# Glider Routing and Trajectory Optimisation in Disaster Assessment

Walton Pereira Coutinho[a,*], Jörg Fliege[a], Maria Battarra[b]

[a] *University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom*
*{w.p.coutinho, j.fliege}@soton.ac.uk*
[b] *University of Bath, Claverton Down, Bath, BA2 7AY, United Kingdom*
*m.battarra@bath.ac.uk*

**Abstract**

In this paper, we introduce the Glider Routing and Trajectory Optimisation Problem (GRTOP), the problem of simultaneously finding optimal routes and trajectories for a fleet of gliders with the aim of surveying a set of locations. We propose a novel Mixed-Integer Nonlinear Programming (MINLP) formulation for the GRTOP, which optimises the routes as well as the trajectories along these routes, while flight dynamics is modelled as constraints. We avoid solving a non-convex problem by linearising the gliders' flight dynamics, converting the proposed MINLP into a Mixed-Integer Second-order Cone Programming (MISOCP) problem. To allow for a more tractable formulation, the dynamical constraints are relaxed and a penalisation is added to the objective function. Several different discretisation techniques are compared. The formulation is tested on instances inspired by risk maps of flooding-prone cities across the UK and on 180 randomly generated instances.

*Keywords:* OR in disaster relief, unmanned gliders, routing, trajectory optimisation

## 1. Introduction

In this paper we propose the Glider Routing and Trajectory Optimisation Problem (GRTOP), which consists of simultaneously finding optimal routes and trajectories for a fleet of unmanned aerial gliders. Gliders are Unmanned Aerial Vehicles (UAVs) without on-board propulsion and, in the GRTOP, they are due to visit a set of locations.

This problem arises from disaster assessment applications, in which camera-equipped gliders survey a number of risky locations in a post-disaster situation. The collected information can be used to assess the severity of the effects in the aftermath of a disaster. This problem was motivated by discussions with the Royal National Lifeboat Institution, the leading UK charity in providing flood rescue response, among other services.

Controlled powered drones have been used for raising emergency response in several scenarios, see for example, Chowdhury et al. (2017) and the recent Grenfell Tower disaster (Laville et al., 2017, June 15). However, these drones are expensive and often require experienced pilots to be operated. In aerial survey operations, a fleet of camera-equipped low cost balloon-launched autonomous gliders (Crispin, 2016) can be 3D printed (Keane et al., 2017) and do not require a specialised team to be operated. We believe this solution concept allows for rapid response to disasters.

[*] Corresponding author

The GRTOP integrates two well known problems in operational research and computational optimisation, namely the Vehicle Routing Problem (VRP) and the Trajectory Optimisation Problem (TOP). The VRP consists of determining the minimum cost routes to serve a set of locations. Routes are ordered sequences of locations and each route is departing and returning to the depot. Distances (or travel times) between locations (and the depot) are typically known in advance. UAVs' routes do not necessarily satisfy trajectory constraints (e.g., flight dynamics). The TOP consists in defining the motion of an object over time (trajectory) such that a measure of performance is optimised while respecting boundary conditions, trajectory constraints and flight dynamics. By ensuring that routes respect trajectory constraints, the GRTOP allows to define flyable routes for a fleet of gliders.

In Coutinho et al. (2018), a review on UAV routing and UAV TOPs is provided. Moreover, the authors introduce a Multi-phase Mixed Integer Optimal Control formulation for the UAV Routing and Trajectory Optimisation Problem (UAVRTOP) and a taxonomy devoted to UAV routing, task assignment, path planning and trajectory optimisation of UAVs. The authors show that there is a lack of research integrating UAV routing and trajectory optimisation.

Many TOPs are non-convex in nature, such as most of the aerospace engineering-related problems (Conway, 2010; Shaw-Cortez & Frew, 2015). Several optimisation techniques based on Optimal Control (OC) and Non-linear Programming (NLP) have been developed to tackle TOPs. We refer the interested reader to Betts (2001) for a complete overview of those methods. NLP-based solution methods are known to be sensitive to initial guesses, i.e., convergence can be only ensured if a proper initialisation is provided (Zhao, 2004). Constructing a good set of initial guesses for flying vehicles often requires expertise on flight dynamics. In order to overcome such difficulties, the non-linear dynamics of UAVs is often linearised, see, e.g., Hajiyev et al. (2015), How et al. (2015) and Harris & Acikmese (2013).

In this paper, we propose a single-phase Mixed-Integer Non-linear Programming (MINLP) formulation for the GRTOP. Mixed-Integer Programming (MIP) has been already applied for solving TOPs and OC problems, see, e.g., Keviczky et al. (2008), Soler et al. (2014), Fügenschuh & Müllenstedt (2015), Yuan et al. (2015) and Maolaaisha (2015). Coutinho et al. (2018) provided a detailed description of other papers dealing with routing and trajectory optimisation in an integrated framework.

Our formulation simultaneously optimises routes and the flight trajectories along these routes. The flight dynamics of the fleet of gliders are modelled as dynamical constraints, while the field of view of the cameras is modelled as second-order cone constraints. We avoid dealing with non-convex dynamical constraints by linearising the gliders' Equations of Motion (EOM), reducing the proposed MINLP into a Mixed-Integer Second-Order Cone Programming (MISOCP) problem. To allow for a more tractable formulation while keeping high quality solutions, we relax the resulting linear dynamical constraints and add a corresponding penalisation term to the objective function. Next, we study several integration methods for solving the EOMs.

Motivated by our application, we consider a number of real-life instances based on flood risk maps of cities in the UK as well as 180 randomly generated instances. We perform computational experiments to test alternative commercial MISOCP solvers and the performance of the integration methods when embedded into the MISOCP problem. Further computational experiments are performed to test different discretisation sizes and associated errors.

The remainder of this paper is organised as follows. In Section 2, we provide a brief introduction to flight dynamics and present the glider's EOMs. A MINLP for the GRTOP is introduced on Section 3. In Section 4, we linearise the glider's EOMs and present different integration methods for the linear dynamics. In Section 5, the resulting MISOCP formulation is then tested on several randomly generated and real-life-based instances. In Section 6, we conclude this work and highlight future research venues.

## 2. Gliders' Flight Dynamics

The four basic forces acting on an aircraft during flight are *thrust, drag, lift* and *weight*, as depicted in Figure 1a. Thrust is the force generated by the on-board propulsion of the aircraft itself. Drag is the air resistance acting upon the airplane's fuselage. Lift is the force generated by airflow through the control surfaces (flaps, ailerons, elevators and rudders), allowing the aircraft to fly. The weight models the force pulling the aircraft to the centre of the Earth. For a glider, thrust is absent, since there is no engine on board.

An aircraft is said to fly in *equilibrium* (a.k.a. steady-state flight) when the basic forces balance each other out. For instance, a powered steady-level flight can be achieved when lift equals weight and thrust equals drag. In simple terms, two types of equilibrium can be described, *static* and *dynamic*. Static equilibrium is related to the absence of velocity (static position). Dynamic equilibrium is related to the absence of acceleration (e.g., an object moving at constant velocity). In order to find steady-state conditions, we assume that gliders fly in dynamic equilibrium.

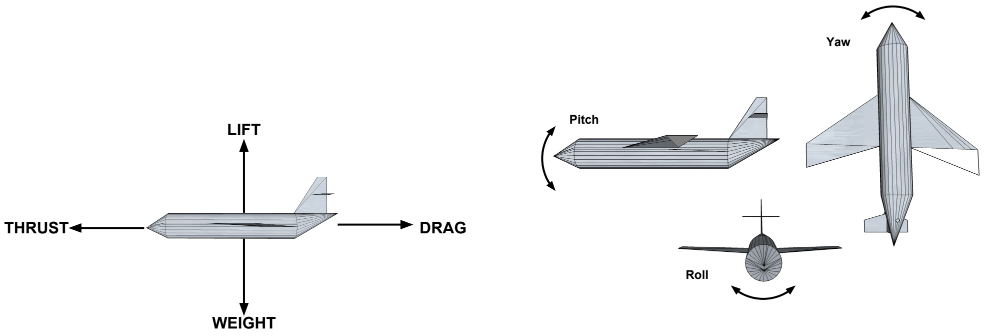
By activating the control surfaces of the aircraft, one can change its angular orientation. Angular orientation can be defined, for example, in terms of Euler angles, namely *pitch, yaw* and *roll*, with respect to a fixed reference frame. Alternative representations can also be applied, e.g., *quaternions* and *rotation matrices*, but they will not be considered in this paper. Figure 1b depicts the planes of actuation of each Euler angle.

The control of an aircraft's horizontal orientation is usually described in terms of the Angle-of-Attack (AoA). The AoA is the difference between the pitch angle and the flight path angle when the flight path angle is referenced to the atmosphere (Boeing, 2000)[1]. We assume that the AoA can be written as a function of the *lift coefficient* ($C_L$). This is a common assumption in the flight dynamics literature (Stengel, 2004). In simple terms, the lift coefficient describes the amount of lift generated by an aircraft's wings. We refer the interested reader to the books by Russell (1996) and Stengel (2004) for a more detailed understanding of aircraft flight dynamics.

### *2.1. Gliders' Equations of Motion*

In this paper, we apply the EOMs for glider dynamic soaring flight presented by Zhao (2004) in order to model the motion of the gliders. These equations have been extensively used by the gliding flight literature, see, for instance, Zhao & Qi (2004), Gordon (2006), Deittert et al. (2009), Sukumar & Selig (2013) and Gao et al. (2014). With the recent developments in UAV technology, this technique has

[1] (Stengel, 2004, chap. 1,p. 3) defines the AoA as the aircraft-relative vertical angle between the centreline of the aircraft and the relative wind. In turn, the relative wind is defined as the aircraft's velocity relative to the surrounding air.



(a) Aerodynamic forces. (b) Planes of actuation of the Euler angles.

Figure 1: Relevant forces and angles in an aircraft's flight.

become popular for autonomous gliding flight. Several studies have acknowledged the use of autonomous gliders for different purposes, see, e.g., Langelaan (2007), Chakrabarty & Langelaan (2011) and Crispin (2016).

The model presented by Zhao (2004) includes the following assumptions: the circumference of the earth is negligible compared to the range of flight, the density of the air can be considered constant, the wind is stationary and the mass of the glider does not change during the flight. Without loss of generality, one can assume that only the horizontal component of the wind is present and that it can be described by a linear profile as a function of the flight altitude.

