5.1 is used to derive facet-defining inequalities for the GMDTSP polytope P by lifting the facet-defining inequalities for the MDTSP polytope Q in [11]. For a given lifting sequence of the set of targets T, say $\{v_1, \ldots, v_n\}$, the procedure is iteratively applied to derive a facet of $P(\{v_{t+1},\ldots,v_n\})$ from a facet of $P(\{v_t,\ldots,v_n\})$ for $t=1,\ldots,n$. Different lifting sequences produce different facets; hence the name, *sequence dependent* lifting. In the rest of the section, we use the lifting procedure to check if the constraints in (5.2)-(5.9) are facet-defining and derive additional facet-defining inequalities for the GMDTSP polytope.

Proposition 5.2. *The following results hold for the GMDTSP polytope P:*

1. $x_e \geq 0$ *defines a facet for every* $e \in E$ *if* $|T| \geq 4$*,*
2. $x_e \leq 1$ *defines a facet if and only if* $e \in \gamma(W)$ *and* $|T| \geq 3$*,*
3. $x_e \leq 2$ *does not define a facet for any* $e \in (D:T)$*,*
4. $y_i \geq 0$ *does not define a facet for any* $i \in T$*,*
5. $y_i \leq 1$ *defines a facet if and only if* $i \notin W$*, and*
6. $\sum_{i\in C_h} y_i \geq 1$ *does not define a facet for any* $h \in \{1,\ldots,m\}$*.*

Proof. We use the facet-defining results of the MDTSP polytope [11] in conjunction with Proposition 5.1 to prove (1)–(3).

1. Observe that for every $e \in E$, $x_e \geq 0$ defines a facet of the MDTSP polytope Q if $|T| \geq 4$. Now for any lifting sequence, Proposition 5.1 produces $\bar{\beta}_v = 0$ for all $v \in T$ and the result follows.
2. Suppose that $e = (i,j)$. If $i,j \in W$ and $|T| \geq 3$, then the claim follows from the forthcoming Proposition 5.3 by choosing $S = \{i,j\}$. Otherwise if $e = (i,j) \in \gamma(T)$, then $x_e \leq 1$ is dominated by $x_e \leq y_i$ if $i \notin W$ and $x_e \leq y_j$ if $j \notin W$.

3. Let $e = (d, i)$ where $d \in D, i \in T$. $x_e \leq 2$ defines a face of the MDTSP polytope Q. Hence neither of the lifted versions of the inequality *i.e.*, $x_e \leq 2$ (if $i \in W$) or $x_e \leq 2y_i$ (if $i \notin W$) defines a facet of P.

4. The inequality $y_i \geq \frac{1}{2}x_e$ for $e \in \delta(i)$ dominates $y_i \geq 0$. Hence, $y_i \geq 0$ does not define a facet for any $i \in T$.

5. Observe that the valid inequality $y_i \leq 1$ induces a face, $P(\{i\}) = \{(\mathbf{x}, \mathbf{y}) \in P : y_i = 1\}$ of P. From the Lemma 5.1, $\dim(P(\{i\})) = \dim(P) - 1$ if and only if $i \notin W$. Hence, $y_i \leq 1$ is facet-defining for P if and only if $i \notin W$. When $i \in W$, the inequality defines an improper face.

6. The constraint $\sum_{i \in C_h} y_i \geq 1$ can be reduced, using the degree constraints in (5.2), to $\sum_{e \in \delta(C_h)} x_e + 2\sum_{e \in \gamma(C_h)} x_e \geq 2$. When $\gamma(C_h) \neq \emptyset$, the constraint $\sum_{e \in \delta(C_h)} x_e + 2\sum_{e \in \gamma(C_h)} x_e \geq 2$ is dominated by $\sum_{e \in \delta(C_h)} x_e \geq 2$. When $\gamma(C_h) = \emptyset$ (*i.e.*, $|C_h| = 1$), the constraint $\sum_{e \in \delta(C_h)} x_e = 2$ is satisfied by any feasible solution in P and hence in this case, it is an improper face. Therefore, $\sum_{i \in C_h} y_i \geq 1$ does not define a facet for any $h \in \{1, \ldots, m\}$.

□

In the next proposition, we prove that the sub-tour elimination constraints in Eq. (5.4) define facets of P. To do so, we apply the lifting procedure in Proposition 5.1 to the MDTSP sub-tour elimination constraints

$$x(\delta(S)) \geq 2 \text{ for all } S \subseteq T.$$

In the process, we derive alternate versions of the sub-tour elimination constraints in Eq. (5.4) which we will refer to as the GSEC. To begin with, we observe that sub-tour elimination constraints given above define facets of the MDTSP poytope Q when $|T| \geq 3$ (see [11]).

Proposition 5.3. *Let $S \subseteq T$ and $|T| \geq 3$. Then the following GSEC is valid and facet-defining for P:*

$$x(\delta(S)) + \bar{\beta}_i(1 - y_i) \geq 2 \textit{ for } i \in S,$$

where

$$\bar{\beta}_i = \begin{cases} 2 & \textit{if } \mu(S) = 0, \\ 0 & \textit{otherwise}; \end{cases}$$

$\mu(S)$ is defined as $\mu(S) = |\{h : C_h \subseteq S\}|$.

Proof. We first observe that the inequality $x(\delta(S)) \geq 2$ with $S \subseteq T$ and $|T| \geq 3$ defines a facet for the MDTSP polytope. We lift this inequality using the lifting procedure in Proposition 5.1. Let $\{v_1, \ldots, v_n\}$ be any lifting sequence of the set of targets such that $v_n = i$. The lifting coefficients $\bar{\beta}_{v_t}$ are computed iteratively for $t = 1, \ldots, n$. For $t = 1, \ldots, n-1$, it is trivial to see that $\bar{\beta}_{v_t} = 0$. Hence, $x(\delta(S)) \geq 2$ defines a facet of $P(\{v_n\})$. As to $\bar{\beta}_{v_n}$, we compute its value by performing the lifting procedure again and obtain a facet of P. We have

$$\bar{\beta}_{v_n} = 2 - \min\left\{x(\delta(S)) : (\mathbf{x}, \mathbf{y}) \in P, \text{ and } y_{v_n} = 0\right\}.$$

Solving for $\bar{\beta}_{v_n}$ using the above equation, we obtain $\bar{\beta}_{v_n} = 2$ if a feasible GMDTSP solution visiting no target in S exists (*i.e.*, no $C_h \subseteq S$ exists) and $\bar{\beta}_{v_n} = 0$ otherwise. □

In summary, the Proposition 5.3 results in the following facet-defining inequalities of P: suppose $S \subseteq T$ with $|T| \geq 3$. Then,

$$x(\delta(S)) \geq 2 \text{ for } \mu(S) \neq 0 \text{ and} \tag{5.13}$$

$$x(\delta(S)) \geq 2y_i \text{ for } \mu(S) = 0,\, i \in S. \tag{5.14}$$

Note that the inequality $x(\delta(S)) \geq 2y_i$ is valid for any $S \subseteq T$. It is facet-defining for P only when $\mu(S) == 0$. When $\mu(S) \neq 0$ it does not define a facet of P as it is dominated by Eq. (5.13). Using the degree constraints in Eq. (5.2), the above GSEC can rewritten as

$$x(\gamma(S)) \leq \sum_{v \in S} y_v - 1 \text{ for } \mu(S) \neq 0 \text{ and} \tag{5.15}$$

$$x(\gamma(S)) \leq \sum_{v \in S \setminus \{i\}} y_v \text{ for } \mu(S) = 0,\, i \in S. \tag{5.16}$$

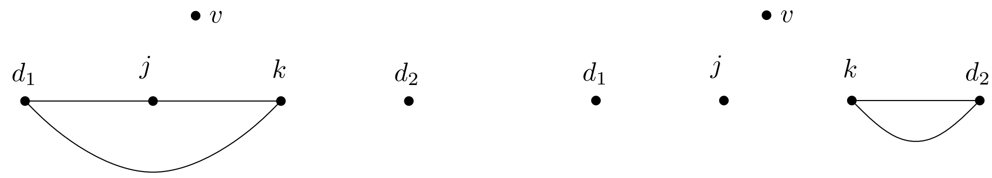
In the forthcoming two propositions, we prove that the path elimination constraints in Eq. (5.5) and (5.6) are facet-defining of P using Proposition 5.1. The corresponding path elimination constraints for the MDTSP polytope Q are as follows: suppose that $j, k \in T$, $D' \subset D$ with $D' \neq \emptyset$, then

$$x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 4 \tag{5.17}$$

$$x(D' : \{j\}) + 2x(\gamma(S \cup \{j,k\})) + x(\{k\} : D \setminus D') \leq 2|S| + 3$$
$$\text{for } S \subseteq T \setminus \{j,k\}, S \neq \emptyset \tag{5.18}$$

We remark that Eq. (5.17) and (5.18) define facets for the MDTSP polytope Q (see [11]).

