Brandenburg
University of Technology
Cottbus - Senftenberg

Mission Planning for Unmanned Aerial Vehicles

Armin Fügenschuh
Daniel Müllenstedt
Johannes Schmidt

Cottbus Mathematical Preprints
COMP # 6(2019)

Mission Planning for Unmanned Aerial Vehicles

Armin Fügenschuh, Daniel Müllenstedt, Johannes Schmidt

April 24, 2019

Abstract

We formulate the mission planning problem for a fleet of unmanned aerial vehicles (UAVs) as a mixed-integer nonlinear programming problem (MINLP). The problem asks for a selection of targets from a list to the UAVs, and trajectories that visit the chosen targets. To be feasible, a trajectory must pass each target at a desired maximal distance and within a certain time window, obstacles or regions of high risk must be avoided, and the fuel limitations must be obeyed. An optimal trajectory maximizes the sum of values of all targets that can be visited, and as a secondary goal, conducts the mission in the shortest possible time. In order to obtain numerical solutions to this model, we approximate the MINLP by a mixed-integer linear program (MILP), and apply a state-of-the-art solver (GUROBI) to the latter on a set of test instances.

Keywords: Mixed-Integer Nonlinear Programming, Trajectory Planning, Unmanned Aerial Vehicles, Linear Approximation.

1 Introduction

The German Armed Forces currently use several types of UASs (unmanned aerial systems) and UAVs (unmanned aerial vehicles), from the smallest one called "MIKADO", which is an electrical quadcopter, over the "LUNA" system to the large "Heron-1" and "Euro Hawk", which is based on the US-american "Global Hawk" system. The current mission planning process involves much user interaction and is not too well supported by decision support systems. At present, these systems only support the computation of a (locally) optimal trajectory between two given points. Anything more complex than this requires the aid of a human planner. The goal of our work is to develop a support system to plan UAV missions, and take as many real-world constraints into account as possible.

The mission planning problem for an unmanned aerial vehicle (UAV) asks for an optimal trajectory that visits a largest possible subset from a list of desired targets. When selected, each target must be traversed in a certain maximal distance and within a certain time interval. In a further variant of this problem, a fleet of potentially inhomogeneous UAVs is given. The UAVs differ with respect to their sensor properties, radar profiles, and operating ranges. Before actually planning the trajectories, a vehicle-to-target assignment has to be carried out. If the targets are surrounded by radar surveillance, then the vehicle's trajectory should be chosen to minimize the risk of detection. Hereto, the flight trajectory should make use of terrain properties. This planning problem is similar to classical vehicle routing problems with time windows (VRPTW) that are analyzed in the field of Operational Research. However, in these models the vehicles only move on a street network that is modeled as a graph. The UAV in contrast can fly freely through the three dimensional space. Moreover the fuel consumption for road based vehicles (cars or lorries) is more easy to calculate compared to UAVs. For the latter, the current weight, the altitude, the speed and climb/descend have an influence on the actual fuel consumption. We formulate the mission planning problem for UAVs as mixed-integer nonlinear programming problem. Binary decision variables are used to represent the decision about which target is inspected by which vehicle at which time. If too many targets are specified, then the model selects the most valuable ones with respect to a predefined ordering of the targets, where a target scoring function for visited targets is given. We formulate this optimization problem as a mixed-integer programming problem and solve it numerically using available solver software. In a parameter study, we vary a single selected parameter while leaving the others constant, in order to analyze the dependency of the running time of the solver on that particular parameter.

2 Survey of the Literature

The desire for unmanned aerial vehicles (UAVs) is as old as the history of human flight itself. For historical surveys we refer to [24, 1, 17]. During the last 15 years their popularity for use in military and civil applications rose. This reflects in the number of scientific publications over the past two decades that focus on the deployment of UAVs. From a mathematical or operations research point-of-view, roughly speaking, one can cluster the approaches by their degree of details on physical and technical properties that is incorporated into the models. On the one side, there exist highly detailed models that try to describe the physical properties of a flying UAV as accurately as possible, see for instance [19, 28]. Such models can be used to control a UAV for a very short segment of its trajectory. The numerical effort for solving such models does not allow the incorporation of decisions on selecting targets or the ordering of targets. On the other side of the range there are models that ignore most of the physical properties or approximate them in a very coarse way. Here, for example, the UAV is flying only in standard altitude and with standard speed, and the fuel consumption is assumed to be constant over time. Because they ignore the physics of flight, models such as the VRPTW mentioned above can include a large number of targets and vehicles and can still be solved to proven global optimality [25, 11]. Between these two extremal sides, there are numerous publications that include several aspects of the real-world in order to be realistic, while neglecting other aspects to become solvable. In the following, we review some of them.

Ma et al. [23] use an ant colony metaheuristic strategy to find a trajectory for a single UAV that traverses towards a single target point and must avoid threat points where the UAV might be detected by radar. The trajectory is smoothed in a post-processing step to reflect turn constraints. Fuel is not considered, and the trajectory is planar (2D).

The work of Schøler [29] is based on visibility graphs in 3D that are used for UAV path planning. He focuses on the collision free path planning around obstacles with minimum length.

Cobano et al. [6] use a genetic algorithm to solve a trajectory planning problem for multiple UAVs that avoids collisions.

Borelli et al. [2] address the conflict resolving problem for multiple UAVs. They numerically compare a nonlinear programming formulation with a mixed-integer linear formulation and demonstrate that the latter is faster the more dominant the disjunctive constraints become.

Robustness aspects for a single UAV are included in the work of Luders [22], that is, the trajectory should still be feasible when small disturbances to the input data occur. Hereto a linear dynamic model for the vehicle motion is assumed, and the whole problem is thus a mixed-integer linear program that can be numerically solved via CPLEX, a commercial optimization solver for mixed-integer linear problems.

Gao et al. [13] use Dubins curves to describe the path of a UAV. A Dubins path is the shortest curve that connects two points in the 2D plane subject to further constraints on the curvature of the path and given initial and terminal tangents to the path [9].

For a given set of waypoints, Forsmo [12] considers 2D path planning for a single UAV with mixed-integer linear programming, where flying around static rectangular obstacles is taken into account. Nonlinear constraints are approximated by piece-wise linear ones. The MILP problems are solved with GUROBI, another commercial optimization solver.

The focus of Evers et al. [10] is on robustness and uncertainty. They use a graph to model the dynamics of the UAV, similar to the traveling salesman problem, thus flight physics are neglected. However, their approach allows online processing of new waypoints that emerge during the flight.

Geiger et al. [14] use a direct collocation method and approximate the trajectory for a single or multiple UAVs by piecewise polynomials. The goal is not to minimize the flight time (as in most other publications), but to maximize the viewing time for the mounted sensor devices. The target can be stationary or moving, and wind can be taken into account.

Ruz et al. [27] take a symmetric adversary into account that uses radar to detect UAVs. Hence they describe a radar model and plan a trajectory that avoids areas of high detection risk. Their model is formulated as a mixed-integer linear program, and applied to a single target and a single UAV.

Lee et al. [21] consider the problem of following a moving ground vehicle with a UAV. If the ground vehicle is slower than the UAV, the path lurches around the ground vehicle's trajectory in a sinusoidal mode. When the vehicle stops, the UAV is performing a rose-shaped curve during its loitering mode. The switching between these two modes is performed ad-hoc in an online way,

depending on the changing behavior of the ground vehicle. Thus no integer model is formulated and solved.

Jun and D'Andrea [16] give a graph-based formulation for the UAV path planning problem that can be solved by shortest-path algorithms (Dijkstra or Bellman-Ford). Sharp edges of the trajectory are smoothed in a post-processing step. The area is discretized, and each finite cell area is endowed with a risk value (a probability map). The objective is to find a risk-minimal path from an origin to a target for a single UAV.

Culligan [7] studies the online trajectory planning problem for UAVs, and formulates this problem as a mixed-integer program. The nonlinear kinematic constraints are linearized by piecewise linear approximation. Obstacles are modeled by rectangular boxes in 3D and avoidance constraints are formulated by binary variables. In a similar way, cost regions are formulated. Obstacles can be static or dynamic. The earth's surface shape can be taken into account to allow flights close to the surface.

Landry et al. [20] solve a similar problem for a robot motion planning problem. Similar to the UAVs, the robot have to move to different locations and are not allowed to meet (collision avoidance). Naturally, height does not play a role in their model. The problem is solved by a combination of discrete and continuous methods.

A different approach is presented by Kress and Royset [18], where the targets are unknown at the beginning of the mission and should be detected by a fleet of UAVs. The targets move around the area with certain probabilities. The goal is to determine a search plan that specifies which part of the area is searched in which (discretized) time period. This problem can be modeled as a mixed-integer linear problem, and is solved using CPLEX within 2 minutes. The physical and technical properties of actual UAVs are mostly ignored. Only collision avoidance constraints and a maximum deployment time is considered.

We contribute to the state-of-the-art by presenting a model that simultaneously takes into account several aspects of the mission and trajectory planning process that were treated independently before: Our model is capable of planning a 4D path (three spatial dimensions plus time) around static or dynamic obstacles for an inhomogeneous fleet of UAVs. The fuel consumption of the UAVs is considered. The selection of waypoints and their assignment to the fleet of UAVs is part of the model, as well as the ordering in which the chosen waypoints are to be visited.

3 A Mathematical Model

We first formulate the UAV trajectory planning problem as a mixed-integer nonlinear program, and reformulate it later as a mixed-integer linear program. We focus on short range UAVs with an operational range of up to 200 km and endurance times of 8 to 12 hours mainly used for surveillance and reconnaissance purpose. Due to the short range it is not necessary to take the curvature of the earth into account. Instead, we are using a two-dimensional flat projection of the relevant portion of the earth, where we can apply a flat two-dimensional x-y-coordinate system (usually, a Universal Transverse Mercator system is chosen) for the surface, and a z-coordinate for the altitude. In the following, $\|\mathbf{v}\|_2 := \sqrt{(v^x)^2 + (v^y)^2}$ and $\|\mathbf{v}\|_3 := \sqrt{(v^x)^2 + (v^y)^2 + (v^z)^2}$ denote the Euclidean norms in $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. We use the notation $\|\mathbf{v}\|_2$ also for a three-dimensional vector $\mathbf{v} = (v^x, v^y, v^z)$, by which we mean that the Euclidean norm is only taken over the first two components v^x, v^y. Moreover, we use the 1-norm $\|\mathbf{v}\|_1 := \sum_{i\in\{x,y,z\}} |v^i|$ and the maximum norm $\|\mathbf{v}\|_\infty := \max_{i\in\{x,y,z\}} |v^i|$. We denote $\mathbf{1} := (1, 1, 1)$.