Let us define the *state* of a glider at time $\tau \in \mathbb{R}_{\geq 0}$ as a *state vector* $\mathbf{y}(\tau) = (x(\tau), y(\tau), h(\tau), v(\tau), \gamma(\tau), \varphi(\tau))^\top$, where the first three components $x(\tau), y(\tau)$ and $h(\tau)$ of $\mathbf{y}(\tau)$ represent the position of the glider and $v(\tau) \in \mathbb{R}_{\geq 0}$ denotes its relative velocity (a.k.a. *airspeed*). Finally, $\gamma(\tau) \in \mathbb{R}$ is the flight path angle and $\varphi(\tau) \in \mathbb{R}$ is the heading angle. The *controls* (or input) to the system are represented by the *control vector* $\mathbf{u}(\tau) = (C_L(\tau), \mu(\tau))^\top$, where $C_L(\tau) \in \mathbb{R}$ is the lift coefficient and $\mu(\tau) \in \mathbb{R}$ the bank angle. A North-East-Down coordinate frame is used to define the inertial position (Shaw-Cortez & Frew, 2015), while the airspeed is modelled in a wind relative frame (Deittert et al., 2009). These are depicted in Figure 2. In the following, the notation “ ˙ ” is used to represent time derivatives of the variables.

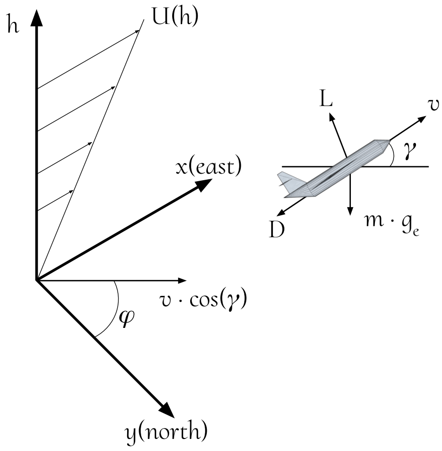


Figure 2: Coordinate frames.

The glider model used in this paper is based on the designs proposed by Bower (2010) and Flanzer

(2012). State and control variables are lower bounded by $\mathbf{y}_{lb} = (x_{lb}, y_{lb}, h_{lb}, v_{lb}, \gamma_{lb}, \varphi_{lb})$ and $\mathbf{u}_{lb} = (C_{L,lb}, \mu_{lb})$, and upper bounded by $\mathbf{y}_{ub} = (x_{ub}, y_{ub}, h_{ub}, v_{ub}, \gamma_{ub}, \varphi_{ub})$ and $\mathbf{u}_{ub} = (C_{L,ub}, \mu_{ub})$, respectively. These values are shown in Table 1. Airspeed is measured in metres per second $(m/s)$ and angles in radians. We compute the *aerodynamic coefficient* by using Equation (1), see Kroo (2001). The *Oswald factor* $e$ is computed using the Matlab function available in Sartorius (2013). Finally, $b_w = 2.49$ is the glider's wing span and $S_w$ the wing area.

$$k_A = \frac{1}{\pi e \frac{b_w^2}{S_w}}. \tag{1}$$

Table 1: Lower and upper bounds on state and control variables.

| | $x$ | $y$ | $h$ | $v$ | $\gamma$ | $\varphi$ | $C_L$ | $\mu$ |
|---|---|---|---|---|---|---|---|---|
| *lb* | *free* | *free* | 0 | 5 | -1.4 | -6.28 | -0.2 | -1.05 |
| *ub* | *free* | *free* | *free* | 40 | 1.4 | 6.28 | 1.17 | 1.05 |

The *wind strength* coefficient ($\beta$) is chosen so as to linearly approximate the average wind gradient profile for the UK, as suggested by Drew et al. (2013), for a reference altitude of approximately 500 metres. The remaining model parameters and their meaning are summarised in Table 2.

The EOMs of a glider can be expressed by Equations (2–12). For the sake of simplicity of notation, we omit the dependence on time ($\tau$) from state, control and auxiliary variables.

Table 2: Environmental and glider constants.

| **Symbol** | **Value** | **Description** | **Unity (IS)** |
|---|---|---|---|
| $\rho$ | 1.22543 | Density of the air at sea level | $(kg/m^3)$ |
| $g_e$ | 9.80665 | Gravity of Earth at sea level | $(m/s^2)$ |
| $\beta$ | 0.02500 | Wind strength | $(s^{-1})$ |
| $C_{D0}$ | 0.01730 | Coefficient of drag at zero-lift | (dimensionless) |
| $k_A$ | 0.03200 | Aerodynamic coefficient | (dimensionless) |
| $m_g$ | 1.99000 | Mass of the glider | $(kg)$ |
| $S_w$ | 0.48500 | Glider's total wing area | $(m^2)$ |

$$m_g\dot{v} = -D - m_g g_e \sin\gamma - m_g\dot{U}\cos\gamma\sin\varphi \tag{2}$$

$$m_g v\dot{\gamma} = L\cos\mu - m_g g_e\cos\gamma + m_g\dot{U}\sin\gamma\sin\varphi \tag{3}$$

$$m_g v\cos\gamma\dot{\varphi} = L\sin\mu - m_g\dot{U}\cos\varphi \tag{4}$$

$$\dot{x} = v\cos\gamma\sin\varphi + U(h) \tag{5}$$

$$\dot{y} = v\cos\gamma\cos\varphi \tag{6}$$

$$\dot{h} = v\sin\gamma, \tag{7}$$

where

$$D = \frac{1}{2}\rho S_w C_D v^2 \tag{8}$$

$$L = \frac{1}{2}\rho S_w C_L v^2 \tag{9}$$

$$C_D = C_{D0} + k_A {C_L}^2 \tag{10}$$

$$U(h) = \beta h \tag{11}$$

$$\dot{U} = \frac{dU(h)}{dt} = \beta v \sin\gamma. \tag{12}$$

In Equations (2–7), the wind's velocity is $U(h)$. The auxiliary variables $D$ and $L$ (Equations (8) and (9)) represent the drag and lift forces, respectively, acting on the glider.

By re-writing Equations (2–7) so that the time derivatives are isolated and by grouping the equations, one can obtain a compact representation of the system dynamics as follows:

$$\dot{\mathbf{y}} = f(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau), \tag{13}$$

where $f(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau)$ corresponds to the right-hand-side of the system of Ordinary Differential Equations (ODEs).

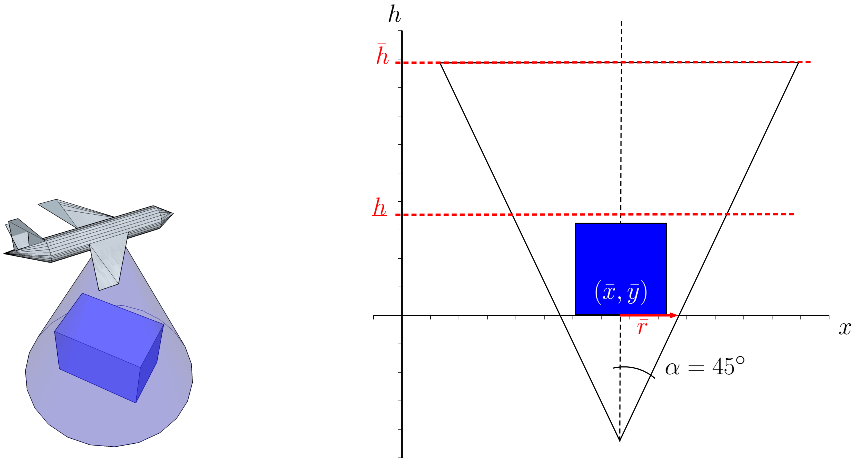
## 3. Problem Definition

In the GRTOP, a fleet of balloon-lifted gliders is required to survey a number of points of interest (waypoints), such as hospitals, schools and residential areas, in order to assess possible damage and people at risk in the aftermath of a disaster. Gliders are launched and are expected to land in one of the predetermined landing zones. Suitable positions of launching sites can be estimated by using the tool "ASTRA High Altitude Balloon Flight Planner" proposed by Sobester et al. (2013), available at Zapponi (2013). This tool allows to simulate the trajectory of a balloon carrying a payload. The user is allowed to choose, for example, the payload weight, the type of balloon and the site from where the balloon is released. Then, according to the appropriate weather forecast, ASTRA provides possible flight trajectories for the balloon. A suitable releasing point should allow balloons to take gliders to a favourable position with respect to the waypoints.

Each glider is equipped with a remote camera able to survey objects positioned within relative ranges. An inverted conic shape is adopted in order to model a cameras' field of view (Figure 3a). This type of geometry has also been used for UAV-camera systems in Ariyur & Fregene (2008), Roelofsen et al. (2016) and Nedjati et al. (2016). We assume that cameras are fixed to the body of the gliders, and we enforce the gliders to fly in level-flight (or "flat") over a waypoint in order to properly photograph the desired object. For the sake of simplicity, we assume that the fleet of gliders is homogeneous and cameras have the same specifications. For each waypoint, the cameras' field of view allows us to define conic-like regions that must be visited by the gliders.

Figure 3b illustrates the geometric representation of a waypoint. In this picture, the object of interest corresponds to the blue box. Each waypoint is entirely described by $(\bar{x}_i, \bar{y}_i, \bar{r}_i, \underline{h}_i, \bar{h}_i), i \in V$, where $(\bar{x}_i, \bar{y}_i)$ represents the position of the object $i$ in the $xy$ plane. Parameter $\bar{r}_i > 0$ represents the radius of a circle in the $xy$ plane enclosing the footprint of the object $i$. Parameters $\underline{h}_i$ and $\bar{h}_i$ denote the minimum and maximum heights in which object $i$ can be photographed, respectively. The last two components define and constrain the quality of the pictures. Without loss of generality, we set the cameras' opening angle $\alpha$ to $45^{\circ}$ (0.7854 rad) while the points of interest are assumed to lie in the same $xy$ plane. Considering the aforementioned assumptions, a glider flying at an altitude $h$ can visit a waypoint $i$ and take a good picture if it touches or enters the truncated cone covering $i$, i.e., if $(x-\bar{x}_i)^2+(y-\bar{y}_i)^2 \leq (h+\bar{r}_i)^2 \tan\alpha = (h+\bar{r}_i)^2$.

