J Intell Robot Syst (2017) 88:513–526
DOI 10.1007/s10846-016-0458-5

Algorithms for Heterogeneous, Multiple Depot, Multiple Unmanned Vehicle Path Planning Problems

Kaarthik Sundar · Sivakumar Rathinam

Received: 31 August 2016 / Accepted: 15 December 2016 / Published online: 26 December 2016

Abstract Unmanned vehicles, both aerial and ground, are being used in several monitoring applications to collect data from a set of targets. This article addresses a problem where a group of heterogeneous aerial or ground vehicles with different motion constraints located at distinct depots visit a set of targets. The vehicles also may be equipped with different sensors, and therefore, a target may not be visited by any vehicle. The objective is to find an optimal path for each vehicle starting and ending 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 length of the paths for all the vehicles is minimized. Two variants of this problem are formulated (one for ground vehicles and another for aerial vehicles) as mixed-integer linear programs and a branch-and-cut algorithm is developed to compute an optimal solution to each of the variants. 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.

Keywords Unmanned vehicles · Branch-and-cut · Heterogeneous vehicles · Site-dependent vehicle routing · Vehicle routing · Dubins vehicles · Reeds-Shepp vehicles

K. Sundar (✉)
Center for Non-Linear Studies, Los Alamos National Laboratory, Los Alamos, NM 87544, USA
e-mail: kaarthik01sundar@gmail.com

S. Rathinam
Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA

1 Introduction

Unmanned aerial vehicles, ground vehicles and underwater vehicles are being used routinely in military applications such as border patrol, reconnaissance and surveillance expeditions. They are also being used for civilian applications like remote sensing, traffic monitoring, weather and hurricane monitoring [1–3]. 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 path planning 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 [4, 5]. 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 payload restrictions. In this case, we partition the set of targets into disjoint subsets: (i) targets to be visited by specific vehicles, and (ii) common targets that can be visited by any of the vehicles.

This article considers two variants of the following heterogeneous, multiple depot, multiple unmanned vehicle path planning problem: given a set of targets and a fleet of heterogeneous vehicles located at distinct depots, an heading angle for visiting each target and each depot, find a path 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, the paths satisfy the motion constraints of the respective vehicles, and the total length of the paths traveled by all the vehicles is a minimum. The first variant of the problem considers ground vehicles where each vehicle can move forwards and backwards at a constant velocity with a bound on its minimum turning radius; a vehicle which satisfies these motion constraints is commonly referred to as the Reed-Shepp vehicle [6] in the literature. The second variant of the problem considers aerial vehicles where each vehicle can only move forwards at a constant velocity with a bound on its minimum turning radius; a vehicle which satisfies these motion constraints is called a Dubins vehicle [7] in the literature. Given the desired heading angles of the vehicles at the target locations, one can compute the length of the shortest path between any two targets for a Reed-Shepp vehicle and a Dubins vehicle using the results in [6] and [7], respectively. The main difference between solving the two variants of the path planning problem lies in the fact that these lengths are symmetric for ground vehicles and asymmetric for aerial vehicles, i.e., for any two targets i and j, if d_{ij} denotes the length of traveling from the i^{th} target to the j^{th} target, then $d_{ij} = d_{ji}$ for the ground vehicle, and d_{ij} may not be equal to d_{ji} for the aerial vehicle. Refer to Figs. 1 and 2 for an illustration of a feasible solution to the ground and aerial vehicle problems, respectively. The ground vehicle variant is referred to as the Multiple Ground Vehicle Path planning Problem (MGVPP) and the aerial vehicle variant is referred to as the Multiple Around Vehicle Path planning Problem (MAVPP).

The MGVPP and MAVPP are generalizations of the multiple depot symmetric [8] and asymmetric multiple traveling salesmen problems, respectively, which are known to be NP-hard. The main contributions of this article are as follows: (i) we develop mixed-integer linear programming formulations for the MGVPP and MAVPP, respectively, (ii) the linear programming relaxation of both the formulations are then strengthened by introducing new valid inequalities, (iii) a transformation method to transform any

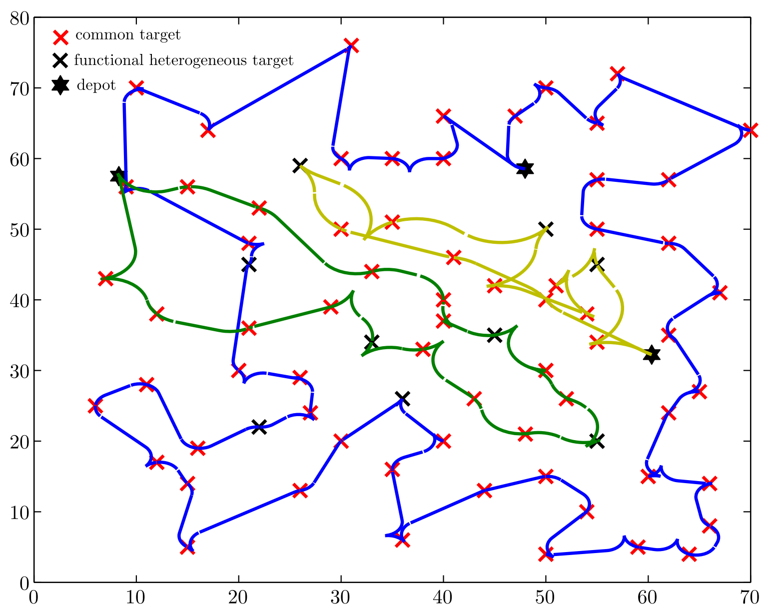

Fig. 1 A feasible solution for a MGVPP with the ground vehicles modeled as Reeds-Shepp vehicles