Proposition 5.4. *Suppose $j, k \in T$ and $D' \subset D$ with $D' \neq \emptyset$. Then the following*



(a) $\bar{\beta}_v = 0$ (b) $\bar{\beta}_j = 2$

Figure 5.1: Tight feasible solutions for proof of Prop. 5.4

path elimination constraint is valid and facet-defining for P:

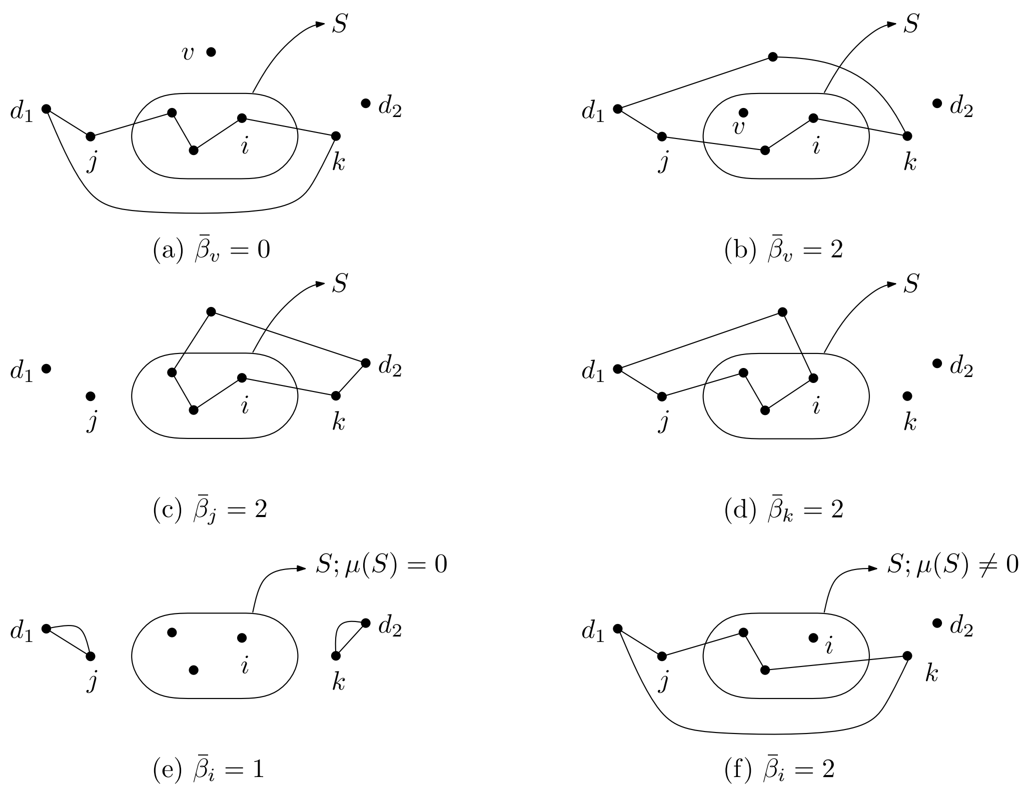
$$x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') + \bar{\beta}_j(1 - y_j) + \bar{\beta}_k(1 - y_k) \leq 4$$

where $\bar{\beta}_j = \bar{\beta}_k = 2$.

Proof. Let $\{v_1, \ldots, v_n\}$ be any lifting sequence of the set of targets such that $v_{n-1} = j$ and $v_n = k$. The lifting coefficients are iteratively computed for $t = 1, 2, \ldots, n$. Coefficients $\bar{\beta}_v$ for $v \in \{v_1, \ldots, v_{n-2}\}$ are easily computed (tight GMDTSP solution is depicted in Fig. 5.1(a), showing that the value of $\bar{\beta}_v$ cannot be increased without producing a violated inequality). Similarly for $t = n-1$ *i.e.*, $v_t = j$, the correctness of the coefficient $\bar{\beta}_j = 2$ can be checked with the help of Fig. 5.1(b). Analogously, we obtain $\bar{\beta}_k = 2$. □

The inequality in Proposition 5.4 can be rewritten as $x(D' : \{j\}) + 3x_{jk} + x(\{k\} : D \setminus D') \leq 2(y_j + y_k)$ which is the path elimination constraint in Eq. (5.5). We have proved that this inequality is valid and defines a facet of P.

Proposition 5.5. *Let $j, k \in T$, $D' \subset D$, $S \subseteq T \setminus \{j, k\}$ and $i \in S$ such that $D' \neq \emptyset$ and $S \neq \emptyset$. Also let $\bar{S} = S \cup \{j, k\}$. Then the following GPEC is valid and*



(a) $\bar{\beta}_v = 0$ (b) $\bar{\beta}_v = 2$

(c) $\bar{\beta}_j = 2$ (d) $\bar{\beta}_k = 2$

(e) $\bar{\beta}_i = 1$ (f) $\bar{\beta}_i = 2$

Figure 5.2: Tight feasible solutions for proof of Prop. 5.5

facet-defining for P:

$$x(D' : \{j\}) + 2x(\gamma(\bar{S})) + x(\{k\} : D \setminus D') + \sum_{v \in T} \bar{\beta}_v (1 - y_v) \leq 2|S| + 3$$

where

$$\bar{\beta}_v = \begin{cases} 0 & \text{if } v \in T \setminus \bar{S}, \\ 2 & \text{if } v \in \bar{S} \setminus \{i\}, \\ 1 & \text{if } v = i \text{ and } \mu(S) = 0, \\ 2 & \text{if } v = i \text{ and } \mu(S) \neq 0; \end{cases}$$

$\mu(S)$ is defined as $\mu(S) = |\{h : C_h \subseteq S\}|$.

Proof. Consider any lifting sequence of the the set of targets $\{v_1, \ldots, v_n\}$ such that each target in the set $S \setminus \{i\}$ follows all the targets in the set $|T \setminus \bar{S}|$ and $v_{n-2} = j$, $v_{n-1} = k$ and $v_n = i$. The coefficients $\bar{\beta}_v = 0$ for $v \in T \setminus \bar{S}$ and $\bar{\beta}_v = 2$ for $v \in S \setminus \{i\}$ are trivial to compute (tight GMDTSP solution is depicted in Fig. 5.2(a) and 5.2(b) respectively, showing that the value of $\bar{\beta}_v$ cannot be increased without producing a violated inequality). The correctness of coefficients $\bar{\beta}_j = 2$ and $\bar{\beta}_k = 2$ can be checked with the help of Fig. 5.2(c) and 5.2(d), respectively.

It remains to compute the value of coefficient $\bar{\beta}_i$. For computing $\bar{\beta}_i$, we have to take into account for the possibility of a GMDTSP solution not visiting any target in the set S. This can happen when $\mu(S) = 0$. In this case, we obtain $\bar{\beta}_i = 1$; see Fig. 5.2(e). Likewise, when $\mu(S) \neq 0$, any GMDTSP solution has to have at least two edges in $\delta(S)$. This leads to $\bar{\beta}_i = 2$; tight GMDTSP solution is shown in Fig. 5.2(f). □

In summary, the Proposition 5.5 results in the following facet-defining inequalities of P: suppose $j, k \in T$, $D' \subset D$, $S \subseteq T \setminus \{j, k\}$, $\bar{S} = S \cup \{j, k\}$ and $i \in S$ such that

$D' \neq \emptyset$ and $S \neq \emptyset$, then

$$x(D' : \{j\}) + 2x(\gamma(\bar{S})) + x(\{k\} : D \setminus D') \leq \sum_{v \in \bar{S}} 2y_v - y_i \text{ for } \mu(S) = 0 \text{ and} \tag{5.19}$$

$$x(D' : \{j\}) + 2x(\gamma(\bar{S})) + x(\{k\} : D \setminus D') \leq \sum_{v \in \bar{S}} 2y_v - 1 \text{ for } \mu(S) \neq 0. \tag{5.20}$$

We note that the above GPEC can be rewritten in cut-set form as

$$x(\delta(\bar{S})) \geq x(D' : \{j\}) + x(\{k\} : D \setminus D') + y_i \text{ for } \mu(S) = 0 \text{ and} \tag{5.21}$$

$$x(\delta(\bar{S})) \geq x(D' : \{j\}) + x(\{k\} : D \setminus D') + 1 \text{ for } \mu(S) \neq 0. \tag{5.22}$$

As we will see in the forthcoming section, the GPEC in the above form are more amicable for developing separation algorithms. Next, we examine the comb inequalities that are valid and facet-defining for the MDTSP polytope. These inequalities were initially introduced for the TSP in [13]. These inequalities were extended and proved to be facet-defining for the MDTSP polytope in [11]. We define a comb inequality using a comb, which is a family $C = (H, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t)$ of $t+1$ subsets of the targets; t is an odd number and $t \geq 3$. The subset H is called the handle and the subsets $\mathcal{T}_1, \ldots, \mathcal{T}_t$ are called teeth. The handle and teeth satisfy the following conditions:

i $H \cap \mathcal{T}_i \neq \emptyset \quad \forall i = 1, \ldots, t$,

ii $\mathcal{T}_i \setminus H \neq \emptyset \quad \forall i = 1, \ldots, t$,

iii $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset \quad 1 \leq i \leq j \leq t$.

The conditions i. and ii. indicate that every tooth T_i intersects the handle H and the condition iii. indicates that no two teeth intersect. We define the size of C as $\sigma(C) := |H| + \sum_{i=1}^{t} |\mathcal{T}_i| - \frac{3t+1}{2}$. Then the comb inequality associated with C is given by

$$x(\gamma(H)) + \sum_{i=1}^{t} x(\gamma(\mathcal{T}_i)) \leq \sigma(C) \tag{5.23}$$

The inequality in Eq. 5.23 is valid and facet-defining for the MDTSP (see [11]). A special case of the comb inequality, called *2-matching* inequality is obtained when $|\mathcal{T}_i| = 2$ for $i = 1, \ldots, t$. In the case of a 2-matching inequality, the size of the comb is $\sigma(C) = |H| + \frac{t+1}{2}$. We apply the lifting procedure in Proposition 5.1 to the inequality in (5.23) and obtain facet-defining inequality for the GMDTSP. The following proposition is adapted from [20]; the proof of the proposition is omitted as it is similar to the proof of the corresponding theorem for GTSP in [20].