Landing zones can be defined in a similar way. The tuple $(\tilde{x}_i, \tilde{y}_i, \tilde{r}_i)$ describes the geometry of a landing zone $i \in L$. The first two components define the position of landing site $i \in L$ on the $xy$ plane and $\tilde{r}_i$ denotes its radius. The shape of landing zones consists of half-spheres with centres in the same $xy$ plane as the waypoints.



(a) Camera model.

(b) Waypoint geometry.

Figure 3: Geometric representation of the cameras' field of view and waypoints.

### 3.1. A Mixed-Integer Non-Linear Programming Formulation

In this section, a mathematical formulation for the GRTOP is proposed. In the following, we assume a fleet $G$ of gliders to be available at a known launching point 0. Let $V$ represent a set of waypoints that have to be visited and $L$ a set of possible landing sites. We are asked to find optimal routes and trajectories for the gliders in $G$ such that the total mission time is minimised.

The motion of each glider is constrained by the system of ODEs (13). We assume that the initial position, denoted by $\mathbf{x}_o$, of each glider is known in advance, i.e., $\mathbf{x}_g(\tau = 0) = \mathbf{x}_o, \forall g \in G$. The subvector $\mathbf{x}_g(\tau) = (x_g, y_g, h_g)$ represents the position of glider $g \in G$ at time $\tau \in [\tau_o, \tau_f]$, where $\tau_o = 0$ is the initial mission time and $\tau_f$ the maximum final mission time. Unlike the initial position, the initial orientation of each glider is hard to estimate. Therefore we assume that the initial orientations can be randomly chosen within the bounds of the state and control variables. We refer to the set of EOMs and initial conditions of each glider as their *dynamical system*.

We solve the GRTOP by means of *direct collocation* (Betts, 2001). In a direct collocation method, a continuous optimal control problem is discretised into a NLP by defining a grid of collocation points over a time interval $[\tau_o, \tau_f]$. Let us define $N$ as the number of collocation points, where each time index $t$ represents a time instant $\tau_t \in [\tau_o, \tau_f]$. From now on, a uniform time grid is adopted, as represented by $\tau_t = \tau_o + \eta t, t \in T, \eta = \frac{\tau_f - \tau_o}{N-1}$, where $T = \{0, \ldots, N-1\}$ is a set of collocation points (time index set). The value of $\eta$ denotes the step size, which is constant in a uniform grid. Without loss of generality, we assume $\tau_o = 0$. In this paper, we apply several different integration schemes to solve the dynamical system, namely, a forward Euler method, a Trapezoidal method, two Adams-Bashforth methods and two Runge-Kutta methods. They are discussed in more details in Section 4. More information about numerical methods for solving ODEs can be found, e.g., in the books by Betts (2001) and Hairer et al. (2011).

Let $\mathbf{y}_{gt}$ and $\mathbf{u}_{gt}$ approximate the continuous state and control vectors $\mathbf{y}_g(\tau_t)$ and $\mathbf{u}_g(\tau_t)$, respectively, of glider $g \in G$ at time index $t$. A simple method for approximating the continuous dynamical system of glider $g$ at time index $t$ can be written as in Equation (14), based on Euler's method,

$$\mathbf{y}_{g(t+1)} = \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, \tau_t), t \in T \setminus \{N-1\}. \tag{14}$$

Equations (14) describe one discretisation method that can be used to represent the non-linear gliders' dynamics in an optimisation problem, namely, the Euler's method. Changing the discretisation method, therefore, implies substituting the discrete system (14) by the corresponding equations of the new method. In our formulation, the dynamical system is relaxed by considering an error term $\varepsilon \in \mathbb{R}_{\geq 0}$, improving the feasibility of the set of dynamical constraints. Next, $\varepsilon$ is added as a penalty to the objective function. This allows for a more tractable formulation, while maintaining the accuracy of trajectories. In Section 5, we show that the values of $\varepsilon$ are small enough to be considered negligible in our examples.

We define the following binary decision variables:

$$a_{git} = \begin{cases} 1, & \text{if glider } g \text{ visits waypoint } i \text{ at time index } t \\ 0, & \text{otherwise.} \end{cases}$$

$$b_{git} = \begin{cases} 1, & \text{if glider } g \text{ lands in the landing site } i \text{ at time index } t \\ 0, & \text{otherwise.} \end{cases}$$

The GRTOP can be formally described by the non-convex MINLP defined by Equations (15-38). The auxiliary variables $d_{git}$ (and $r_{git}$) denote the distance between glider $g$ and waypoint $i$ (landing site $i$) at time index $t$. We denote $\theta_t$ as the "cost" of flying between time steps 0 and $t$. For convenience, we have set $\theta_t = t$ in our computational experiments.

$$\min \quad \sum_{g\in G}\sum_{i\in L}\sum_{t\in T} \theta_t b_{git} + \varepsilon \tag{15}$$

$$\text{s.t.} \quad \sum_{\substack{g\in G \\ g\leq i}}\sum_{t\in T} a_{git} \geq 1, \forall i \in V \tag{16}$$

$$d_{git}^2 \geq (\bar{x}_i - x_{gt})^2 + (\bar{y}_i - y_{gt})^2, \forall g \in G, \forall i \in V, \forall t \in T \tag{17}$$

$$d_{git} \leq (h_{gt} + \bar{r}_i) + M(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{18}$$

$$h_{gt} \leq \bar{h}_i a_{git} + h_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{19}$$

$$h_{g\tilde{t}} \geq h^{\min} a_{git} + h_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T, \forall \tilde{t} \leq t \tag{20}$$

$$\gamma_{gt} \leq \hat{\gamma} a_{git} + \gamma_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{21}$$

$$\gamma_{gt} \geq -\hat{\gamma} a_{git} + \gamma_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{22}$$

$$\mu_{gt} \leq \hat{\mu} a_{git} + \mu_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{23}$$

$$\mu_{gt} \geq -\hat{\mu} a_{git} + \mu_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{24}$$

$$\sum_{i \in L} \sum_{t \in T} b_{git} = 1, \forall g \in G \tag{25}$$

$$\sum_{\substack{\tilde{t} \in T \\ \tilde{t} \leq t}} b_{gj\tilde{t}} \leq 1 - a_{git}, \forall g \in G, j \in L, i \in V, t \in T \tag{26}$$

$$r_{git}^2 \geq (\tilde{x}_i - x_{gt})^2 + (\tilde{y}_i - y_{gt})^2 + h_{gt}^2, \forall g \in G, i \in L, t \in T \tag{27}$$

$$r_{git} \leq \tilde{r}_i + M(1 - b_{git}), \forall g \in G, i \in L, t \in T \tag{28}$$

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, \tau_t) + \mathbf{1}\varepsilon, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{29}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, \tau_t) - \mathbf{1}\varepsilon, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{30}$$

$$\mathbf{x}_{g0} = \mathbf{x}_o, \forall g \in G \tag{31}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{gt} \leq \mathbf{y}_{ub}, \forall g \in G, \forall t \in T \tag{32}$$

$$\mathbf{u}_{lb} \leq \mathbf{u}_{gt} \leq \mathbf{u}_{ub}, \forall g \in G, \forall t \in T \tag{33}$$

$$a_{git} \in \{0,1\}, \forall g \in G, \forall i \in V, \forall t \in T \tag{34}$$

$$b_{git} \in \{0,1\}, \forall g \in G, \forall i \in L, \forall t \in T \tag{35}$$

$$d_{git}, r_{git} \in \mathbb{R}, \forall g \in G, \forall i \in V, \forall t \in T \tag{36}$$

$$\mathbf{y}_{gt} \in \mathbb{R}^6, \mathbf{u}_{gt} \in \mathbb{R}^2, \forall g \in G, \forall t \in T \tag{37}$$

$$\varepsilon \in \mathbb{R}_{\geq 0}. \tag{38}$$

The constants and "big-$M$" terms in the model are defined as follows. The $M$ constant has been computed as the space diagonal of the smallest cuboid containing the waypoints, landing sites and launching point. This is an upper bound on the distance between a glider and any waypoint and landing site at any time. The value $h^{\min}$ is defined as the minimum allowed flight altitude before landing, i.e., $h^{\min} = \max_i\{\underline{h}_i \; | i \in V\}$, assuming that $h^{\min} < \min_i\{\bar{h}_i \; | i \in V\} - c$, with $c \in \mathbb{R}_{\geq 0}$ properly chosen. By setting this $h^{\min}$ value, we prevent gliders from flying lower than the "tallest" waypoint before having visited all waypoints. The values $\hat{\gamma} > 0$ and $\hat{\mu} > 0$ define small flight path and bank angles forcing the glider to fly "flat" in the cone covering an object. In our computational experiments the conditions for "flat" flight have been set as $\hat{\gamma} = \hat{\mu} = 0.03$ (in radians). Finally, we denote by $\mathbf{1}$ a vector of ones with the same length as $\mathbf{y}_{gt}$.

The objective function (15) accounts for a linear combination of the mission time and the error. Throughout this paper, we we will also refer to this objective as minimising the *routing cost.* The minimisation of the mission time forces the gliders to land as soon as possible. The second term of the objective function minimises the error associated with the discretisation of the continuous dynamical system. Constraints (16) state that every waypoint should be visited at least once. The inequality $g \leq i$ underneath the first summation in Constraints (16) breaks some of the symmetry of our model by ensuring that waypoint 1 can only be visited by glider 1, waypoint 2 can only by visited by gliders 1 and 2, and so on. Constraints (17) and (18) make sure that gliders fly within the cone above each waypoint in order to take pictures. Constraints (19) and (20) ensure that the gliders respect minimum and maximum surveying heights. Constraints (21) to (24) enforce gliders to be "flat" when taking pictures. Constraints (25) ensure that each glider lands in exactly one landing site and Constraints (26) make sure that gliders do not land before all waypoints are visited. Constraints (27) and (28) guarantee that gliders land within pre-assigned regions. We highlight that constraints (17) and (27) are second-order cone constraints. The dynamics of each glider are taken into account in Constraints (29) and (30) by applying a relaxation to Euler's method. Alternative formulations can be achieved by applying different discretisation methods. They will be presented in Section 4. By relaxing the dynamical system, we allow for a more tractable optimisation problem. Since the dynamical system is represented by a set of non-convex ODEs, Constraints (29) and (30) are non-convex and the model is a MINLP. Constraints (31) define the initial positions of each glider and Constraints (32–33) define bounds on the state and control variables. Finally, Constraints (34–38) define the domain of the variables.