3.1 Sets

The following sets describe an instance of the problem. Given is a set of UAVs $\mathcal{U} := \{1, \dots, U\}$ which represents the (inhomogeneous) fleet, a set of waypoints $\mathcal{W} := \{1, \dots, W\}$ which are to be covered by a UAV in a fly by maneuver, a set of restricted air spaces $\mathcal{Q} := \{1, \dots, Q\}$ which are not to be traversed by the UAVs, a set of altitude bands $\mathcal{L}_u := \{1, \dots, L_u\}$ and throttle bands $\mathcal{V}_u := \{1, \dots, V_u\}$ (for each $u \in \mathcal{U}$ which are needed to account for varying fuel consumption rates of the UAVs depending on the chosen altitude and speed, and a set of discrete time steps $\mathcal{T} := \{0, \dots, T\}$. As abbreviations we set $\mathcal{T}^- := \mathcal{T}\backslash\{T\}$, and $\mathcal{L}_{0,u} := \mathcal{L}_u \cup \{0\}$.

3.2 Parameters

The following parameters need to be specified. They describe the technical properties of the UAVs and geographical environment of the mission.

3.2.1 UAV Technical Parameters

The minimum and maximum velocity (true air speed) of UAV u are denoted by $\underline{v}_u, \overline{v}_u \in \mathbb{R}_+$, respectively. These limits apply for horizontal flight maneuvers. For the vertical direction, limits are imposed by a maximum climb rate $\overline{v}_u^{z,+}$ and a maximum descend rate $\overline{v}_u^{z,-}$. Its maximum acceleration is $\overline{a}_u \in \mathbb{R}_+$. Its minimum and maximum flight altitude is $\underline{h}_u$ and $\overline{h}_u$, respectively. The (type-independent) minimum safety distance between two airborne UAVs to avoid collisions is $\varepsilon = (\varepsilon^x, \varepsilon^y, \varepsilon^z) \in \mathbb{R}_+^3$.

The initial fuel at start is denoted by $F_u \in \mathbb{R}_+$; we assume the UAVs with maximal fuel at start. The fuel consumption rate varies with speed and altitude; it is usually largest for flying fast in low altitude and lowest for flying slow in high altitude. To approximate the fuel consumption, we assume a discrete model for the fuel consumption, depending on the altitude band and speed limit. (This model neglects environmental parameters such as the temperature and air moisture, and UAV in-flight parameters, most importantly its current weight.) For this we define the boundaries $H_{u,0} < H_{u,1} < \ldots < H_{u,L_u}$ for the altitudes in $\mathcal{L}_{0,u}$, and speed limits $\theta_{u,1} < \theta_{u,2} < \ldots < \theta_{u,V_u}$ for the discrete speeds in $\mathcal{V}_u$. The minimum value $H_{u,0} = \underline{h}_u$ is (close to) zero, and $H_{u,L_u} = \overline{h}_u$ is the maximum flight altitude of the UAVs. Similarly, we have $\theta_{u,1} = \underline{v}_u$ and $\theta_{u,V_u} = \overline{v}_u$. By $\eta_{u,i,j} \in \mathbb{R}_+$ we denote the fuel consumption parameters for UAV u in altitude band $i \in \mathcal{L}_u$ and speed limit θ_j with $j \in \mathcal{V}_u$. There is a fuel surplus when climbing, which is given by $\xi_u \in \mathbb{R}_+$.

3.2.2 Mission Parameters

The start and end location (launch and recovery point) for each UAV u are the coordinate vector $\mathbf{R}_u^0, \mathbf{R}_u^T \in \mathbb{R}^3$, respectively. We assume that both locations have zero altitude: $R_u^{0,z} = R_u^{T,z} = 0$. A number of waypoints $\mathbf{p}_w \in \mathbb{R}^3$ for $w \in \mathcal{W}$ are given that should all be visited, if time and fuel conditions allow. Depending on the sensor/actor technique and the desired mission goal, a maximal operational distance to the waypoint w for UAV u is given by $\delta_{u,w}$. Each waypoint must be reached within a time window $\mathcal{T}_w \subseteq \mathcal{T}$. When reaching a waypoint within the prescribed distance and given time window, a score value S_w is added to the objective.

The ground control station for UAV u is located at the coordinates $\mathbf{G}_u \in \mathbb{R}^3$, and the UAV can be controlled up to a maximum distance (range) $\varrho_u \in \mathbb{R}_+$. Hereby we assume that there are no obstacles to the line-of-sight in all directions. Otherwise, one can specify those regions without UHF/VHF connection as restricted air spaces (see below).

3.2.3 Environmental Parameters

Restricted air spaces are three dimensional rectangular regions that are not allowed to be entered by the UAV. This feature can be used to capture regions without radio connection between the UAV and the ground control station, mountains or high buildings, or adversary radar or air defense systems. For each restricted air space $q \in \mathcal{Q}$ two coordinate vectors of the lower and upper bound for each coordinate are specified: $\underline{\mathbf{c}}_q(t), \overline{\mathbf{c}}_q(t) \in \mathbb{R}^3$. Each restricted airspace may vary its location over time t.

The wind velocity is given by $\mathbf{w}(t) = (w^x(t), w^y(t), 0) \in \mathbb{R}^3$. Hereby we assume that wind may vary over time, but is constant for the whole flight area.

3.2.4 Further Parameters

The step size of the time discretization is given by Δt, i.e., the wall-clock time that passes between t and $t+1$, for $t \in \mathcal{T}^-$.

A parameter $M \in \mathbb{R}_+$ symbolizes a sufficiently large value, which is needed for coupling binary variables to real-valued decision variables using the big-M-method.

3.3 Variables and Bounds

The trajectory for each UAV $u \in \mathcal{U}$ is described by a set of coordinate vectors (real-valued or continuous variables) $\mathbf{r}_u(t) = (r_u^x(t), r_u^y(t), r_u^z(t)) \in \mathbb{R}^3$ for each discrete time step $t \in \mathcal{T}$. The horizontal velocity of vehicle u at time step t is denoted by the continuous variables $\mathbf{v}_u(t) = (v_u^x(t), v_u^y(t)) \in \mathbb{R}^2$, and its horizontal acceleration is modeled by the continuous variables $\mathbf{a}_u(t) = (a_u^x(t), a_u^y(t)) \in \mathbb{R}^2$. The vertical velocity is separated into a climb rate $v_u^{z,+} \in [0, \overline{v}_u^{z,+}]$ and a descend rate $v_u^{z,-} \in [0, \overline{v}_u^{z,-}]$. (Acceleration with respect to the vertical axis is not considered.) The vertical velocity is separated into two, since only a climb will burn additional fuel, whereas a descend comes "for free" by the force of gravity.

To model the fuel consumption, the following variables are introduced. The current amount of available fuel on board of UAV u at time step t is given by the continuous variable $g_u(t)$. The binary variable $s_{u,i,j}(t) \in \{0,1\}$ is equal to 1 if and only if UAV u is flying in altitude band $i \in \mathcal{L}_u$ with speed at most θ_j for $j \in \mathcal{V}_u$ in time step t.

The mission time is defined by an interplay of the following three families of binary decision variables. The variables $b_u^+(t) \in \{0,1\}$ are set to 0 before the start of the task for UAV u, and set to 1 once the mission has started and also in all subsequent time steps. Opposed to that the variables $b_u^-(t) \in \{0,1\}$ are set to 1 before the end of the mission, and set to 0 once the mission has ended and in all subsequent time steps thereafter. The resulting binary variables $b_u(t) = b_u^+(t) + b_u^-(t) - 1$ then indicate if UAV u is airborne in time step t, which is the case if and only if both binary variables $b_u^+(t)$ and $b_u^-(t)$ are equal to 1.

The proximity of a UAV to waypoints, restricted airspaces, and other UAVs is modeled using the following binary decision variables. The binary variables $d_{u,w}(t) \in \{0,1\}$ indicate if UAV u reaches waypoint w at time step $t \in \mathcal{T}_w$. The binary variables $\underline{\mathbf{f}}_{u,q}(t), \overline{\mathbf{f}}_{u,q}(t) \in \{0,1\}^3$ for each $u \in \mathcal{U}, q \in \mathcal{Q}$ report if the vehicle is within a restricted airspace. The binary variables $\underline{\mathbf{e}}_{u,u'}(t)$, $\overline{\mathbf{e}}_{u,u'}(t) \in \{0,1\}^3$ for $u, u' \in \mathcal{U}, u < u'$ and $t \in \mathcal{T}$ keep track if vehicle u is too close to u' with respect to safety requirements.

3.4 Constraints

The following constraints define a feasible assignment of waypoints to UAVs and the UAV flight itself.

3.4.1 Flight Area

The flight starts and ends for each UAV at the specified coordinates:

$$\mathbf{r}_u(0) = \mathbf{R}_u^0, \quad \forall\, u \in \mathcal{U}, \tag{1}$$

$$\mathbf{r}_u(T) = \mathbf{R}_u^T, \quad \forall\, u \in \mathcal{U}. \tag{2}$$

The UAV must keep a connection to the ground control unit (otherwise, an emergency landing procedure is initiated). Hence a range limit is imposed (which can be disobeyed when a satellite link is used for communication):

$$\|\mathbf{r}_u(t) - \mathbf{G}_u\|_2 \leq \varrho_u, \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{3}$$

The altitude cannot be higher than the maximum altitude, and is zero if the UAV is not airborne:

$$r_u^z(t) \leq \overline{h}_u \cdot b_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{4}$$

The altitude has to be a least the minimum altitude once the UAV is airborne:

$$r_u^z(t) \geq \underline{h}_u \cdot b_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{5}$$

3.4.2 Flight Dynamics

The flight dynamic is described by a point model for the UAV [3]. By Newton's law of motion we have that the acceleration as the derivative of the velocity, and the velocity as the derivative of the location:

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{r}}{dt^2}.$$

Since we are discretizing the time and use a finite difference approach, Newton's law enters our model in the following form: The position is updated by a second order difference equation, where the influence of wind is also taken into account:

$$r_u^{x|y}(t+1) = r_u^{x|y}(t) + \Delta t \cdot (v_u^{x|y}(t) + w^{x|y}(t) \cdot b_u(t)) + \frac{(\Delta t)^2}{2} \cdot a_u^{x|y}(t), \quad \forall\, u \in U, t \in \mathcal{T}^-. \tag{6}$$

The underlying avionic model is that of the TC-TH-W vector triangle, where TH is the true heading vector, W is the wind vector, and TC is the true course vector, W is the wind vector, and TC is the true course vector, resulting from a vector addition TC = TH + W; for more details on this we refer to [15].

