2014 European Control Conference (ECC)
June 24-27, 2014. Strasbourg, France

Mixed-Integer SOCP Formulation of the Path Planning Problem for Heterogeneous Multi-Vehicle Systems

Martin Klaučo, Slavomír Blažek, Michal Kvasnica, Miroslav Fikar

Abstract— **We consider the path planning problem for heterogeneous multi-vehicle systems. In such a setup an agile vehicle, which can move quickly but has limited operating range, is carried by a carrier vehicle that moves slowly but has large range. The objective is to devise an optimal path for the multi-vehicle system such that all desired points are visited as quickly as possible, while respecting all physical constraints. We show how to translate the mixed-integer nonlinear formulation of such a problem into a mixed-integer second-order cone problem that can be solved much more efficiently. The translation process employs basic concepts of propositional logic and is not conservative. Efficacy of the proposed formulation is demonstrated on a large case study.**

I. Introduction

Path planning problems are of imminent importance in many applications, such as in parcel delivery, mail collection, waste management, or delivery of goods, to name just a few. The main objective is to devise an optimal path which minimizes the cost of operation through minimizing the distance traveled. If only a single vehicle is considered, the problem is fairly well studied and usually boils down to solving a traveling salesman problem (TSP), see e.g. [9].

More recently, significant attention started to be devoted for optimal path planning for multiple vehicles, often in the heterogenous arrangement. Such a heterogenous vehicle in fact consists of multiple parts that can move individually, but are somehow coupled. Examples include, but are not limited to, ships that carry a helicopter for delivery of goods to off-shore oil rigs, transportation systems that involve a train and a truck or a car (the train carries the truck/car to a major city where the vehicle detaches and performs goods deliver/pickup before reconnecting with the train at some other place), or mail delivery systems where a human uses a car to travel long distances, but has to walk when servicing a pedestrian precinct. Achieving path planning for such heterogenous vehicles is challenging, and is often approached by heuristic approaches that do not guarantee optimality.

One such application is presented in [11] where the authors tackle the network planning and vehicle routing problems for the Austrian parcel service. Key issues addressed in the reference include favorable placement of hubs and depots, allocation of customers to particular service areas and, most importantly, determination of suitable transportation routes. The task was formulated as a mixed-integer optimization problem, however in order to arrive at a tractable schemes, the authors had to resort to several restrictive assumptions, approximations, and heuristic rules that undermine optimality of the solution.

The vehicle routing problem for a heterogenous vehicle system was also studied in [10] where the waste collection and waste unloading for the city of Hanoi was considered. The reference employs a heuristic method to find suitable routes between a predefined set of pick-up and drop-off locations.

An heuristic approach was also employed by [6] to devise routing for distribution of soft drinks in the Coca-Cola company. The authors employed a constructive procedure where vehicles with larger capacities are first used to distribute goods at larger distances before dispatching the cargo to several smaller vehicles that have smaller fuel consumption until all customers are served, and the total cost of transportation is lowered. The same reference also discusses similar applications in the maritime industry. A similar application is also discussed in [3], which also elaborates on the downsides of heuristic approaches.

An approach to obtaining a truly optimal routing of a heterogenous vehicle was recently proposed by [4]. There the authors consider a system composed of two vehicles: a carrier ship with a low maximal speed but large (in fact, unlimited) range, that carries an agile vehicle (e.g. a helicopter) which moves quickly but has limited range. Moreover, the carrier provides refuelling and therefore the helicopter can take off multiple times. The objective is to devise an optimal plan of takeoff and landing points such that a set of points q_i, $i = 1, \dots, n$ can be visited in minimum time. Assuming that the ordering of points to be visited is fixed, the authors have proposed a mixed-integer nonlinear (MI-NLP) formulation of the path planning problem. The crucial practical limitation of such an approach is the induced computational complexity. Specifically, the exponential complexity of MI-NLP problems allows to tackle only scenarios with low number of points. In particular, the reference mentions a case with $n = 7$ points as the largest practical scenario. To address more complex cases, the authors proposed a set of heuristic rules with the inherent downside of potentially arriving at a suboptimal solution.