Alternative objective functions can also be considered within our GRTOP framework. Equity objectives have gained attention in the VRP and disaster management literature (Matl et al., 2018). In contrast with the objective function (15), which allows gliders to be recovered as soon as possible (therefore, allowing for quicker relaunching), equity objectives prioritise a quicker (or fairer) service.

In disaster assessment operations, for example, one might consider minimising the latest arrival of aid, a.k.a. *makespan* or min-max, (Golden et al., 2014). This objective is commonly used when a low response time is necessary. In our model, this objective can be modelled by introducing the auxiliary variable $f$ and adding one more set of constraints as follows.

$$\min \quad f + \varepsilon \tag{39}$$

$$\text{s.t.} \quad f \geq \sum_{i\in L}\sum_{t\in T}\theta_t b_{git}, \forall g \in G \tag{40}$$

Another common equity objective in the VRP literature is the minimisation of the *latency*. The latency of a customer is defined as the total distance travelled up to that customer (Kara et al., 2008). In the GRTOP, we define the latency of a waypoint as the total flight time up to that waypoint. The minimum latency problem consists of finding routes such that the sum of the latencies of all waypoints is minimised. This objective allows for a reduction on the times in which waypoints are photographed. In our formulation, this can be achieved by replacing the objective function with Expression (41)

$$\min \quad \sum_{g\in G}\sum_{i\in L}\sum_{t\in T}\theta_t a_{git} + \varepsilon. \tag{41}$$

## 4. Linearisation and Discretisation of the Glider's Dynamics

The model (15–38) combines routing and trajectory optimisation decisions in a non-convex formulation. Local optimisation software for non-linear optimisation often requires high quality initial guesses. In order to avoid this issue, we transform the MINLP into a more tractable convex model by linearising the gliders' EOMs. This simplification is usually preferred in the literature when the dynamics are highly non-linear (Ahmed et al., 2015; Hajiyev et al., 2015; How et al., 2015). In the following sections, we present the procedure for linearising the gliders' flight dynamics and the discretisation methods we applied for solving the resulting linear system.

### *4.1. Equilibrium flight and linearisation*

A classic approach for linearising a system of ODEs consists of assuming the system operates in a *steady-state* condition, a.k.a. in equilibrium conditions. The equivalent linear system is then modelled assuming perturbations from this steady-state. Alternative techniques involve, for example, sequential and input-output linearisation. An interested reader can refer to the books by Russell (1996) and Stengel (2004) for more methods of finding steady-state conditions.

We denote by $\mathbf{y}_{eq}$ and $\mathbf{u}_{eq}$ the steady states and controls, respectively, of the glider dynamics. In a steady flight, the resultant forces and moments acting on the vehicle are zero. In other words, let us define $\mathbf{y}_{eq} = (x_{eq}, y_{eq}, h_{eq}, v_{eq}, \gamma_{eq}, \varphi_{eq})$ and $\mathbf{u}_{eq} = (C_{L,eq}, \mu_{eq})$ as the state and control variables such that

$$\dot{\mathbf{y}} = f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau) = \mathbf{0}. \tag{42}$$

In order to find an analytic solution to Equation (42), the following assumptions are made as in Stengel (2004) and Langelaan (2007):

- Steady gliding flight, i.e., the equilibrium is achieved by matching the wind force with the drag, and the lift with the weight.
- The flight path and bank angles, $\gamma$ and $\mu$, respectively, are very small. Therefore, $\sin\gamma \approx \gamma$, $\cos\gamma \approx 1$, $\sin\mu \approx \mu$ and $\cos\mu \approx 1$. It is also assumed that $\varphi = 0$.
- The air mass is stable and the wind velocity is constant.
- The lift coefficient is constant.

From these assumptions, the EOMs (2–12) can be simplified to the following equations (43–45).

$$\dot{v} = -D/m_g - g_e\gamma = 0 \tag{43}$$

$$\dot{\varphi} = -L\mu/m_g v = 0 \tag{44}$$

$$\dot{\gamma} = L/m_g v - g_e/v = 0. \tag{45}$$

The optimal static lift coefficient is expected to minimise the drag-to-lift $(D/L)$ ratio. Therefore:

$$\frac{\partial(D/L)}{\partial C_{L,eq}} = -\frac{C_{D0}}{C_{L,eq}^2} + k_A = 0 \implies C_{L,eq} = \sqrt{\frac{C_{D0}}{k_A}}. \tag{46}$$

By substituting Equation (9) into Equation (45) we find:

$$\frac{\rho S_w C_L v}{2m_g} - \frac{g_e}{v} = 0. \tag{47}$$

The equilibrium velocity can be calculated by substituting the value of the equilibrium lift coefficient $C_{L,eq}$ into Equation (47):

$$v_{eq} = \sqrt{\frac{2m_g g_e}{\rho S_w C_{L,eq}}}. \tag{48}$$

The value of the equilibrium flight path angle $(\gamma_{eq})$ can be found in a similar way. By substituting Equation (8) into Equation (43) we find:

$$\frac{-\rho S_w C_D v^2}{2m_g} - g_e\gamma = 0. \tag{49}$$

From Equations (46), (48) and (49), the value of $\gamma_{eq}$ can be computed as in Equation (50)

$$\gamma_{eq} = -2\sqrt{k_A C_{D0}}. \tag{50}$$

Let us define the following new variables as perturbations around state and control variables as $\delta\mathbf{y}(\tau) = \mathbf{y}(\tau) - \mathbf{y}_{eq}$ and $\delta\mathbf{u}(\tau) = \mathbf{u}(\tau) - \mathbf{u}_{eq}$, respectively. Applying first order Taylor's expansion to the system (13) around the steady-state conditions and discarding the higher order terms gives:

$$T(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \delta\mathbf{y}, \delta\mathbf{u}, \tau) = f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau) + \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{y}}\delta\mathbf{y}(\tau) + \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{u}}\delta\mathbf{u}(\tau). \tag{51}$$

By definition, the first term of Equation (51) equals zero for the equilibrium condition. Matrices $A = \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{y}}$ and $B = \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{u}}$ denote the Jacobians of the dynamics (13) with respect to state and control variables. This linear system of ODEs can be re-written in state-space form as in Equation (52)

$$T(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \delta\mathbf{y}, \delta\mathbf{u}, \tau) = A\delta\mathbf{y}(\tau) + B\delta\mathbf{u}(\tau), \tag{52}$$

where the system's matrices $A$ and $B$ have been found by computing the derivatives of the glider's EOM (2–12) at the steady-state conditions $\mathbf{y}_{eq}$ and $\mathbf{u}_{eq}$:

$$A = \begin{bmatrix} 0 & 0 & 0.025 & 0 & 0 & 9.44023 \\ 0 & 0 & 0 & 0.99889 & 0.44456 & 0 \\ 0 & 0 & 0 & -0.04704 & 9.44023 & 0 \\ 0 & 0 & 0 & -0.09766 & -9.79579 & 0.01110 \\ 0 & 0 & 0 & 0.21947 & -0.04881 & 0.00006 \\ 0 & 0 & 0 & 0 & -0.02506 & 0 \end{bmatrix}, \mathbf{y}_{eq} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 9.45068 \\ -0.04705 \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.62763 & 0 \\ 1.41127 & 0 \\ 0 & 1.03882 \end{bmatrix}, \mathbf{u}_{eq} = \begin{bmatrix} 0.73527 \\ 0 \end{bmatrix}.$$

In our approach, we approximate the nonlinear dynamical system defined in Equation (13) by the linear system of equations (52), such that:

$$\dot{\mathbf{y}} \approx A\delta\mathbf{y}(\tau) + B\delta\mathbf{u}(\tau). \tag{53}$$

## *4.2. Discretisation methods*

In this paper, we solve the gliders' EOMs by means of a *direct collocation* method (Betts, 2001). This is accomplished by discretising the linear EOMs represented by Equations (53) and adding the resulting expressions as constraints in the GRTOP. Since the flight time interval $[\tau_o, \tau_f]$ and step size $\eta$ will be fixed for all methods presented here, the linearity of the EOMs is maintained. Due to the linearity of the EOMs, the MINLP formulation for the GRTOP, defined by Equations (15–38), is reduced to a MISOCP formulation that can be solved by commercial optimisation software without the need of initial guesses to converge.

### *4.2.1. Euler's method*

The forward Euler's method is a first-order explicit numerical approach for solving ODEs (Hairer et al., 2011). Let $\mathbf{y}_t$ and $\mathbf{u}_t$ approximate $\mathbf{y}(\tau_t)$ and $\mathbf{u}(\tau_t)$, respectively, at time $\tau_t \in [\tau_o, \tau_f]$ such that:

$$\dot{\mathbf{y}} \approx \frac{\mathbf{y}_{t+1} - \mathbf{y}_t}{\eta}.$$

By using this approximation on Equation (53), we can write:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \eta(A\delta\mathbf{y}_t + B\delta\mathbf{u}_t). \tag{54}$$

This dynamical system can be reformulated in terms of the discretised state and control variables in Equation (55). In a direct collocation method, these equations are used as constraints in an optimisation problem.

$$\mathbf{y}_{t+1} = (\eta A + I)\mathbf{y}_t + \eta B\mathbf{u}_t - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}). \tag{55}$$

The feasibility of this dynamical system can be improved by considering an error $\varepsilon$ and relaxing the equality in Equation (55) such that:

$$\mathbf{y}_{t+1} \leq (\eta A + I)\mathbf{y}_t + \eta B\mathbf{u}_t - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}) + \mathbf{1}\varepsilon \tag{56}$$

$$\mathbf{y}_{t+1} \geq (\eta A + I)\mathbf{y}_t + \eta B\mathbf{u}_t - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}) - \mathbf{1}\varepsilon. \tag{57}$$

We penalise $\varepsilon$ in our objective function in order to maintain the quality of our solutions. Finally, we replace Constraints (29–30) by Constraints (56–57) when applying Euler's method for solving the dynamical system in our formulation. This strategy is repeated for all discretisation methods presented in Section 4.2.

