Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 2, 297–308
DOI: 10.1515/amcs-2016-0021

AN OPTIMAL PATH PLANNING PROBLEM FOR HETEROGENEOUS MULTI–VEHICLE SYSTEMS

MARTIN KLAUČO [a,*], SLAVOMÍR BLAŽEK [a], MICHAL KVASNICA [a]

[a] Institute of Information Engineering, Automation and Mathematics
Slovak University of Technology in Bratislava, Radlinského 9, Bratislava, Slovakia
e-mail: {martin.klauco,slavomir.blazek,michal.kvasnica}@stuba.sk

A path planning problem for a heterogeneous vehicle is considered. Such a vehicle consists of two parts which have the ability to move individually, but one of them has a shorter range and is therefore required to keep in a close distance to the main vehicle. The objective is to devise an optimal path of minimal length under the condition that at least one part of the heterogeneous system visits all desired waypoints exactly once. Two versions of the problem are considered. One assumes that the order in which the waypoints are visited is known *a priori*. In such a case we show that the optimal path can be found by solving a mixed-integer second-order cone problem. The second version assumes that the order in which the waypoints are visited is not known *a priori*, but can be optimized so as to shorten the length of the path. Two approaches to solve this problem are presented and evaluated with respect to computational complexity.

Keywords: path planning, multi-vehicle system, mixed-integer programming.

1. Introduction

Vehicle routing problems have become widely investigated in recent days. The main interest in these problems stems from the need for cost reduction of entire logistic operations. Such cost reduction can be achieved by optimizing the trajectories of vehicles performing services, thus minimizing mainly the amount of fuel expenditures. Another benefit of such optimization is a decreased travelling time, so a logistic company can perform more deliveries during the same operational time. Routing problems for single vehicle systems have been widely studied. Usually, when an optimal solution needs to be found for such problems, the travelling salesman problem (TSP) is often considered, and it is solved as a mixed-integer problem (Applegate, 2006; Miller *et al*., 1960).

Recently, solving path planning problems for a single vehicle system has become insufficient. Many companies operate several vehicles which need to be coordinated. In many applications, these vehicles are arranged in a heterogeneous fashion. This heterogeneous vehicle consists of several parts which can be detached from the main vehicle and perform tasks separately. After performing individual tasks, these vehicles must return back to the main vehicle for refuelling or for resupplying. Many such applications can be found in the work of Hoff *et al*. (2010). There the authors mention the strategy of food delivery in island countries, where a ship travels with a supply car on board. This supply car, after disembarking from the ship, makes routes on the island delivering goods, and then returns back to the ship for resupplying. The referenced survey also describes soft drink distribution by Coca-Cola. This company employs vehicles with large capacities to deliver goods over long distances before using small vehicles to deliver drinks to specific locations. This arrangement shows a decrease in the overall cost of operation. A similar application is discussed by Fagerholt (1999), but with the focus on maritime industry.

Another application of such a heterogeneous multi-vehicle system is garbage collection. A study was published by Tung and Pinnoi (2000) concerning garbage collection in the city of Hanoi. Here, multiple vehicles with different capacities must be coordinated together in order to minimize the cost of operation, while collecting all garbage in designated areas in the city. Path planning for a heterogeneous vehicle was also considered by Mathew *et al*. (2014), who address the problem of

*Corresponding author

goods delivery in an urban location where a truck is carrying a quadcopter. This quadcopter then carries specific packages to desired locations, thus speeding up the process of delivery. The authors show that solving such an optimization problem is hard. Thus they again resort to a heuristic solver to arrive at a solution to the path planning problem.

In all the aforementioned references the authors focus on heuristic approaches to find the path for such heterogeneous vehicles. Heuristic solutions are in many cases easy to find, but there is no guarantee that the resulting solution is the best one, i.e., optimal. Furthermore, there is no clue how far the solution found by heuristics is from the true optimal solution.

A paper on finding an optimal solution to a path planning problem for heterogeneous vehicles was published by Garone *et al*. (2012), who consider a heterogeneous vehicle consisting of two separate vehicles: a ship and a helicopter. In this setting, the ship is assumed to have an unlimited range of operation but moves slowly compared with the helicopter. The helicopter, on the other hand, is fast but has a limited range of operation due to a limited fuel tank. The objective is to devise an optimal path for the heterogeneous vehicle by calculating coordinates of take-off and landing points for the helicopter, whose objective is to visit intermediate waypoints. The authors proposed to formulate a mixed-integer non-linear problem (MI-NLP) for finding an optimal path. The MI-NLP formulation considered has several disadvantages. First, the order waypoints to be visited are assumed to be known *a priori*. Second, the MI-NLP optimization problem is difficult to solve and scales badly with an increasing number of waypoints. The authors show that even for a small number of waypoints ($n = 7$) the time to solve such an MI-NLP problem reaches several hours. The authors therefore devise a set of heuristic rules, thus arriving at a suboptimal solution. These heuristic rules were still devised for a fixed order of points.

A computationally tractable mixed-integer second-order cone programming (MI-SOCP) formulation for finding an optimal path for heterogeneous vehicles was proposed in our earlier work (Klaučo *et al*., 2014). The MI-SOCP formulation is much more favorable in terms of computational time and allows devising an optimal path even for a larger number of points ($n = 100$). In this paper we extend our previous work and show how to devise the optimal path in the situation where the order of waypoints is not known *a priori*, but can be optimized so as to reduce the overall mission time. Two strategies to achieve such a goal are presented. The truly optimal scenario solves an extended MI-SOCP problem with additional binary variables, which optimize the ordering of waypoints. Moreover, a suboptimal strategy based on a TSP ordering is designed which neglects the heterogeneity of the vehicle. An extensive case study is presented to evaluate the induced loss of optimality.