In this paper we improve upon the method of [4] by showing that the particular MI-NLP problem can in fact be formulated, in a non-conservative fashion, as a mixed-integer problem with second-order cone constraints (MI-SOCP). Although still a combinatorial problem, the MI-SOCP formulation scales much better with increasing problem size and allows to provide optimal path planning for scenarios

The authors are with Slovak University of Technology in Bratislava, Slovakia; {martin.klauco, slavomir.blazek, michal.kvasnica, miroslav.fikar}@stuba.sk

with hundreds of points. This is due to the fact that once the integer components are fixed during a branch-and-bound or a branch-and-cut procedure, the local subproblems are always convex. To achieve such a conversion we employ basic concepts of propositional logic [12] that are well established in the field of optimization-based control of hybrid systems [1]. The rest of this paper is organized as follows. First we present the formulation of the path-planning problem for a heterogenous vehicle in Section II. In the subsequent section we review the MI-NLP formulation of [4] and provide interpretation of individual decision variables. The translation of the MI-NLP problem into a MI-SOCP problem is then presented in Section IV, before demonstrating the efficacy on several examples in Section V. Conclusions and outline of future work is provided in Section VI.

II. Problem Statement

We consider a vehicle system that consists of a carrier vehicle with low maximal speed and large range, and an agile vehicle (e.g. a helicopter) that moves quickly, but has limited range. In particular, the carrier is assumed to move with a constant velocity v_c (the suffix c denotes the carrier) and has unlimited range. The agile vehicle either rests on the carrier, or is airborne. When airborne, the helicopter is assumed to travel at a constant velocity v_h and its range on one fuel load is limited by $t_\text{h,max}$, expressed as the amount of time the helicopter can be airborne without refuelling. Whenever the helicopter rests on the carrier, refuelling to maximum capacity takes place. We assume that such a refuelling is instantaneous.

The heterogeneous vehicle starts at the point q_s and is required to visit each point $q_1, \ldots, q_n$ exactly once *in the order* from q_1 to q_n, after which the fleet proceeds to the final point q_f. The objective is to minimize the mission completion time while taking into account constraints on maximal range of the agile vehicle. We furthermore assume that each point q_i is visited by the agile part of the heterogeneous vehicle (i.e., by the helicopter). Such a requirement frequently occurs in practice e.g. when the helicopter is used to inspect conditions of wind towers or of off-shore drilling towers, or to rescue drowning people.

Formally, we are interested in solving the following problem:

Problem 2.1: Given are: starting point $q_\text{s} \in \mathbb{R}^2$, final point $q_\text{f} \in \mathbb{R}^2$, intermediate points $q_i \in \mathbb{R}^2$, $i = 1, \ldots, n$ to visit, carrier's speed v_c, helicopter's speed v_h, and helicopter's range $t_\text{h,max}$. Determine:

- index sets $\mathcal{I}_1, \ldots, \mathcal{I}_m$ with $\mathcal{I}_i \subseteq \{1, \ldots, n\}$ denoting which points q_i the helicopter visits during one flyover;
- set of takeoff and landing points $\{\tau_i, \ell_j\}$ such that the helicopter lifts off from the carrier at position τ_i, visits points $q_i, \ldots, q_j$ (indexed by $\mathcal{I}_i$), before landing at the carrier at position ℓ_j,

such that:

- the mission completion time is minimized,

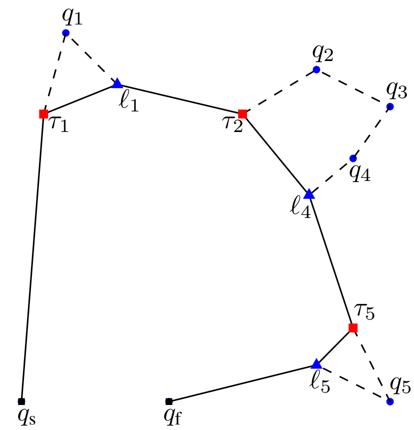

Fig. 1. Illustration of an optimal path for the heterogenous vehicle. τ_i and ℓ_j denote the takeoff and landing points for the helicopter, respectively. Solid line shows trajectory of the carrier, dashed lines visualize the path of the helicopter that needs to visit points $q_1, \ldots, q_5$ in a consecutive order. Due to a restricted range, however, the helicopter needs to perform intemediate stops for refuelling.

- for each takeoff-landing phase, the associated index set $\mathcal{I}_i$ contains only indices of points q_i which are in the helicopter's range,
- each point q_i is visited exactly once, i.e., the index sets $\mathcal{I}_i$ are mutually exclusive $\mathcal{I}_j \cap \mathcal{I}_k = \emptyset$ for all $j \neq k$, while their union satisfies $\bigcup_i \mathcal{I}_i = \{1, \ldots, n\}$.