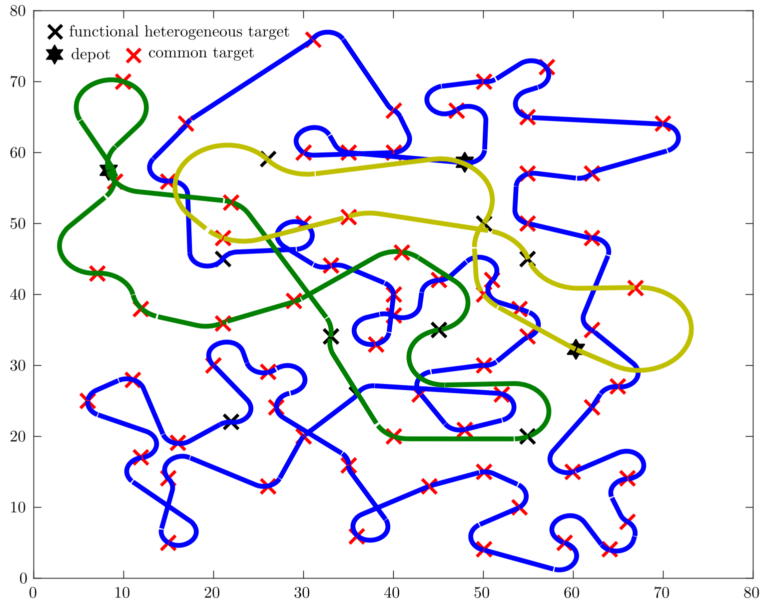

Fig. 2 A feasible solution for a MAVPP with the aerial vehicles modeled as Dubins vehicles

result of the MVAPP to the MVGPP is presented, (iv) a branch-and-cut algorithm to compute an optimal solution to any instance of the MGVPP and MAVPP based on the formulations is elaborated, and (v) finally, extensive computational results to corroborate the effectiveness of the proposed algorithms are presented. To the best of our knowledge, this is the first article to develop algorithms to compute an optimal solution for these variants of the problem. The reminder of the article is organized as follows. In Section 2, we discuss the relevant literature. In Sections 3–4, we formulate the two problems as a mixed-integer linear programs and present additional valid inequalities to strengthen the linear programming relaxations. A branch-and-cut algorithm based on the formulations is described in Section 5, and Section 6 presents computational results on several classes of test instances.

2 Related Work

The single vehicle variant of the MGVPP and MAVPP is the symmetric and asymmetric traveling salesman problem (STSP and ATSP), respectively. Over the past two decades, several methods including exact algorithms, heuristics, and approximation algorithms have been developed to address the STSP and the ATSP [9]. The MGVPP and MAVPP reduce to the multiple depot symmetric and asymmetric multiple traveling salesmen problems (MDSMTSP and MDAMTSP) when all the vehicles are homogeneous. Authors in [8, 10] present an algorithm to solve the symmetric version to optimality. Another variant of the MDSMTSP that has received considerable attention in the literature is the single depot symmetric multiple traveling salesman problem, which we abbreviate as MSTSP. In the MSTSP, there are m homogeneous ground 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 MSTSP and its variations, Kara and Bektas present some integer linear programming formulations in [11]. Bektas reviews the applications, exact and heuristic solution procedures and transformations of MSTSP to the STSP in [12]. A branch-and-bound-based method for large-scale MSTSP may be found in [13]. To the best of our knowledge, there are no computationally efficient algorithms in the literature to solve any variant of the multiple vehicle asymmetric versions, discussed thus far, to optimality. The asymmetric travel distances is a specific feature of the aerial vehicles, in particular, fixed wing aircrafts that feature in applications pertaining to surveillance missions using Dubins vehicle models (see [14–19] and references therein).

The MGVPP can also be considered as a special case of the multiple depot vehicle routing problem (MDVRP). The MDVRP consists of finding a set of routes for based on a set of given depots to serve the demand of a set of customers with multiple homogeneous vehicles of limited capacity. Laporte et al. [20] study variants of this problem with asymmetric travel distances and propose branch-and-bound algorithms to solve the problem to optimality. More recently, authors in [21] have developed an exact solution framework to solve different vehicle routing problems that can be applied to the MDVRP as well. Taillard introduced and developed a column generation heuristic for the vehicle routing problems (VRPs) using an heterogeneous ground vehicle fleet in [22]. He 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 algorithm available in the literature to solve any variant of heterogeneous VRPs to optimality.

Baldacci et al. [23] 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 [24, 25].

Riddhi et al. present an approximation algorithm for the 2-depot heterogeneous hamiltonian path problem in [26]. This is the first paper that considers both functional and structural heterogeneous ground vehicles. Apart from [26], we are not aware of any literature that addresses multiple depot routing problem with a functional and structural heterogeneous fleet of ground or aerial vehicles and develops algorithms to solve them to optimality. The main focus of this article is the development of an exact algorithm based on a branch-and-cut method [27, 28] for solving the MGVPP and MAVPP. We also present a computational study for the algorithm in order to evaluate its performance.

3 Mathematical Formulation

This section is presented in two parts: the first part develops an integer linear programming formulation for the MGVPP after introducing the relevant notation and the second part develops a similar formulation for the MAVPP. Let T denote the set of targets and we have a heterogeneous fleet of n vehicles initially stationed at distinct depots. Let $D = \{d_1, d_2, \ldots, d_n\}$ represent the set of the depots. We will refer to the set $V = D \cup T$ as the set of vertices. Associated with each vertex $i \in V$ is an orientation angle δ_i which is the angle at which any vehicle has to arrive and depart from the vertex. 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. The notations that are introduced thus far are common to both aerial and ground vehicles.