The paper is organized as follows. First, we formally state the two main problems to be solved in Section 2. Then, in Section 3 we show how to solve the path-planning problem for a known ordering of waypoints. In particular, we first review the MI-NLP formulation of Garone *et al*. (2012) before stating the computationally more favorable MI-SOCP formulation, followed by a discussion of possible real-world extensions. The content of this section is based on our previous work (Klaučo *et al*., 2014). The main novel results are summarized in Section 4, where we show two strategies to devise the optimal plan for the situation where the ordering of waypoints is optimized. The paper is concluded by an extensive case study in Section 5, which discusses computational aspects and suboptimality.

2. Problem statement

The main obstacle in solving such path planning problems is the order of locations which one of the vehicles must visit. In this paper we tackle not only the problem of finding an optimal path for a heterogeneous vehicle when the order of points is known, but we also propose a way to obtain an optimal order of points.

We consider a heterogeneous vehicle system that consists of a carrier vehicle with a low maximal speed and a large range, and an agile vehicle (e.g., a helicopter) with a large maximal speed, but with a limited range. In particular, the carrier is assumed to move with a constant velocity v_{c} (the suffix c denotes the carrier) and has an unlimited range. The agile vehicle either rests on the carrier or is airborne. While airborne, the helicopter is assumed to travel at a constant velocity v_{h} and its range is limited by $t_{\mathrm{h,max}}$, i.e., the 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 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, after which the fleet proceeds to the final destination q_{f}. While the starting and the finishing points have to be visited by the ship, the intermediate points can also be visited by the agile part of the vehicle. Moreover, the agile vehicle can visit multiple intermediate points during one airborne phase. The objective is to devise an optimal route which minimizes the mission time, while taking into account constraints on the maximal range of the agile vehicle.

Formally, we can state this problem as follows.

Problem 1. Given are an ordered set of intermediate points $q_i \in \mathbb{R}^2$, $i = 1, \ldots, n$, a starting point $q_{\mathrm{s}} \in \mathbb{R}^2$, a final point $q_{\mathrm{f}} \in \mathbb{R}^2$, the carrier's speed v_{c}, the helicopter's speed v_{h}, and the helicopter's range. 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,
- a 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,
- 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 intermediate point q_i is visited exactly once *in the order* $i = 1, \ldots, n$, 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 therefore $m = 1$ (when the helicopter's range allows visiting all points $q_1, \ldots, q_n$ in one shot), while the maximum is $m = n$.

To give the reader a flavor of what the individual variables in Problem 1 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 $q_\mathrm{s} \to \tau_1 \to \ell_1 \to \tau_2 \to \ell_4 \to \tau_5 \to \ell_5 \to q_\mathrm{f}$. 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 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 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 heterogeneous vehicle returns to the port located at q_f.

Note that in Problem 1 we assume that the order of points is given. Such a situation frequently occurs in rescue operations, where the ordering represents priorities in which the targets must be rescued. However, in a different type of applications the ordering of intermediate points can change if it allows shortening the mission time. Therefore in this paper we also address the following extension of Problem 1.

Problem 2. Given are a starting point q_s, a final location q_f and a intermediate points q_i, $i = 1, \ldots, n$. Determine an optimal route of a minimal length and an optimal ordering in which the points q_i are visited such that the heterogeneous vehicle visits all intermediate points exactly once.

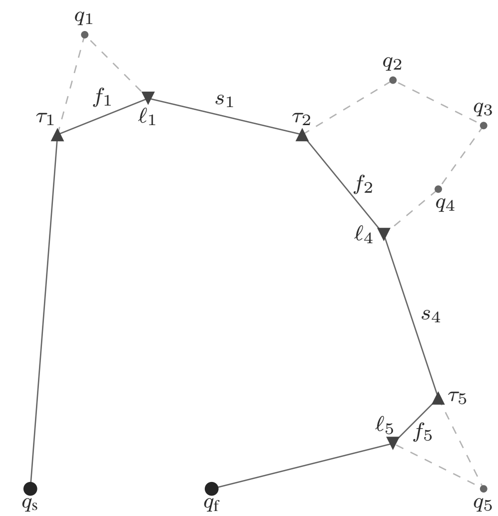

Fig. 1. Illustration of an optimal path for the heterogeneous vehicle. τ_i and ℓ_j denote the takeoff and landing points for the helicopter, respectively. The solid line shows the trajectory of the carrier and the dashed lines visualize the path of the helicopter that needs to visit points $q_1, \ldots,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

While in Problem 1 the index i in q_i refers to the fixed order of points visited by the agile part of the heterogeneous vehicle, in Problem 2 these indices are decision variables and are optimized.

3. Solution to Problem 1

3.1. Non-linear formulation. In this section we review the mixed-integer non-linear formulation of Problem 1 as suggested by Garone *et al.* (2012). 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-th 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:

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

It is easy to verify that α of (1) satisfies (2) and corresponds to three airborne phases: the first one visits only q_1 (and gives $\mathcal{I}_1 = \{1\}$), the second takeoff covers $q_2,\dots,q_4$ (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 us to somehow mitigate the exponential complexity of the resulting mixed-integer formulation.