□

Note that while the helicopter is airborne and visiting points $q_i, \ldots, q_j$, the carrier follows the straight path from τ_i to ℓ_j. The minimal number of takeoff-landing sequences is $m = 1$ (when the helicopter's range allows to visits all points $q_1, \ldots, q_n$ in one shot), while the maximum is $m = n$.

To give the reader a flavour of what the individual variables represent, consider the case depicted in Fig. 1. Here, the task is to visit 5 points $q_1, \ldots, q_5$, starting at q_s and finishing at q_f. In the particular scenario depicted in Fig. 1 the carrier follows the route $q_\text{s} \to \tau_1 \to \ell_1 \to \tau_2 \to \ell_4 \to \tau_5 \to \ell_5 \to q_\text{f}$. When at position τ_1, the helicopter lifts off and visits q_1 alone (hence $\mathcal{I}_1 = \{1\}$) before returning to the carrier at point ℓ_1 for refuelling. From here the two vehicles continue together until the point τ_2 is reached. Here, the helicopter lifts off again and, this time, visits points q_2, q_3, q_4, which corresponds to $\mathcal{I}_2 = \{2, 3, 4\}$. Meanwhile, the carrier continues directly to ℓ_4, where it meets with the helicopter. The platoon then continues to point τ_5 where the helicopter separates again to visit q_5 (with $\mathcal{I}_5 = \{5\}$), before returning to the carrier, which in the meantime travelled to ℓ_5. From there the heterogenous vehicle returns to the port located at q_f.

III. Mixed-Integer NLP Formulation of Problem 2.1

In this section we review the mixed-integer nonlinear formulation of Problem 2.1 as suggested by [4]. Let us consider a binary matrix α with m rows corresponding to the airborne phases and to the sets $\mathcal{I}_1, \ldots, \mathcal{I}_m$. Then $\alpha_{i,j} = 1$ is interpreted as follows: during the k-the airborne phase, the helicopter sequentially flies over points $q_i, \ldots, q_j$ without any intermediate landings. In the example shown in Fig. 1 the matrix would take the following form:

$$\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

To guarantee that each point q_i is visited exactly once, the following constraints must be added:

$$\sum_{i=1}^{k} \sum_{j=k}^{n} \alpha_{i,j} = 1, \quad k = 1, \ldots, n. \tag{2}$$

It is easy to verify that α of (1) satisfies (2), and corresponds to three airborne phases: first one visits only q_1 (and gives $\mathcal{I}_1 = \{1\}$), the second takeoff covers $q_2, \ldots, q_4$ (which corresponds to $\mathcal{I}_2 = \{2, 3, 4\}$), and in the last run the helicopter visits q_5 alone with $\mathcal{I}_5 = \{5\}$. The advantage of the introduced semantics for α, enforced by (2), is that at most n elements of α can be equal to one. This allows to somehow mitigate the exponential complexity of the resulting mixed-integer formulation.

To each takeoff-landing sequence we furthermore associate the flyover time $f_{i,j} \geq 0$ as the time required for the helicopter to travel from the corresponding takeoff point τ_i via $q_i, \ldots, q_j$ to the touchdown point ℓ_j. The time spent airborne is restricted by helicopter's range by

$$\alpha_{i,j} f_{i,j} \leq t_{\text{h,max}}. \tag{3}$$

The multiplication by $\alpha_{i,j}$ with $\alpha_{i,j} \in \{0, 1\}$ provides that the constraint will only become active if $\alpha_{i,j} = 1$, which corresponds to selection of $q_i, \ldots, q_j$ as flyover points. If $\alpha_{i,j} = 0$, the constraint is inactive. Moreover, the flyover time must be selected such that the carrier can travel from τ_i to ℓ_j for rendezvous. Assuming the carrier moves on a straight line at a fixed speed v_c, the following constraint must be satisfied:

$$\alpha_{i,j} \|\tau_i - \ell_j\| \leq v_\text{c} f_{i,j}. \tag{4}$$

Otherwise the helicopter would arrive to the rendezvous point ℓ_j before the carrier and could thus run out of fuel while waiting. Finally, the flyover time is bounded from below by the time it takes the helicopter to travel the total distance from τ_i via $q_i, \ldots, q_j$ to ℓ_j at a fixed speed v_h, i.e.,