3.1 Ground Vehicle Problem Formulation

Each vehicle in the heterogeneous fleet of ground vehicles is modeled as Reed-Sheep vehicles with distinct value for its minimum turn radius. The kinematic constraints of a Reeds-Shepp vehicle stationed at depot d_k is given by: $v_x = v\cos\theta,\ v_y = v\sin\theta, \dot{\theta} \leq u_k$, where v_x and v_y are the x and y components of the velocities, respectively, and $\dot{\theta}$ and u_k are the angular velocity and the maximum yaw-rate of the vehicle. u_k is different across vehicles and $|v|$ is assumed to be a constant across vehicles. When $|v|$ is constant across vehicles the u_k's can be mapped bijectively to the vehicles' minimum turn radius values. Given these vehicles, the problem is formulated on 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. For each $(i, j) = e \in E$

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

for each vehicle $v \in \{1, \ldots, n\}$, let c_e^v be the length of the minimum distance path for the vehicle v to traverse the edge E subject to its kinematic constraints and with an angle of departure and arrival of δ_i and δ_j at i and j, respectively. The length can be computed either using geometry [6] or as a solution to an optimal control problem [29]. Since a ground vehicle can move both forward and backward, the length of the path from i to j is equal to the path from j to i for fixed orientations δ_i and δ_j; hence, the problem is formulated on an undirected graph.

We now present a mathematical formulation for the MGVPP, inspired by the models for the standard routing problems in [9, 30]. 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 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 vehicle can start at its depot and return to the depot after visiting a single 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.

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 $x^k(\bar{E}) = \sum_{(i,j)\in\bar{E}} x_{ij}^k$. Using the above notations, the MGVPP is formulated as an integer linear program as follows:

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

$$x^k(\delta(i)) = 2y_i^k \qquad \forall i \in T, k \in \{1, \ldots, n\}, \tag{2}$$

$$x^k(\delta(S)) \geq 2y_i^k \qquad \forall i \in S, S \subseteq T, k \in \{1, \ldots, n\}, \tag{3}$$

$$\sum_{k=1}^{n} y_i^k = 1 \qquad \forall i \in T, \tag{4}$$

$$y_i^k = 1 \qquad \forall k \in \{1, \ldots, n\}, i \in R_k, \tag{5}$$

$$x_e^k \in \{0, 1, 2\} \qquad \forall e \in \{(d_k, j) : j \in T\}, k \in \{1, \ldots, n\}, \tag{6}$$

$$x_e^k \in \{0, 1\} \qquad \forall e \in \{(i, j) : i \in T, j \in T\}, k \in \{1, \ldots, n\} \text{ and} \tag{7}$$

$$y_i^k \in \{0, 1\} \qquad \forall i \in T, k \in \{1, \ldots, n\}. \tag{8}$$

In the above formulation, the constraints in Eq. 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. 4 ensure that each target $i \in T$ is visited by some vehicle. The constraints in Eq. 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. 5 are the vehicle–target assignment constraints for the functional heterogeneous targets. Constraints in Eqs. 6, 7, and 8 are the integrality restrictions on the decision variables. If the integrality restrictions in constraints (6), (7), and (8) are relaxed, then we call that model a linear programming relaxation of the formulation.

3.2 Aerial Vehicle Problem Formulation

Similar to the ground vehicle model, each aerial vehicle is assumed to have a different value of its minimum turn radius and all the n vehicles are assumed to travel with the same velocity. Given these assumptions, the problem is formulated on a directed graph $\tilde{G} = (V, A)$, A is a set of directed edges between any two vertices. We remark that unlike the formulation for the ground vehicles, here (i, j) and (j, i) are two different edges in the set A. For each directed edge $(i, j) \in A$ and each vehicle v, let d_{ij}^v denote the minimum length path for v to traverse the edge from i to j with angles of departure and arrival δ_i and δ_j respectively. This length can be computed using the well-known result of Dubins [7]. Similar to the ground vehicle, for each vehicle $v \in \{1, \ldots, n\}$, we associate with each edge (i, j) a variable x^v_{ij}, whose value is 1 when the vehicle traverses the edge and 0 otherwise. Similarly, for each vehicle $v \in \{1, \ldots, n\}$, we associate with each target $i \in T$ a binary variable y^v_i, which takes a value 1 when the target i is visited by vehicle v and 0 otherwise.

Apart from the notations introduced for ground vehicles, for any $S \subset V$, we define $\delta^+(S) = \{(i, j) \in A : i \in S,\ j \notin S\}$, $\delta^-(S) = \{(i, j) \in A : i \notin S,\ j \in S\}$. If $S = \{i\}$, we simply write $\delta^+(i)$ and $\delta^-(i)$ instead of $\delta^+(\{i\})$ and $\delta^-(\{i\})$, respectively. Using the above notations and those introduced in Section 3.1, the MAVPP is formulated as an integer linear program as follows:

$$\min \sum_{k=1}^{n} \sum_{(i,j)\in A} d^k_{ij} x^k_{ij} \quad \text{subject to:} \tag{9}$$

$$x^k(\delta^+(i)) = y^k_i \quad \forall i \in T, k \in \{1, \ldots, n\}, \tag{10}$$

$$x^k(\delta^-(i)) = y^k_i \quad \forall i \in T, k \in \{1, \ldots, n\}, \tag{11}$$

$$x^k(\delta^+(S)) \geq y^k_i \quad \forall i \in S, S \subseteq T, k \in \{1, \ldots, n\}, \tag{12}$$

$$\sum_{k=1}^{n} y^k_i = 1 \quad \forall i \in T, \tag{13}$$

$$y^k_i = 1 \quad \forall k \in \{1, \ldots, n\}, i \in R_k, \tag{14}$$

$$x^k_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, k \in \{1, \ldots, n\} \text{ and} \tag{15}$$

$$y^k_i \in \{0, 1\} \quad \forall i \in T, k \in \{1, \ldots, n\}. \tag{16}$$

The constraints in Eqs. 10 and 11 ensure that the in-degree and out-degree of a target visited by vehicle k is one. The remaining constraints in Eqs. 12–16 are similar to the constraints introduced for the ground vehicle. The linear programming relaxation of the above integer linear program is obtained by relaxing the integrality restrictions on the constraints (15) and (16). In the following section, we strengthen the linear programming relaxations of the formulations corresponding to ground and aerial vehicles by introducing additional valid inequalities (a valid inequality is a linear inequality in the variables defining the problem that does not remove any feasible solutions to the problem).