With 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 $q_i,\dots,q_j$ to the touchdown point ℓ_j. The time spent airborne is restricted by the helicopter's range by

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

The multiplication by $\alpha_{i,j} \in \{0,1\}$ guarantees that the constraint will only become active if $\alpha_{i,j} = 1$, which corresponds to selection of $q_i,\dots,q_j$ 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 along 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 above by the time it takes the helicopter to travel the total distance from τ_i via $q_i,\dots,q_j$ 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 $q_i,\dots,q_j$, 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:

(a) the time the fleet travels from the starting point q_s to the first takeoff point τ_1, represented by $(1/v_\text{c})\|q_\text{s} - \tau_1\|$,

(b) the time consumed by the carrier alone to travel from one takeoff point to the next landing point, given by $\sum_{i=1}^{n}\sum_{j=i}^{n} f_{i,j}$,

(c) the 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

$$\alpha_{i,j}\|\ell_j - \tau_{j+1}\| \leq v_\text{c} s_{i,j}, \tag{7}$$

(d) the time of the 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

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

Then a solution to Problem 1 can be obtained by solving an optimization problem of the form

$$\min t_\text{m} \text{ subject to } (2)-(7), \tag{9}$$

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 the takeoff and landing points in the two-dimensional Euclidean 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), which result in $f_{i,j} \geq 0$ (and $s_{i,j} \geq 0$) for $\alpha_{i,j} = 0$.

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

Remark 1. Once the optimal solution to (9) is obtained, the equivalence between α and index sets $\mathcal{I}$ from Problem 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 $\mathcal{I}_i = \{i,\dots,j\}$, where j is the index of the column for which $\alpha_{i,j} = 1$. Due to (2), the index j will be a singleton for each i.

Remark 2. The problem (9) is always feasible. In the worst case, each flyover point q_i is individually visited by navigating the carrier to its neighborhood from where the point can be reached by the agile part of the vehicle. This scenario corresponds to α being an identity matrix. Feasibility stems from the fact that the carrier's range is assumed to be unlimited, and because there are no constraints on the overall mission time.

3.2. Mixed-integer SOCP formulation of Problem 1.

The main limitation of the mixed-integer non-linear programming (MI-NLP) formulation of the optimization problem (9) stems from its computational complexity. Specifically, Garone *et al*. (2012) demonstrated that the problem is solvable, in reasonable time, just for a small number of intermediate points q_i, i.e., for a small value of n. Specifically, the largest case reported in the reference was for $n = 7$. 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), which can be solved efficiently for *hundreds* of points. Although MI-SOCP problems are non-convex due to the presence of integer variables, once these integers are fixed in a branch-and-bound algorithm, the problem becomes a convex SOCP. On the other hand, in MI-NLP problems even the "inner" problems are non-convex.

We start by reminding that the non-trivial part of (9) covers non-linear constraints where various decision variables multiply each other. However, a closer look at (3)−(5) and (7) reveals that such non-linear 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 the 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 $\alpha_{i,j} = 0$ the 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 (Williams, 1993).

Lemma 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 1 to (10) allows us 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 the (19) are convex in corresponding decision variables. In particular, due to employing Euclidean norms, (19) can be written as a set of *second-order cone* constraints (see Boyd and Vandenberghe, 2009).

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

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

$$\text{subject to (18)−(19),} \tag{20b}$$

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

with t_{m} as in (8) and the constraints imposed for each $i, j \in \{0, \ldots, n\}$. The problem (20) is a mixed-integer second-order cone program that can be solved, e.g. by the Gurobi solver (Gurobi Optimization, 2013), which employs the branch-and-cut method to efficiently eliminate infeasible combinations of binary variables, thus avoiding exploration of an exponential number of cases.

Remark 3. It is important to note that (20) is a *non-conservative* reformulation 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 lower-bounded by zero in (20c).

To solve (20) as efficiently as possible, the value of M in (18) and (19) has to be chosen as low as possible. As noted, e.g., by Kvasnica (2008), non-tight values of the M constants can easily increase the computational time of (20) by several orders of magnitude. Therefore, it is important to derive the tightest possible values of M employed in (18) and (19). As noted by Williams (1993), 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 limit 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, $i = 1, \dots, n$, 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, $\Omega_f = \{f \mid 0 \leq f \leq t_{\text{h,max}}\}$, which follows from (3). Therefore, the tightest M in (19a) can be computed from (22) by evaluating $\|\tau_i - \ell_j\| - v_{\text{c}} f_{i,j}$ 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.

Remark 4. A direct consequence of Remarks 2 and 3 is that the MI-SOCP formulation (20) is always feasible, provided the value of M is chosen per (21).

3.3. Extensions of (9).

The solution to Problem 1 can therefore be obtained by solving (20) as an MI-SOCP problem. In this section we present several modifications of (20) which reflect real-life scenarios.