$$\alpha_{i,j} \left(\|\tau_i - q_i\| + d_{i,j} + \|q_j - \ell_j\| \right) \leq v_\text{h} f_{i,j}, \tag{5}$$

where $d_{i,j}$ denotes the total distance of the piecewise-linear path of minimal length connecting points $q_1, \ldots, q_j$, i.e.,

$$d_{i,j} = \sum_{k=i}^{j-1} \|q_k - q_{k+1}\|. \tag{6}$$

Note that the matrix $d \in \mathbb{R}^{n \times n}$ with entries $d_{i,j}$ as in (6) can be pre-computed off-line, and is treated as a matrix of constants since positions of points q_i are fixed a-priori.

The total mission time t_m to be minimized is composed of four parts:

1) the time the fleet travels from the starting point q_s to the first takeoff point τ_1, represented by $1/v_\text{c} \|q_\text{s} - \tau_1\|$,
2) time consumed by the carrier alone to travel from one takeoff point to the next landing point, given by $\sum_{i=1}^{n} \sum_{j=i}^{n} f_{i,j}$,
3) time the carrier and the helicopter travel together from the previous landing point to the next takeoff point, i.e., $\sum_{i=1}^{n} \sum_{j=i}^{n-1} s_{i,j}$ where $s_{i,j} \geq 0$ relates to α via
$$\alpha_{i,j} \|\ell_j - \tau_{j+1}\| \leq v_\text{c} s_{i,j}, \tag{7}$$
4) time of fleet travel from the last landing point to the final destination at q_f, i.e., $1/v_\text{c} \|\ell_n - q_\text{f}\|$.

Hence the mission time is given by

$$\begin{aligned} t_\text{m} =& 1/v_\text{c} \left(\|q_\text{s} - \tau_1\| + \|\ell_n - q_\text{f}\| \right) + \\ &+ \sum_{i=1}^{n} \sum_{j=i}^{n} f_{i,j} + \sum_{i=1}^{n} \sum_{j=i}^{n-1} s_{i,j}. \end{aligned} \tag{8}$$

Then a solution to Problem 2.1 can be obtained by solving an optimization problem of the form

$$\min t_\text{m} \text{ s.t. } (2) - (7), \tag{9}$$

with decision variables $\alpha \in \{0, 1\}^{n \times n}$, $f \in \mathbb{R}^{n \times n}$, $f_{i,j} \geq 0$, $s \in \mathbb{R}^{n \times n}$, $s_{i,j} \geq 0$, $\tau \in \mathbb{R}^{2 \times n}$, and $\ell \in \mathbb{R}^{2 \times n}$. Note that each column of τ and of ℓ denotes coordinates of takeoff and landing points in the 2-dimensional Euclidian space. It is important to notice that, since $f_{i,j}$ and $s_{i,j}$ are minimized by (8), if $\alpha_{i,j} = 0$ is an optimal solution to (9), then $f_{i,j} = 0$ and $s_{i,j} = 0$ are feasible optimal choices. This follows from (4), (5), and (7) that result in $f_{i,j} \geq 0$ (and $s_{i,j} \geq 0$) for $\alpha_{i,j} = 0$.

Problem (9) is a mixed-integer nonlinear programming problem. The integer component is due to presence of binary decision variables α, and the nonlinearity is due to products between $\alpha_{i,j}$ and continuous decision variables in (3), (4), (5), and (7).

Remark 3.1: Once the optimal solution to (9) is obtained, the equivalence between α and index sets $\mathcal{I}$ from Problem 2.1 can be recovered as follows: let i be the index of a row of α that contains at least one non-zero entry. Then $\mathcal{I}_i = \{i, \ldots, j\}$, where j is such that $\alpha_{i,j} = 1$. □

IV. Mixed-Integer SOCP Formulation of Problem 2.1

The main limitation of the mixed-integer nonlinear programming (MI-NLP) formulation of the optimization problem (9) stems from its computational complexity. Specifically, the authors in [4] demonstrated that the problem is solvable, in reasonable time, just for a small number of points q_i. The largest scenario considered in the reference contained mere seven points. In this section we show how to *equivalently* reformulate the MI-NLP (9) as a mixed-integer problem with second-order cone constraints (MI-SOCP), that can be solved efficiently for *hundreds* of points.