4 Additional Valid Inequalities

In this section, we develop different classes of valid inequalities for the MGVPP and MAVPP. First, we introduce a transformation method to convert any valid inequality for the MAVPP to a valid inequality for the MGVPP. Suppose, we are given a valid inequality for the MAVPP, we can construct an MGVPP counterpart by setting $x^k_e = x^k_{ij} + x^k_{ji}$ for all $e \in E$, $e = (i, j)$. Hence, throughout the rest of the article we will present proof of validity for inequalities only for the MAVPP, unless otherwise stated. Now, we present an alternate characterization of the sub-tour elimination constraints in Eqs. 3 and 12 for the MGVPP and MAVPP, respectively.

Proposition 1 *The sub-tour elimination constraints in Eqs. 3 and 12 for each vehicle k and any $S \subseteq T$ can be equivalently stated as $x^k(\gamma(S)) \leq \sum_{j\in S\setminus\{i\}} y^k_j$ for every $i \in S$ and $k \in \{1, \ldots, n\}$. This alternate form is also referred to as the cycle form of the sub-tour elimination constraints and holds for the both the aerial and ground vehicles.*

Proof We present the proof for the constraints corresponding to MAVPP. The proof for MGVPP can be obtained by the transformation method. For a fixed vehicle k and a fixed subset $S \subseteq T$, any feasible solution to the MAVPP satisfies the following equality:

$$\begin{aligned}
&\sum_{j\in S}\left(x^k(\delta^+(j)) + x^k(\delta^-(j))\right) = x^k(\delta^+(S)) \\
&\quad + x^k(\delta^-(S)) + 2x^k(\gamma(S)) \\
\Rightarrow\ &\sum_{j\in S} 2y^k_j = x^k(\delta^+(S)) + x^k(\delta^-(S)) \\
&\quad + 2x^k(\gamma(S)) \quad \text{(using Eq. (10) and (11))} \\
\Rightarrow\ &2x^k(\gamma(S)) \leq \sum_{j\in S} 2y^k_j \\
&\quad - 2y^k_i \quad \forall i \in S \ \text{(from Eq. (12))} \\
\Rightarrow\ &x^k(\gamma(S)) \leq \sum_{j\in S\setminus\{i\}} y^k_j \quad \forall i \in S.
\end{aligned}$$

We remark that the cycle form of the sub-tour elimination constraints is very sparse and performs computationally better. □

We first derive trivial inequalities that are special cases of the sub-tour elimination constraints in Eqs. 3 and 12 for the MGVPP and MAVPP, respectively.

Proposition 2 *The sub-tour elimination in Eq. 12 reduces to the following constraints when $S = \{i, j\}$ i.e., $|S| = 2$:*

$$x_{ij}^k + x_{ji}^k \le y_i^k \text{ and } x_{ij}^k + x_{ji}^k \le y_j^k. \quad (17)$$

Proof When $S = \{i, j\}$, $x^k(\delta^+(S)) = x^k(\delta^+(i)) + x^k(\delta^+(j)) - x_{ij}^k - x_{ji}^k$. Using the above equality and Eq. 10, we obtain the required inequalities. □

Similarly for the MGVPP, when $S = \{i, j\}$, the constraint in Eq. 3 reduces to $x_e^k \le y_i^k$ and $x_e^k \le y_j^k$ where $e = (i, j)$.

The inequalities that are valid for a ATSP (STSP) are also valid for the MAVPP (MGVPP) when suitably modified for the multiple vehicle setting. We particularly examine the *2-matching inequalities* available for the STSP [8, 9]. Specifically, we consider the following inequality for every vehicle k for the MAVPP:

$$x^k(\gamma(H)) + x^k(\mathcal{T}) \le \sum_{i \in H} y_i^k + \frac{|\mathcal{T}| - 1}{2} \quad (18)$$

for all $H \subseteq T$ and $\mathcal{T} \subset \left(\delta^+(H) \cup \delta^-(H)\right)$. 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,
- $|\mathcal{T}| \ge 3$ and odd.

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

Proposition 3 *The 2-matching inequality in Eq. 18 is valid for any feasible solution to the MAVPP.*

Proof For any $H \subseteq T$ and $\mathcal{T} \subset \left(\delta^+(H) \cup \delta^-(H)\right)$ satisfying the conditions stated previously, we have the following equality that is satisfied by any feasible solution to the MAVPP:

$$\begin{aligned}
&2x^k(\gamma(H)) + x^k(\delta^+(H)) + x^k(\delta^-(H)) \\
&= \sum_{i \in H} x^k(\delta(i)) \\
&\Rightarrow 2x^k(\gamma(H)) + x^k(\mathcal{T}) \\
&+ x^k\left((\delta^+(H) \cup \delta^-(H)) \setminus \mathcal{T}\right) \\
&= 2\sum_{i \in H} y_i.
\end{aligned}$$