The velocity is updated by a first order difference equation:

$$v_u^{x|y}(t+1) = v_u^{x|y}(t) + \Delta t \cdot a_u^{x|y}(t), \quad \forall\, u \in U, t \in \mathcal{T}^-. \tag{7}$$

For vertical movements (climb and descend) the change in altitude depends on the climb and descend velocity (upwind or downwind effects are neglected, but could be similarly taken into account as in equation (6)):

$$r_u^z(t+1) = r_u^z(t) + \Delta t(v_u^{z,+} - v_u^{z,-}), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{8}$$

The horizontal velocity for airborne UAVs is limited by certain lower and upper bounds:

$$\underline{v}_u \cdot b_u(t) \leq \|\mathbf{v}_u\|_2, \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T} \tag{9}$$

$$\|\mathbf{v}_u\|_2 \leq \overline{v}_u \cdot b_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{10}$$

The vertical acceleration of airborne UAVs is only limited from above:

$$\|\mathbf{a}_u\|_2 \leq \overline{a}_u \cdot b_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{11}$$

The start and the end of the mission are reported by the corresponding binary variables:

$$b_u^-(t+1) \leq b_u^-(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}^-, \tag{12}$$

$$b_u^+(t) \leq b_u^+(t+1), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}^-. \tag{13}$$

3.4.3 Waypoints

If a waypoint $w \in \mathcal{W}$ is met, then the UAV must fly by within a certain maximal distance:

$$\|\mathbf{r}_u(t) - \mathbf{p}_w\|_3 \leq \delta_{u,w} + M \cdot (1 - d_{u,w}(t)), \quad \forall\, u \in \mathcal{U}, w \in \mathcal{W}, t \in \mathcal{T}. \tag{14}$$

At most one UAV can claim a waypoint:

$$\sum_{u \in \mathcal{U}, t \in \mathcal{T}_w} d_{u,w}(t) \leq 1, \quad \forall\, w \in \mathcal{W}. \tag{15}$$

3.4.4 Restricted Airspaces

A rectangular area where the UAV is not allowed to fly through is modeled as follows:

$$\overline{\mathbf{c}}_q(t) - M \cdot \overline{\mathbf{f}}_{u,q}(t) \leq \mathbf{r}_u(t), \quad \forall\, u \in \mathcal{U}, q \in \mathcal{Q}, t \in \mathcal{T}, \tag{16}$$

$$\mathbf{r}_u(t) \leq \underline{\mathbf{c}}_q(t) + M \cdot \underline{\mathbf{f}}_{u,q}(t), \quad \forall\, u \in \mathcal{U}, q \in \mathcal{Q}, t \in \mathcal{T}, \tag{17}$$

$$\mathbf{1} \cdot \underline{\mathbf{f}}_{u,q}(t) + \mathbf{1} \cdot \overline{\mathbf{f}}_{u,q}(t) \leq 5, \quad \forall\, u \in \mathcal{U}, q \in \mathcal{Q}, t \in \mathcal{T}. \tag{18}$$

Constraints (16), (17) require that if the UAV u is staying inside the forbidden box $[\underline{c}_q^x, \overline{c}_q^x] \times [\underline{c}_q^y, \overline{c}_q^y] \times [\underline{c}_q^z, \overline{c}_q^z]$, then all six binary variables from $\underline{\mathbf{f}}_{u,q}(t)$ and $\overline{\mathbf{f}}_{u,q}(t)$ must be set to 1, which is forbidden by constraints in (18).

3.4.5 Collision Avoidance

If $U > 1$, i.e., more than one UAV is flying at the same time, then for each pair of UAVs $u, u' \in \mathcal{U}$ with $u < u'$ it has to be ensured that their trajectories are sufficiently separated from each other in order to avoid a collision. That can be seen as each UAV defining a restricted airspace that no other UAV must enter:

$$\mathbf{r}_{u'}(t) + \varepsilon - M \cdot \overline{\mathbf{e}}_{u,u'}(t) \leq \mathbf{r}_u(t), \quad \forall\, u, u' \in \mathcal{U} : u < u', t \in \mathcal{T}, \tag{19}$$

$$\mathbf{r}_u(t) \leq \mathbf{r}_{u'}(t) - \varepsilon + M \cdot \underline{\mathbf{e}}_{u,u'}(t), \quad \forall\, u, u' \in \mathcal{U} : u < u', t \in \mathcal{T}, \tag{20}$$

$$\mathbf{1} \cdot \underline{\mathbf{e}}_{u,u'}(t) + \mathbf{1} \cdot \overline{\mathbf{e}}_{u,u'}(t) \leq 7 - b_{u_1}(t) - b_{u_2}(t), \quad \forall\, u, u' \in \mathcal{U} : u < u', t \in \mathcal{T}. \tag{21}$$

3.4.6 Fuel Consumption

We use an approximative fuel consumption model that takes into account several altitude bands. In general, the higher the altitude the less fuel is consumed by the UAV. Moreover, the higher the velocity, the more fuel it consumes. At start, the UAV's fuel level is initialized with the start fuel amount:

$$g_u(0) = F_u, \quad \forall\, u \in \mathcal{U}. \tag{22}$$

During the flight, the fuel is reduced for climbing as well as for cruising, where the latter consumption depends on the momentary speed and altitude band in which the UAV operates:

$$g_u(t+1) = g_u(t) - \Delta_t \cdot \left(\xi_u \cdot v_u^{z,+} + \sum_{i \in \mathcal{L}_u, j \in \mathcal{V}_u} \eta_{u,i,j} \cdot s_{u,i,j}(t) \right), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}^-. \tag{23}$$

An airborne UAV choses precisely one speed band and one altitude band per time step, and a grounded UAV zero:

$$\sum_{i \in \mathcal{L}_u, j \in \mathcal{V}_u} s_{u,i,j}(t) = b_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{24}$$

This selection influences the actual speed of the UAV:

$$\sum_{j \in \mathcal{V}_u} \theta_{u,j} \cdot \left(\sum_{i \in \mathcal{L}_u} s_{u,i,j}(t) \right) = \|\mathbf{v}_u(t)\|, \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{25}$$

Similarly it influences the actual altitude of the UAV:

$$\sum_{i \in \mathcal{L}_u} H_{u,i} \cdot \left(\sum_{j \in \mathcal{V}_u} s_{u,i,j}(t) \right) \geq r_u^z(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}, \tag{26}$$

$$\sum_{i \in \mathcal{L}_u} H_{u,i-1} \cdot \left(\sum_{j \in \mathcal{V}_u} s_{u,i,j}(t) \right) \leq r_u^z(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{27}$$

3.5 Objective

The primary goal is to maximize the score for reaching the waypoints. On a subordinate level it is desired to finish the mission as soon as possible with the least amount of fuel flying in high altitude at low speed. This reflects the following objective function:

$$\max \sum_{u \in \mathcal{U}, w \in \mathcal{W}, t \in \mathcal{T}_w} S_w \cdot d_{u,w}(t) \tag{28}$$

$$+ \frac{1}{M} \sum_{u \in \mathcal{U}, t \in \mathcal{T}} \left(-t \cdot b_u(t) + g_u(t) + r_u^z(t) - \|\mathbf{v}_u(t)\|_2 \right). \tag{29}$$

4 Linearizing the Model

The model contains several nonlinear constraints. All nonlinear constraints in the model – (3), (9), (10), (11), (14) – involve the Euclidean norms $\|\cdot\|_2$ and $\|\cdot\|_3$. Moreover, in almost all nonlinear constraints this norm is bounded from above, with (9) being the only exception. In order to apply a linear mixed-integer solver, we linearize them and thus obtain an approximation.

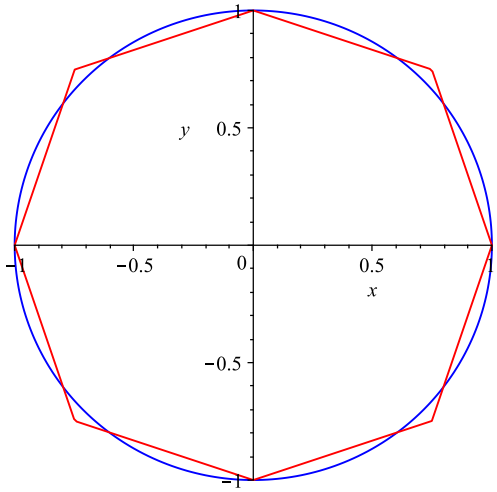

Figure 1: Euclidean norm approximation in two dimensions; blue graph: $\|\mathbf{v}\|_2 = 1$, red graph: $\lambda_2\|\mathbf{v}\|_1 + (1-\lambda_2)\|\mathbf{v}\|_\infty = 1$.

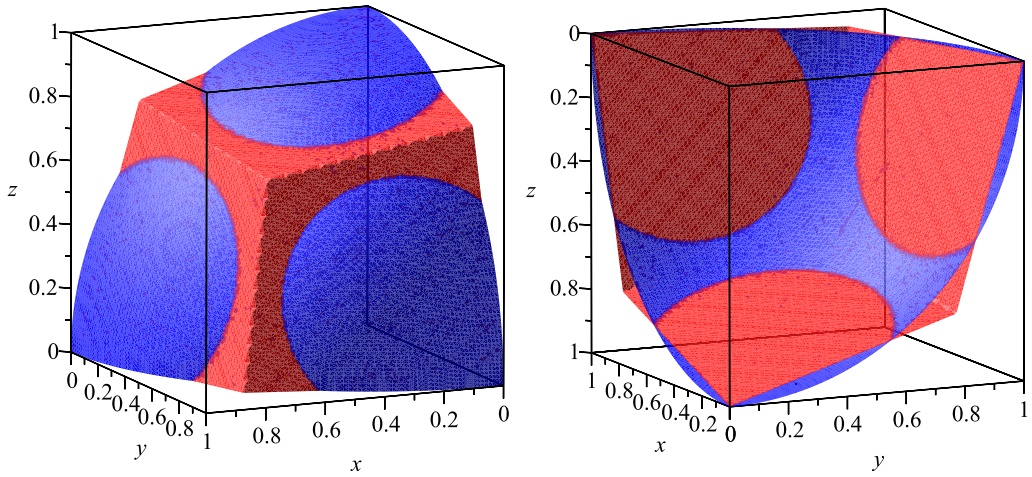

Figure 2: Euclidean norm approximation in three dimensions; blue surface: $\|\mathbf{v}\|_3 = 1$, red surface: $\lambda_3\|\mathbf{v}\|_1 + (1-\lambda_3)\|\mathbf{v}\|_\infty = 1$ (shown only for the positive octant).