We start by reminding that the non-trivial part of (9) are nonlinear constraints where various decision variables multiply each other. However, a closer look at (3)−(5) and (7) reveals that such nonlinear terms only involve multiplication between a binary variable $\alpha_{i,j}$ and a convex function. Take (3) as an example. The constraint can be equivalently written as a logic relation of the form

$$(\alpha_{i,j} = 1) \Rightarrow f_{i,j} \leq t_{\text{h,max}}. \tag{10}$$

Note that regardless of value of $\alpha_{i,j}$, the flyover time $f_{i,j}$ is assumed to be lower-bounded by $f_{i,j} \geq 0$ for any combination of i and j. Similarly, (4) can be written as

$$(\alpha_{i,j} = 1) \Rightarrow \|\tau_i - \ell_j\| \leq v_{\text{c}} f_{i,j}, \tag{11}$$

which introduces a strictly positive lower bound on $f_{i,j}$ if $\alpha_{i,j} = 1$. Note that for $\alpha_{i,j} = 0$ constraint (4) yields $0 \leq v_{\text{c}} f_{i,j}$, which is again equivalent to the lower bound $f_{i,j} \geq 0$. Continuing along the same lines, (5) is equivalent to

$$(\alpha_{i,j} = 1) \Rightarrow (\|\tau_i - q_i\| + d_{i,j} + \|q_j - \ell_j\|) \leq v_{\text{h}} f_{i,j} \tag{12}$$

and (7) can be written as

$$(\alpha_{i,j} = 1) \Rightarrow \|\ell_j - \tau_{j+1}\| \leq v_{\text{c}} s_{i,j} \tag{13}$$

with the sailing time being lower-bounded by $s_{i,j} \geq 0$.

The advantage of rewriting (3)−(5) and (7) as a set of implication rules in (10)−(13) is that they can be further simplified into a set of constraints that are convex in decision variables $\alpha_{i,j}$, $f_{i,j}$, $s_{i,j}$, τ_i and ℓ_j using basic rules of propositional logic [12].

Lemma 4.1: Consider a binary variable $\delta \in \{0, 1\}$, continuous variables $x \in \mathbb{R}^m$, and an arbitrary function $g : \mathbb{R}^m \to \mathbb{R}$. Then

$$(\delta = 1) \Rightarrow g(x) \leq 0 \tag{14}$$

iff

$$g(x) \leq M(1 - \delta), \tag{15}$$

is satisfied for some constant M.

Proof: We start by noting that, given two logic statements Y_1 and Y_2,

$$(Y_1 \Rightarrow Y_2) \Leftrightarrow (\overline{Y}_1 \vee Y_2), \tag{16}$$

where $\overline{Y}_1$ is the negation of Y_1 and $\vee$ is the logic "or" operator. Moreover, it is easy to verify that

$$([\delta = 1] \vee [g(x) \leq 0]) \Leftrightarrow (g(x) \leq M\delta). \tag{17}$$

Then (15) follows directly from (16) and (17) by considering the negation of δ as $\overline{\delta} = 1 - \delta$ (recall that δ is a binary variable). ■

Applying Lemma 4.1 to (10) allows to rewrite the logic implication as

$$f_{i,j} - t_{\text{h,max}} \leq M(1 - \alpha_{i,j}), \tag{18}$$

with the lower bound $f_{i,j} \geq 0$. Note that (18) is linear in the continuous decision variables $f_{i,j}$ and in the binary variables $\alpha_{i,j}$. Similarly, (11)−(13) can be converted into

$$\|\tau_i - \ell_j\| - v_{\text{c}} f_{i,j} \leq M(1 - \alpha_{i,j}), \tag{19a}$$

$$(\|\tau_i - q_i\| + d_{i,j} + \|q_j - \ell_j\|) - v_{\text{h}} f_{i,j} \leq M(1 - \alpha_{i,j}) \tag{19b}$$

$$\|\ell_j - \tau_{j+1}\| - v_{\text{c}} s_{i,j} \leq M(1 - \alpha_{i,j}). \tag{19c}$$

Note that all constraints in (19) are convex in corresponding decision variables. In particular, due to employing Euclidian norms, (19) can be written as a set of *second-order cone* constraints, see [2].