Proposition 5.6. *Suppose* $\mu(S) = |\{h : C_h \subseteq S\}|$ *for* $S \subseteq T$ *and let* $C = (H, \mathcal{T}_1, \ldots, \mathcal{T}_t)$ *be a comb. For* $i = 1, \ldots, t$*, let* a_i *be any target in* $\mathcal{T}_i \cap H$ *if* $\mu(\mathcal{T}_i \cap H) = 0$*;* $a_i = 0$ *(a dummy value) otherwise; and let* b_i *be any target in* $\mathcal{T}_i \backslash H$ *if* $\mu(\mathcal{T}_i \backslash H) = 0$*;* $b_i = 0$ *otherwise. Then the following comb inequality is valid and facet-defining for the GMDTSP polytope* P*:*

$$x(\gamma(H)) + \sum_{i=1}^{t} x(\gamma(\mathcal{T}_i)) + \sum_{v \in T} \bar{\beta}_v (1 - y_v) \leq \sigma(C), \tag{5.24}$$

where $\bar{\beta}_v = 0$ *for all* $v \in T \setminus (H \cup \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t)$*,* $\bar{\beta}_v = 1$ *for all* $v \in H \setminus (\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t)$

and for $i = 1, \ldots, t$*:*

$$\begin{aligned}
\bar{\beta}_v &= 2 \quad \textit{for } v \in \mathcal{T}_i \cap H, v \neq a_i; \\
\bar{\beta}_{a_i} &= 1 \quad \textit{if } a_i \neq 0; \\
\bar{\beta}_v &= 1 \quad \textit{for } v \in \mathcal{T}_i \setminus H, v \neq b_i; \\
\bar{\beta}_{b_i} &= 0 \quad \textit{if } b_i \neq 0.
\end{aligned}$$

Proof. See [20]. □

5.3.1 Additional valid inequalities specific to multiple depot problems

In this section, we will examine a special type of comb inequality called the T-comb inequalities. The T-comb inequalities were introduced in [11] and proved to be valid and facet-defining for the MDTSP polytope. These inequalities are specific to problems involving multiple depots and hence, are important for the GMDTSP. A T-comb inequality C is defined by an handle H and teeth $\mathcal{T}_1, \ldots, \mathcal{T}_t$ such that the following conditions are satisfied:

i. $H \cap \mathcal{T}_i \neq \emptyset \quad \forall i = 1, \ldots, t,$

ii. $\mathcal{T}_i \setminus H \neq \emptyset \quad \forall i = 1, \ldots, t,$

iii. $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset \quad 1 \leq i \leq j \leq t,$

iv. $\mathcal{T}_i \cap D \neq \emptyset \quad \forall i = 1, \ldots, t,$

v. $H \subset T,$

vi. $H \setminus \cup_{i=1}^{t} \mathcal{T}_i \neq \emptyset,$

vii. $D \setminus \cup_{i=1}^{t} \mathcal{T}_i \neq \emptyset.$

The difference between the T-comb inequalities and the comb inequalities defined in Eq. (5.23) is that, the number of teeth are allowed to be even ($t \geq 1$) and each teeth must contain a depot. The comb size in this case is given by $\sigma(C) = |H| + \sum_{i=1}^{t} |\mathcal{T}_i| - (t+1)$. In this paper, we will only examine the T-comb inequalities with $|\mathcal{T}_i| = 2$ for every $i \in \{1, \ldots, t\}$; the size of the comb in this case reduces to $\sigma(C) = |H| + t - 1$ and the corresponding T-comb inequality is given by

$$x(\gamma(H)) + \sum_{i=1}^{t} x(\gamma(\mathcal{T}_i)) \leq |H| + t - 1, \tag{5.25}$$

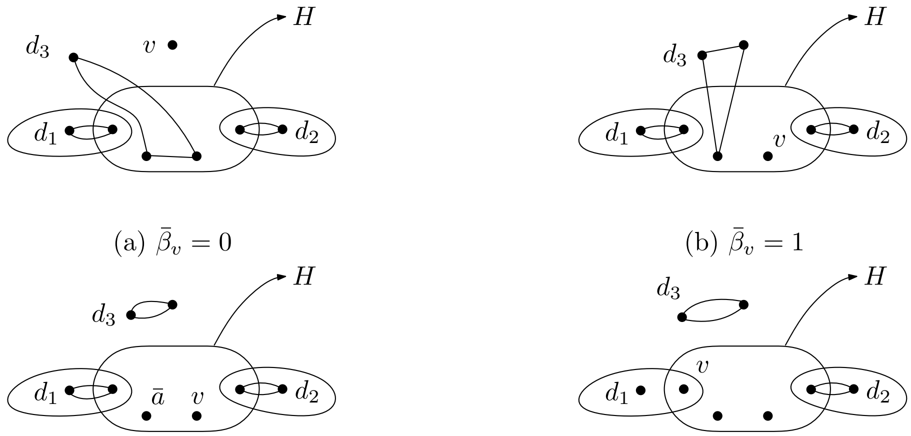
The inequality in Eq. (5.25) is valid and facet-defining for the MDTSP when $t \geq 2$. Again, we apply the lifting procedure in Proposition 5.1 to the inequality in (5.25) and obtain facet-defining inequality for the GMDTSP.

Proposition 5.7. *Let $C = (H, \mathcal{T}_1, \ldots, \mathcal{T}_t)$ be a T-comb with $|\mathcal{T}_i| = 2$ for every $i \in \{1, \ldots, t\}$ and $t \geq 2$. Also suppose $|H \setminus \cup_i \mathcal{T}_i| > 1$ (the proposition can be trivially extended to the case where $|H \setminus \cup_i \mathcal{T}_i| = 1$). Let $\bar{a}$ be any target in $H \setminus \cup_i \mathcal{T}_i$. Then the following T-comb inequality is valid and facet-defining for the GMDTSP polytope P:*

$$x(\gamma(H)) + \sum_{i=1}^{t} x(\gamma(\mathcal{T}_i)) + \sum_{v \in T} \bar{\beta}_v (1 - y_v) \leq |H| + t - 1, \tag{5.26}$$

where $\bar{\beta}_v = 0$ for all $v \in T \setminus (H \cup \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t)$, $\bar{\beta}_v = 1$ for all $v \in H \setminus (\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t \cup \{\bar{a}\})$, $\bar{\beta}_{\bar{a}} = 0$, and $\bar{\beta}_v = 2$ for all $v \in \mathcal{T}_i \cap H, i = 1, \ldots, t$.

Proof. Consider any lifting sequence for the set of targets T in the following order: (i) targets in the set $T \setminus (H \cup \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t)$, (ii) $v \in H \setminus (\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_t \cup \{\bar{a}\})$, (iii) $\bar{a}$, and (iv) $v \in \mathcal{T}_i \cap H, i = 1, \ldots, t$. The lifting coefficients $\bar{\beta}_v = 0$ and $\bar{\beta}_v = 1$ for the sets in (i) and (ii) respectively, are trivial to compute (tight feasible GMDTSP solutions are depicted in Fig. 5.3(a) and 5.3(b), respectively). Similarly, tight feasible GMDTSP



(a) $\bar{\beta}_v = 0$ (b) $\bar{\beta}_v = 1$

(c) $\bar{\beta}_{\bar{a}} = 0$ when $y_v = 0$ (d) $\bar{\beta}_v = 2$

Figure 5.3: Tight feasible solutions for proof of Prop. 5.7

solutions for the cases where $\bar{\beta}_{\bar{a}} = 0$ and $\bar{\beta}_v = 2$ (cases (iii) and (iv)) are shown in Fig. 5.3(c) and 5.3(d), respectively. $\square$

In the above proposition, for the case when $|H \setminus \cup_i \mathcal{T}_i| = 1$, the facet-defining inequality is given by

$$x(\gamma(H)) + \sum_{i=1}^{t} x(\gamma(\mathcal{T}_i)) \leq \sum_{i=1}^{t} \sum_{v \in H \cap \mathcal{T}_i} 2y_v. \tag{5.27}$$

5.4 Separation algorithms

In this section, we discuss the algorithms that are used to find violated families of all the valid inequalities introduced in Sec. 5.3. We denote by $G^* = (V^*, E^*)$ the *support graph* associated with a given fractional solution $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{R}^{|E| \cup |T|}$ *i.e.,* G^* is a capacitated undirected graph with vertex set $V^* := \{i \in T : y_i^* > 0\} \cup D$ and $E^* := \{e \in E : x_e^* > 0\}$ with edge capacities x_e^* for each edge $e \in E^*$.

5.4.1 Separation of generalized sub-tour elimination constraints

We first develop a separation algorithm for constraints in Eq. (5.14): $x(\delta(S)) \geq 2y_i$ for $\mu(S) = 0, i \in S$ and $S \subseteq T$. Given a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the most violated constraint of the form (5.14) can be obtained by computing a minimum capacity cut $(S, V^* \setminus S)$ with $i \in S$ and $D \subseteq V^* \setminus S$ on the graph G^*. The minimum capacity cut can be obtained by computing a maximum flow from i to t, where t is an additional vertex connected with each depot in the set D through an edge having very large capacity. The algorithm is repeated for every target $i \in T \cap V^*$ and the target set S obtained during each run of the algorithm defines a violated inequality if the capacity of the cut is strictly less than $2y_i^*$. This procedure can be implemented in $O(|T|^4)$ time.

Now we consider the constraint in Eq. (5.13): $x(\delta(S)) \geq 2$ for $\mu(S) \neq 0$ and $S \subseteq T$. Given a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the most violated inequality (5.13) in this case is obtained by computing a minimum capacity cut $(S, V^* \setminus S)$ with a cluster $C_h \subseteq S$ and $D \subseteq V^* \setminus S$ on the graph G^*. This is in turn achieved by computing a maximum $s - t$ flow on G^*, where s and t are additional vertices connected with each $j \in C_h$ and each $d \in D$ respectively through an edge having very large capacity. The algorithm is repeated for every cluster C_h and the set S obtained on each run of the algorithm defines a violated inequality if the capacity of the cut is strictly less than 2. The time complexity of this procedure is $O(m|T|^3)$, where m is the number of clusters.

We remark that the violated inequality of the form (5.14) using the above algorithm, is not necessarily facet-defining as the set S computed using the algorithm might have $\mu(S) \neq 0$. When this happens, we reject the inequality in favour of its dominating and facet-defining inequality in Eq. (5.13).

5.4.2 Separation of path elimination constraints

We first discuss the procedure to separate violated constraints in Eq. (5.5). Consider every pair of targets $j, k \in V^* \cap T$. We rewrite the constraint in (5.5) as $x(D' : \{j\}) + x(\{k\} : D \setminus D') \leq 2(y_k + y_j) - 3x_{jk}$. Given j, k and a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the RHS of the above inequality is a constant and is equal to $2(y_k^* + y_j^*) - 3x_{jk}^*$. We observe that the LHS of the inequality is maximized when $D' = \{d \in D : x_{jd}^* \geq x_{kd}^*\}$. Furthermore, when $D' = \emptyset$ or $D' = D$, no path constraint in Eq. (5.5) is violated for the given pair of vertices. With $D' = \{d \in D : x_{jd}^* \geq x_{kd}^*\}$, if $x^*(D' : \{j\}) + x^*(\{k\} : D \setminus D')$ is strictly greater than $2(y_k^* + y_j^*) - 3x_{jk}^*$, the path constraint in Eq. (5.5) is violated for the pair of vertices j, k and the subset of depots D'. This procedure can be implemented in $O(|T|^2)$.