4.1 Approximating Euclidean Norms by Combinations of Other Norms

We approximate the Euclidean norm by a linear combination of the 1-norm and the infinity-norm, see Celebi et al. [4] and the references therein for a survey of this topic. Using Rhodes' approximation approach [26], we obtain the parameters $\lambda_2 := 0.3364$ and $\lambda_3 := 0.2980$ from solving the quartic (forth order) equation $1 - 2\sqrt{\lambda - \lambda^2} = \sqrt{1 + \lambda^2(i-1)} - 1$ for $\lambda \in [0, \frac{1}{2}]$ (for $i \in \{2, 3\}$ being the respective dimension of the space), and taking the smaller of the two real roots as solution λ_i (the two remaining roots are complex). These values are used in an approximation formula introduced by Chaudhuri et al. [5]:

$$\|\mathbf{v}\|_i \approx \lambda_i\|\mathbf{v}\|_1 + (1-\lambda_i)\|\mathbf{v}\|_\infty, \quad \forall\, i \in \{2,3\}. \tag{30}$$

Figure 1 and Figure 2 show a comparison between the Euclidean norm and the approximated norm in two and three dimensions, respectively.

4.2 Linear Constraints for Norm Approximations

To include the approximated norms in our model formulation, it is necessary to describe them by linear constraints. We demonstrate this approach for the constraints (3) in detail. For the other constraints involving norms it can be done in an analogue way.

For each $u \in \mathcal{U}, t \in \mathcal{T}$ we introduce the variables $OR_u(t), \overline{OR}_u(t) \in \mathbb{R}^+$, and additionally for each coordinate direction the variables $OR_u^x(t), OR_u^y(t) \in \mathbb{R}^+$. (The abbreviation OR stands for "operational range".) The absolute difference is then measured in each coordinate by the constraints

$$r_u^i(t) - G_u^i \leq OR_u^i(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}, i \in \{x, y\}, \tag{31}$$

$$G_u^i - r_u^i(t) \leq OR_u^i(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}, i \in \{x, y\}. \tag{32}$$

Then the larger of the two values is taken as the maximum by the constraints

$$OR_u^i(t) \leq \overline{OR}_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{33}$$

We can then define the approximation of the Euclidean norm by the constraints

$$OR_u(t) = \lambda_2 \sum_{i \in \{x,y\}} OR_u^i(t) + (1 - \lambda_2)\overline{OR}_u(t), \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{34}$$

Then (3) is replaced by

$$OR_u(t) \leq \varrho_u, \quad \forall\, u \in \mathcal{U}, t \in \mathcal{T}. \tag{35}$$

4.3 Minimum Velocity

This approximation scheme is only valid if an upper bound constraint is imposed on the Euclidean norm. A lower bound constraint leads to a non-convex feasible region, and thus needs to be treated by further binary variables in order to deal with this disjunction. This applies to constraints (9) in our model.

One possible way to deal with that is to exclude small velocities in the same way as restricted air spaces, where additional binary variables are introduced to restrain the velocity from being smaller than $\underline{v}$. Note that this gives only a rectangular box around the origin, which is a rather coarse approximation in comparison with the upper bound, where the round shape of the Euclidean norm is approximated which a higher precision by an octagon (in 2D) or a deltoidal icositetrahedron (in 3D).

However, it turned out that this introduction of further combinatorial complexity is in fact not necessary for two reasons. First, in almost all cases the solutions of various different model instances do not make use of too small velocities: Since the fuel is always computed using the value of the velocity's approximated norm, the optimal solution tends to use a velocity that is close to this approximation, because a smaller velocity would be a waste (of time). Only in cases where time windows actually force a UAV to slow down significantly in order to reach a certain waypoint not too early, the choice of small velocities is actually justified. From a practical point of view, the solution can still be used, because a small velocity is achieved by flying a loop maneuver with the UAV, which reduces its effective speed towards the next waypoint, while keeping its actual speed above the necessary lower bound.

5 Input Data

We consider the following four different types of UAVs:

UAV-1. Heron TP UAV [Eitan] - Israel (Air Force), since 2012.
UAV-2. RQ-5A Hunter UAV - United States (Army), since 1996.
UAV-3. Brevel KZO UAV - Germany (Army), since 2006.
UAV-4. RQ-7B Shadow 200 UAV - United States (Marines), since 2008.

The following technical parameters are taken from the database of C:MA/NO [8]. Table 1 shows an overview of the general parameters. Note that the database does only contain a maximum climb rate, but not a maximum descend rate. Hence we assumed that both rates are equal, to fill the gap. The fuel consumption depending on the altitude band and the speed is given in Tables 2-5. There is no climb fuel surplus given in the database; the values for ξ_u in Table 1 are estimated from Jameson [15], who gives the values of 4.2 gal/hr for cruise and 4.4 gal/hr for climb, hence a fuel surplus of 0.2 gal/hr or roughly 5% (for a RQ-5A Hunter UAV). Our values for climb fuel are based on this estimation.

description	param.	unit	UAV-1	UAV-2	UAV-3	UAV-4
min speed	$\underline{v}_u$	km/h	130	167	120	111
max speed	$\overline{v}_u$	km/h	232	204	213	194
max climb rate	$\overline{v}_u^{z,+}$	m/s	8	2	3	3
max descend rate	$\overline{v}_u^{z,-}$	km/min	8	2	3	3
max altitude	$\overline{h}_u$	km	13.716	4.572	4.572	4.572
max fuel	F_u	kg	2000	150	65	35
climb fuel	ξ_u	kg/min	0.2	0.02	0.03	0.01
max oper. range	ϱ_u	km	185	185	185	185

Table 1: Technical parameters of the UAVs.

altitude band & throttle	altitude [km]	speed [km/h]	fuel cons. [kg/min]
band 1, loiter speed	0.001 - 3.658	130	2.08
band 1, cruise speed	0.001 - 3.658	204	2.6
band 1, military speed	0.001 - 3.658	232	5.23
band 2, loiter speed	3.658 - 7.315	130	1.53
band 2, cruise speed	3.658 - 7.315	204	1.91
band 2, military speed	3.658 - 7.315	232	3.85
band 3, loiter speed	7.315 - 10.972	130	1.06
band 3, cruise speed	7.315 - 10.972	204	1.33
band 3, military speed	7.315 - 10.972	232	2.66
band 4, loiter speed	10.972 - 13.716	130	0.7
band 4, cruise speed	10.972 - 13.716	204	0.88
band 4, military speed	10.972 - 13.716	232	1.78

Table 2: Performance details ($\eta_{u,i,j}$) for UAV-1.

altitude band & throttle	altitude [km]	speed [km/h]	fuel cons. [kg/min]
band 1, loiter speed	0.001 - 3.658	167	0.19
band 1, cruise speed	0.001 - 3.658	194	0.23
band 1, military speed	0.001 - 3.658	204	0.44
band 2, loiter speed	3.658 - 4.572	167	0.14
band 2, cruise speed	3.658 - 4.572	194	0.17
band 2, military speed	3.658 - 4.572	204	0.32

Table 3: Performance details ($\eta_{u,i,j}$) for UAV-2.

altitude band & throttle	altitude [km]	speed [km/h]	fuel cons. [kg/min]
band 1, loiter speed	0.001 - 3.658	167	0.29
band 1, cruise speed	0.001 - 3.658	194	0.38
band 1, military speed	0.001 - 3.658	204	0.6
band 2, loiter speed	3.658 - 4.572	167	0.22
band 2, cruise speed	3.658 - 4.572	194	0.28
band 2, military speed	3.658 - 4.572	204	0.44

Table 4: Performance details ($\eta_{u,i,j}$) for UAV-3.

altitude band & throttle	altitude [km]	speed [km/h]	fuel cons. [kg/min]
band 1, loiter speed	0.001 - 3.658	111	0.08
band 1, cruise speed	0.001 - 3.658	185	0.11
band 1, military speed	0.001 - 3.658	194	0.23
band 2, loiter speed	3.658 - 4.572	111	0.07
band 2, cruise speed	3.658 - 4.572	185	0.08
band 2, military speed	3.658 - 4.572	194	0.17

Table 5: Performance details ($\eta_{u,i,j}$) for UAV-4.

We set the horizontal safety distance between two UAVs to $\varepsilon^x, \varepsilon^y := 2$ km and the vertical separation to $\varepsilon^z := 0.5$ km for all pairs of UAVs, independent of their type.

The sensor range with respect to a waypoint $\delta_{u,w}$ is set a value in the second-highest altitude band (that is, 10 km for UAV-1 and 3 km for UAV-2-4). This forces the UAVs to leave the optimum (heighest) altitude band and the preferred flight at maximum altitude in order to come into the enforced range within the target. After making the contact, the UAV will usually climb up again, unless another waypoint is nearby, so that the saving of fuel from flying in a higher band would be overcompensated by the fuel that is spent on climbing up.

The flight area is a square with a side length of 300 km each. Each UAV has a hub (or launch/recovery point), which is its mission start and end location. The model allows that the start can be at different coordinates from the end, but we do not make use of that in our test instances. The coordinates are near the middle of the upper, lower, left, and right boundary of the area. The ground control unit is placed 50 km away from the hub.

A number of waypoints is generated randomly with a uniform distribution within the area with a minimum distance of 10km among each other. Then the earliest and latest point in time for claiming the waypoint by an UAV is also chosen randomly with a uniform distribution on the time window $\mathcal{T}$. It is made sure that each time window has a certain minimal width of at least 5 time steps and a maximal width of at most $0.75T$.