First, the objective function (20a) can be extended to account for minimization of the helicopter's flyover time, which is proportional to fuel consumption. This can be done by adding the term

$$\sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} \tag{23}$$

to (20a). Second, it might be desirable to minimize the number of take-offs, which might induce a physical stress on the helicopter. This can be achieved by including

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j} \tag{24}$$

into (20a). To allow the designer to assign priorities to individual objectives, the individual cost terms are associated with penalties $\gamma_{\text{f}} \geq 0$ and $\gamma_{\text{a}} \geq 0$ for (23) and (24), respectively. The overall cost function in (20a) then becomes

$$\min_{\alpha, f, s, \tau, \ell} t_{\text{m}} + \gamma_{\text{f}} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} + \gamma_{\text{a}} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j}. \tag{25}$$

The constraints of (20) can be extended as well. One natural extension is to limit the total number of takeoffs of the helicopter. This can be achieved by adding the constraint

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j} \leq m_{\text{max}}, \tag{26}$$

where m_{max} is the maximal desirable number of takeoffs. However, choosing a small value of m_{max} could lead to infeasibility of (20) if the helicopter's action radius, represented by $t_{\text{h,max}}$ in (3) and (18), does not allow the helicopter to reach all targets during m_{max} takeoffs. To overcome this limitation, we propose to use a soft version of (26):

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j} \leq m_{\text{max}} + z. \tag{27}$$

Here, $z \geq 0$ is a new continuous optimization variable which represents violation of the hard constraint. To discourage the optimization problem from violating the hard constraint unless absolutely necessary, the variable z must be penalized by $\gamma_{\text{z}} z$ in the objective function with γ_{z} being sufficiently high (Kerrigan and Maciejowski, 2000).

4. Solution to Problem 2

Determining an optimal route of minimal length by solving the MI-NLP (9) or its MI-SOCP version (20) requires that the order in which the intermediate waypoints q_i, $i = 1, \dots, n$, are to be visited be known *a priori* (Klaučo *et al.*, 2014). In this section we show how to find the optimal route subject to the assumption that the order is not given, but can be optimized so as to decrease the mission time. Two approaches can be used. One is to neglect the heterogeneity of the vehicle and regard it as having homogeneous dynamics. Then an extension of the well-known (Miller *et al.*, 1960) traveling salesman problem (TSP) can be used to deduce the optimal ordering of the intermediate waypoints. However, such a solution would only be suboptimal for heterogeneous dynamics. Therefore, in this section we present a truly optimal solution which takes the specifics of the vehicle into account. To optimize the ordering, one approach is to optimize over the indices of the intermediate waypoints in (5) and (6). An alternative strategy is to introduce a new set of waypoints, which will correspond to an optimal permutation of the original ones. The latter approach is used next. The suboptimal TSP-based approach will be discussed in Section 4.2.

4.1. Optimal solution to Problem 2. Let us first aggregate the intermediate waypoints q_i column-wise into the matrix $Q \in \mathbb{R}^{2\times n}$:

$$Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}. \tag{28}$$

Subsequently, we introduce a binary permutation matrix $P \in \{0,1\}^{n\times n}$ with the row and column sums equal to one:

$$\sum_{i=1}^{n} P_{i,j} = 1, \quad j = 1,\dots,n, \tag{29a}$$

$$\sum_{j=1}^{n} P_{i,j} = 1, \quad i = 1,\dots,n. \tag{29b}$$

Then

$$\widetilde{Q} = QP \tag{30}$$

denotes the matrix of permuted intermediate waypoints.

Let $\tilde{q}_j$ denote the j-th column of

$$\widetilde{Q} = \begin{bmatrix} \tilde{q}_1 & \tilde{q}_2 & \cdots & \tilde{q}_n \end{bmatrix}. \tag{31}$$

Then if $P_{i,j} = 1$, the i-th original waypoint q_i will in fact be visited as the j-th one. The advantage of this formulation is that the order of the permuted waypoints, i.e., $\tilde{q}_j$, $j = 1,\dots,n$, is now fixed and the optimization of the ordering was moved to the binary permutation matrix.

With this change, the constraint (5) becomes

$$\alpha_{i,j}\left(\|\tau_i - \tilde{q}_i\| + \tilde{d}_{i,j} + \|\tilde{q}_j - \ell_j\|\right) \le v_\text{h} f_{i,j}, \tag{32}$$

which must hold for all $i = 1,\dots,n$ and $j = 1,\dots,n$. Here, $\tilde{d}_{i,j}$ is the Euclidean distance between $\tilde{q}_i$ and $\tilde{q}_j$. Note that, in Section 3, these distances were pre-computed and considered constants, cf. (6). However, in this section the points $\tilde{q}_j$ depend on the choice of the binary permutation matrix P, which is a decision variable. Therefore, the distances must be computed inside the optimization problem via

$$\tilde{d}_{i,j} = \sum_{k=i}^{j-1} \|\tilde{q}_k - \tilde{q}_{k+1}\|. \tag{33}$$

The optimized variables in the new problem are $\tilde{q}_i$, $i = 1,\dots,n$, as the permuted waypoints, P as the binary permutation matrix, α as the binary matrix of takeoff/landing sequences, τ_i and ℓ_j as the takeoff and landing coordinates, respectively, as well as the matrices f and s, which represent the flyover and cruise times, respectively. They can be computed by solving

$$\min_{\widetilde{Q},P,\alpha,f,s,\tau,\ell} \; t_\text{m} \tag{34a}$$

subject to

$$f_{i,j} - t_\text{h,max} \le M(1-\alpha_{i,j}), \tag{34b}$$

$$\|\tau_i - \ell_j\| - v_\text{c} f_{i,j} \le M(1-\alpha_{i,j}), \tag{34c}$$

$$\tilde{d}_{i,j} = \sum_{k=i}^{j-1} \|\tilde{q}_k - \tilde{q}_{k+1}\|, \tag{34d}$$

$$\left(\|\tau_i - \tilde{q}_i\| + \tilde{d}_{i,j} + \|\tilde{q}_j - \ell_j\|\right) - v_\text{h} f_{i,j} \le M(1-\alpha_{i,j}), \tag{34e}$$

$$\|\ell_j - \tau_{j+1}\| - v_\text{c} s_{i,j} \le M(1-\alpha_{i,j}), \tag{34f}$$