We also have $x^k(\mathcal{T}) \le |\mathcal{T}|$ for the set $\mathcal{T}$ (since $x_{ij}^k \le 1$ for any $(i, j), i, j \in T$). Adding this inequality to the above equality, we obtain,

$$\begin{aligned}
&2x^k(\gamma(H)) + 2x^k(\mathcal{T}) + x^k\left((\delta^+(H) \cup \delta^-(H)) \setminus \mathcal{T}\right) \\
&\qquad \le 2\sum_{i \in H} y_i^k + |\mathcal{T}| \\
&\Rightarrow 2x^k(\gamma(H)) + 2x^k(\mathcal{T}) \le 2\sum_{i \in H} y_i^k + |\mathcal{T}|.
\end{aligned}$$

The last inequality follows because $x^k\left((\delta^+(H) \cup \delta^-(H)) \setminus \mathcal{T}\right) \ge 0$. Further simplification using the facts that the L.H.S. of the last inequality is even and $|\mathcal{T}|$ is odd yields:

$$\begin{aligned}
&2x^k(\gamma(H)) + 2x^k(\mathcal{T}) \le 2\sum_{i \in H} y_i^k + (|\mathcal{T}| - 1) \\
&\Rightarrow x^k(\gamma(H)) + x^k(\mathcal{T}) \le \sum_{i \in H} y_i^k + \frac{|\mathcal{T}| - 1}{2}.
\end{aligned}$$

Hence the 2-matching inequality is valid for the MAVPP. □

The above 2-matching constraints are also valid for the MGVPP; the proof follows from the transformation method. The constraints in Eq. 18 are also equivalent to the *blossom's inequality* for the 2-matching problem and a special case of the *comb inequality* for the STSP [31]. Equation 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.

5 Branch-and-Cut Algorithm

In this section, we briefly present the main ingredients of a branch-and-cut algorithm that is used to solve the two formulations presented in the Section 3 to optimality. The formulations developed in the Section 3 can be provided to off-the-shelf commercial branch-and-cut solvers to obtain an optimal solution to the MGVPP and MAVPP. But, observe that MGVPP and MAVPP contain the sub-tour elimination constraints ((3) and (12), respectively) which enforce any feasible solution to the corresponding problems not to contain any sub-tours of the targets. The number of such constraints is exponential and it may not be computationally efficient to enumerate all these constraints and provide them to these solvers. Furthermore, the 2-matching valid inequalities developed to tighten the linear programming relaxations of the MGVPP and

MAVPP are also exponential in number. To address this issue, we use the following approach: we relax these constraints from the formulation, and whenever the solver obtains an integer solution feasible to this relaxed problem (or a fractional solution with integrality constraints dropped), we check if any of these constraints are violated by the feasible solution, integer or fractional. If so, we add the infeasible constraint and continue solving the original problem. This process of adding constraints to the problem sequentially has been observed to be computationally efficient for the STSP and some of its variants [10, 32]. The algorithms used to identify violated constraints are referred to as separation algorithms. For the sake of completeness, a detailed pseudo-code of the branch-and-cut algorithm for the MAVPP is given in Algorithm 1. The pseudo-code for the MGVPP is similar and hence, is not presented.

Algorithm 1 Pseudo-code of the branch-and-cut algorithm (MAVPP)

```
τ̄ ← optimal solution to an instance of the MVAPP;
STEP 1 (Initialization);
    Iteration count: t ← 1;
    Initial upper bound on the optimal objective: ᾱ ← +∞;
    Initial sub-problem: (9)–(11), (13)–(14), and 0 ≤ x^k_ij, y^k_i ≤ 1;
    The initial subproblem is solved and inserted in a list ℒ;
STEP 2 (Termination check and subproblem selection);
if ℒ = ∅ then
    STOP;
else
    Select a subproblem from ℒ with the lowest objective value [27];
end
STEP 3 (Subproblem solution);
    t ← t + 1;
    Let α be the solution objective value;
if α ≥ ᾱ then
    Go to STEP 2;
else
    if solution is feasible for MAVPP, set ᾱ ← α, update τ̄ and go to STEP 2;
end
STEP 4 (Constraint separation and generation) Given a fractional solution
  (x, y), introduce violated sub-tour elimination constraints (12), connectivity
  constraints (17), and 2-matching constraints (18). If no constraints can be
  generated using the current fractional solution, then go to STEP 5, else go to
  STEP 3;
STEP 5 (Branching) Create two subproblems by branching on a fractional y^k_i or
  x^k_ij variable. First, select a fractional y^k_i variable, based on the strong
  branching rule [33]. If all these variables are integer, then select a fractional x^k_ij
  variable using the same rule. Then insert both the subproblems in the list ℒ
  and go to STEP 2;
```

In the following paragraphs, we detail the separation algorithms used to dynamically identify violated constraints given a fractional solution. We present the algorithms for the MAVPP and the transformation method in Section 4 enables using the algorithms without any modification on a transformed graph for the MGVPP. For every vehicle k, $G_k^* = (V_k^*, A_k^*)$ denotes the *support graph* associated with a given fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, i.e., $V_k^* := \{i \in T : y_i^{k*} > 0\} \cup \{d_k\}$ and $A_k^* := \{(i, j) \in A : x_{ij}^{k*} > 0\}$. Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of the decision variable values in MAVPP.

Separation of Constraints (12) and (17): As shown previously Section 4, the inequalities in Eq. 12 reduce to Eq. 17 when $|S| = 2$. For every vehicle k, the violation of the inequality in Eq. 17 can be verified by examining the inequality for every pair of targets in the set V_k^*. Next, we examine the strongly connected components in G_k^*. Each strongly 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 strongly connected component C that contains a depot d_k, $i \in C \setminus \{d_k\}$, and a fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, 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 directed graph $\bar{G}_k = (\bar{V}_k, \bar{A}_k)$, 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 $\bar{A}_k = A_k^*$. Every edge $(i, j) \in \bar{A}_k$ is assigned a capacity x_{ij}^{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 $y_i^{k*} = 0$ need not be considered. This algorithm can be repeated for every vehicle to generate the violated sub-tour elimination constraints. Once the set S' that defines a violated sub-tour elimination constraint is obtained, its corresponding cycle form given by Proposition 1 is added back to the formulation and the branch-and-cut algorithm is continued.