A certain number of uniformly distributed rectangular no-flight zones is generated. Here it is ensured that these rectangles do not contain waypoints (which would trivially exclude a waypoint from the problem instances). Furthermore, the zones have a certain minimum size, which is related to length of a time step and the maximum speed of the fastest UAV. In short, any UAV should not be allowed to cross the no-flight zone, because this condition is only checked at the discrete time steps (this would be an artifact of the time discretization).

The direction of the wind is generated randomly, where its magnitude is always 45 km/h, which is a strong breeze (Beaufort number 6). Since UAVs are typically slower than manned aircraft, they are more susceptible to higher wind speeds. With this value for the wind speed, we further challenge the mission planning.

6 Computational Results

The linearized version of the above model is a mixed-integer linear programming problem, hence numerical solvers such as CPLEX, GUROBI or XPRESS can be deployed to find feasible solutions and an estimation of the solution's quality, which ideally is a certificate of optimality. The hardware environment we used was an Apple MacBookPro Laptop running an Intel Core i7 at 3.10 GHz clock speed and 16 GB RAM. On this hardware, we made experiments with all three solvers, and found out that GUROBI was the fastest and CPLEX the slowest (and XPRESS in the middle, but closer to GUROBI than to CPLEX). Hence we focused on using GUROBI, and all results presented in this section were obtained from that solver. All instances were generated with the modeling language AMPL (Version 20180822).

6.1 Analyzing the CPU Time Sensitivity

In this section we analyze the runtime behavior and its sensitivity with respect to the variance of those input parameters that determine the size of an instance in terms of numbers of variables and constraints. In general, the larger an instance, the more time it consumes for solving it to proven optimality. Here we aim at a closer look in order to find out to what extend a state-of-the-art off-the-shelf mixed-integer linear solver (such as GUROBI) can be applied to compute solutions for the mission planning problem within a decent amount of CPU time.

6.1.1 Number of Waypoints

In the first set of experiments, we fix the number of UAVs (2), the number of no-flight areas (2), the number of time steps (40), and only vary the number of waypoints between 4 and 30. For each number of waypoints, we randomly generate 41 instances. The CPU time for each run, together with three common mean values is reported in Figure 3. The total CPU time of all 1,107 runs together is 1,886,300 seconds. For up to 10 waypoints, a solution could be found in short time (less than 1,000 seconds). Starting at 11 waypoints, the CPU time grows significantly, and

some instances could not be solved within the given time limit of 3,600 seconds; in these cases the remaining gap is reported as a number between 0 and 100%. Each additional waypoint increases the solution time on average. However, there is a large spread for the CPU time from the slowest to the fastest solvable instance, and even for the largest instance size (having 30 waypoints) there are some that can be solved within a very short amount of time.

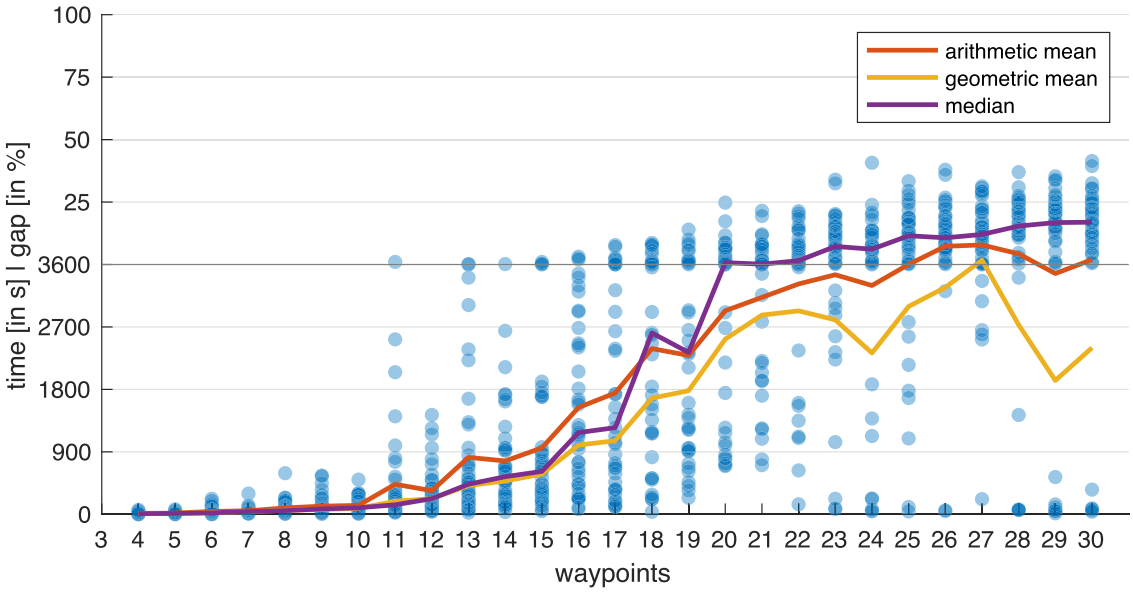

Figure 3: Solution time depending on the number of waypoints.

6.1.2 Number of Restricted Areas

In these experiments, we fix the number of UAVs (2), the number of waypoints (15), the number of time steps (40), and vary the number of restricted areas between 0 (none) and 5, where 41 instances were generated each. Interestingly, it turns out that the CPU time is in fact independent from the number of restricted airspaces, as can be seen in Figure 4. The total amount of CPU time for solving all 246 instances was 183,620 s.

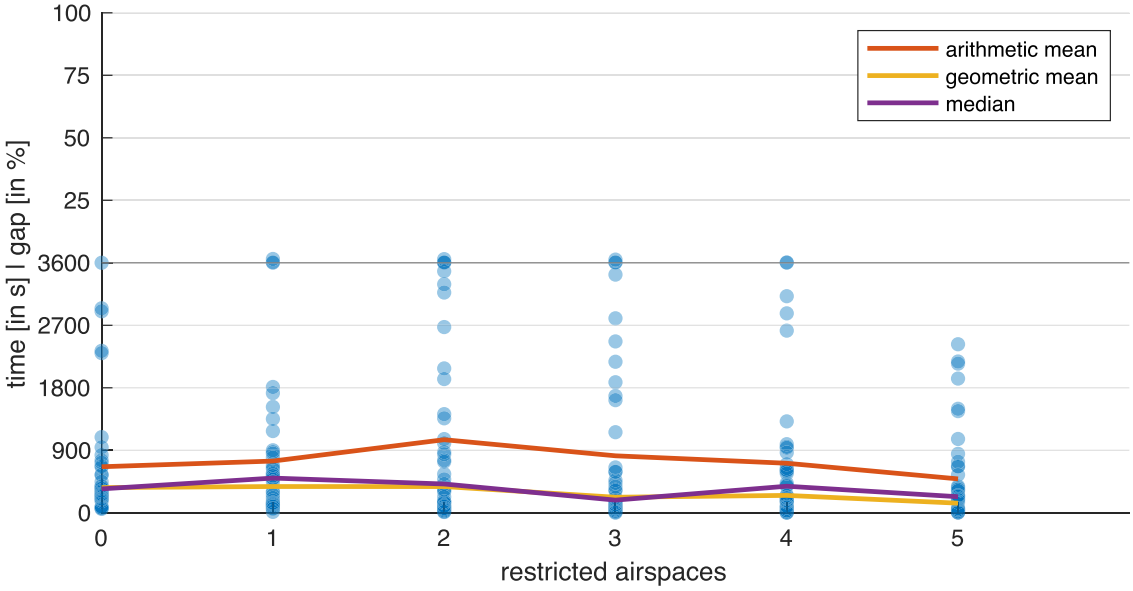

Figure 4: Solution time depending on the number of restricted airspaces.

6.1.3 Number of UAVs

In a final set of experiments, we vary the number of UAVs between 1 and 4, while keeping the number of waypoints (15), the number of time steps (40), and the number of restricted areas (2). In Figure 5 one can see that the number of UAVs is, similar to the number of waypoints, a driving force for the CPU time. The total amount of CPU time for solving all 164 instances was 293,400 s.

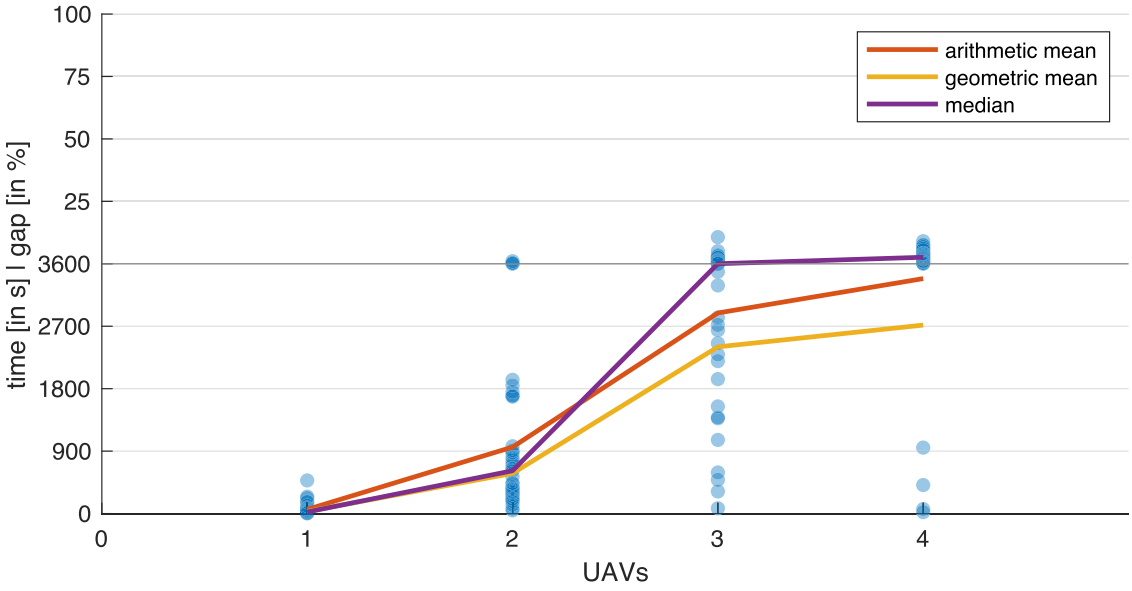

Figure 5: Solution time depending on the number of UAVs.