$$\sum_{i=1}^{n} P_{i,j} = 1, \tag{34g}$$

$$\sum_{j=1}^{n} P_{i,j} = 1, \tag{34h}$$

$$\widetilde{Q} = QP, \tag{34i}$$

$$f_{i,j} \ge 0, \quad s_{i,j} \ge 0, \tag{34j}$$

$$P_{i,j} \in \{0,1\}, \quad \alpha_{i,j} \in \{0,1\}. \tag{34k}$$

In the objective function (34a), t_m is defined per (8), taking the permuted waypoints into account indirectly via $f_{i,j}$, which represent the flyover time from $\tilde{q}_i$ to $\tilde{q}_j$. In the constraints, (34b) stands for the limited flyover time and is identical to (18), which in turn is a big-M version of (3). Next, (34c) and (34f) are the same as in (19a) and in (19c) and represent the big-M versions of (4) and (7), respectively. These two constraints ensure that the agile part of the heterogeneous vehicle meets the ship at the landing point. The permuted waypoints $\tilde{q}_j$ (taken as the corresponding columns of the matrix $\widetilde{Q}$) enter the optimization problem via (34e), which is a big-M version of (32), together with (30) and (33) embedded as (34i) and (34d), respectively.

The optimization problem in (34) is still an MI-SOCP since (34b), (34c), and (34f) are second-order cone inequality constraints, (34d) is merely a substitution for $\tilde{d}_{i,j}$ into (34e), and (34g)–(34j) are linear constraints. However, compared with (20), the problem in (34) has n^2 additional binary variables due to the permutation matrix P. On the other hand, (34) optimizes the ordering of waypoints while (20) assumes an a-priori known order.

Once the optimal solution to (34) is computed, the optimal ordering can be easily extracted from the binary permutation matrix P as described before. We remark that the optimal takeoff and landing coordinates τ_i and ℓ_j, respectively, are already adjusted to the optimal ordering.

4.2. Suboptimal solution to Problem 2. A suboptimal solution to Problem 2 can be obtained by neglecting the heterogeneous dynamics, i.e., assuming that the vehicle is homogeneous and moves at a fixed speed. Then a standard TSP algorithm of Miller *et al*. (1960) can be extended to devise an optimal ordering of points for the homogeneous vehicle. Needless to say that such an ordering need not be optimal for the heterogeneous setup. We will investigate its suboptimality in Section 5.

Let the matrix $\widetilde{Q} = QP$ of permuted waypoints be as in (30) and (31), with Q defined per (28). Moreover, introduce a binary permutation matrix $P \in \{0,1\}^{n\times n}$, which optimally reorders the waypoints q_i, $i = 1,\dots,n$. The total distance traveled by the homogeneous vehicle is

$$l = \|q_{\mathrm{s}} - \tilde{q}_1\| + \left(\sum_{i=1}^{n-1} \|\tilde{q}_i - \tilde{q}_{i+1}\|\right) + \|\tilde{q}_n - q_{\mathrm{f}}\|. \quad (35)$$

The objective is to devise an optimal selection of P such that the total distance is minimized subject to the constraint that each of the intermediate waypoints q_i, $i = 1,\dots,n$, is visited exactly once. This, however, is equivalent to P satisfying (29). The search for an optimal permutation matrix (and hence for the optimal reordering of waypoints) can be cast as

$$\min_{\widetilde{Q},P} \text{ (35) subject to (29), (30),} \quad (36)$$

which is a mixed-integer linear program (MILP). Once solved, the i-th column of $\widetilde{Q}$, defined per (30) and (31), is the waypoint which is to be visited as the i-th one on the path $q_{\mathrm{s}} \to \tilde{q}_1 \to \cdots \to \tilde{q}_n \to q_{\mathrm{f}}$. Once the optimal TSP ordering of waypoints is obtained from (36), a suboptimal solution to Problem 2 can be reached by considering the TSP ordering in (20). Hence, two mixed-integer problems need to be solved. However, MILPs of the form (36) are usually cheaper to solve compared with MI-SOCPs since all constraints are linear. In the next section we analyze the suboptimality of this approach on a case study.

5. Case study

In this section we first analyze how the computational complexity of the MI-SOCPs (20) and (34) scales with an increasing number of intermediate waypoints. We demonstrate that the approach of this paper scales much better compared with the MI-NLP formulation of Garone *et al*. (2012). However, the formulation in (34), which also optimizes the ordering, is significantly more involved compared with (20), where the ordering is assumed to be fixed. In such a case the suboptimal TSP-based approach of Section 4.2 can also be used on much more favorable computational terms. Therefore, in Section 5.2 we analyze its suboptimality as well.

All computations were carried out on a 6-core `X5660` 2.8 GHz Intel CPU with 16 GB of RAM using Matlab 2014a. All optimization problems were formulated using YALMIP (Löfberg, 2004). The MI-SOCP problems (20), (34), and (36) were solved by CPLEX (ILOG, 2007). The MI-NLP problem (9) was solved by YALMIP's global branch-and-bound solver with `fmincon` as a subsolver, which was configured to perform at most 1×10^4 iterations with at most 5×10^4 function evaluations.