Search for optimal takeoff-landing sequences from (9) can thus be equivalently formulated as

$$\min_{\alpha, f, s, \tau, \ell} \; t_{\text{m}} \tag{20a}$$

$$\text{s.t. } (18) - (19), \tag{20b}$$

$$f_{i,j} \geq 0, \; s_{i,j} \geq 0, \; \alpha_{i,j} \in \{0, 1\}, \tag{20c}$$

with t_{m} as in (8) and the constraints imposed for each $i, j \in \{0, \ldots, n\}$. Problem (20) is a mixed-integer second-order cone program that can be solved e.g. by the GUROBI solver [5], which employs the branch-and-cut method to efficiently eliminate infeasible combinations of binary variables, thus avoiding exploration of an exponential number of cases.

It is important to note that (20) is a non-conservative version of (9). If $\alpha_{i,j} = 1$, then the constraints of (20) are the same as in (9), which can be seen from (18)−(19). If $\alpha_{i,j} = 0$, then the corresponding optimal values of $f_{i,j}$ and $s_{i,j}$ will be zero because they are minimized in (8) and since they are lower-bounded by zero in (20c).

To solve (20) as efficiently as possible, the value of M has to be chosen as low as possible. As noted e.g. in [7], non-tight values of the M constants can easily increase the computational time of (20) by several order of magnitudes. Therefore it is important to derive the tightest possible values of M employed in (18)−(19). As noted by [12], the tightest value of M that can be employed in (15) is given by

$$M = \max_{x \in \Omega} g(x), \tag{21}$$

where Ω is the (bounded) domain of the function $g(\cdot)$. In (18), such an M is trivially given as $M = t_{\text{h,max}}$. In (19a) the lowest value of M is

$$M = \max_{\tau_i, \ell_j, f_{i,j}} (\|\tau_i - \ell_j\| - v_{\text{c}} f_{i,j}). \tag{22}$$

Since the function $\|\tau_i - \ell_j\| - v_{\text{c}} f_{i,j}$ is convex in τ_i, ℓ_j, and in $f_{i,j}$, the maximum is attained at one of the vertices of the corresponding domain $\Omega = \Omega_\tau \times \Omega_\ell \times \Omega_f$. Here, Ω_τ and Ω_ℓ are subsets of $\mathbb{R}^2$ that delimit the search space for takeoff and landing points, respectively. In practice these sets can be obtained as the smallest box that contains the points q_i, $i = 1, \ldots, n$ to be visited. Such a box can be easily computed as $\Omega_\tau = \Omega_\ell = \{x \mid \min_i q_i \leq x \leq \max_i q_i\}$, where the minima and maxima are taken element-wise over coordinates of points q_i. Finally, $\Omega_f = \{f \mid 0 \leq f \leq t_{\text{h,max}}\}$,

TABLE I

LOCATIONS OF POINTS $q_1, \dots, q_{10}$ FOR THE EXAMPLE IN SECTION V-A. ALL ENTRIES ARE IN KILOMETERS.

i	1	2	3	4	5	6	7	8	9	10
x_i	0	0	1	46	48	46	50	50	20	30
y_i	40	50	49	50	48	46	48	35	5	0

which follows from (3). Therefore the tightest M in (19a) can be computed from (22) by evaluating $\|\tau_i - \ell_j\| - v_c f_{i,j}$ at each vertex of $\Omega_\tau \times \Omega_\ell \times \Omega_f$, followed by retaining the maximal value. Tight values of M in (19b) and in (19c) can be obtained accordingly.

V. EXAMPLES

A. Illustrative Example with 10 Points

First we consider a case with 10 points q_i, $i = 1, \dots, 10$, whose coordinates in the 2-dimensional Euclidian plane are provided in Table I. The heterogenous vehicle consists of a carrier that travels at a constant speed $v_c = 18$ km/h and has an unlimited range. The carrier carries a helicopter whose speed is $v_h = 90$ km/h, but has limited time of operation $t_{\text{h,max}} = 21$ min. The fleet starts at point $q_s = [0,\ 0]^T$ and is required to finish at $q_f = [50,\ 0]^T$.

We have formulated the search for optimal takeoff and landing points τ_i, ℓ_j (which completely characterize the path of the carrier), along with the matrix α (that determines which points the helicopter will visit in one flyover) as a mixed-integer SOCP problem (20). The problem was formulated using YALMIP [8], which automatically determines optimal values of M in (18)−(19). The resulting MI-SOCP was then solved by GUROBI on a 2.8 GHz machine with 16 GB of memory.

Optimal solution to (20) which, according to the discussion in Section IV is equivalent to the solution of (9), was calculated in mere 2.2 seconds. The optimal path of the carrier and of the helicopter, along with location of optimal takeoff and landing points, is shown in Fig. 2.