### *4.2.2. Trapezoidal method*

The Trapezoidal method is a second-order implicit approach for solving ODEs based on the trapezoidal rule for computing integrals (Hairer et al., 2011). The equation defining the trapezoidal method can be derived from both Runge-Kutta and Adams-Bashforth methods, and for the GRTOP it can be defined as follows:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{1}{2}\eta(A\delta\mathbf{y}_t + B\delta\mathbf{u}_t + A\delta\mathbf{y}_{t+1} + B\delta\mathbf{u}_{t+1}). \tag{58}$$

By re-writing this system in terms of the original variables, we find the following discrete linear system:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{1}{2}\eta(A(\mathbf{y}_{t+1} + \mathbf{y}_t) + B(\mathbf{u}_{t+1} + \mathbf{u}_t)) - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}). \tag{59}$$

In order to use Equation (59) as a constraint in our formulation, we apply the same relaxation method as presented in Section 4.2.1.

### *4.2.3. Runge-Kutta methods*

The Runge-Kutta methods are a family of numerical methods for solving initial value problems. They consist of sampling intermediate values between subsequent time steps in order to cancel out lower order error terms (Hairer et al., 2011). In this paper, we apply a fourth-order Runge-Kutta method ($RK4$) in order to discretise the glider dynamics. The coefficients of the $RK4$ method for the linear glider dynamics are defined by Equations (60–63) in terms of the original discrete state and control variables.

$$k_1 = A(\mathbf{y}_t - \mathbf{y}_{eq}) + B(\mathbf{u}_t - \mathbf{u}_{eq}) \tag{60}$$

$$k_2 = A(\mathbf{y}_t + \frac{1}{2}\eta k_1 - \mathbf{y}_{eq}) + B(\hat{\mathbf{u}} - \mathbf{u}_{eq}) \tag{61}$$

$$k_3 = A(\mathbf{y}_t + \frac{1}{2}\eta k_2 - \mathbf{y}_{eq}) + B(\hat{\mathbf{u}} - \mathbf{u}_{eq}) \tag{62}$$

$$k_4 = A(\mathbf{y}_t + \eta k_3 - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}), \tag{63}$$

where the auxiliary variable $\hat{\mathbf{u}}$ represents an interpolation of the control variables between time steps $t$ and $t+1$. Here, two interpolation methods have been used. The first one consists of a linear interpolation (Equation (64)) and the second one consist of an exponential smoothing (Equation (65)), which weights the control history up to time step $t$.

$$\hat{\mathbf{u}} = \frac{\mathbf{u}_{t+1} + \mathbf{u}_t}{2} \tag{64}$$

$$\hat{\mathbf{u}} = \frac{1}{2}\sum_{k=0}^{k\leq t}\frac{1}{2^k}\mathbf{u}_{t-k} \tag{65}$$

Discretised EOMs can then be defined by the following equation:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{\eta}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{66}$$

In order to use Equation (66) as a constraint in our formulation, we apply the same relaxation method as presented in Section 4.2.1.

### *4.2.4. Adams-Bashforth methods*

Linear multistep methods use information from previous steps to determine the current values of the state vector (Hairer et al., 2011). Unlike Runge-Kutta methods, multistep methods do not required interpolation of the control variables at intermediate steps since calculations are based on predetermined collocation points (in case of a direct collocation method). The Adams-Bashforth (AB) methods are a family of explicit integrators that compute the value of the current state from a linear combination of the values of previous states. In this article, a third-order Adams-Bashforth method ($AB3$) and a fourth-order Adams-Bashforth method ($AB4$) are presented, in terms of the original state and control vectors, in the form of Equations (67) and (68).

$$\begin{aligned}\mathbf{y}_{t+3} = \mathbf{y}_{t+2} + \frac{1}{12}\eta(23(&A(\mathbf{y}_{t+2} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+2} - \mathbf{u}_{eq}))\\ -16(&A(\mathbf{y}_{t+1} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}))\\ +5(&A(\mathbf{y}_{t} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t} - \mathbf{u}_{eq}))\end{aligned} \tag{67}$$

$$\begin{aligned}\mathbf{y}_{t+4} = \mathbf{y}_{t+3} + \frac{1}{24}\eta(55(&A(\mathbf{y}_{t+3} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+3} - \mathbf{u}_{eq}))\\ -59(&A(\mathbf{y}_{t+2} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+2} - \mathbf{u}_{eq}))\\ +37(&A(\mathbf{y}_{t+1} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}))\\ -9(&A(\mathbf{y}_{t} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t} - \mathbf{u}_{eq}))\end{aligned} \tag{68}$$

Unlike single step methods, the third- and fourth-order AB methods require 3 and 4 initial values at the first iteration, respectively. This can be accomplished by running a single step method in order to find these initial values and then continuing the solution process with a multistep AB method. In this paper, we apply the Euler method in order to compute the initial values that are necessary for the Adams-Bashforth methods. In order to use Equations (67) and (68) as constraints in our formulation, we apply the same relaxation method as presented in Section 4.2.1.

## 5. Computational Experiments

The computational experiments described in the next sections have been implemented in the AMPL modelling language (version 20150516) and solved with CPLEX 12.7, Gurobi 7.0 and Xpress 8.0 on an Intel Core i7-4770 CPU with 3.40GHz and 16GB of RAM running under Linux Mint 17 64 bits (kernel 3.13.0-24). These solvers were chosen due to availability of licenses and their popularity in the research community (Mittelmann, 2017). They were set to their standard configurations, with a time limit of 1 hour of execution each. Unless stated otherwise, the objective considered in our experiments was minimising the total flight time.

### *5.1. Generation of test instances*

A number of test instances have been generated as follows. We have considered instances having $n \in \{2, \ldots, 10\}$ waypoints and $m \in \{1, 2\}$ landing zones. The number of available gliders has been defined as $\lfloor n/2 \rfloor$, but limited to at most 3 gliders in our examples. Two classes of instances have been created. The so-called *small range* instances (represented by "S" in the instances' name) have been defined over an area of $1km^2$ and the so-called *medium range* instances (represented by "M" in the instances' name) over an area of $25km^2$. We assume a square shape for each area. In addition, each combination of number of waypoints and landing zones received 5 different random instances, so as to have a diversity of test cases. These instances are grouped in our tables by the number of waypoints, e.g., the group GRTOP-S10 represents all small range instances with 10 waypoints.

The geometry of waypoints and landing zones has been defined by the parameters in Table 3. Let $\mathcal{U}[a, b]$ denote the continuous uniform distribution from $a$ to $b$. The launching altitude $h_o$ has been chosen from $\mathcal{U}[500, 600]$ for the small range instances and $\mathcal{U}[1000, 2000]$ for the medium range instances, with the values of the limits given in metres. In Table 3, the value of $R$ represents the length of a side of the area containing the waypoints, $R = 1km$ for the small-ranged instances and $R = 5km$ for the medium range ones. All the other values in Table 3 are given in metres. Instances can be downloaded from `http://dx.doi.org/10.5258/SOTON/D0583`.

The positions of waypoints were not constrained. Overlaps were allowed except when they generate duplicates and waypoints were randomly assigned within the boundaries of the area. However, landing zones were not allowed to overlap with waypoints.

Table 3: Limits of parameters defining the geometry of waypoints and landing zones in the generated instances.

| Parameter | $a$ | $b$ | Parameter | $a$ | $b$ |
|---|---|---|---|---|---|
| $\bar{x}$ | 0 | $R$ | $\tilde{x}$ | 0 | $R$ |
| $\bar{y}$ | 0 | $R$ | $\tilde{y}$ | 0 | $R$ |
| $\bar{r}$ | 10 | 25 | $\tilde{r}$ | 10 | 25 |
| $\underline{h}$ | 50 | 100 | $x_o$ | 0 | $R$ |
| $\bar{h}$ | 200 | 300 | $y_o$ | 0 | $R$ |

### *5.2. Comparing the performance of different solvers*

Table 4 shows a summary of the results for the small range instances. In this table, *Group* denotes the nine groups of 10 instances (organised according to the number of waypoints in each instance). The performance of each solver is shown on columns *CPLEX*, *Gurobi* and *Xpress*.

In Table 4, the *Status* column shows the tuple $(s_1, s_2, s_3) \in \mathbb{Z}^3$, representing the possible output statuses from AMPL as explained on AMPL (1998), where $s_1$ denotes the number of instances finished with status `solved`, $s_2$ represents the number of instances finished with status `solved?` and `limit` (i.e, with a gap greater than 0), and $s_3$ represents the number of instances finished with status `failure`.

Column *Error* corresponds to the normalised average error for each group. This has been calculated by averaging the error in each group and then dividing this average by the smallest error among the solvers for the same group. We use normalised average errors in order to make a comparison between different strategies easier. Column *Gap(%)* shows the average optimality gap at the end of the optimisation. Column *CPU(s)* represents the average computing time in seconds. Finally, column *Tree Size* shows the average number of explored branch-and-bound nodes for each group.

In order to test the solvers, we have chosen the Euler discretisation method. The number of collocation points $N$ has been set to 30. The flight time horizon of each instance has been estimated by using the steady-state velocity and the largest range of that instance, i.e., $\tau_f = \max_{i \in V}\{\bar{x}_i, \bar{y}_i\}/v_{eq}$.

The solver Xpress outperforms the other solvers in most aspects. Best results overall in each column (except from the second) have been highlighted in boldface. Averages are shown for each solver. The relationship between CPU times and tree sizes indicates that Xpress is noticeably faster on processing the second-order cone relaxations during the branch-and-bound search. Due to a large number of failures and worse performance for computing Second-Order Cone Programming (SOCP) relaxations, CPLEX presents smaller average tree sizes on most cases. For our problem, Xpress has been capable of solving 76% of the instances to optimality as opposed to 32% and 24% of the instances that have been solved by CPLEX and Gurobi, respectively. One can notice that CPLEX has failed to find solutions to 15 problems in total. For the reasons exposed above, the solver Xpress has been chosen for the computational experiments presented in the next sections.

### *5.3. Comparing the performance of different discretisation methods*

In this section, we compare the performance of the numerical integration methods presented in Section 4. Table 5 summarises the results for the small range instances. The remaining columns refer to the aforementioned discretisation methods, namely the *Euler* method, Trapezoidal method ($TRP$), the third- and fourth-order Adams-Bashforth methods, $AB3$ and $AB4$, respectively, and both versions of the fourth-order Runge-Kutta method $1RK4$ and $2RK4$, where the former refers to the RK4 method with linear control interpolation of Equation (64) and the latter to the RK4 method with the interpolation described in Equation (65). Table 5 has been subdivided for each algorithm performance measure, namely, Status, Error, Gap(%), CPU(s) and Tree, as in Table 4. Overall averages are shown at the end of each subdivision. The discretisation size and flight time horizon estimation have been kept as in the previous section.