6.2 Analyzing a Particular Solution

We take a closer look on one of solutions from above, with the following settings: 2 UAVs, 15 waypoints, 2 restricted airspaces, and 40 timesteps (where each time step represents 10 minutes in real life). We run this experiment in two ways, one with zero wind-speed, and one with a wind of 45 km/h from North-West. Figure 6 shows the locations where UAV-1 and UAV-2 start and end their missions, denoted by Hub-1 and Hub-2, respectively. The ground control units are GC-1 and GC-2, respectively, and have a radius of 185 km, indicated by the two blue circles with center at GC-1 and GC-2. The two restricted areas are shown as red rectangles and denoted by F-1 and F-2. The waypoints WP-1, ..., WP-15 are randomly distributed in the area. The dark grey lines show the flight paths of the UAVs, and the arrows indicate the flight direction. The total length of these trajectories are 751.49 km for UAV-1 and 369.52 km for UAV-2 in the no-wind scenario (top of Figure 6) and 692.62 km for UAV-1 and 366.26 km for UAV-2 in the presence of wind (bottom of Figure 6). Note that UAV-1 has a higher maximum speed compared to UAV-2, and consequently it covers a larger distance and collected more waypoints. This solution is a proven global optimal one, which was found after 363 s (no wind) and 1986 s (with wind).

The details of the waypoints and their time windows can be found in Figure 7. The time window in which a waypoint can be visited is indicated by a thick dark line, and the moment a UAV has a fly-by is marked by a red diamond symbol. UAV-1 is able to collect 9 waypoints (no wind) and 8 waypoints (with wind), UAV-2 collects 4 waypoints in either way, and the remaining 2 (no wind) or 3 (with wind) waypoints are too far-off and have too narrow time windows to be covered from any of the two UAVs.

The third dimension (flight altitude) can be seen as blue lines in Figure 8. In order to save as much fuel as possible, the UAVs have to fly in the highest possible altitude. We set the range of sensors of the UAVs on purpose in such way that the UAV has to fly in a much lower and thus suboptimal altitude in order to be close enough to the target. These changes in altitude can clearly be seen in Figure 8, where the waypoints are marked as red squares on the horizontal axis, and the altitude goes particularly down when a UAV is over a waypoint, and goes up again after the contact.

Figure 9 shows the velocity of the UAVs. The fuel consumption is lowest when flying at the lowest possible speed. However, the tight time windows of the waypoints (see Figure 7) force the UAVs to take higher speeds in order to reach them in time.

The fuel consumption can be found in Figure 10 in two ways: as instantaneous fuel consumption rate in kg/min (red line), and as the remaining amount of fuel on board in kg (blue line). Note that there is one waypoint (WP-7) that cannot be covered in the presence of wind, hence the trajectory of UAV-1 is approx. 60 km shorter and the fuel consumption (see Figure 10) is less by approx. 200 kg. Compared to a 8% shorter trajectory (751.49 km vs. 692.62 km), the amount of consumed fuel is about 42% less (450 kg vs. 260 kg), which cannot be explained by the trajectory length alone. When comparing the velocities in Figure 9, one can see that UAV-1 must fly with higher velocities on average in order to reach all 9 waypoints (no wind). Once it is decided that only 8 of these waypoints can be covered (with wind), the trajectory is shorter and additionally can be traversed at a lower speed. Both effects add up to the aforementioned fuel savings. For UAV-2 however, the waypoints are the same in both cases (with or without wind), and here the trajectories are of almost identical length (369.52 km vs. 366.26 km), and also the consumed fuel differs only a little (29 kg vs. 33 kg). The surplus of approx. 10% can be explained from flying against the wind, which also results in higher velocities for UAV-2 on average in the presence of wind.

7 Conclusions

We studied the flight trajectory and mission planning problem for a fleet of UAVs and gave a mixed-integer nonlinear formulation, where the nonlinearities are due to the Euclidean norm, which can be linearized by approximations using the 1-norm and the max-norm. Numerical tests revealed to what instance sizes current state-of-the-art MILP solvers are able to deal with on standard desktop computer hardware. It turned out that typical instance sizes (for our project partner), which are planning problems with up to 15 waypoints and 2 UAVs, can be routinely solved within one hour on a standard computer using standard software. For larger problems with more waypoints or more UAVs, one has to take suboptimal solutions into account. As long as the gap is small, such solutions are still useful in practice.

Acknowledgement. This research was supported by the German Federal Ministry of Education and Research (BMBF) project "E-Motion" and the German Research Foundation (DFG) project FU860/1-1. We thank our colleagues from the Planungsamt der Bundeswehr, Ottobrunn, and from the Aufklärungslehrbataillon 3, Lüneburg, for fruitful discussions. Many thanks to the anonymous reviewer for carefully reading the manuscript and his various helpful comments for improving it.

References

[1] J. D. Blom. Unmanned Aerial Systems : a historical perspective. Technical report, Combat Studies Institute Press, US Army Combined Arms Center, Fort Leavenworth, Kansas, 2010.

[2] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L. T. Biegler. MILP and NLP Techniques for Centralized Trajectory Planning of Multiple Unmanned Air Vehicles. In *IEEE American Control Conference 2006, Minneapolis, MN*, 2006.

[3] E. Brommundt, G. Sachs, and D. Sachau. *Technische Mechanik - Eine Einführung.* Oldenbourg Wissenschaftsverlag GmbH, 2007.

[4] M. E. Celebi, F. Celiker, and H. A. Kingravi. On Euclidean Norm Approximations. *Pattern Recognition*, 44(2):278–283, 2011.

[5] D. Chaudhuri, C. A. Murthy, and B.B. Chaudhuri. A Modified Metric to Compute Distance. *Pattern Recognition*, 25(7):667–677, 1992.

[6] J. A. Cobano, D. Alejo, R. Conde, and A. Ollero. A new method for UAV trajectory planning under uncertainties. In *Proceedings of the Workshop on Research, Development and Education on Unmanned Aerial Systems (RED-UAS). Seville, November 30 and December 1*, 2011.

[7] K. F. Culligan. Online Trajectory Planning for UAVs Using Mixed Integer Linear Programming. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2006.

[8] D. Drandis. Command: Modern Air/Naval Operations 1.14.7, Database Build 476. Matrix Games, Januar 9, 2019.

[9] L. E. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[10] L. Evers, A. I. Barros, H. Monsuur, and A. Wagelmans. UAV Mission Planning: From Robust to Agile. In V. Zeimpekis, G. Kaimakamis, and N. J. Daras, editors, *Military Logistics*, Interfaces Series 56, pages 1–17. Springer International Publishing, Switzerland, 2015.

[11] M. Faied, A. Mostafa, and A. Girard. Vehicle Routing Problem Instances: Application to Multi-UAV Mission Planning. In *AIAA Guidance, Navigation, and Control Conference, 2 - 5 August 2010, Toronto, Ontario Canada*, 2010.

[12] E. J. Forsmo. Optimal Path Planning for Unmanned Aerial Systems. Master's thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics, 2012.

[13] X.-Z. Gao, Z.-X. Hou, X.-F. Zhu, J.-T. Zhang, and X.-Q. Chen. The Shortest Path Planning for Manoeuvres of UAV. *Acta Polytechnica Hungarica*, 10(1):221–239, 2013.

[14] B. R. Geiger, J. F. Horn, A. M. DeLullo, L. N. Long, and A. F. Niessner. Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming. Technical Report 2006-6199, American Institute of Aeronautics and Astronautics, 2006.

[15] T. Jameson. A Fuel Consumption Algorithm for Unmanned Aircraft Systems. Technical Report ARL-TR-4803, Army Research Laboratory, 2009.

[16] M. Jun and R. D'Andrea. Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments. In S. Butenko, R. Murphey, and P. Pardalos, editors, *Cooperative Control: Models, Applications and Algorithms*, pages 95–111. Kluwer, 2004.

[17] J. F. Keane and S. S. Carr. A Brief History of Early Unmanned Aircraft. *Johns Hopkins APL Technical Digest*, 32(3):558 – 571, 2013.

[18] M. Kress and J.O. Royset. Aerial Search Optimization Model (ASOM) for UAVs in Special Operations. *Military Operations Research*, 13(1):23–33, 2008.

[19] E. Ładyżyńska-Kozdraś. Modeling and Numerical Simulation of Unmanned Aircraft Vehicle Restricted by Non-holonomic Constraints. *Journal of Theoretical and Applied Mechanics*, 50(1):251–268, 2012.

[20] C. Landry, W. Welz, and M. Gerdts. A coupling of discrete and continuous optimization to solve kinodynamic motion planning problem. Technical report, WIAS Preprint 1900, 2013.

[21] J. Lee, R. Huang, A. Vaughn, X. Xiao, J. K. Hedrick, M. Zennaro, and R. Sengupta. Strategies of Path-Planning for a UAV to Track a Ground Vehicle. In *Proceedings of the 2nd annual Autonomous Intelligent Networks and Systems Conference (AINS), Menlo Park, CA*, 2003.

[22] B. Luders. Robust Trajectory Planning for Unmanned Aerial Vehicles in Uncertain Environments. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008.

[23] G. Ma, H. Duan, and S. Liu. Improved Ant Colony Algorithm for Global Optimal Trajectory Planning of UAV under Complex Environment. *International Journal of Computer Science & Applications*, 4(3):57–68, 2007.

[24] L. R. Newcome. *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*. General Publication S. American Institute of Aeronautics and Astronautics, 2004.

[25] A. J. Pohl and G. B. Lamont. Multi-Objective UAV Mission Planning Using Evolutionary Computation. In *Proceedings of the INFORMS 2008 Winter Simulation Conference, Miami, FL*, 2008.

[26] F. Rhodes. On the Metrics of Chaudhuri, Murthy and Chaudhuri. *Pattern Recognition*, 28(5):745–752, 1995.

[27] J. J. Ruz, O. Arévalo, G. Pajares, and J. M. de la Cruz. UAV Trajectory Planning for Static and Dynamic Environments. In Thanh Mung Lam, editor, *Aerial Vehicles*. InTech, Rijeka, Croatia, 2009.

[28] S. Sarwar, S. ur Rehman, and S. F. Shah. Mathematical Modelling of Unmanned Aerial Vehicles. *Mehran University Research Journal of Engineering & Technology*, 32(4):615–622, 2013.

[29] F. Schøler. *3D Path Planning for Autonomous Aerial Vehicles in Constrained Spaces*. PhD thesis, Section of Automation & Control, Department of Electronic Systems, Aalborg University, 2012.