Separation of 2-Matching Constraints (18): We discuss exact and heuristic separation procedures for the 2-matching constraints. Using a construction similar to the one proposed in [34] 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, so we use the following simple heuristic proposed by the authors in [35, 36]. 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 $(i, j) \in \big(\delta^+(H) \cup \delta^-(H)\big)$ with $x_{ij}^{k*} = 1$. We reject the inequality if the number of teeth is even. The procedure can be implemented in $O(|V_k^*| + |A_k^*|)$ time and can be repeated for each vehicle k.

6 Computational Results

The branch and cut algorithm was implemented in C++ (clang++ 7.0.2) 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 Macbook Pro, 2.9 GHz Intel Core i5 processor with 16 GB RAM. The computation times reported are expressed in seconds, and we imposed a time limit of 3600 seconds for each run of the algorithm. The performance of the algorithm was tested on instances generated using the traveling salesman problem library [37].

Instance Generation We generated 36 MGVPP and 36 MAVPP instances using four TSPLIB instances [37] 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 and their desired heading angles at the vertices were uniformly randomly generated. The minimum turn radius of the vehicles were generated according to the following procedure: for each instance, the grid size g was computed to be the maximum of the coordinates of all the vertices in the instances; now the minimum turn radius was computed using the formula $3 \cdot k \cdot g/100$ where $k = 1, \cdots, n$. 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

Table 1 MGVPP computational results

name	n	$\|R\|$	%-LB	nodes	time	time-2%	cuts
bays29	3	1	100.00	1	0.26	0.25	716
bays29	3	3	100.00	1	0.38	0.37	992
bays29	3	5	100.00	1	0.14	0.14	521
bays29	4	1	100.00	1	0.45	0.44	1108
bays29	4	3	100.00	1	0.35	0.34	834
bays29	4	5	100.00	1	0.11	0.11	498
bays29	5	1	99.92	4	0.83	0.73	1190
bays29	5	3	99.90	4	0.55	0.51	946
bays29	5	5	100.00	1	0.06	0.06	228
eil51	3	1	99.86	4	2.81	2.48	1185
eil51	3	3	99.45	10	6.03	4.74	2001
eil51	3	5	99.63	7	1.84	1.76	1415
eil51	4	1	99.56	23	4.71	2.35	1655
eil51	4	3	99.61	6	6.46	4.58	2530
eil51	4	5	99.72	5	5.05	3.99	2121
eil51	5	1	99.40	15	9.53	5.85	2699
eil51	5	3	99.63	6	7.88	6.96	3175
eil51	5	5	99.74	4	3.86	3.78	2158
eil76	3	1	99.41	23	68.34	52.74	4605
eil76	3	3	99.25	8	24.86	22.97	3926
eil76	3	5	99.28	27	28.10	16.55	3531
eil76	4	1	99.05	95	125.50	57.54	5600
eil76	4	3	99.51	24	47.06	38.59	4443
eil76	4	5	99.29	20	32.63	21.29	4624
eil76	5	1	99.10	5	201.10	99.49	7220
eil76	5	3	98.92	104	114.57	86.30	5669
eil76	5	5	100.00	1	14.01	17.65	4396
eil101	3	1	99.95	4	12.42	12.32	2697
eil101	3	3	99.70	15	50.00	57.82	4295
eil101	3	5	99.48	87	178.34	93.68	7310
eil101	4	1	99.53	91	271.73	138.51	9360
eil101	4	3	99.36	141	310.26	277.73	8448
eil101	4	5	99.48	100	279.12	122.10	9102
eil101	5	1	99.83	18	53.82	47.14	5496
eil101	5	3	99.33	1092	1168.27	286.52	16849
eil101	5	5	99.28	119	267.61	232.26	10634

chosen from the set $\{1, 3, 5\}$. The minimum length path for each pair of vertices was computed using the results of Reeds-Shepp [6] and Dubins [7] for the ground and aerial vehicles, respectively. Hence, for each TSPLIB instance, we generated 9 MGVPP and 9 MAVPP instances with all possible combinations of n

Table 2 MAVPP computational results

name	n	$\|R\|$	%-LB	nodes	time	time-2%	cuts
bays29	3	1	98.83	34	2.68	0.87	524
bays29	3	3	97.47	49	0.27	0.22	186
bays29	3	5	96.60	52	0.75	0.72	239
bays29	4	1	94.43	68	1.72	1.61	575
bays29	4	3	95.62	112	1.08	0.66	275
bays29	4	5	97.72	63	0.33	0.20	180
bays29	5	1	92.26	600	6.17	3.48	1554
bays29	5	3	97.21	45	0.69	0.49	243
bays29	5	5	99.52	2	0.05	0.03	86
eil51	3	1	99.69	17	0.66	0.32	686
eil51	3	3	97.93	1347	10.80	1.12	1660
eil51	3	5	98.52	369	3.88	4.00	1530
eil51	4	1	98.65	473	8.87	7.63	3446
eil51	4	3	98.55	106	5.97	3.16	850
eil51	4	5	98.52	266	3.95	2.54	546
eil51	5	1	97.17	1096	15.66	4.59	3065
eil51	5	3	97.77	91	4.45	0.92	811
eil51	5	5	97.92	670	7.61	0.83	1286
eil76	3	1	99.18	57	1.70	0.84	361
eil76	3	3	98.55	166	5.08	4.76	955
eil76	3	5	98.99	254	3.79	2.06	1038
eil76	4	1	97.34	2472	149.87	26.96	7075
eil76	4	3	98.22	2726	234.22	14.30	9097
eil76	4	5	99.08	543	12.40	12.20	2285
eil76	5	1	97.83	1394	60.30	25.62	3647
eil76	5	3	97.84	3684	410.46	105.29	9333
eil76	5	5	98.13	1243	41.66	26.33	3299
eil101	3	1	99.03	83	4.92	1.24	576
eil101	3	3	98.22	3368	1373.00	25.10	15479
eil101	3	5	98.82	236	26.43	10.40	2620
eil101	4	1	99.12	25	3.17	2.95	412
eil101	4	3	98.38	1428	164.66	5.81	3927
eil101	4	5	99.07	257	22.94	4.26	1913
eil101	5	1	99.51	51	14.33	6.53	305
eil101	5	3	98.31	421	95.90	6.67	5208
eil101	5	5	97.77	487	147.38	52.03	3413