From the results in Table 5, one can notice that the solver is more effective in proving optimality when the Euler and Trapezoidal methods used, finding 68 (75.6%) and 69 (76.7%) optimal solutions, respectively, against 62 (68.9%) and 61 (67.8%) optimal solutions found by using the 1RK4 and 2RK4 methods. Together, they also produce smaller gaps for the instances that were not solved within the provided time limit. One can also verify that the ratio between the average number of branch-and-bound nodes and average CPU times is larger for these methods. This fact indicates that the Euler and Trapezoidal methods generate relaxations that are easier to solve during the tree search. On the other hand, Runge-Kutta methods outperform all the others in terms of error. The Runge-Kutta methods present error magnitudes that are up to 16 times smaller on average than the largest errors, at the expense of presenting higher average gaps for the instances that were not solved to optimality. There is

Table 4: Summary of the results for different solvers.

| Group | Status | Error | Gap(%) | CPU(s) | Tree |
|---|---|---|---|---|---|
| | | | CPLEX | | |
| GRTOP-S2 | (10,0,0) | **1.000** | **0.00%** | 22.348 | 1274.30 |
| GRTOP-S3 | (8,0,2) | 1.007 | **0.00%** | 99.081 | 2428.75 |
| GRTOP-S4 | (2,0,8) | 1.034 | **0.00%** | 1199.889 | 26176.00 |
| GRTOP-S5 | (3,4,3) | **1.000** | 18.57% | 2171.628 | 12652.43 |
| GRTOP-S6 | (3,6,1) | 1.403 | 32.33% | 2997.753 | 20993.56 |
| GRTOP-S7 | (1,8,1) | 1.736 | 35.78% | 3175.324 | 13763.67 |
| GRTOP-S8 | (2,8,0) | 4.017 | 51.30% | 3397.745 | 12404.38 |
| GRTOP-S9 | (0,10,0) | 3.285 | 54.50% | 3600.500 | 11439.10 |
| GRTOP-S10 | (0,10,0) | 3.230 | 65.90% | 3600.589 | 12096.90 |
| avg. | - | 1.97 | 28.71% | 2251.65 | 12581.01 |
| | | | Gurobi | | |
| GRTOP-S2 | (10,0,0) | **1.000** | **0.00%** | 44.125 | **2831.40** |
| GRTOP-S3 | (9,1,0) | 1.028 | 2.70% | 492.016 | **7698.60** |
| GRTOP-S4 | (1,9,0) | 1.681 | 22.10% | 3351.221 | **40894.40** |
| GRTOP-S5 | (1,9,0) | 2.256 | 26.00% | 3250.099 | 40628.50 |
| GRTOP-S6 | (1,9,0) | 2.763 | 44.00% | 3379.685 | 42566.10 |
| GRTOP-S7 | (0,10,0) | 2.731 | 39.40% | 3600.206 | 48218.50 |
| GRTOP-S8 | (0,10,0) | 2.736 | 42.90% | 3600.184 | 39544.80 |
| GRTOP-S9 | (0,10,0) | 2.946 | 39.60% | 3600.166 | 35009.00 |
| GRTOP-S10 | (0,10,0) | 4.482 | 56.90% | 3600.204 | 31395.40 |
| avg. | - | 2.40 | 30.40% | 2768.66 | 32087.41 |
| | | | Xpress | | |
| GRTOP-S2 | (10,0,0) | **1.000** | **0.00%** | **7.124** | 800.60 |
| GRTOP-S3 | (10,0,0) | **1.000** | **0.00%** | **14.024** | 3250.30 |
| GRTOP-S4 | (10,0,0) | **1.000** | **0.00%** | **104.985** | 18475.50 |
| GRTOP-S5 | (9,1,0) | 1.020 | **0.10%** | **727.232** | **186538.20** |
| GRTOP-S6 | (7,3,0) | **1.000** | **2.40%** | **1294.716** | **140235.50** |
| GRTOP-S7 | (9,1,0) | **1.000** | **0.60%** | **1207.613** | **138074.90** |
| GRTOP-S8 | (5,5,0) | **1.000** | **3.40%** | **2226.280** | **200746.40** |
| GRTOP-S9 | (7,3,0) | **1.000** | **5.20%** | **2157.578** | **173651.90** |
| GRTOP-S10 | (1,9,0) | **1.000** | **12.30%** | **3256.597** | **201550.40** |
| avg. | - | **1.00** | **2.67%** | **1221.79** | **118147.08** |

Table 5: Summary of the results for different discretisation methods.

| Group | Euler | TRP | AB3 | AB4 | 1RK4 | 2RK4 |
|---|---|---|---|---|---|---|
| Status | | | | | | |
| GRTOP-S2 | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) |
| GRTOP-S3 | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) |
| GRTOP-S4 | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) | (10,0,0) |
| GRTOP-S5 | (9,1,0) | (10,0,0) | (10,0,0) | (9,1,0) | (8,2,0) | (7,3,0) |
| GRTOP-S6 | (7,3,0) | (9,1,0) | (5,5,0) | (4,6,0) | (7,3,0) | (6,4,0) |
| GRTOP-S7 | (9,1,0) | (8,2,0) | (5,5,0) | (3,7,0) | (6,4,0) | (6,4,0) |
| GRTOP-S8 | (5,5,0) | (5,5,0) | (5,5,0) | (3,7,0) | (6,4,0) | (6,4,0) |
| GRTOP-S9 | (7,3,0) | (6,4,0) | (2,8,0) | (1,9,0) | (3,7,0) | (3,7,0) |
| GRTOP-S10 | (1,9,0) | (1,9,0) | (2,8,0) | (2,8,0) | (2,8,0) | (3,7,0) |
| Error | | | | | | |
| GRTOP-S2 | 6.076 | 6.071 | 6.040 | 6.079 | **1.000** | 1.077 |
| GRTOP-S3 | 4.438 | 4.565 | 4.633 | 4.565 | **1.000** | 1.007 |
| GRTOP-S4 | 6.825 | 6.963 | 7.134 | 7.207 | **1.000** | 1.045 |
| GRTOP-S5 | 3.866 | 4.015 | 3.928 | 3.951 | **1.000** | 1.088 |
| GRTOP-S6 | 9.525 | 9.454 | 9.518 | 9.789 | **1.000** | 1.309 |
| GRTOP-S7 | 5.854 | 5.659 | 6.264 | 6.164 | 1.033 | **1.000** |
| GRTOP-S8 | 14.396 | 14.608 | 15.477 | 16.025 | 1.028 | **1.000** |
| GRTOP-S9 | 8.114 | 8.194 | 9.018 | 8.868 | **1.000** | 1.108 |
| GRTOP-S10 | 12.478 | 13.235 | 12.937 | 12.141 | **1.000** | 1.382 |
| avg. | 7.952 | 8.085 | 8.328 | 8.310 | **1.007** | 1.113 |
| Gap(%) | | | | | | |
| GRTOP-S2 | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** |
| GRTOP-S3 | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** |
| GRTOP-S4 | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** | **0.00%** |
| GRTOP-S5 | 0.10% | **0.00%** | **0.00%** | 0.90% | 3.00% | 0.40% |
| GRTOP-S6 | 2.40% | **0.50%** | 4.10% | 3.00% | 6.50% | 9.20% |
| GRTOP-S7 | **0.60%** | 1.30% | 5.10% | 9.50% | 10.20% | 3.90% |
| GRTOP-S8 | 3.40% | **3.20%** | 6.20% | 13.50% | 9.80% | 8.30% |
| GRTOP-S9 | 5.20% | **4.80%** | 15.40% | 12.30% | 17.00% | 15.60% |
| GRTOP-S10 | **12.30%** | 16.10% | 16.50% | 20.50% | 34.20% | 34.40% |
| avg. | **2.67%** | 2.88% | 5.26% | 6.63% | 8.97% | 7.98% |
| CPU(s) | | | | | | |
| GRTOP-S2 | 7.12 | 9.09 | 13.62 | 20.55 | 6.96 | **6.14** |
| GRTOP-S3 | **14.02** | 17.10 | 38.15 | 39.37 | 16.93 | 49.12 |
| GRTOP-S4 | **104.99** | 151.17 | 338.46 | 714.07 | 306.09 | 530.00 |
| GRTOP-S5 | 727.23 | **146.84** | 328.67 | 919.91 | 859.51 | 1282.27 |
| GRTOP-S6 | 1294.72 | **939.52** | 2423.59 | 2778.01 | 1602.54 | 1878.44 |
| GRTOP-S7 | **1207.61** | 1693.21 | 2517.73 | 3157.09 | 1952.83 | 1774.54 |
| GRTOP-S8 | 2226.28 | 2159.28 | 2571.27 | 3071.74 | **1813.62** | 1958.09 |
| GRTOP-S9 | **2157.58** | 2789.33 | 3276.83 | 3386.65 | 2959.64 | 2660.28 |
| GRTOP-S10 | 3256.60 | 3277.13 | 3216.06 | 3274.96 | 3089.89 | **2938.90** |
| avg. | **1221.79** | 1242.52 | 1636.04 | 1929.15 | 1400.89 | 1453.09 |
| Tree | | | | | | |
| GRTOP-S2 | 800.60 | **491.50** | 1536.60 | 2208.70 | 1349.80 | 605.00 |
| GRTOP-S3 | 3250.30 | **2772.60** | 9364.90 | 6327.00 | 2937.30 | 13045.80 |
| GRTOP-S4 | **18475.50** | 24006.80 | 47125.80 | 87958.70 | 74628.00 | 88431.30 |
| GRTOP-S5 | 186538.20 | **17292.70** | 36597.40 | 85667.00 | 104206.00 | 197635.70 |
| GRTOP-S6 | 140235.50 | **88382.10** | 165818.60 | 140593.00 | 114531.70 | 98954.10 |
| GRTOP-S7 | 138074.90 | 157636.90 | 146485.20 | 133274.00 | 126049.60 | **89096.20** |
| GRTOP-S8 | 200746.40 | 144065.70 | 122573.20 | 100995.50 | 95772.00 | **95153.00** |
| GRTOP-S9 | 173651.90 | 174789.50 | 114985.30 | **98945.70** | 127546.60 | 100475.20 |
| GRTOP-S10 | 201550.40 | 174476.00 | 110871.60 | **82600.00** | 112147.00 | 113462.60 |
| avg. | **118147.08** | 87101.53 | 83928.73 | 82063.29 | 84352.00 | 88539.88 |

a clear trade-off between computational performance and solution accuracy among the lower and higher order integration methods. Nonetheless, the third- and fourth-order Adams-Bashforth methods perform quite poorly for our problem, given the large error values and considerable gaps compared to the lower order methods.