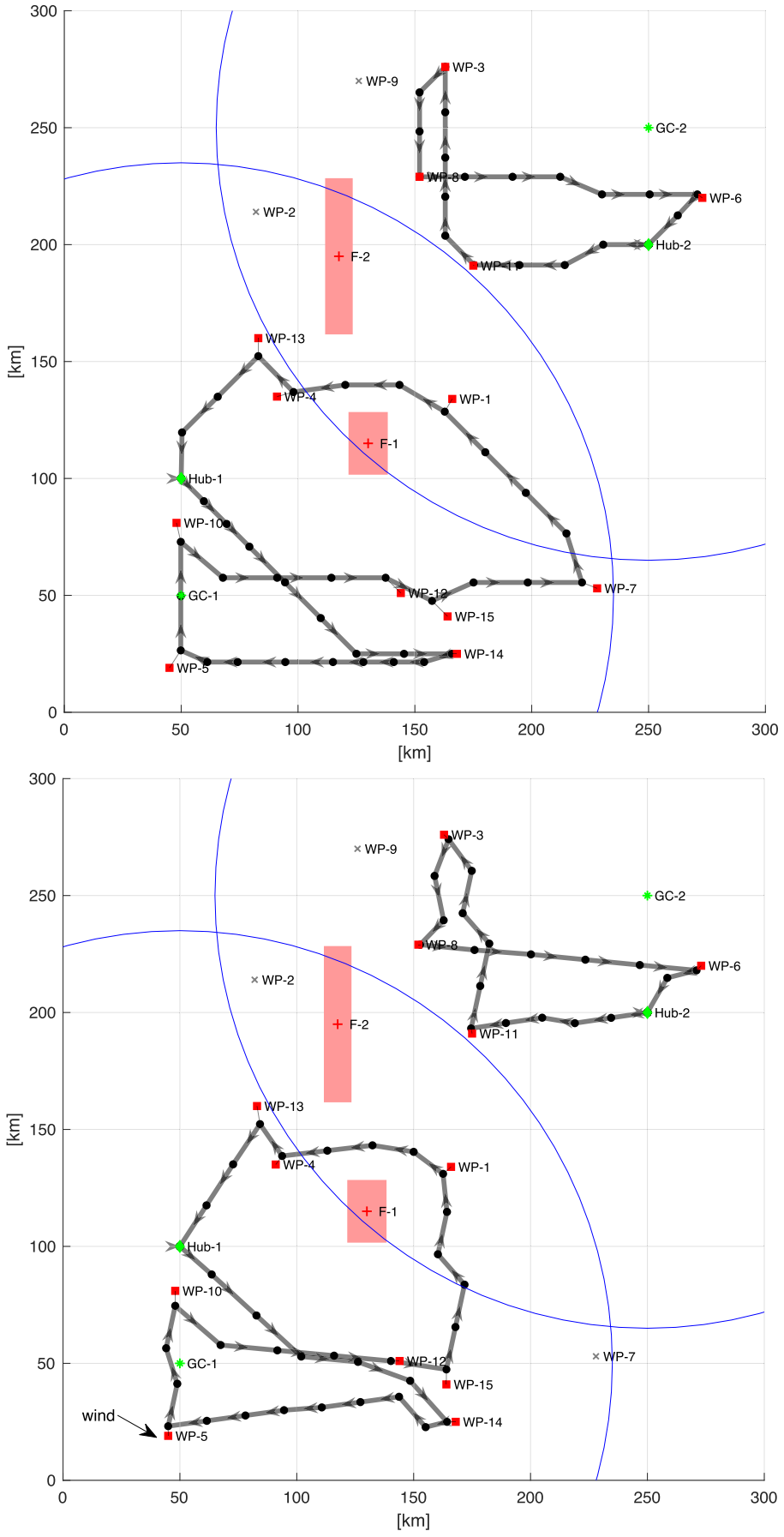

Figure 6: Mission planning for UAV-1 and UAV-2, 15 waypoints, 2 restricted airspaces. Top: without wind, bottom: with wind.

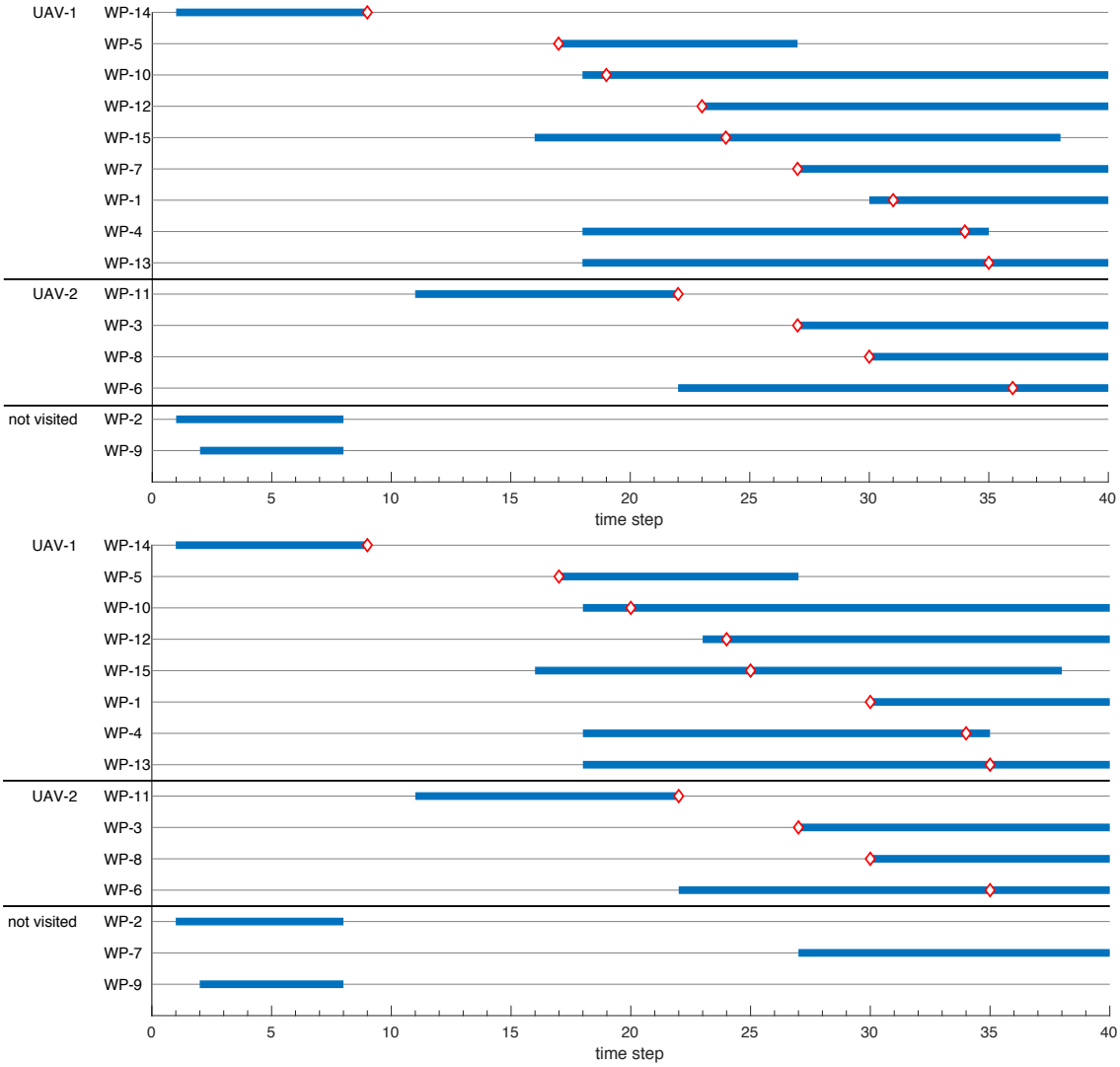

Figure 7: Time windows of waypoints and visiting times for UAV-1 and UAV-2 for mission in Figure 6. Top: without wind, bottom: with wind.