In particular, with the data in Table I, the optimal path consists of 7 takeoff-landing sequences. In the first one, the helicopter takes off from the carrier at position τ_1, visits q_1 alone and returns for refuelling at position ℓ_1. The second airborne phase covers points q_2 and q_3. The fleet then continues to τ_4, where the helicopter separates to visit points q_4, q_5, and q_6, before landing at the carrier at position ℓ_6. In the next part, point q_7 is visited. Notice that it's not possible for the helicopter to travel from q_4 all the way to q_7 and still land safely due to limits on its maximal operational range. In the final three airborne phases, the helicopter visits, individually, points q_8, q_9, q_{10}, making a refuelling stop after each visited point. Finally, the heterogenous vehicle proceeds to the final destination q_f. The minimum mission completion time as calculated by solving (20) was $t_m^\star = 6.248$ hours, which corresponds to the total traveled distance of 112.464 kilometers. Optimal values of corresponding takeoff and landing points are reported in Table II. The associated binary matrix $\alpha \in \{0,1\}^{10\times 10}$ contained 7 non-zero entries (which

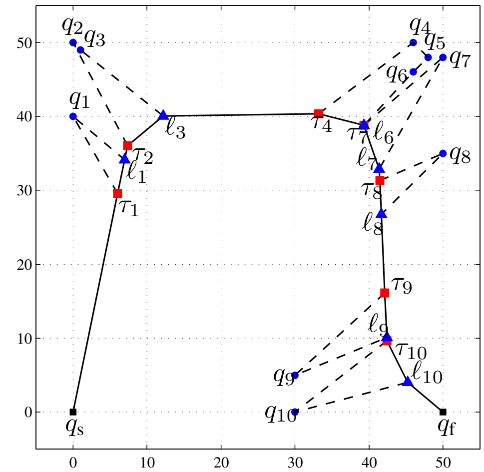

Fig. 2. Optimal path profile for the example in Section V-A. Solid line shows the path of the carrier, dashed line represents the trajectory of the helicopter. q_i are the points to be visited, τ_i are takeoff points and ℓ_j represent landings for refuelling. Both axis are in kilometers.

TABLE II

POSITIONS OF OPTIMAL TAKEOFF AND LANDING POINTS FOR THE EXAMPLE IN SECTION V-A. ALL ENTRIES ARE IN KILOMETERS.

Point	x-coordinate	y-coordinate
τ_1	6.0269	29.5898
ℓ_1	6.9511	34.1272
τ_2	7.3433	36.0529
ℓ_3	12.1979	40.0683
τ_4	33.2130	40.3264
ℓ_6	39.3277	38.8097
τ_7	39.3277	38.8097
ℓ_7	41.3947	32.8585
τ_8	41.4646	31.3118
ℓ_8	41.6700	26.7668
τ_9	42.1510	16.1206
ℓ_9	42.4233	10.0939
τ_{10}	42.4431	9.6556
ℓ_{10}	45.2319	4.0065

corresponds to the 7 airborne phases). Specifically, the non-zero entries were at $\alpha_{1,1}$, $\alpha_{2,3}$, $\alpha_{4,6}$, $\alpha_{7,7}$, $\alpha_{8,8}$, $\alpha_{9,9}$, and at $\alpha_{10,10}$. These values correspond to index sets of points visited during one flyover given by $\mathcal{I}_1 = \{1\}$, $\mathcal{I}_2 = \{2, 3\}$, $\mathcal{I}_3 = \{4, 5, 6\}$, $\mathcal{I}_4 = \{7\}$, $\mathcal{I}_5 = \{8\}$, $\mathcal{I}_6 = \{9\}$, $\mathcal{I}_7 = \{10\}$. These sets, together with the associated takeoff and landing points τ_i, ℓ_j, represent the complete optimal navigation plan for the heterogeneous vehicle.

B. Complexity Analysis

Next we have analysed how the mixed-integer SOCP formulation (20) scales with increasing number of flyover points q_i. To perform the analysis, we have randomly distributed n points for $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ in the box-shaped domain with sides $[0, \min\{50, 5n\}]$ km. For each n we have generated three sets of points randomly distributed in the box. Subsequently for each scenario we have devised the optimal navigation plan by solving (20) and measured the total computation time.