5.1. Computational complexity analysis.

First we analyze the computational complexity of the MI-SOCP formulation (20) as a function of the number of waypoints points q_i. To perform the analysis, we randomly distributed n points for $n \in \{10, 20, 30, \dots, 90, 100\}$ in a box-shaped domain with length of sides $\max\{50, 2.5n\}$ km. For each value of n we generated three sets of points randomly distributed in the box. The carrier's speed was set to $v_{\mathrm{c}} = 18\,\mathrm{km\,h^{-1}}$ and the speed of the agile vehicle was set to $v_{\mathrm{h}} = 90\,\mathrm{km\,h^{-1}}$ with a maximal flyover time $t_{\mathrm{h,max}} = 25\,\mathrm{min}$. Subsequently, for each scenario we devised an optimal navigation plan by solving (20) and measured the total computation time. Note that in this analysis the ordering of waypoints was considered fixed in advance, and therefore the procedure solves Problem 1.

The obtained results are shown graphically in Fig. 2. The solid line shows the average computation time for each value of n, the number of waypoints to be visited. Moreover, the runtimes reported for each value of n are the average among three sets of randomly generated points. As can be observed, for $n \leq 50$, it takes seconds to minutes to devise an optimal path. For $50 \leq n \leq 100$, the computation time reaches an hour to several hours. The non-monotonic behavior of Fig. 2 is due to the random nature of the generated points. Therefore, the computational time for obtaining the path plan for $n = 70$ can be shorter than the time for solving the MI-SOCP when 50 randomly distributed points are considered. Moreover, non-monotonicity is to be expected since the mixed-integer problem is non-linear and its solution depends on the branching strategy. Sometimes the solver employs a more efficient branching procedure. However, this is entirely problem-dependent.

Table 1 compares the complexity of the MI-NLP problem in (9) with the proposed MI-SOCP formulation in (20). Specifically, we solved both the problems for $n \in \{4, 5, 6, 7\}$ randomly generated points. We remark that the runtimes were obtained using the same HW/SW setup for both formulations. As can be observed, the MI-SOCP approach of this paper is significantly more effective than the MI-NLP formulation by Garone *et al*. (2012), and thus allows solving larger path-planning problems for heterogeneous vehicles.

Next we analyze how the order-optimizing formulation (34) scales with an increasing number of waypoints. Note that this formulation solves Problem 2.

Table 1. Solver runtimes required to solve the proposed MI-SOCP formulation (20) and the MI-NLP setup in (9).

n	MI-SOCP time [s]	MI-NLP time [s]
4	0.03	317.4
5	0.07	467.1
6	0.13	951.9
7	0.14	1755.3

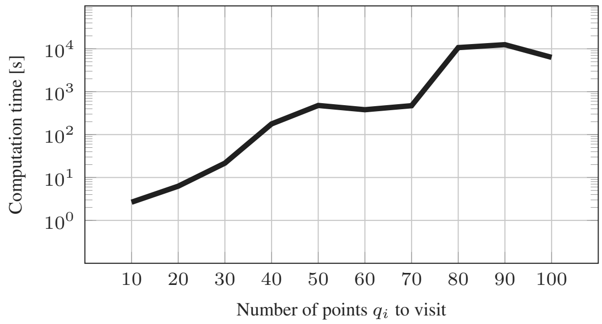

Fig. 2. Time required to obtain an optimal solution to (20) as a function of n and the number of points q_i to visit.

In this case, $n = \{5, 7, 9, 10\}$ randomly situated waypoints were considered. Compared with (20), the order-optimizing problem (34) has n^2 additional binary optimization variables due to the permutation matrix P being used in (30) and is therefore more involved. Individual calculation times are reported in Table 2. As can be seen, the computation time for $n = 10$ points is about 7.5 hours, a significant increase from several seconds required in (20), where the order is assumed to be fixed.

The second alternative to optimize the order of waypoints is to use the suboptimal procedure of Section 4.2. Here, first the TSP (36) is solved by neglecting the heterogeneous properties of the vehicle. Subsequently, the problem (20) is solved with the TSP ordering. The computational complexity of (36), which is formulated and solved as a mixed-integer linear problem, is shown in Fig. 3. As can be seen, the suboptimal ordering of waypoints can be obtained in less than one hour even for $n = 100$ waypoints. The necessary tradeoff is the induced loss of optimality compared with the solution of the truly optimal version (34). The amount of suboptimality is analyzed in the next section.

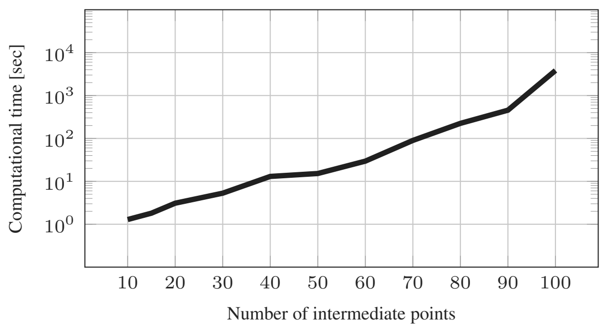

Fig. 3. Calculation time required to solve (36) as a function of an increasing number of intermediate waypoints.

Table 2. Solver times required for solving the order-optimizing problem (34).

n	time [s]
5	1
7	41
9	691
10	27624

5.2. Suboptimality analysis. As pointed out in Section 4.2, the TSP ordering obtained by solving (36) neglects the heterogeneous properties of the vehicle and is therefore suboptimal. In this section we analyze the induced loss of optimality compared with the solution of (34), which optimizes the ordering while assuming heterogeneous properties.