For constraints in Eq. (5.19) and (5.20), we present two separation algorithms that are very similar to the algorithms presented in Sec. 5.4.1. We will use the equivalent constraints in Eq. (5.21) and (5.22) to develop the algorithms. We first consider the path elimination constraint in Eq. (5.22). Given j, k and a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, we first compute D' to maximize $x^*(D' : \{j\}) + x^*(\{k\} : D \setminus D') := \mathcal{L}$. Now, the most violated constraint of the form (5.22) can be obtained by computing a minimum capacity cut $(\bar{S}, V^* \setminus \bar{S})$ with $j, k \in \bar{S}$, a cluster $C_h \subseteq \bar{S} \setminus \{j, k\}$ and $D \subseteq V^* \setminus \bar{S}$. This algorithm is repeated for every target $j, k \in T$ and cluster C_h such that $j, k \notin C_h$ and the target set $S = \bar{S} \setminus \{j, k\}$ obtained during each run of the algorithm defines a violated inequality if the capacity of the cut is strictly less than $\mathcal{L} + 1$. The time complexity of this algorithm is $O(m|T|^4)$. Similarly, the most violated constraint of the form (5.21) can be obtained by computing a minimum capacity cut $(\bar{S}, V^* \setminus \bar{S})$, with $i, j, k \in \bar{S}$ and $D \subseteq V^* \setminus \bar{S}$ on the graph G^*. This algorithm is repeated for very triplet of targets in V^* and the set $S = \bar{S} \setminus \{j, k\}$

defines a violated inequality if the capacity of the cut is strictly less than $\mathcal{L} + y_i^*$. The time complexity of the algorithm is $O(|T|^5)$.

Similar to the separation of the sub-tour elimination constraints, we remark that the violated inequality of the form (5.21), computed using the above algorithm is not necessarily facet-defining as the set S might have $\mu(S) \neq 0$. When this happens, we reject the inequality in favour of its dominating and facet-defining inequality in Eq. (5.22).

5.4.3 Separation of comb inequalities

For the comb-inequalities in Eq. (5.24), we use the separation procedures discussed in [23]. We first consider the special case of the comb inequalities with $|\mathcal{T}_i| = 2$ for $i = 1, \ldots, t$ *i.e.,* the 2-matching inequalities. Using a construction similar to the one proposed in [66] for the b-matching problem, the separation problem for the 2-matching inequalities can be transformed into a minimum capacity off cut problem; hence this separation problem is exactly solvable in polynomial time. But this procedure is computationally intensive, and so we use the following heuristic proposed by [29]. Given a fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$, the heuristic considers a graph $\bar{G} = (\bar{V}, \bar{E})$ where $\bar{V} = V^* \cap T$ and $\bar{E} = \{e : 0 < x_e^* < 1\}$. Then, we consider each connected component H of $\bar{G}$ as a handle of a possibly violated 2-matching inequality whose two-vertex teeth correspond to edges $e \in \delta(H)$ with $x_e^* = 1$. We reject the inequality if the number of teeth is even. The time complexity of this algorithm is $O(|\bar{V}| + |\bar{E}|)$. As for the comb inequalities, we apply the same procedure after shrinking each cluster into a single supernode.

5.4.4 Separation of T-comb inequalities

We present a separation heuristic similar to the one used in [11] to identify violated T-comb inequalities of the form Eq. (5.26) and (5.27). We first build a set of teeth, each containing a distinct depot according to the following procedure: a tooth $\mathcal{T}_i$ is built by starting with a set containing a depot $d \in D$; a target $v \in T$ is added to $\mathcal{T}_i$ such that $x(\delta(\mathcal{T}_i))$ is a minimum. Then, for every subset of this set of teeth such that: (i) they are pairwise disjoint, (ii) belong to the same connected component of the support graph $G^* = (V^*, E^*)$, and (iii) do not together contain all the targets of that connected component, an appropriate handle H is built as follows: assume H is the set of all the targets in the connected component and remove the targets in $H \setminus (\mathcal{T}_i \cup \cdots \cup \mathcal{T}_t)$ sequentially. Every time a target is removed, the T-comb inequality of the appropriate form is checked for violation. The time complexity of this algorithm is $O(|T|)$.

5.5 Branch-and-cut algorithm

In this section, we describe important implementation details of the branch-and-cut algorithm for the GMDTSP. The algorithm is implemented within a CPLEX 12.4 framework using the CPLEX callback functions [34]. The callback functions in CPLEX enable the user to completely customize the branch-and-cut algorithm embedded into CPLEX, including the choice of node to explore in the enumeration tree, the choice of branching variable, the separation and the addition of user-defined cutting planes and the application of heuristic methods.

The lower bound at the root node of the enumeration tree is computed by solving the LP relaxation of the formulation in Sec. 5.2 that is further strengthened using the cutting planes described in Sec. 5.3. The initial linear program consisted of all constraints in (5.1)-(5.9), except (5.4), (5.5) and (5.6). For a given LP solution, we identify violated inequalities using the separation procedures detailed in Sec. 5.4 in the following order: (i) sub-tour elimination constraints in Eq. (5.13), (ii) sub-tour elimination constraints in Eq. (5.14) (iii) path elimination constraints in Eq.

(5.5), (5.19) and, (5.20), (iv) generalized comb constraints in Eq. (5.24), and (v) T-comb constraints in Eq. (5.26) and (5.27). This order of adding the constraints to the formulation was chosen after performing extensive computational experiments. Furthermore, we disabled the separation of all the cuts embedded into the CPLEX framework because enabling these cuts increased the average computation time for the instances. Once the new cuts generated using these separation procedures were added to the linear program, the tighter linear program was resolved. This procedure was iterated until either of the following conditions was satisfied: (i) no violated constraints could be generated by the separation procedures, (ii) the current lower bound of the enumeration tree was greater or equal to the current upper bound. If no constraints are generated in the separation phase, we create subproblems by branching on a fractional variable. First, we select a fractional y_i variable, based on the *strong branching* rule [1]. If all these variables are integers, then we select a fractional x_e variable using the same rule. As for the node-selection rule, we used the best-first policy for all our computations,*i.e.*, select the subproblem with the lowest objective value.

5.5.1 Preprocessing

In this section, we detail a preprocessing algorithm that enables the reduction of size of the GMDTSP instances whose edge costs satisfy the triangle inequality *i.e.,* for distinct $i, j, k \in T$, $c_{ij} + c_{jk} \geq c_{ik}$. A similar algorithm is presented in [48, 9] for the asymmetric generalized traveling salesman problem and generalized vehicle routing problem respectively. In a GMDTSP instance where the edge costs satisfy the triangle inequality, the optimal solution would visit exactly one target in each cluster. We utilize this structure of the optimal solution and reduce the size of a given GMDTSP instance, if possible. To that end, we define a target $i \in T$ to be *dominated* if there exits a target $j \in C_{h(i)}$, $j \neq i$ such that

1. $c_{pi} + c_{iq} \geq c_{pj} + c_{jq}$ for any $p, q \in T \setminus C_{h(i)}$,
2. $c_{di} \geq c_{dj}$ for all $d \in D$, and
3. $c_{di} + c_{ip} \geq c_{dj} + c_{jp}$ for any $d \in D, p \in T \setminus C_{h(i)}$.

Proposition 5.8. *If a dominated target is removed from a GMDTSP instance satisfying triangle inequality, then the optimal cost to the instance does not change.*

Proof. Let $i \in T$ be a dominated vertex. If the target i is not visited in the optimal solution, then its removal does not change the optimal cost. So, assume that $i \in T$ is visited by the optimal solution. Since the edge costs of the instance satisfy the triangle inequality, exactly one target in each cluster is visited by the optimal solution. We now claim that it is possible to exchange the target i with a target $j \in C_{h(i)}$ without increasing the cost of the optimal solution. This follows from the definition of a dominated target. □

The preprocessing checks if a target is dominated and removes the target if it is found so. Then the other targets are checked for dominance relative to the reduced instance. The time complexity of the algorithm is $O(|T|^5)$.

5.5.2 LP rounding heuristic

We discuss an *LP-rounding* heuristic that aides to generate feasible solutions at the root node and to speed up the convergence of the branch-and-cut algorithm. The heuristic constructs a feasible GMDTSP solution from a given fractional LP solution. It is used only at the root node of the enumeration tree. The heuristic is based on a transformation method in [63]. We are given $\mathbf{y}^*$, the vector of fractional y_i values (denoted by y_i^f) for each target i. The algorithm proceeds as follows: for each cluster

C_k and every target $i \in C_k$, the heuristic sets the value of y_i to 0 or 1 according to the condition $y_i^f \geq 0.5$ or $y_i^f < 0.5$ respectively. If every target $i \in C_k$ has $y_i^f < 0.5$, then we set the value of $y_j = 1$ where $j = \text{argmax}\{y_i^f : i \in C_k\}$. Once we have assigned the y_i value for each target i, we define the set $\Pi := \{i \in T : y_i = 1\}$. We then solve a MDTSP on the set of vertices $\Pi \cup D$. A heuristic based on the transformation method in [63] and LKH heuristic [31] is used to solve the MDTSP.

5.6 Computational results

In this section, we discuss the computational results of the branch-and-cut algorithm. The algorithm was implemented in C++ (gcc version 4.6.3), using the elements of Standard Template Library (STL) in the CPLEX 12.4 framework. As mentioned in Sec. 5.5, the internal CPLEX cut generation was disabled, and CPLEX was used only to mange the enumeration tree. All the simulations were performed on a Dell Precision T5500 workstation (Intel Xeon E5360 processor @2.53 GHz, 12 GB RAM). The computation times reported are expressed in seconds, and we imposed a time limit of 7200 seconds for each run of the algorithm. The performance of the algorithm was tested on a total of 116 instances, all of which were generated using the generalized traveling salesman problem library [23, 30].

5.6.1 Problem instances

All the computational experiments were conducted on a class of 116 test instances generated from 29 GTSP instances. The GTSP instances are taken directly from the GTSP Instances Library [30]. For each of the 29 instances, GMDTSP instances with $|D| \in \{2, 3, 4, 5\}$ were generated by assuming the first $|D|$ targets in a GTSP instance to be the set of depots; these depots were then removed from the target clusters. The number of targets in the instances varied from 14 to 105, and the maximum number of target clusters was 21. Hence we had 4 GMDTSP instances for each of the 29 GTSP instances totalling to 116 test instances. We also note that for 64/116 instances, the edge costs do not satisfy the triangle inequality and for the remaining 52 instances, the edge costs satisfy the triangle inequality. The name of the generated instances are the same but for a small modification to spell out the number of depots in the instances. The naming conforms to the format `GTSPinstancename-D`, where `GTSPinstancename` corresponds to the GTSP instance name from the library (the first and the last integer in the name corresponds to the number of clusters and the number of targets in the GTSP instance respectively) and `D` corresponds the number of depots in the instance.