and $|R_i|$ which resulted in a total for 72 instances. The Tables 1–2 summarize the computational behavior of the branch-and-cut algorithm for all the 72 instances. The column headings are defined as follows:

name: instance name;
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;
time-2%: time taken in seconds to compute a feasible solution that is within 2 % of the optimal;
cuts: total number of violated sub-tour elimination and 2-matching constraints that were added to the formulation dynamically.

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 Tables 1 and 2 indicate that the lower bound obtained at the root node of the enumeration tree is very tight, typically within 0.5 % and 1.5 % of the optimum for the MGVPP and MAVPP, respectively. Hence the proposed integer linear programming formulations for the MGVPP and MAVPP is by itself very tight. The optimal solution for the instance *eil*76 with $n = 3$ and $|R| = 3$ is shown in the Figs. 1 and 2. Though the maximum computation over all the instances for the MGVPP and MAVPP are 1168 and 1373 seconds, respectively, a feasible solution for all the instances that is within 2 % of the optimal solution can be obtained relatively fast, typically within 300 seconds for any instance of the problem with up to 100 targets. The average time taken in seconds for each instance to compute a feasible solution within 2 % of the optimal is shown in the Fig. 3. The plot indicates that even in cases where the computational time is a constraint, the algorithm can give near optimal feasible solutions within a couple of minutes. A convergence plot showing the objective values and the lower bound for the MGVPP instance *eil101-5-3* is shown in the Fig. 4; initially, for around 37,000 iterations, no feasible solutions were generated by the branch-and-cut algorithm, as illustrated by the blue curve in the figure. This behaviour varies with the instances – for many an instance a feasible solution with a very small objective value was generated much earlier in the branch-and-cut algorithm. Overall, we were able to solve all the 72 TSPLIB based instances, with the largest instance involving 101 targets, 5 vehicles and 5 functional heterogeneous targets per vehicle.

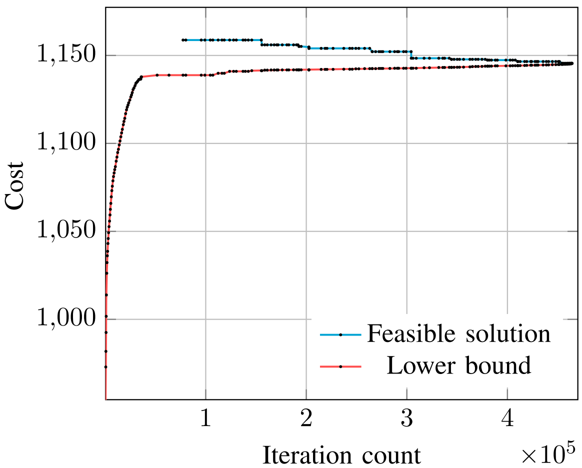

Fig. 4 Convergence plot for the MGVPP instance *eil101-5-3*. This instance was chosen since the time taken to solve this instance to optimality was the maximum among all the MGVPP instances. Iteration count is the cumulative iteration count of the algorithm solving the subproblems

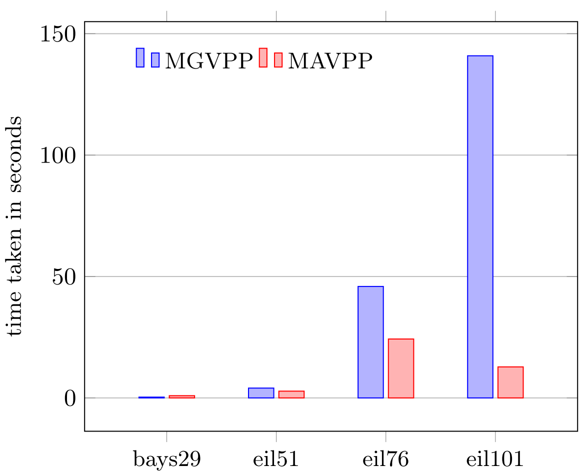

Fig. 3 Average time taken to obtain a feasible solution that is within 2 % of the optimal for the MGVPP and MAVPP

7 Conclusion

In this paper, we have presented an exact algorithm for the heterogeneous multiple depot multiple unmanned vehicle path planning problems that arises in the context of monitoring a set of targets and collecting relevant data. Two integer linear programming formulations, one for the ground vehicles and one of the aerial vehicles, including 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 balancing the distances that each vehicle travels i.e., a min-max version of the problem and developing similar algorithms for problems involving both aerial and ground vehicles working together on the same mission.