Table 3. Coordinates of an unordered list of visiting points.

x-coordinate	y-coordinate
0.00	50.00
27.00	25.00
35.00	15.00
15.00	10.00
22.50	25.00
50.00	50.00
48.00	48.00

We considered 7 intermediate points in a grid of $50 \times 50\,\mathrm{km}$. Coordinates of these points can be found in Table 3. In the test scenario coordinates of the starting and the finish point were fixed as $q_\mathrm{s} = [0\ 0]^T$ and $q_\mathrm{f} = [50\ 0]^T$. We set the maximum time of helicopter flyover to $t_{\mathrm{h,max}} = 25\,\mathrm{min}$, the speed of the agile vehicle to $v_\mathrm{h} = 90\,\mathrm{km\,h^{-1}}$ and the speed of the ship to $v_\mathrm{c} = 18\,\mathrm{km\,h^{-1}}$.

First, we devised the optimal mission plan by formulating the order-optimizing problem (34) as an MI-SOCP in the work of Löfberg (2004) and then solving it by CPLEX. The time required to obtain the optimal solution was $37\,\mathrm{s}$. The path can be seen in Fig. 4(a). The minimum mission time was $5.8319\,\mathrm{h}$.

Then, we neglected the heterogeneity of the vehicle and optimized the order of waypoints by solving the TSP (36). The TSP ordering was then used in (20) to devise the plan for the heterogeneous vehicle. The time needed for solving the TSP was $3.5\,\mathrm{s}$. Subsequently, solving the MI-SOCP (20) took $8.1\,\mathrm{s}$. The path plan given by this procedure is shown in Fig. 4(b). The mission time in this case was $5.8502\,\mathrm{h}$, which is by $65\,\mathrm{s}$ (or by 0.3%) longer than in the truly optimal case.

The difference between the globally optimal order of points and the TSP solution can be seen in Fig. 4. In particular, in the optimal mission each waypoint is visited individually, except for q_5 and q_6 from Fig. 4(a), which are covered by a single flyover. On the other hand, in the suboptimal TSP-based case, the waypoints q_2 and q_3 in Fig. 4(b) are visited together while all other points are covered individually.

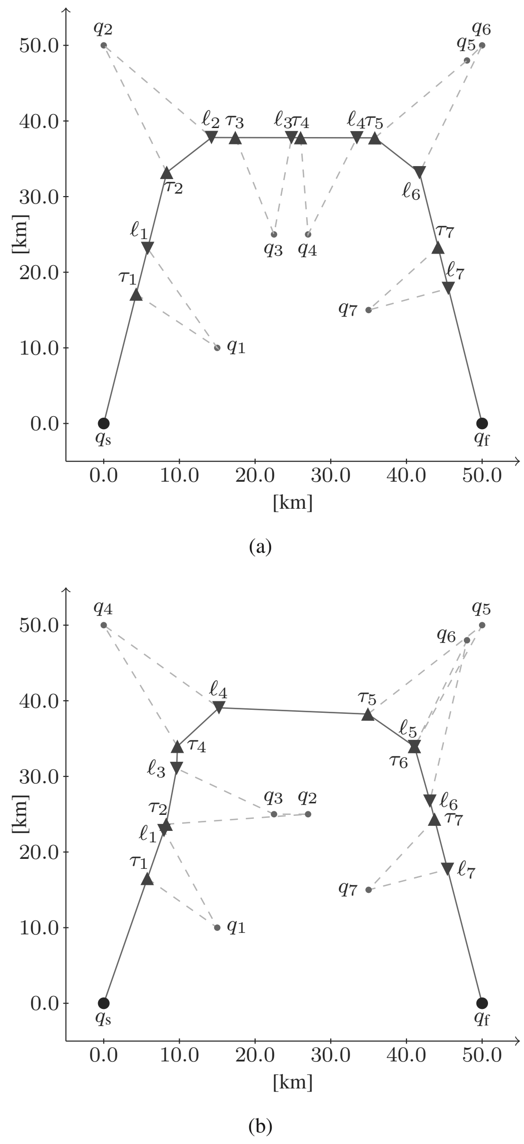

Fig. 4. Optimal path plan compared with the suboptimal path plan given by TSP ordering. The solid line represents the trajectory traveled by the carrier and the dashed lines represent the path of the agile vehicle. Takeoffs are denoted as τ_i and landing points as ℓ_i. Optimal path plan (a), suboptimal path plan (b).

To give a better assessment of the suboptimality of the TSP-based approach, we also investigated 50 random test scenarios, each with 7 randomly distributed points. For each test case the heterogeneously-optimal solution was obtained by solving the MI-SOCP problem (34). Subsequently, this globally optimal solution was compared with the path obtained for the TSP ordering. Recall that this path is obtained by first solving the MI-LP (36), followed by using this ordering in the MI-SOCP (20), which yields the path. The suboptimality of the TSP ordering is expressed as

$$\sigma = \frac{t_{\mathrm{TSP}} - t_{\min}}{t_{\min}} \times 100\%, \tag{37}$$

where t_{TSP} is the mission time determined by solving the MI-SOCP problem (20) with the ordering of points given by the TSP algorithm. Mission time $t_{\min}$ is the shortest time required for servicing targets determined by solving the MI-SOCP problem (34). Results of this analysis are shown in Fig. 5.