The results are tabulated in Tables 5.1 and 5.2. For more detailed computational results, the readers are refered to [77]. The following nomenclature is used in the Table 5.1

name: problem instance name (format: `GTSPinstancename-D`);
%LB: percentage LB/opt, where objective value of the LP relaxation computed at the root node of the enumeration tree;
%UB: percentage UB/opt, where cost of the best feasible solution generated by the LP-rounding heuristic generated at the root node of the enumeration tree;
sec1: total number of constraints (5.13) generated;
sec2: total number of constraints (5.14) generated;
4pec: total number of constraints (5.5) generated;
pec: total number of constraints (5.19) and (5.20) generated;
comb: total number of constraints (5.24), (5.26), and (5.27) generated;
nodes: total number of nodes examined in the enumeration tree.

The Table 5.2 gives the computational time for each separation routine and the overall the branch-and-cut algorithm. The nomenclature used in Table 5.2 are as follows:

name: problem instance name (format: `GTSPinstancename-D`);

total-t: CPU time, in seconds, for the overall execution of the branch-and-cut algorithm;

sep-t: overall CPU time, in seconds, spent for separation;

sec-t: CPU time, in seconds, spent for the separation of constraints (5.13) and (5.14);

4pec-t: CPU time, in seconds, spent for the separation of constraints (5.5);

pec-t: CPU time, in seconds, spent for the separation of constraints (5.19) and (5.20);

comb-t: CPU time, in seconds, spent for the separation of constraints (5.24), (5.26), and (5.27);

%pec: percentage of separation time spent for the separation of path elimination constraints (5.19) and (5.20).

Table 5.1: Branch-and-cut statistics.

name	opt	LB	%LB	UB	%UB	sec1	sec2	4pec	pec	comb	nodes
3burma14-2	1939	1939.00	100.00	1939	100.00	51	8	0	2	0	0
3burma14-3	1664	1664.00	100.00	1664	100.00	11	15	0	2	0	0
3burma14-4	1296	1296.00	100.00	1296	100.00	8	14	0	0	0	0
3burma14-5	562	562.00	100.00	562	100.00	1	20	0	0	0	0
4br17-2	31	31.00	100.00	54	174.19	7	4	0	0	1	3
4br17-3	31	31.00	100.00	31	100.00	7	7	0	0	0	0
4br17-4	19	19.00	100.00	19	100.00	5	14	0	0	0	0
4br17-5	19	19.00	100.00	19	100.00	5	20	0	4	0	0
4gr17-2	958	846.33	88.34	965	100.73	22	187	8	335	0	97
4gr17-3	738	722.88	97.95	794	107.59	3	43	1	53	4	6
4gr17-4	611	611.00	100.00	611	100.00	2	14	0	3	0	0
4gr17-5	513	513.00	100.00	513	100.00	1	25	0	0	0	0
4ulysses16-2	4695	4695.00	100.00	4695	100.00	36	18	0	0	0	0

Table 5.1 – continued from previous page

name	opt	LB	%LB	UB	%UB	sec1	sec2	4pec	pec	comb	nodes
4ulysses16-3	4695	4695.00	100.00	4695	100.00	53	20	0	0	0	0
4ulysses16-4	4695	4695.00	100.00	4695	100.00	50	27	0	0	0	0
4ulysses16-5	3914	3884.00	99.23	4188	107.00	22	27	0	7	0	3
5gr21-2	1679	1531.67	91.22	1985	118.23	419	367	12	2158	0	449
5gr21-3	1024	1024.00	100.00	1024	100.00	6	32	0	2	0	0
5gr21-4	953	953.00	100.00	953	100.00	9	20	0	1	0	0
5gr21-5	780	780.00	100.00	780	100.00	4	9	0	2	0	0
5gr24-2	377	340.53	90.33	828	219.63	25	169	0	366	0	13
5gr24-3	377	318.00	84.35	569	150.93	37	181	0	524	32	42
5gr24-4	371	325.17	87.65	753	202.96	39	157	8	303	6	26
5gr24-5	362	308.17	85.13	739	204.14	12	99	7	222	0	87
5ulysses22-2	5199	5199.00	100.00	5199	100.00	70	71	2	126	1	0
5ulysses22-3	5311	5310.50	99.99	5442	102.47	45	82	0	1	0	3
5ulysses22-4	5021	5021.00	100.00	5021	100.00	45	39	0	0	0	0
5ulysses22-5	3913	3913.00	100.00	3913	100.00	37	27	0	1	0	0
6bayg29-2	711	624.50	87.83	905	127.29	82	312	0	1526	0	148
6bayg29-3	684	582.50	85.16	841	122.95	70	809	3	3489	28	301
6bayg29-4	583	527.50	90.48	811	139.11	25	91	0	171	7	24
6bayg29-5	565	520.79	92.17	1888	334.16	40	103	0	360	6	21
6bays29-2	849	761.46	89.69	1194	140.64	123	178	0	1466	0	296
6bays29-3	830	777.68	93.70	1092	131.57	80	145	1	959	17	48
6bays29-4	691	650.60	94.15	847	122.58	30	92	3	238	20	6
6bays29-5	622	591.55	95.10	1052	169.13	30	99	1	258	3	10
6fri26-2	480	471.50	98.23	541	112.71	54	184	1	519	0	15
6fri26-3	486	466.00	95.88	510	104.94	167	166	0	1923	3	388
6fri26-4	440	414.57	94.22	446	101.36	92	128	0	355	9	38
6fri26-5	436	411.56	94.39	473	108.49	66	91	2	520	2	41
9dantzig42-2	413	413.00	100.00	413	100.00	114	300	0	0	0	0
9dantzig42-3	351	351.00	100.00	358	101.99	82	328	0	10	1	3
9dantzig42-4	350	345.75	98.79	396	113.14	81	272	1	442	33	6
9dantzig42-5	348	344.29	98.93	348	100.00	82	203	2	346	45	12
10att48-2	4924	4284.05	87.00	5510	111.90	456	945	0	7563	0	268
10att48-3	4913	4539.33	92.39	6054	123.22	177	880	8	10115	154	1406
10att48-4	4428	3980.11	89.89	5685	128.39	197	738	2	8555	138	879
10att48-5	4204	3897.97	92.72	5515	131.18	87	690	9	12826	1077	594
10gr48-2	1708	1707.00	99.94	1708	100.00	88	186	1	259	0	2

Table 5.1 – continued from previous page

name	opt	LB	%LB	UB	%UB	sec1	sec2	4pec	pec	comb	nodes
10gr48-3	1638	1628.14	99.40	2345	143.16	74	220	4	1011	0	14
10gr48-4	1645	1629.23	99.04	2197	133.56	86	185	0	958	1	33
10gr48-5	1638	1471.48	89.83	2243	136.94	108	405	5	2163	30	179
10hk48-2	6401	6209.83	97.01	6753	105.50	357	418	7	3018	0	82
10hk48-3	5872	5567.49	94.81	6211	105.77	234	364	1	2549	0	75
10hk48-4	5642	5044.00	89.40	6359	112.71	269	474	1	2370	3	69
10hk48-5	5641	5145.17	91.21	6702	118.81	282	399	0	3455	14	27
11berlin52-2	3500	3425.00	97.86	4010	114.57	121	288	0	1	1	17
11berlin52-3	3500	3376.17	96.46	3963	113.23	142	311	1	753	66	20
11berlin52-4	3500	3280.00	93.71	3699	105.69	88	241	1	426	3	25
11berlin52-5	3500	3273.92	93.54	4169	119.11	131	160	0	599	26	26
11eil51-2	175	174.50	99.71	175	100.00	148	522	2	1071	0	3
11eil51-3	174	168.83	97.03	174	100.00	138	269	3	1160	54	11
11eil51-4	175	165.24	94.42	183	104.57	175	273	11	1837	18	74
11eil51-5	170	166.44	97.91	170	100.00	71	214	2	479	6	8
12brazil58-2	14939	14939.00	100.00	14939	100.00	141	278	3	834	0	0
12brazil58-3	14930	14840.50	99.40	15240	102.08	140	298	1	967	57	18
12brazil58-4	13082	12680.46	96.93	16148	123.44	147	397	1	1447	126	40
12brazil58-5	12613	11958.93	94.81	15546	123.25	153	1049	1	583	50	98
14st70-2	304	288.01	94.74	307	100.99	392	576	2	3147	3	81
14st70-3	301	292.57	97.20	312	103.65	313	600	6	2846	12	17
14st70-4	298	287.25	96.39	298	100.00	182	372	4	1404	4	19
14st70-5	298	282.28	94.73	325	109.06	313	670	9	3883	5	163
16eil76-2	198	198.00	100.00	198	100.00	223	436	0	945	0	0
16eil76-3	197	197.00	100.00	197	100.00	174	258	3	727	6	0
16eil76-4	197	197.00	100.00	197	100.00	147	360	4	941	20	0
16eil76-5	188	180.42	95.97	196	104.26	233	386	5	1132	25	27
20gr96-2†	29966	28357.03	94.63	30821	102.85	823	1220	1	3540	0	62
20gr96-3†	29621	29263.93	98.79	30768	103.87	876	1326	2	3382	529	50
20gr96-4	28705	27650.63	96.33	30121	104.93	866	1754	6	4268	7	144
20gr96-5	28598	27768.50	97.10	29976	104.82	676	1269	1	2087	1	52
20kroA100-2	9630	9265.75	96.22	9769	101.44	746	1080	5	3481	0	66
20kroA100-3	9334	8935.25	95.73	9535	102.15	532	915	0	2801	0	92
20kroA100-4	8897	8539.03	95.98	10243	115.13	935	1241	2	4490	0	126
20kroA100-5	8827	8477.39	96.04	9020	102.19	520	1028	4	2480	0	47
20kroB100-2	9800	9492.00	96.86	10382	105.94	510	955	4	3025	0	30