We have extended our computational results for the smallest and largest instances of type “S” in Table 6. In this table, the first column shows the instance names and remaining columns present the error and CPU times for each discretisation methods presented in Section 4. The settings for the experiments shown in Table 6 remain the same as in the ones shown in Table 5.

From the results presented in Table 6, it can be seen that the magnitudes of the errors can be considered acceptable even for the lower order methods. To support our claim, we have further investigated which state variables are most affected by the errors when using the Euler’s method. This has been accomplished by performing two modifications in our formulation. The first one consists of using a vector $\boldsymbol{\varepsilon} = (\varepsilon_x, \varepsilon_y, \varepsilon_h, \varepsilon_v, \varepsilon_\gamma, \varepsilon_\varphi), \boldsymbol{\varepsilon} \in \mathbb{R}^6$, to represent the error for each state variable in the dynamical equations. For example, the generic discretisation method presented in Constraints (29) and (30) can be re-written in the form of Constraints (69) and (70), respectively. It means that more variables will be added to the MISOCP formulation for the GRTOP.

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) + \boldsymbol{\varepsilon}, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{69}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) - \boldsymbol{\varepsilon}, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{70}$$

The second modification follows from the first one as the objective function (15) needs to be re-written as in Equation (71) (similar modifications can be achieved for the objectives regarding makespan and latency). While the first term remains the same, the second term of the new objective function sums up the individual errors for each state variable.

$$\min \quad \sum_{g \in G} \sum_{i \in L} \sum_{t \in T} \theta_t b_{git} + \mathbf{1}^\top \boldsymbol{\varepsilon}. \tag{71}$$

Table 7 shows the results of this reformulation for a subset of instances (using Euler’s method for discretising the dynamics). The main source of errors are the position components of the state vector. This can be explained by the fact that those components have the largest magnitude among all state variables, varying roughly between 0 and 1000. Even though the error associated to the translational dynamics is comparatively higher, they only represent a small fraction of the magnitudes of the position variables. For example, the error $\varepsilon_y = 11.58$ associated to the $y$ variable for the small-ranged instance `grtopS_21_1` only represents 1.16% of the range of this state variable. The last two columns of Table 7 shows the results from the original formulation with a single error variable using Euler’s method.

Finally, we analyse how the error and CPU times behave as the number of collocation points $N$ increases. This results are shown in Figure 4. For this experiment, the following number of collocation points have been adopted $N = \{30, 45, 60, 75, 90, 105\}$. The flight time interval has been computed as in the previous sections. Since the first term of the objective function (15) might affect the value of $\varepsilon$, we have eliminated this term from the objective function in order to properly assess the behaviour of the error term when varying the discretisation size. We highlight that these experiments were performed with the original MISOCP formulation (with a single error variable) and Euler’s method. For the sake

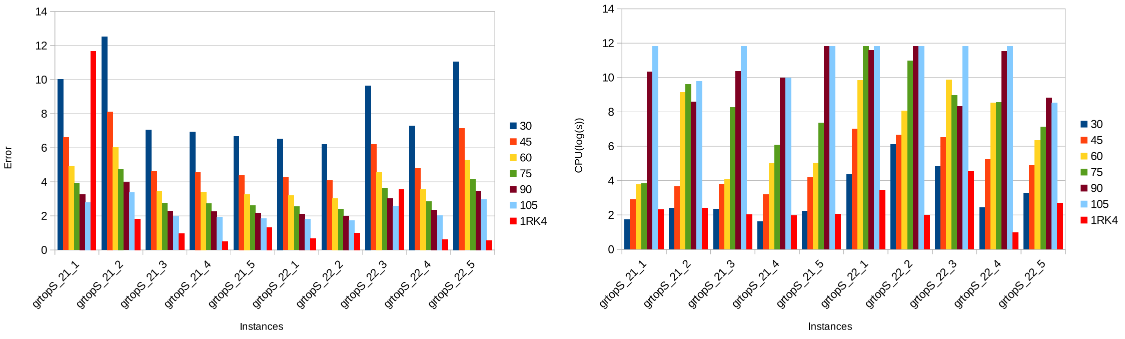
Table 6: Detailed results for error and CPU(s).

| Name | Euler | | TRP | | AB3 | | AB4 | | 1RK4 | | 2RK4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | error | CPU(s) | error | CPU(s) | error | CPU(s) | error | CPU(s) | error | CPU(s) | error | CPU(s) |
| grtopS_21_1 | 15.54 | 5.0 | 15.54 | 6.4 | 15.05 | 8.6 | 14.59 | 13.2 | 11.68 | 5.0 | 11.62 | 5.2 |
| grtopS_21_2 | 18.63 | 6.4 | 16.57 | 8.5 | 15.46 | 12.4 | 16.53 | 19.0 | 1.83 | 5.3 | 1.66 | 4.7 |
| grtopS_21_3 | 12.54 | 5.7 | 13.43 | 7.0 | 13.09 | 9.9 | 13.12 | 12.5 | 0.96 | 4.1 | 0.96 | 6.4 |
| grtopS_21_4 | 12.02 | 3.7 | 12.96 | 6.6 | 12.74 | 8.4 | 13.76 | 12.1 | 0.50 | 3.9 | 0.56 | 3.0 |
| grtopS_21_5 | 12.77 | 5.5 | 12.63 | 8.2 | 13.38 | 10.1 | 12.51 | 19.1 | 1.32 | 4.2 | 1.35 | 3.3 |
| grtopS_22_1 | 12.19 | 7.9 | 14.23 | 13.3 | 12.95 | 23.8 | 12.83 | 22.8 | 0.68 | 11.0 | 1.48 | 12.8 |
| grtopS_22_2 | 12.24 | 6.6 | 12.95 | 9.5 | 13.24 | 12.2 | 12.27 | 20.2 | 1.01 | 4.0 | 1.04 | 5.2 |
| grtopS_22_3 | 13.07 | 11.5 | 13.12 | 12.5 | 14.64 | 23.9 | 13.72 | 41.3 | 3.55 | 23.6 | 4.54 | 9.9 |
| grtopS_22_4 | 14.06 | 9.4 | 12.91 | 8.9 | 13.50 | 15.3 | 13.66 | 29.4 | 0.63 | 2.0 | 0.63 | 4.1 |
| grtopS_22_5 | 14.95 | 9.6 | 13.58 | 10.0 | 13.15 | 11.6 | 15.10 | 15.9 | 0.56 | 6.5 | 0.63 | 7.0 |
| avg. | **13.80** | **7.1** | **13.79** | **9.1** | **13.72** | **13.6** | **13.81** | **20.5** | **2.27** | **7.0** | **2.45** | **6.1** |
| grtopS_101_1 | 23.04 | 3601.6 | 26.40 | 3600.6 | 29.45 | 2603.9 | 25.66 | 2863.3 | 1.37 | 1732.4 | 1.24 | 2049.6 |
| grtopS_101_2 | 23.17 | 3600.5 | 20.50 | 3600.4 | 20.78 | 3600.4 | 21.30 | 3600.3 | 1.24 | 3600.4 | 1.22 | 3600.4 |
| grtopS_101_3 | 30.60 | 3600.5 | 25.93 | 3600.2 | 30.49 | 3600.5 | 23.68 | 3600.4 | 4.11 | 3600.5 | 3.50 | 3600.4 |
| grtopS_101_4 | 23.39 | 160.3 | 23.27 | 367.2 | 24.80 | 753.3 | 24.36 | 1083.1 | 0.83 | 363.1 | 1.01 | 889.0 |
| grtopS_101_5 | 24.94 | 3600.7 | 28.66 | 3600.5 | 26.31 | 3600.4 | 22.48 | 3600.4 | 2.22 | 3600.3 | 6.89 | 3600.3 |
| grtopS_102_1 | 25.91 | 3600.5 | 22.28 | 3600.4 | 26.11 | 3600.5 | 24.67 | 3600.4 | 1.64 | 3600.3 | 0.78 | 3600.5 |
| grtopS_102_2 | 24.60 | 3600.6 | 33.02 | 3600.6 | 27.00 | 3600.5 | 32.97 | 3600.4 | 2.85 | 3600.5 | 8.35 | 3600.7 |
| grtopS_102_3 | 30.08 | 3600.5 | 26.27 | 3600.4 | 27.11 | 3600.2 | 25.29 | 3600.4 | 4.09 | 3600.4 | 1.69 | 1247.1 |
| grtopS_102_4 | 22.01 | 3600.5 | 27.28 | 3600.5 | 21.67 | 3600.6 | 19.75 | 3600.4 | 1.07 | 3600.4 | 2.20 | 3600.6 |
| grtopS_102_5 | 26.27 | 3600.3 | 35.81 | 3600.6 | 29.63 | 3600.3 | 26.99 | 3600.5 | 0.93 | 3600.6 | 1.26 | 3600.4 |
| avg. | **25.40** | **3256.6** | **26.94** | **3277.1** | **26.33** | **3216.1** | **24.71** | **3275.0** | **2.04** | **3089.9** | **2.81** | **2938.9** |

Table 7: Analysis of the error for the individual components of the state vector.

| Name | $x$ | $y$ | $h$ | $v$ | $\gamma$ | $\varphi$ | Sum | CPU(s) | $\varepsilon$ | CPU(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| grtopS_21_1 | 0.00 | 11.58 | 0.00 | 0.00 | 0.68 | 0.00 | 12.26 | 6.34 | 15.54 | 4.98 |
| grtopS_21_2 | 0.00 | 0.01 | 0.00 | 0.00 | 1.22 | 0.14 | 1.37 | 10.67 | 18.63 | 6.36 |
| grtopS_21_3 | 0.00 | 0.79 | 0.00 | 0.00 | 1.14 | 0.00 | 1.93 | 6.58 | 12.54 | 5.70 |
| grtopS_21_4 | 0.00 | 0.38 | 0.00 | 0.00 | 0.66 | 0.00 | 1.04 | 2.77 | 12.02 | 3.68 |
| grtopS_21_5 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.99 | 3.77 | 12.77 | 5.54 |
| grtopS_22_1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.75 | 0.00 | 0.75 | 6.47 | 12.19 | 7.85 |
| grtopS_22_2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.89 | 0.00 | 0.89 | 5.01 | 12.24 | 6.59 |
| grtopS_22_3 | 0.00 | 0.59 | 0.00 | 0.00 | 1.20 | 0.04 | 1.83 | 12.80 | 13.07 | 11.50 |
| grtopS_22_4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.66 | 0.00 | 0.66 | 1.47 | 14.06 | 9.42 |
| grtopS_22_5 | 0.00 | 0.47 | 0.00 | 0.00 | 0.59 | 0.35 | 1.40 | 5.97 | 14.95 | 9.62 |

of comparison, we added a column representing the results from using the 1RK4 method with $N = 30$. Figure 4a shows the magnitude of the error for each number of collocation points on the small range instances of group GRTOP-S2 (expressed in the horizontal axis). One can notice that the error decreases as the number of collocation points increases, as expected. The opposite happens with the CPU times. In Figure 4b, these have been plotted in *log* scale.