Obtained results are shown graphically in Fig. 3. The solid line shows the average computation time for each n (the

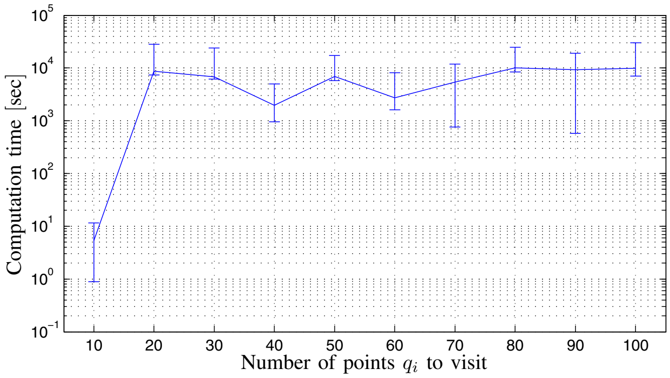

Fig. 3. Time required to obtain an optimal solution to (20) as a function of n, the number of points q_i to visit. The error bars represent the minimal and maximal computational times for each of the 3 random scenarios considered for each n. The y axis is in logarithmic scale.

number of points q_i to visit). The error bars represent the spread of computation time for individual random scenarios. As can be observed, the mixed-integer SOCP formulation in (20) scales rather well with increasing number of points, despite of the theoretical worst-case exponential complexity of the problem. This demonstrates good performance of state-off-the-art solvers (represented by GUROBI) when provided with an efficient problem formulation. Specifically, even for 100 points we were able to obtain the optimal solution to Problem 2.1 in about 3 hours of computation. This is in a stark contrast to the method of [4], which is based on the MI-NLP formulation (9), and where the reported computation time exceeds one hour already for 7 points.

VI. Conclusion

We have proposed an alternative formulation of the path planning problem for heterogenous vehicles. By exploiting the fact that nonlinearities in constraints are only due to products between a binary variable and a convex function we have shown how to employ propositional logic to simplify such constraints into a set of mixed-integer second-order cone constraints. The resulting mixed-integer SOCP problem (20) can then be solved much more efficiently compared to its mixed-integer nonlinear counterpart (9). By means of a large case study we have demonstrated that complexity of the MI-SOCP formulation scales favourably with increasing problem size. In particular, while the MI-NLP problem could be solved for only up to 7 points (as reported by [4]), our approach allows to calculate optimal navigation plan for hundreds of points.

The outstanding limitation of our approach (shared with the method of [4]) is that we assume that the points q_i are visited in a consecutive order from q_1 to q_n. While this is often a realistic requirement, there are also relevant applications where the ordering of points is not fixed, but instead should also be optimized. Incorporating such a requirement into (20) would require letting even the indices i and j to be optimization variables. Moreover, one would require such variables to be integers. One option to cope with such an extension would be to encode each integer in the range from 1 to n in the binary encoding (which would require $\lceil \log_2 n \rceil$ binaries for each index), or using unary encoding (assigning n binaries to each integer). Needless to say, doing so would tremendously increase complexity of the optimization. An alternative way would be to combine the path-planning problem (20) with the traveling salesman (TSP) formulations. Since TSP problems can be formulated as mixed-integer linear programs [9], in our future research we plan to look at how to combine the two formulations as to achieve path planning where the ordering of points to be visited is optimized.

Acknowledgments

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11 and the contribution of the Slovak Research and Development Agency under the project APVV 0551-11.

References

[1] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] K. Fagerholt. Optimal fleet design in a ship routing problem. *International Transactions in Operational Research*, 6(5):453–464, 1999.

[4] E. Garone, J.-F. Determe, and R. Naldi. A travelling salesman problem for a class of heterogeneous multi-vehicle systems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1166–1171, 2012.

[5] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2013.

[6] A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Lokketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041 – 2061, 2010.

[7] M. Kvasnica. *Efficient Software Tools for Control and Analysis of Hybrid Systems*. PhD thesis, ETH Zurich, ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland, 20.2.2008 2008.

[8] J. Löfberg. YALMIP, 2004. Available from http://users.isy.liu.se/johanl/yalmip/.

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

[10] D.V. Tung and A. Pinnoi. Vehicle routing–scheduling for waste collection in Hanoi. *European Journal of Operational Research*, 125(3):449–468, 2000.

[11] M. Wasner and G. Zäpfel. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, 90(3):403 – 419, 2004.

[12] H.P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, Third Edition, 1993.