Table 5.1 – continued from previous page

name	opt	LB	%LB	UB	%UB	sec1	sec2	4pec	pec	comb	nodes
20kroB100-3†	10218	9197.41	90.01	10300	100.80	903	1120	1	5373	0	130
20kroB100-4	9564	9293.31	97.17	9637	100.76	361	714	0	2323	0	20
20kroB100-5	9226	8525.71	92.41	11708	126.90	739	1058	10	7225	0	119
20kroC100-2†	10089	9548.13	94.64	10089	100.00	420	974	0	1551	0	3
20kroC100-3	9244	9130.82	98.78	9346	101.10	494	1006	0	1940	1	8
20kroC100-4	9292	9061.20	97.52	9342	100.54	307	707	2	1132	3	10
20kroC100-5	9252	8991.89	97.19	10437	112.81	380	956	3	2181	0	19
20kroD100-2†	9353	8497.63	90.85	9381	100.30	886	1525	4	3221	6	65
20kroD100-3	8813	8130.12	92.25	11404	129.40	1284	1664	5	11642	24	212
20kroD100-4	8772	8283.74	94.43	8823	100.58	577	1067	11	3230	3	67
20kroD100-5	8677	8233.85	94.89	9247	106.57	478	732	1	3277	0	45
20kroE100-2	9526	9290.65	97.53	10207	107.15	599	1098	7	4461	0	45
20kroE100-3	9262	9153.61	98.83	9854	106.39	612	1048	7	3974	19	26
20kroE100-4	9262	9147.56	98.76	11046	119.26	513	1032	3	3410	4	21
20kroE100-5	9081	8900.07	98.01	9707	106.89	391	925	3	2802	0	32
20rat99-2	505	504.33	99.87	521	103.17	507	951	0	0	0	7
20rat99-3	504	498.23	98.85	543	107.74	528	977	4	1582	1	20
20rat99-4	501	490.67	97.94	515	102.79	958	1259	5	10214	0	2383
20rat99-5	487	477.67	98.08	506	103.90	688	967	4	4320	0	376
20rd100-2†	3459	3380.39	97.73	3714	107.37	742	1406	0	4119	0	42
20rd100-3	3383	3218.89	95.15	3384	100.03	657	1456	2	4238	1	55
20rd100-4	3298	3167.38	96.04	3398	103.03	530	889	2	2651	0	29
20rd100-5	3234	3109.99	96.17	3327	102.88	559	1056	6	4114	1	64
21eil101-2	248	245.41	98.96	255	102.82	387	812	0	1476	0	20
21eil101-3	248	243.04	98.00	267	107.66	570	982	4	2371	6	37
21eil101-4	233	230.2759	98.83	251	107.73	432	629	3	2586	0	15
21eil101-5	232	226.33	97.56	257	110.78	275	527	0	1483	2	16
21lin105-2	8358	8316.43	99.50	8726	104.40	652	1122	0	0	0	16
21lin105-3†	8304	8164.21	98.32	8619	103.79	870	1298	3	25572	22	7103
21lin105-4	7827	7695.17	98.32	8365	106.87	619	941	2	888	12	89
21lin105-5†	8052	7568.64	94.00	8110	100.72	745	1166	1	2419	6	145

†optimality was not verified within a time-limit of 7200 seconds.

Table 5.2: Algorithm computation times.

name	total-t	sep-t	sec-t	4pec-t	pec-t	comb-t	%pec
3burma14-2	0.07	0.00	0.00	0.00	0.00	0.00	3.13
3burma14-3	0.02	0.00	0.00	0.00	0.00	0.00	2.68
3burma14-4	0.02	0.00	0.00	0.00	0.00	0.00	1.97
3burma14-5	0.02	0.00	0.00	0.00	0.00	0.00	3.50
4br17-2	0.03	0.00	0.00	0.00	0.00	0.00	1.14
4br17-3	0.01	0.00	0.00	0.00	0.00	0.00	0.00
4br17-4	0.02	0.00	0.00	0.00	0.00	0.00	0.00
4br17-5	0.04	0.01	0.00	0.00	0.00	0.00	68.52
4gr17-2	1.16	0.33	0.10	0.00	0.22	0.01	65.71
4gr17-3	0.23	0.05	0.01	0.00	0.04	0.00	74.03
4gr17-4	0.02	0.00	0.00	0.00	0.00	0.00	0.00
4gr17-5	0.01	0.00	0.00	0.00	0.00	0.00	0.00
4ulysses16-2	0.05	0.00	0.00	0.00	0.00	0.00	1.71
4ulysses16-3	0.05	0.00	0.00	0.00	0.00	0.00	2.04
4ulysses16-4	0.08	0.00	0.00	0.00	0.00	0.00	1.93
4ulysses16-5	0.13	0.02	0.01	0.00	0.02	0.00	72.63
5gr21-2	12.89	3.63	1.00	0.00	2.54	0.09	69.98
5gr21-3	0.04	0.00	0.00	0.00	0.00	0.00	2.28
5gr21-4	0.02	0.00	0.00	0.00	0.00	0.00	2.86
5gr21-5	0.07	0.00	0.00	0.00	0.00	0.00	2.81
5gr24-2	1.81	0.45	0.07	0.00	0.38	0.00	84.82
5gr24-3	3.51	0.92	0.18	0.00	0.73	0.01	79.17
5gr24-4	2.89	0.76	0.11	0.00	0.64	0.01	83.80
5gr24-5	1.63	0.38	0.12	0.00	0.25	0.01	65.26
5ulysses22-2	0.77	0.18	0.04	0.00	0.13	0.00	74.26
5ulysses22-3	0.43	0.03	0.03	0.00	0.00	0.00	0.64
5ulysses22-4	0.18	0.02	0.02	0.00	0.00	0.00	0.75
5ulysses22-5	0.06	0.01	0.01	0.00	0.00	0.00	1.82
6bayg29-2	18.69	4.97	0.73	0.00	4.17	0.08	83.79
6bayg29-3	20.50	5.66	1.31	0.00	4.19	0.15	74.10
6bayg29-4	1.26	0.31	0.06	0.00	0.24	0.01	77.32
6bayg29-5	1.19	0.27	0.08	0.00	0.18	0.01	68.11
6bays29-2	21.40	6.19	0.96	0.00	5.14	0.08	83.16
6bays29-3	10.60	2.78	0.33	0.00	2.43	0.02	87.50
6bays29-4	1.22	0.30	0.05	0.00	0.24	0.01	80.74

Table 5.2 – continued from previous page

name	total-t	sep-t	sec-t	4pec-t	pec-t	comb-t	%pec
6bays29-5	0.97	0.22	0.04	0.00	0.18	0.00	79.98
6fri26-2	5.55	1.34	0.12	0.00	1.22	0.01	90.53
6fri26-3	18.32	5.55	1.11	0.00	4.31	0.13	77.68
6fri26-4	3.75	0.92	0.12	0.00	0.78	0.01	85.23
6fri26-5	3.26	0.83	0.12	0.00	0.70	0.01	84.67
9dantzig42-2	1.07	0.28	0.27	0.00	0.00	0.01	0.38
9dantzig42-3	1.26	0.34	0.16	0.00	0.18	0.00	51.77
9dantzig42-4	5.15	1.29	0.22	0.00	1.05	0.01	81.81
9dantzig42-5	7.97	1.93	0.20	0.00	1.71	0.01	88.71
10att48-2	280.75	80.02	6.73	0.00	72.88	0.41	91.08
10att48-3	243.27	71.62	9.29	0.00	60.66	1.67	84.70
10att48-4	203.20	59.39	7.56	0.00	50.63	1.19	85.26
10att48-5	130.36	38.93	5.95	0.00	31.74	1.23	81.55
10gr48-2	9.25	2.26	0.21	0.00	2.04	0.01	90.50
10gr48-3	31.81	7.87	0.54	0.00	7.30	0.03	92.72
10gr48-4	39.36	9.62	0.60	0.00	8.96	0.06	93.10
10gr48-5	43.79	11.76	1.39	0.00	10.17	0.20	86.48
10hk48-2	273.81	69.58	3.29	0.00	66.15	0.14	95.07
10hk48-3	170.99	43.05	1.76	0.00	41.19	0.10	95.66
10hk48-4	35.98	9.64	1.04	0.00	8.51	0.09	88.28
10hk48-5	92.75	24.49	1.57	0.00	22.84	0.08	93.27
11berlin52-2	2.28	1.06	1.03	0.00	0.00	0.02	0.37
11berlin52-3	67.95	16.48	0.95	0.00	15.48	0.05	93.91
11berlin52-4	27.96	7.19	0.44	0.00	6.72	0.04	93.41
11berlin52-5	19.57	5.17	0.46	0.00	4.66	0.05	90.16
11eil51-2	200.63	48.72	1.39	0.00	47.29	0.03	97.08
11eil51-3	100.95	24.48	0.98	0.00	23.47	0.03	95.85
11eil51-4	142.50	37.00	1.94	0.00	34.95	0.11	94.45
11eil51-5	33.19	8.25	0.36	0.00	7.87	0.02	95.42
12brazil58-2	33.00	7.94	0.96	0.00	6.95	0.03	87.51
12brazil58-3	56.51	13.29	0.93	0.00	12.31	0.06	92.60
12brazil58-4	32.61	8.62	1.00	0.00	7.53	0.09	87.35
12brazil58-5	3.48	1.06	0.52	0.00	0.44	0.10	41.55
14st70-2	876.36	222.60	6.73	0.00	215.47	0.39	96.80
14st70-3	1071.01	264.38	4.16	0.00	260.10	0.12	98.38
14st70-4	354.16	87.56	1.86	0.00	85.61	0.08	97.78