References

1. Frew, E.W., Brown, T.X.: Networking issues for small unmanned aircraft systems. J. Intell. Robot. Syst. **54**, 21–37 (2009)
2. Curry, J.A., Maslanik, J., Holland, G., Pinto, J.: Applications of aerosondes in the arctic. Bull. Amer. Meteorol. Soc. **85**(12), 1855–1861 (2004)
3. Zajkowski, E.J.T., Dunagan S: Small UAS communications mission. In: Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference. Salt Lake City (2006)
4. Levy, D., Sundar, K., Rathinam, S.: Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. Math. Prob. Eng. (2014)
5. Sundar, K., Venkatachalam, S., Rathinam, S.: Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem. In: 2016 American Control Conference (ACC), pp. 6489–6494 (2016)
6. Reeds, J., Shepp, L.: Optimal paths for a car that goes both forwards and backwards. Pac. J. Math. **145**(2), 367–393 (1990)
7. Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. Amer. J. Math. **79**(3), 497–516 (1957)
8. Benavent, E., Martínez, A.: Multi-depot Multiple TSP: A polyhedral study and computational results. Ann. Oper. Res. **207**(1), 7–25 (2013)
9. Lawler, E.L., Lenstra, J.K., Kan, A.R., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, vol. 3. New York, Wiley (1985)
10. Sundar, K., Rathinam, S.: Generalized multiple depot traveling salesmen problem–polyhedral study and exact algorithm. Comput. Oper. Res. **70**, 39–55 (2016)
11. Kara, I., Bektas, T.: Integer linear programming formulations of multiple salesman problems and its variations. Eur. J. Oper. Res. **174**(3), 1449–1458 (2006)
12. Bektas, T.: The multiple traveling salesman problem: An overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006)
13. Gavish, B., Srikanth, K.: An optimal solution method for large-scale multiple traveling salesmen problems. Oper. Res. **34**(5), 698–717 (1986)
14. Tang, Z., Ozguner, U.: Motion planning for multitarget surveillance with mobile sensor agents. IEEE Trans. Robot. **21**(5), 898–908 (2005)
15. Rathinam, S., Sengupta, R., Darbha, S.: A resource allocation algorithm for multivehicle systems with nonholonomic constraints. IEEE Trans. Autom. Sci. Eng. **4**(1), 98–104 (2007)
16. Savla, K., Frazzoli, E., Bullo, F.: Traveling salesperson problems for the dubins vehicle. IEEE Trans. Autom. Control **53**(6), 1378–1391 (2008)
17. Le Ny, J., Feron, E., Frazzoli, E.: On the dubins traveling salesman problem. IEEE Trans. Autom. Contr. **57**(1), 265–270 (2012)
18. Manyam, S.G., Rathinam, S., Darbha, S.: Computation of lower bounds for a multiple depot, multiple vehicle routing problem with motion constraints. J. Dyn. Syst. Measur. Control **137**(9), 094501 (2015)
19. Manyam, S., Rathinam, S.: On tightly bounding the dubins traveling salesmans optimum. arXiv preprint (2015)
20. Laporte, G., Nobert, Y., Taillefer, S.: Solving a family of multi-depot vehicle routing and location-routing problems. Transp. Sci. **22**(3), 161–172 (1988)
21. Baldacci, R., Mingozzi, A.: A unified exact method for solving different classes of vehicle routing problems. Mathem. Program. **120**(2), 347–380 (2009)
22. Taillard, É.D.: A heuristic column generation method for the heterogeneous fleet VRP. RAIRO-Oper. Res. **33**(01), 1–14 (1999)
23. Baldacci, R., Battarra, M., Vigo, D.: Routing a heterogeneous fleet of vehicles, in The vehicle routing problem: Latest advances and new challenges. Springer, 3–27 (2008)
24. Nag, B., Golden, B.L., Assad, A.: Vehicle routing with site dependencies. Veh. Rout. Methods Stud., 149–159 (1988)
25. Chao, I.-M., Golden, B.L., Wasil, E.A.: A new algorithm for the site-dependent vehicle routing problem, in Advances in computational and stochastic optimization, logic programming, and heuristic search. Springer, 301–312 (1998)
26. Doshi, R., Yadlapalli, S., Rathinam, S., Darbha, S.: Approximation algorithms and heuristics for a 2-depot, heterogeneous hamiltonian path problem. Int. J. Robust Nonlin. Control **21**(12), 1434–1451 (2011)
27. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization, vol. 18. Wiley, New York (1988)
28. Labbé, M., Laporte, G., Martín, I.R., González, J.J.S.: The ring star problem: Polyhedral analysis and exact algorithm. Networks **43**(3), 177–189 (2004)
29. Sussmann, H.J., Tang, G.: Shortest paths for the reeds-shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control. Rutgers Center Syst. Control Tech. Report **10**, 1–71 (1991)
30. Toth, P., Vigo, D.: The Vehicle Routing Problem. Siam (2001)
31. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: A computational study. Princeton University Press (2011)
32. Sundar, K., Rathinam, S.: An exact algorithm for a heterogeneous, multiple depot, multiple traveling salesman problem. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 366–371. IEEE (2015)

33. Achterberg, T., Koch, T., Martin, A.: Branching rules revisited. Oper. Res. Lett. **33**(1), 42–54 (2005)
34. Padberg, M.W., Rao, M.R.: Odd minimum cut-sets and b-matchings. Math. Oper. Res. **7**(1), 67–80 (1982)
35. Fischetti, M., Salazar Gonzalez, J.J., Toth, P.: A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper. Res. **45**(3), 378–394 (1997)
36. Sundar, K., Rathinam, S.: Multiple depot ring star problem: A polyhedral study and an exact algorithm. J. Global Optim., 1–25 (2016)
37. Reinelt, G.: TSPLIB - a traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991)

Kaarthik Sundar is currently a post-doctoral researcher at Center for Non-Linear Studies in Los Alamos National Laboratory, Los Alamos. He received his M.S. in Electrical Engineering and Ph.D. in Mechanical Engineering, both from Texas A&M University, in 2012 and 2016, respectively. His research interests include, combinatorial optimization and optimal control with applications to unmanned vehicles, infrastructure systems, and scheduling.

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