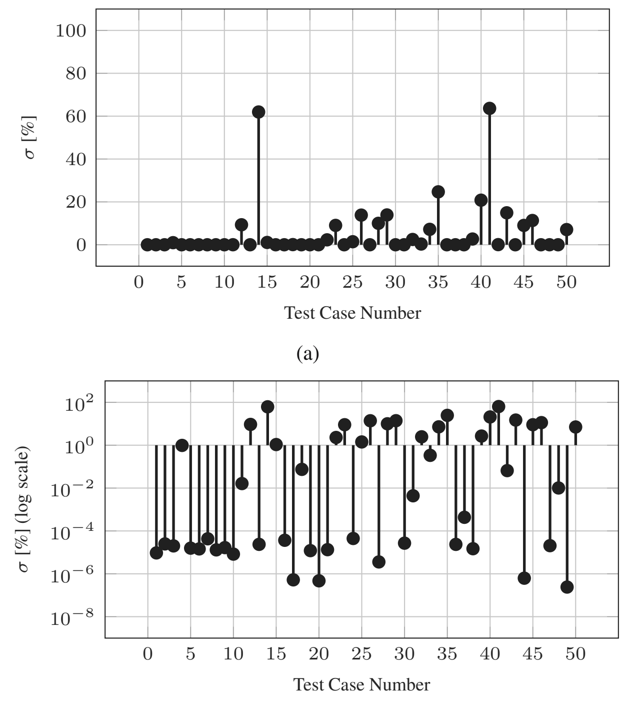

Fig. 5. Illustration of the suboptimality of the TSP path with respect to the optimal order of points. Plots with a linear and a log scale are offered to better illustrate the differences near zero. Suboptimality plotted on a linear scale (a) and on a logarithmic scale (b).

As can be seen in 31 out of 50 cases, which amounts to 62%, the suboptimality of the TSP ordering is less than 1% of the global optimum. In 15 cases (or in 30%) the suboptimality is between 1% and

20%. However, in 4 randomly generated cases the suboptimality reaches as high as 60%. These results indicate the expected suboptimality of the TSP-based approach if the computational complexity of the truly optimal MI-SOCP (34) is prohibitive for a large number of waypoints.

6. Conclusions

This paper suggests a solution to a path planning problem for heterogeneous multi-vehicle systems. The main contribution of this paper is twofold. First, if the order of waypoints to be visited is fixed, we show how to devise an optimal mission plan by solving a mixed-integer second-order cone optimization problem (20). This problem is considerably simpler to solve than the mixed-integer non-linear programming formulation suggested by Garone *et al*. (2012). The second contribution is that if the order of waypoints is to be optimized, then an extended version of the MI-SOCP procedure, reported as Eqn. (34), can be used to simultaneously optimize the ordering as well as the mission path. This new problem, however, is computationally challenging since it introduces additional binary variables. To mitigate the computational load, we also suggest to approach the problem of optimizing the ordering in two phases. First, the TSP-based ordering is found by neglecting the heterogeneity, followed by computing the path from (20). An extensive case study is provided which suggests that the suboptimality of this two-phase approach ranges from 0% to 63%.

Acknowledgment

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant no. 1/0403/15, and the financial support of the Slovak Research and Development Agency under the project APVV-15-0007.

References

Applegate, D. (2006). *The Traveling Salesman Problem: A Computational Study*, Princeton Series in Applied Mathematics, Princeton University Press, Princeton, NJ.

Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization*, 7th Edn., Cambridge University Press, New York, NY.

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

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

Gurobi Optimization (2013). Gurobi optimizer reference manual, `http://www.gurobi.com`.

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

ILOG (2007). 11.0 users manual, ILOG CPLEX Division, Incline Village, NV.

Kerrigan, E. and Maciejowski, J. (2000). Soft constraints and exact penalty functions in model predictive control, *Control 2000 Conference, Cambridge, UK*.

Klaučo, M., Blažek, S., Kvasnica, M. and Fikar, M. (2014). Mixed-integer SOCP formulation of the path planning problem for heterogeneous multi-vehicle systems, *European Control Conference 2014, Strasbourg, France*, pp. 1474–1479.

Kvasnica, M. (2008). *Efficient Software Tools for Control and Analysis of Hybrid Systems*, Ph.D. thesis, ETH Zurich, Zurich.

Löfberg, J. (2004). YALMIP, `http://users.isy.liu.se/johanl/yalmip/`.

Mathew, N., Smith, S. and Waslander, S. (2014). Optimal path planning in cooperative heterogeneous multi-robot delivery systems, *11th International Workshop on the Algorithmic Foundations of Robotics, Istanbul, Turkey*.

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

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

Williams, H. (1993). *Model Building in Mathematical Programming*, 3rd Edn., John Wiley & Sons, Hoboken, NJ.

Martin Klaučo was born in 1989. He received two M.Sc. degrees, one in 2012 in electrical engineering from the Denmark University of Technology and the other in 2013 in process control from the Slovak University of Technology in Bratislava. He is currently a Ph.D. student at the Institute of Information Engineering, Automation and Mathematics at the Faculty of Chemical and Food Technology, Slovak University of Technology.

Slavomír Blažek was born in 1987. He received his diploma at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. He is currently a programmer and an analyst of bank systems in the Intesa Sanpaolo group, as well as a Ph.D. student at the Institute of Information Engineering, Automation and Mathematics at the Faculty of Chemical and Food Technology, Slovak University of Technology.

Michal Kvasnica was born in 1977. He received his diploma in chemical engineering from the Slovak University of Technology in Bratislava and a Ph.D. in electrical engineering from the Swiss Federal Institute of

Technology in Zurich. Since 2011 he has been an associate professor at the Slovak University of Technology in Bratislava. His research interests are in model predictive control, the modeling of hybrid systems, and the development of software tools for control. He is a co-author and a developer of the MPT Toolbox for explicit model predictive control.

Received: 24 June 2015
Revised: 5 November 2015
Accepted: 16 January 2016