Table 5.2 – continued from previous page

name	total-t	sep-t	sec-t	4pec-t	pec-t	comb-t	%pec
14st70-5	429.46	113.03	5.51	0.00	106.96	0.57	94.63
16eil76-2	160.97	38.04	1.72	0.00	36.27	0.04	95.36
16eil76-3	71.48	17.47	0.80	0.00	16.64	0.03	95.24
16eil76-4	173.67	43.19	1.11	0.00	42.03	0.05	97.31
16eil76-5	274.12	69.50	1.87	0.00	67.52	0.12	97.15
20gr96-2†	7200.00	1901.87	44.02	0.00	1857.29	0.56	97.66
20gr96-3†	7200.00	1862.37	38.38	0.00	1823.22	0.77	97.90
20gr96-4	5467.42	1428.08	48.45	0.00	1378.35	1.28	96.52
20gr96-5	6495.00	1643.50	35.00	0.00	1607.79	0.71	97.83
20kroA100-2	4291.87	1091.52	22.62	0.00	1068.47	0.42	97.89
20kroA100-3	4225.89	1060.29	14.82	0.00	1044.91	0.56	98.55
20kroA100-4	5057.47	1300.82	28.19	0.00	1271.60	1.04	97.75
20kroA100-5	6368.98	1606.81	20.13	0.00	1585.98	0.70	98.70
20kroB100-2	3389.43	841.28	12.24	0.00	828.79	0.25	98.52
20kroB100-3†	7200.04	1838.03	33.81	0.00	1803.14	1.08	98.10
20kroB100-4	3120.43	778.88	9.44	0.00	769.15	0.29	98.75
20kroB100-5	3397.49	883.26	24.75	0.00	857.50	1.01	97.08
20kroC100-2†	7200.00	1821.34	15.18	0.00	1805.91	0.25	99.15
20kroC100-3	3052.62	747.14	10.82	0.00	736.09	0.23	98.52
20kroC100-4	1009.37	250.86	4.82	0.00	245.88	0.16	98.01
20kroC100-5	2839.31	713.70	11.93	0.00	701.39	0.38	98.28
20kroD100-2†	7200.00	1852.91	33.91	0.00	1818.46	0.54	98.14
20kroD100-3	6287.9	1671.43	50.47	0.00	1619.66	1.30	96.90
20kroD100-4	4716.98	1190.26	18.79	0.00	1170.92	0.55	98.38
20kroD100-5	2669.25	671.32	13.10	0.00	657.78	0.44	97.98
20kroE100-2	4718.14	1204.19	24.14	0.00	1179.63	0.41	97.96
20kroE100-3	4737.91	1147.37	24.29	0.00	1122.59	0.49	97.84
20kroE100-4	2624.53	641.08	17.04	0.00	623.69	0.35	97.29
20kroE100-5	1892.52	476.91	10.32	0.00	466.24	0.35	97.76
20rat99-2	65.57	12.65	12.55	0.00	0.02	0.09	0.15
20rat99-3	2416.98	583.46	14.15	0.00	569.01	0.30	97.52
20rat99-4	6091.56	1414.13	140.03	0.00	1245.85	28.26	88.10
20rat99-5	3165.79	747.76	46.84	0.00	693.47	7.45	92.74
20rd100-2†	7200.00	1846.05	37.12	0.00	1808.40	0.52	97.96
20rd100-3	3815.24	969.42	23.26	0.00	945.69	0.47	97.55
20rd100-4	3273.97	826.82	16.76	0.00	809.60	0.46	97.92

Table 5.2 – continued from previous page

name	total-t	sep-t	sec-t	4pec-t	pec-t	comb-t	%pec
20rd100-5	2513.41	643.81	15.04	0.00	628.22	0.55	97.58
21eil101-2	2100.39	519.56	10.63	0.00	508.75	0.19	97.92
21eil101-3	4245.95	1069.99	18.31	0.00	1051.25	0.43	98.25
21eil101-4	906.82	227.88	7.48	0.00	220.15	0.25	96.61
21eil101-5	682.82	172.40	4.07	0.00	168.13	0.19	97.52
21lin105-2	86.33	21.14	20.93	0.00	0.03	0.18	0.15
21lin105-3†	7200.00	2047.88	380.14	0.00	1566.72	101.02	76.50
21lin105-4	3609.22	903.74	19.51	0.00	883.49	0.74	97.76
21lin105-5†	7200.00	1890.67	45.87	0.00	1843.24	1.56	97.49

†optimality was not verified within a time-limit of 7200 seconds.

The results indicate that the proposed branch-and-cut algorithm can solve instances involving up to 105 targets with modest computation times. The preprocessing algorithm in Sec. 5.5.1 was applied to 53/116 instances. The time taken by the preprocessing algorithm is not included in the overall computation time. The preprocessing algorithm reduced the size of these instances by 6 targets on average and the maximum reduction obtained was 14 targets. We observe that the instances that have a larger number of violated path elimination constraints take considerably large amount of computation time. The last column in table 5.2, whose average is 73%, indicates the percentage of separation time spent for finding violated path elimination constraints. This is not surprising because the time complexity for identifying violated path elimination constraints in (5.19) and (5.20) given a fractional solution, is $O(|T|^5)$ and $O(m|T|^4)$ respectively. The average number of T-comb inequalities that were generated in the enumeration tree were larger for some of the bigger instances (see table 5.1). They were effective, especially in tightening the lower bound for the instances that were not solved to optimality; for the instances where violated T-comb inequalities were separated out, the average linear programming relaxation gap improvement was 18%. They were also useful in reducing the computation times for larger instances despite increasing the computation times for smaller instances. Overall, we were able to solve 108/116 instances to the optimality with the largest instance involving 105 targets, 21 clusters and 5 depots. For the instances not solved to optimality within the time limit of 7200 seconds, the LP-rounding heuristic was effective in generating feasible solutions within 2.1% of the best feasible solution, on average.

5.7 Conclusion

In summary, we have presented an exact algorithm for the GMDTSP, a problem that has several practical applications including maritime transportation, health-care logistics, survivable telecommunication network design, and routing unmanned vehicles to name a few. A mixed-integer linear programming formulation including several classes of valid inequalities was proposed the facial structure of the polytope of feasible solutions was studied in detail. All the results were used to develop a branch-and-cut algorithm whose performance was corroborated through extensive numerical experiments on a wide range of benchmark instances from the standard library. The largest solved instance involved 105 targets, 21 clusters and 4 depots. Future work can be directed towards development of branch-and-cut approaches accompanied with a polyhedral study to solve the asymmetric counterpart of the problem.

6. CONCLUSION AND FUTURE WORK

In conclusion, this thesis has tried and succeeded to an extent to address a few challenges, combinatorial in nature, that arise in using multiple small unmanned and autonomous vehicles for monitoring and data gathering applications. In particular, we identified four distinct challenges namely, communication capabilities, dynamics, different sensing capabilities, and fuel restrictions of each of the vehicles and formulate combinatorial optimization problems, one for each challenge. We have developed numerically efficient algorithms to compute an optimal solution to each problem using a general branch-and-cut paradigm that has been used to solve combinatorial optimization problems, more specifically, mixed-integer linear programming problems. We note that this paradigm has been successfully used in the literature for over a decade to solve mixed-integer linear programs that frequent in other fields. The problems considered in this thesis are formulated in a way to make them suitable for applying this framework directly. Furthermore, some theoretical results developed in this work generalize some that are already available and can be adapted and used to solve variety of other problems of similar nature.

We have managed to just scrape the surface in addressing a few issues standalone that occur in these applications, let alone considering these challenges together. Future work can be focussed towards combining these challenges, formulating similar problems and studying the scalability of the developed algorithms. Some immediate generalizations that can be addressed in the framework presented in this thesis include:

i introducing capacitated and asymmetric versions of the MDRSP,

ii extending the approach to include vehicles with different dynamics in the HMDMTSP and the GMDTSP,

iii considering vehicles with different fuel capacities in the FCMDVRP,

iv imposing a global connectivity constraints for the FCMDVRP, and

v combining the HMDMTSP and the FCMDVRP to a single problem.

REFERENCES

[1] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

[2] David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. *The traveling salesman problem: a computational study*. Princeton University Press, 2011.

[3] R. Baldacci and M. Dell'Amico. Heuristic algorithms for the multi-depot ring-star problem. *European Journal of Operational Research*, 203(1):270–281, May 2010.

[4] R. Baldacci, M. Dell'Amico, and J. Salazar González. The capacitated m -ring-star problem. *Operations Research*, 55(6):1147–1162, December 2007.

[5] Roberto Baldacci, Maria Battarra, and Daniele Vigo. Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges*, pages 3–27. Springer, 2008.

[6] Roberto Baldacci and Aristide Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2009.

[7] Joaquín Bautista, Elena Fernández, and Jordi Pereira. Solving an urban waste collection problem using ants heuristics. *Computers & Operations Research*, 35(9):3020–3033, 2008.

[8] Tolga Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006.

[9] Tolga Bektas, Günes Erdogan, and Stefan Røpke. Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, 45(3):299–316, 2011.

[10] José-Manuel Belenguer, Enrique Benavent, Christian Prins, Caroline Prodhon, and Roberto Wolfler Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931–941, June 2011.

[11] Enrique Benavent and Antonio Martínez. Multi-depot multiple TSP: a polyhedral study and computational results. *Ann Oper Res*, 207(1):7–25, August 2013.

[12] I-Ming Chao, Bruce L Golden, and Edward A Wasil. A new algorithm for the site-dependent vehicle routing problem. In *Advances in computational and stochastic optimization, logic programming, and heuristic search*, pages 301–312. Springer, 1998.

[13] Václav Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Mathematical Programming*, 5(1):29–40, 1973.

[14] C. E. Corrigan, G. C. Roberts, M. V. Ramana, D. Kim, and V. Ramanathan. Capturing vertical profiles of aerosols and black carbon over the indian ocean using autonomous unmanned aerial vehicles. *Atmospheric Chemistry and Physics*, 8(3):737–747, 2008.

[15] J. A. Curry, J. Maslanik, G. Holland, and J. Pinto. Applications of aerosondes in the arctic. *Bulletin of the American Meteorological Society*, 85(12):1855 – 1861, 2004.

[16] Martin Desrochers and Gilbert Laporte. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

[17] Riddhi Doshi, Sai Yadlapalli, Sivakumar Rathinam, and Swaroop Darbha. Approximation algorithms and heuristics for a 2-depot, heterogeneous hamiltonian path problem. *International Journal of Robust and Nonlinear Control*, 21(12):1434–1451, 2011.

[18] Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

[19] Dominique Feillet, Pierre Dejax, and Michel Gendreau. Traveling salesman problems with profits. *Transportation science*, 39(2):188–205, 2005.

[20] Matteo Fischetti, Juan José Salazar González, and Paolo Toth. The symmetric generalized traveling salesman polytope. *Networks*, 26(2):113–123, 1995.