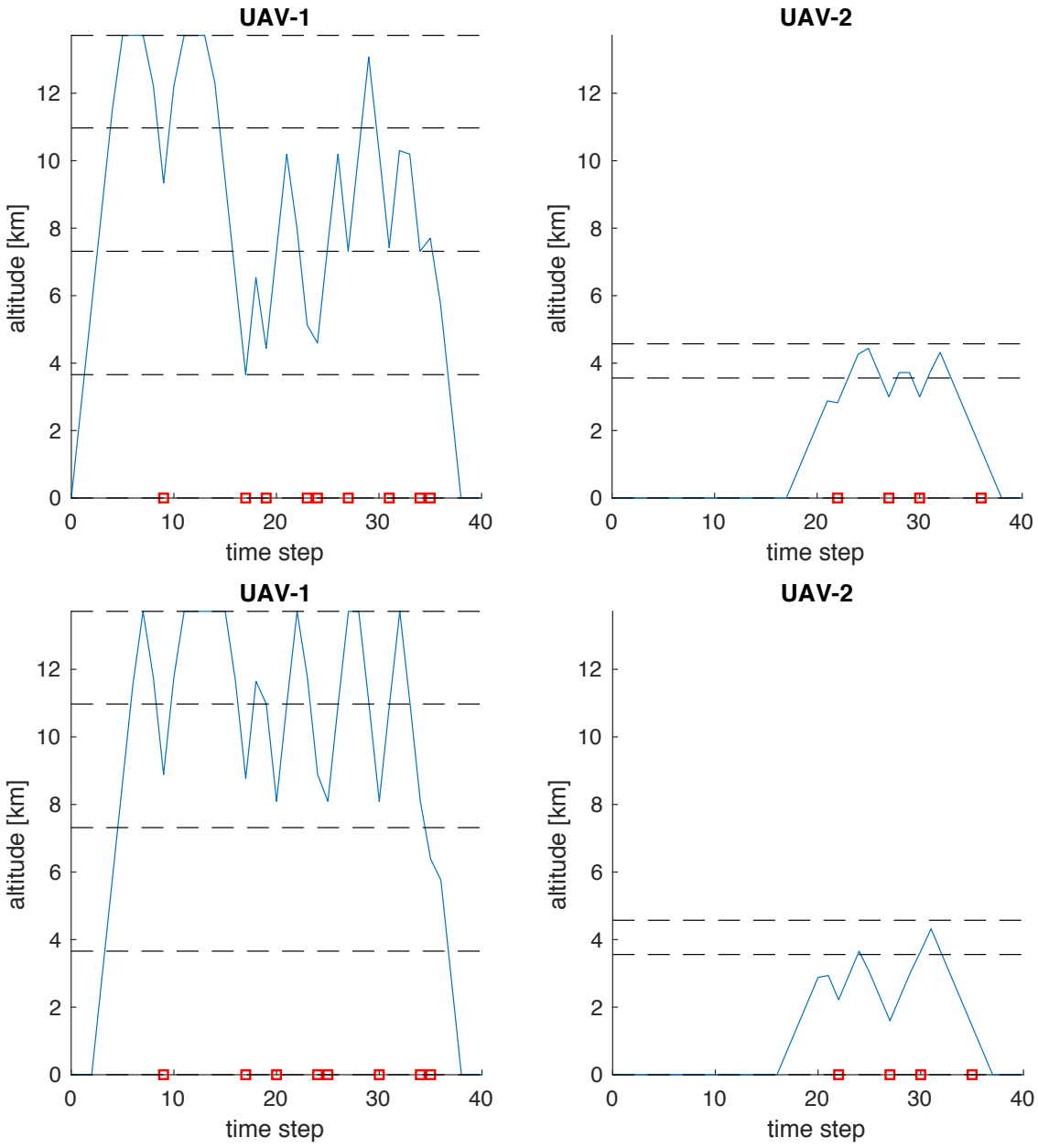

Figure 8: Altitudes of UAV-1 and UAV-2 for mission in Figure 6. Top: without wind, bottom: with wind.

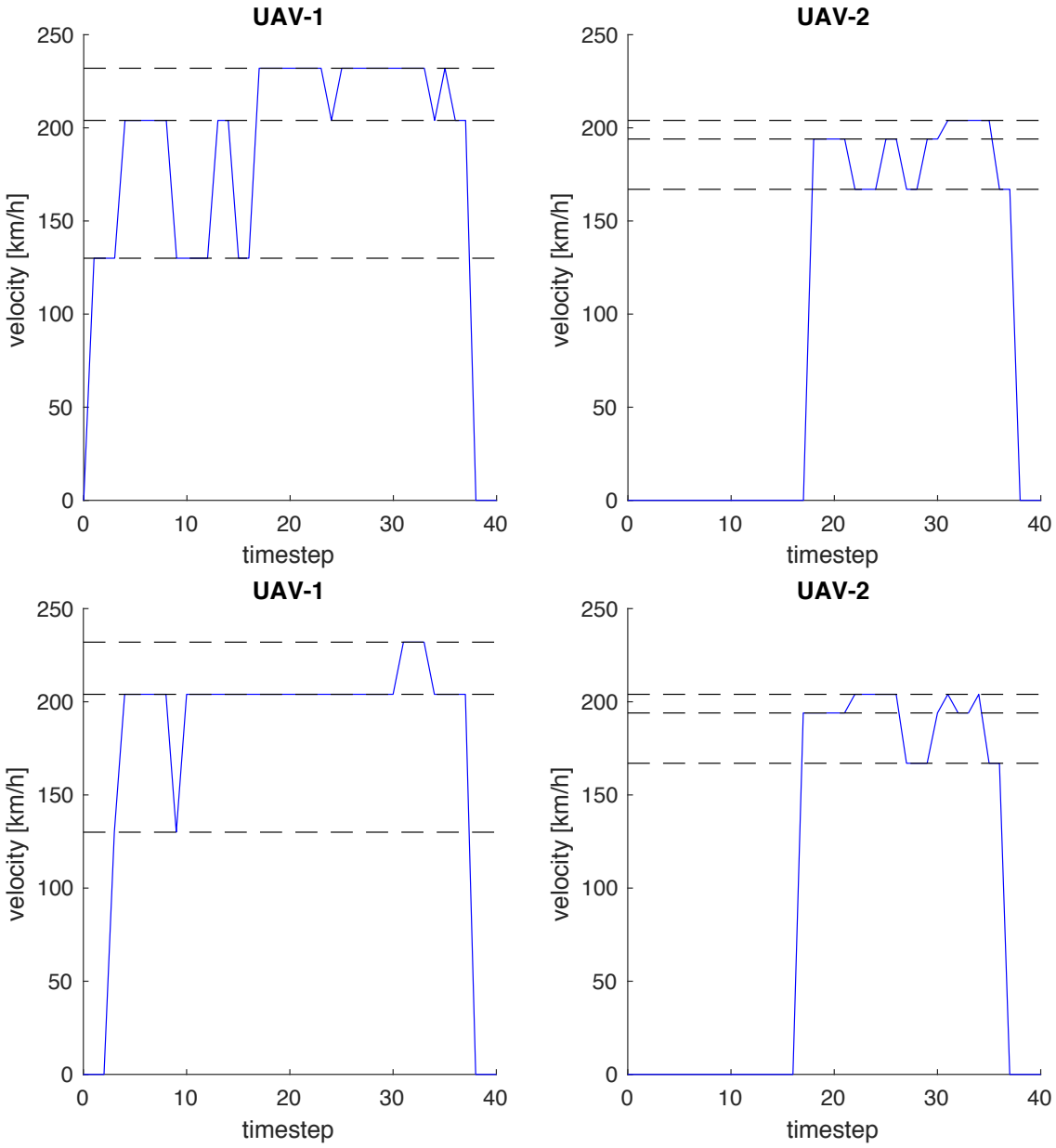

Figure 9: Velocities of UAV-1 and UAV-2 for mission in Figure 6. Top: without wind, bottom: with wind.

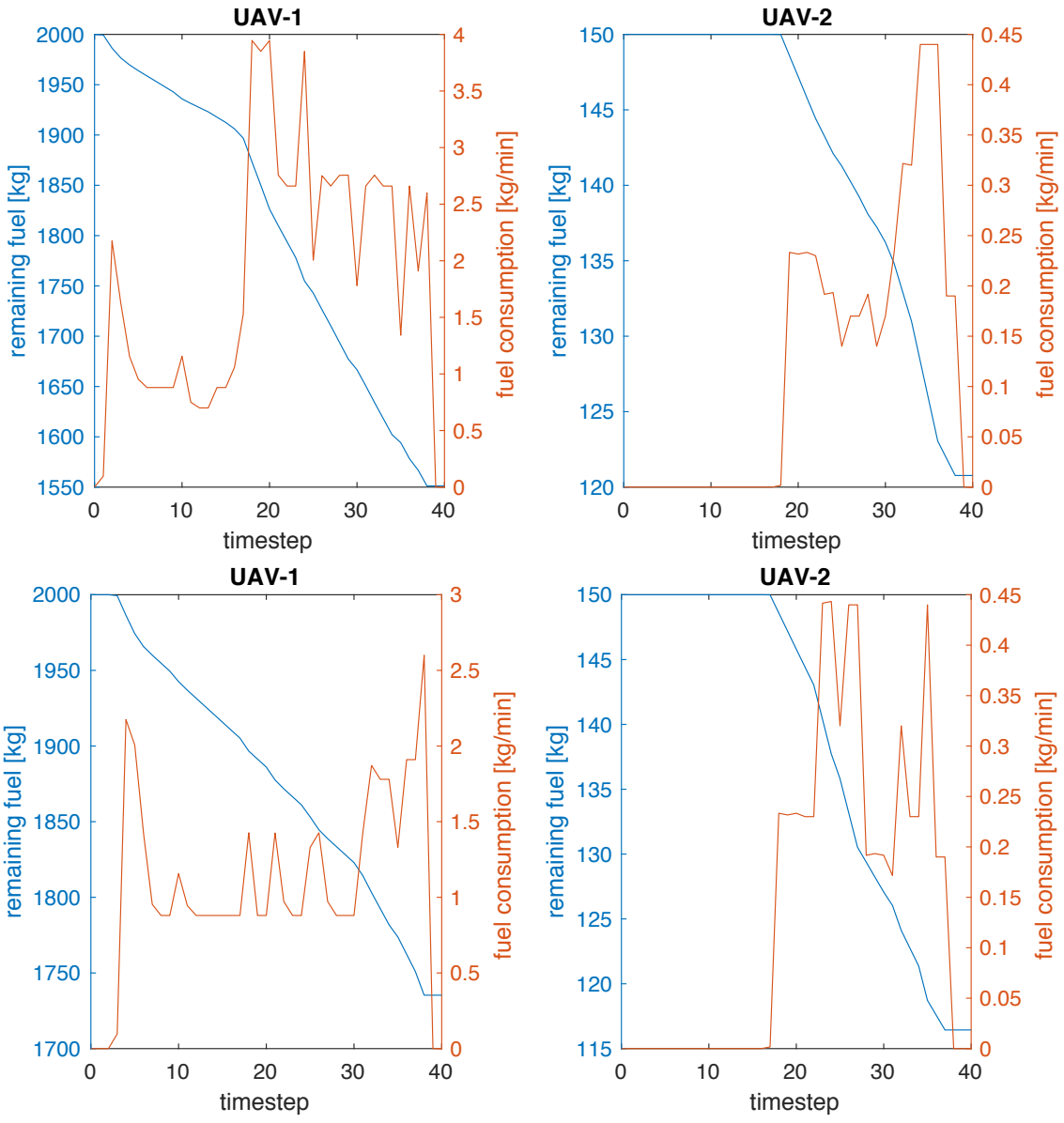

Figure 10: Fuel consumption rate and remaining fuel of UAV-1 and UAV-2 for mission in Figure 6. Top: without wind, bottom: with wind.

IMPRESSUM

Brandenburgische Technische Universität Cottbus-Senftenberg
Fakultät 1 | MINT - Mathematik, Informatik, Physik, Elektro- und Informationstechnik
Institut für Mathematik
Platz der Deutschen Einheit 1
D-03046 Cottbus

Professur für Ingenieurmathematik und Numerik der Optimierung
Professor Dr. rer. nat. Armin Fügenschuh

E fuegenschuh@b-tu.de
T +49 (0)355 69 3127
F +49 (0)355 69 2307

Cottbus Mathematical Preprints (COMP), ISSN (Print) 2627-4019
Cottbus Mathematical Preprints (COMP), ISSN (Online) 2627-6100

www.b-tu.de/cottbus-mathematical-preprints
cottbus-mathematical-preprints@b-tu.de
doi.org/10.26127/btuopen-4828