[21] Matteo Fischetti, Juan Jose Salazar Gonzalez, and Paolo Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148, 1998.

[22] Matteo Fischetti, Andrea Lodi, and Paolo Toth. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. 2001.

[23] Matteo Fischetti, Juan Jose Salazar Gonzalez, and Paolo Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45(3):378–394, 1997.

[24] Eric W. Frew and Timothy X. Brown. Networking issues for small unmanned aircraft systems. *Journal of Intelligent and Robotic Systems*, 54:21–37, March

2009.

[25] Bezalel Gavish and Kizhanathan Srikanth. An optimal solution method for large-scale multiple traveling salesmen problems. *Operations Research*, 34(5):698–717, 1986.

[26] Gianpaolo Ghiani, Francesca Guerriero, Gilbert Laporte, and Roberto Musmanno. Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and length restrictions. *Journal of Mathematical Modelling and Algorithms*, 3(3):209–223, 2004.

[27] Gianpaolo Ghiani and Gennaro Improta. An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research*, 122(1):11–17, 2000.

[28] Stefan Gollowitzer, Luis Gouveia, Gilbert Laporte, Dilson Lucas Pereira, and Adam Wojciechowski. A comparison of several models for the hamiltonian p-median problem. *Networks*, 63(4):350–363, 2014.

[29] Martin Grötschel and M. W. Padberg. Polyhedral theory. *The traveling salesman problem*, pages 251–305, 1985.

[30] Gregory Gutin and Daniel Karapetyan. A memetic algorithm for the generalized traveling salesman problem. *Natural Computing*, 9(1):47–60, 2010.

[31] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[32] Gerhard Hiermann, Jakob Puchinger, and Richard F Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Technical report, 2014.

[33] Alessandro Hill and Stefan Voss. An equi-model matheuristic for the multi-depot ring star problem. Working papers, University of Antwerp, Faculty of Applied Economics, 2014.

[34] IBM ILOG CPLEX. ILOG CPLEX. *ILOG CPLEX 12.4 Advanced Reference Manual*, 2012.

[35] Imdat Kara. On the miller-tucker-zemlin based formulations for the distance constrained vehicle routing problems. In *ICMS international conference on Mathematical Science*, volume 1309, pages 551–561. AIP Publishing, 2010.

[36] Imdat Kara. Arc based integer programming formulations for the distance constrained vehicle routing problem. In *Logistics and Industrial Informatics (LINDI), 2011 3rd IEEE International Symposium on*, pages 33–38. IEEE, 2011.

[37] Imdat Kara and Tolga Bektas. Integer linear programming formulations of multiple salesman problems and its variations. *European Journal of Operational Research*, 174(3):1449–1458, 2006.

[38] Safia Kedad-Sidhoum and Viet Hung Nguyen. An exact algorithm for solving the ring star problem. *Optimization*, 59(1):125–140, 2010.

[39] Samir Khuller, Azarakhsh Malekian, and Julián Mestre. To fill or not to fill: the gas station problem. In *Algorithms–ESA 2007*, pages 534–545. Springer, 2007.

[40] Martine Labbé, Gilbert Laporte, Inmaculada Rodríguez Martín, and Juan José Salazar González. The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, 43(3):177–189, 2004.

[41] Martine Labbé, Gilbert Laporte, Inmaculada Rodríguez Martín, and Juan José Salazar González. Locating median cycles in networks. *European Journal of Operational Research*, 160(2):457–470, 2005.

[42] Martine Labbé, M., Gilbert Laporte, and Inmaculada Rodríguez-Martín. Path, tree and cycle location. In Teodor Gabriel Crainic and Gilbert Laporte, editors, *Fleet Management and Logistics*, pages 187–204. Springer US, January 1998.

[43] Henry AL Labordere. Record balancing problem-A dynamic programming solution of a generalized traveling salesman problem. *Revue Francaise D Informatique De Recherche Operationnelle*, 3(NB 2):43, 1969.

[44] G. Laporte, Y. Nobert, and D. Arpin. An exact algorithm for solving a capacitated location-routing problem. *Ann Oper Res*, 6(9):291–310, September 1986.

[45] Gilbert Laporte. Location routing problems. 1987.

[46] Gilbert Laporte, Ardavan Asef-Vaziri, and Chelliah Sriskandarajah. Some applications of the generalized traveling salesman problem. *Journal of the Operational Research Society*, pages 1461–1467, 1996.

[47] Gilbert Laporte, Martin Desrochers, and Yves Nobert. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14(1):161–172, 1984.

[48] Gilbert Laporte, Hélène Mercure, and Yves Nobert. Generalized traveling salesman problem through n sets of nodes: the asymmetrical case. *Discrete Applied Mathematics*, 18(2):185–197, 1987.

[49] Gilbert Laporte, Yves Nobert, and Serge Taillefer. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation science*, 22(3):161–172, 1988.

[50] Eugene L Lawler, Jan Karel Lenstra, AHG Rinnooy Kan, and David B Shmoys. *The traveling salesman problem: a guided tour of combinatorial optimization*,

volume 3. Wiley New York, 1985.

[51] Y. Lee, Steve Y Chiu, and J. Sanchez. A branch and cut algorithm for the steiner ring star problem. *Management Science and Financial Engineering*, 4(1):21–34, 1998.

[52] David Levy, Kaarthik Sundar, and Sivakumar Rathinam. Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. *Mathematical Problems in Engineering*, 2014, 2014.

[53] Chung-Lun Li, David Simchi-Levi, and Martin Desrochers. On the distance constrained vehicle routing problem. *Operations research*, 40(4):790–799, 1992.

[54] Satyanarayana G Manyam, Sivakumar Rathinam, Swaroop Darbha, David Casbeer, and Phil Chandler. Routing of two unmanned aerial vehicles with communication constraints. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 140–148. IEEE, 2014.

[55] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

[56] Derek Mitchell, Micah Corah, Nilanjan Chakraborty, Katia Sycara, and Nathan Michael. Multi-robot long-term persistent coverage with fuel constrained robots. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1093–1099. IEEE, 2015.

[57] José A. Moreno Pérez, J. Marcos Moreno-Vega, and Inmaculada Rodríguez Martín. Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operational Research*, 151(2):365–378, December 2003.

[58] Barindra Nag, Bruce L Golden, and Arjang Assad. Vehicle routing with site dependencies. *Vehicle Routing: Methods and Studies*, pages 149–159, 1988.

[59] Viswanath Nagarajan and R Ravi. Approximation algorithms for distance constrained vehicle routing problems. *Networks*, 59(2):209–214, 2012.

[60] Alan Nawoj. Energy tradeoffs between collection-based routing protocols in wireless sensor networks. 2004.

[61] George L Nemhauser and Laurence A Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[62] Charles Edward Noon and James C Bean. An efficient transformation of the generalized traveling salesman problem. *Ann Arbor*, 1001:48109–2117, 1989.

[63] Paul Oberlin, Sivakumar Rathinam, and Swaroop Darbha. A transformation for a heterogeneous, multiple depot, multiple traveling salesman problem. In *American Control Conference, 2009. ACC'09.*, pages 1292–1297. IEEE, 2009.

[64] Paul Oberlin, Sivakumar Rathinam, and Swaroop Darbha. Today's travelling salesman: Depot, multiple uav routing problem. *IEEE robotics & automation magazine*, 17(4):70–77, 2010.

[65] Manfred W Padberg. A Note on Zero-One Programming. *Operations Research*, 23(4):833–837, 1975.

[66] Manfred W Padberg and M Ram Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

[67] Michael Polacek, Karl F Doerner, Richard F Hartl, and Vittorio Maniezzo. A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics*, 14(5):405–423, 2008.

[68] James Reeds and Lawrence Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific journal of mathematics*, 145(2):367–393, 1990.

[69] Gerhard Reinelt. TSPLIB - a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[70] Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

[71] SS Srivastava, Santosh Kumar, RC Garg, and Prasenjit Sen. Generalized traveling salesman problem through n sets of nodes. *CORS journal*, 7:97–101, 1969.

[72] PB Sujit, Daniel E Lucani, and Joao B Sousa. Bridging cooperative sensing and route planning of autonomous vehicles. *Selected Areas in Communications, IEEE Journal on*, 30(5):912–922, 2012.

[73] Kaarthik Sundar and Sivakumar Rathinam. Route planning algorithms for unmanned aerial vehicles with refueling constraints. In *American Control Conference (ACC), 2012*, pages 3266–3271. IEEE, 2012.

[74] Kaarthik Sundar and Sivakumar Rathinam. A primal-dual heuristic for a heterogeneous unmanned vehicle path planning problem. *International Journal of Advanced Robotic Systems*, 10(349):3, 2013.

[75] Kaarthik Sundar and Sivakumar Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *Automation Science and Engineering, IEEE Transactions on*, 11(1):287–294, 2014.

[76] Kaarthik Sundar and Sivakumar Rathinam. An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem. In *Unmanned*

Aircraft Systems (ICUAS), 2015 International Conference on, pages 366–371. IEEE, 2015.

[77] Kaarthik Sundar and Sivakumar Rathinam. Generalized multiple depot traveling salesmen problempolyhedral study and exact algorithm. *Computers & Operations Research*, 70:39 – 55, 2016.

[78] Kaarthik Sundar, PB Sujit, Sivakumar Rathinam, Daniel Enrique Lucani Roetter, and João Sousa. Algorithms for collecting data from cooperating sensor motes using unmanned vehicles. In *2015 Indian Control Conference (ICC)*. IEEE, 2015.

[79] Éric D Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO-Operations Research*, 33(01):1–14, 1999.

[80] Onur Tekdas, Volkan Isler, Jong Hyun Lim, and Andreas Terzis. Using mobile robots to harvest data from sensor fields. *Wireless Communications, IEEE*, 16(1):22–28, 2009.

[81] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. Siam, 2001.

[82] US DOE. Alternate Fuels Data Center.

[83] US EPA. Environment Protection Agency, Inventory of US Greenhouse Gas Emissions and Sinks: 1990-2007, EPA 430-R-09-004, 2009.

[84] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[85] Jiefeng Xu, Steve Y. Chiu, and Fred Glover. Optimizing a ring-based private line telecommunication network using tabu search, March 1999.

[86] Eilers J Zajkowski T, Dunagan S. Small UAS communications mission. In *Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference, Salt Lake City, UT*, 2006.