(a) Magnitude of the error.

(b) CPU(s) in log scale.

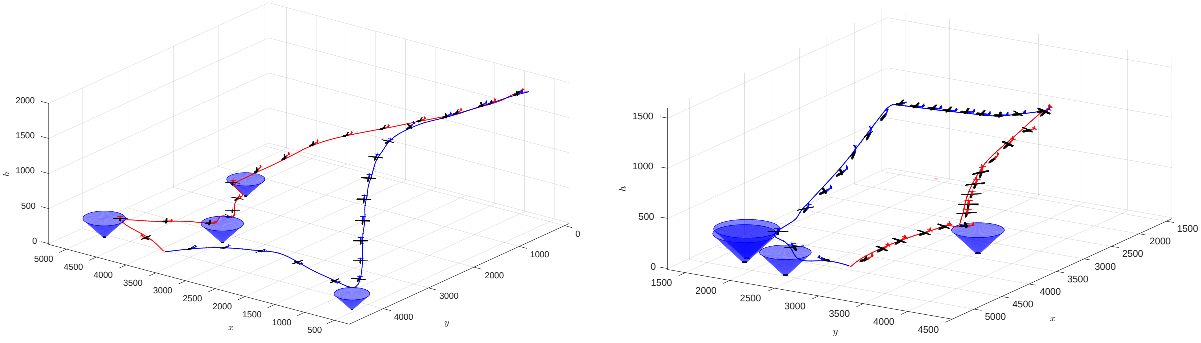
Figure 4: Behaviour of error and CPU times for several values of $N$.

### 5.4. *Results for medium range instances*

In this section, we present the results of our model for a subset of medium range instances. Here, we have chosen Euler's discretisation method due to its better average performance among the lower-order methods (Section 5.3). The number of collocation points $N$ has been set to 60 and the flight time horizon $\tau_f$ has been chosen by the same procedure as described in Section 5.2.

In Table 8, we show the results of our model for instances with $3, 4, 5$ and 6 waypoints. One will notice that running times for the instances solved to optimality have substantially increased compared to the small range instances, as well as the gap for the instances were optimality could not be proved. This happens because of the larger discretisation size that we have applied. Our choice on larger discretisation sizes seeks to guarantee the convergence of the integration methods. We also highlight that errors remain small compared to the range of the instances. The largest error for the results in Table 8 (89.83) represents only 1.8% of the magnitude of position variables related to this instance.

Figure 5 depicts the optimal solution for two instances, `grtopM_41_5` and `grtopM_42_3`. For illustration purposes, we have approximated the trajectories between collocation points by using splines. In the solution of instance `grtopM_41_5`, the flight times are 156s and 152s, for the first and second gliders, respectively, and the step size equals 4.34s. In the solution of instance `grtopM_42_3`, the flight times are 139s and 85s, for the first and second gliders, respectively, and the step size equals 3.39s. We have also provided video animations of feasible and optimal solutions of several instances as supplementary material and through the website Coutinho (2017).



(a) Solution of `grtopM_41_5`.

(b) Solution of `grtopM_42_3`.

Figure 5: Depiction of the optimal solutions of two medium range instances.

### 5.5. *Routing and trajectory optimisation for disaster assessment*

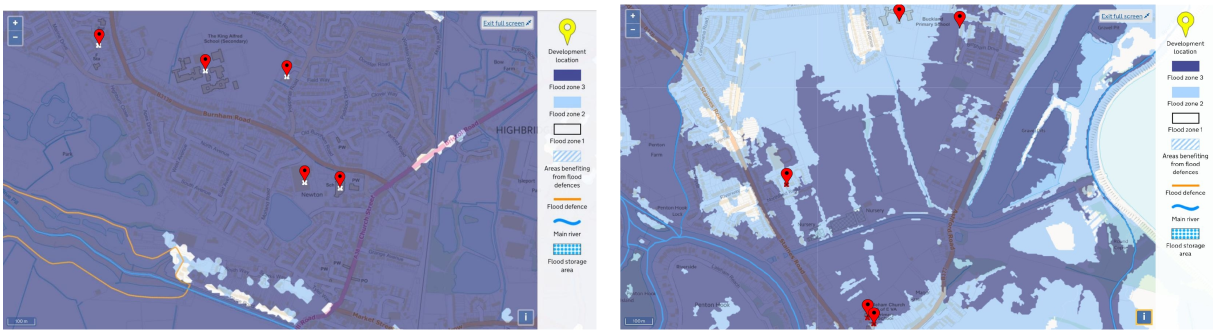
A number of instances based on UK cities prone to flooding has been created. They represent hypothetical flooding scenarios in the cities of Boston, Highbridge, London, Moore (Warrington) and Portsmouth. The so-called *UK instances* have been constructed as follows. We searched for flood risk-prone cities in the UK by using risk information maps provided at Government Digital Service (2017).

For each city, waypoints have been placed in locations with a high concentration of potential flooding victims, such as hospitals, schools, nurseries, residential areas, asylums and industries (which could become possible environmental hazards in a disaster situation), using Google maps. Next, the geographic coordinates of waypoints were collected and converted into Euclidean coordinates. The geometry of the

Table 8: Results for a number of medium range instances.

| Name | Fleet | UB | LB | Error | CPU(s) | Tree | Gap(%) |
|---|---|---|---|---|---|---|---|
| grtopM_31_1 | 1 | 94.49 | 94.49 | 38.49 | 52.84 | 6559 | 0.00% |
| grtopM_31_2 | 1 | 83.43 | 83.43 | 28.43 | 99.77 | 10871 | 0.00% |
| grtopM_31_3 | 1 | 105.56 | 105.56 | 46.56 | 150.12 | 50459 | 0.00% |
| grtopM_31_4 | 1 | 135.85 | 135.85 | 76.85 | 35.71 | 4239 | 0.00% |
| grtopM_31_5 | 1 | 108.36 | 108.36 | 49.36 | 40.19 | 5323 | 0.00% |
| grtopM_32_1 | 1 | 73.59 | 73.59 | 24.59 | 167.19 | 15641 | 0.00% |
| grtopM_32_2 | 1 | 103.17 | 103.17 | 44.18 | 1098.67 | 262303 | 0.00% |
| grtopM_32_3 | 1 | 124.55 | 124.55 | 65.55 | 102.86 | 10618 | 0.00% |
| grtopM_32_4 | 1 | 95.88 | 75.81 | 37.88 | 3600.83 | 783809 | 21.00% |
| grtopM_32_5 | 1 | 91.79 | 91.79 | 33.80 | 260.57 | 38517 | 0.00% |
| grtopM_41_1 | 2 | 138.36 | 138.35 | 59.36 | 1943.31 | 115060 | 0.00% |
| grtopM_41_2 | 2 | 124.20 | 102.43 | 42.20 | 3600.47 | 185657 | 18.00% |
| grtopM_41_3 | 1 | 113.71 | 87.80 | 33.71 | 3600.79 | 334907 | 23.00% |
| grtopM_41_4 | 2 | 96.20 | 96.20 | 25.20 | 3080.92 | 306189 | 0.00% |
| grtopM_41_5 | 2 | 100.45 | 100.44 | 31.45 | 3174.82 | 192406 | 0.00% |
| grtopM_42_1 | 2 | 92.30 | 85.59 | 34.30 | 3600.32 | 122582 | 7.00% |
| grtopM_42_2 | 2 | 86.12 | 65.83 | 27.12 | 3600.28 | 99994 | 24.00% |
| grtopM_42_3 | 2 | 109.12 | 109.12 | 45.12 | 301.81 | 10467 | 0.00% |
| grtopM_42_4 | 2 | 153.41 | 97.07 | 71.41 | 3600.48 | 102879 | 37.00% |
| grtopM_42_5 | 1 | 129.94 | 129.94 | 54.94 | 2376.52 | 177920 | 0.00% |
| grtopM_51_1 | 2 | 160.34 | 99.19 | 82.34 | 3600.70 | 272174 | 38.00% |
| grtopM_51_2 | 2 | 122.69 | 107.50 | 43.69 | 3600.44 | 146891 | 12.00% |
| grtopM_51_3 | 2 | 154.80 | 146.37 | 72.80 | 3600.26 | 167945 | 5.00% |
| grtopM_51_4 | 2 | 116.08 | 105.20 | 28.08 | 3600.47 | 250171 | 9.00% |
| grtopM_51_5 | 2 | 137.60 | 88.60 | 58.60 | 3600.51 | 293565 | 36.00% |
| grtopM_52_1 | 2 | 148.99 | 148.99 | 59.99 | 1083.69 | 26234 | 0.00% |
| grtopM_52_2 | 2 | 93.96 | 79.15 | 41.96 | 3600.53 | 168447 | 16.00% |
| grtopM_52_3 | 2 | 119.81 | 85.83 | 36.81 | 3600.53 | 151592 | 28.00% |
| grtopM_52_4 | 2 | 141.57 | 95.62 | 51.57 | 3600.67 | 147272 | 32.00% |
| grtopM_52_5 | 2 | 159.38 | 91.63 | 75.38 | 3600.77 | 236883 | 43.00% |
| grtopM_61_1 | 3 | 168.25 | 107.26 | 37.25 | 3600.59 | 121569 | 36.00% |
| grtopM_61_2 | 3 | 126.14 | 112.61 | 49.14 | 3600.48 | 136656 | 11.00% |
| grtopM_61_3 | 3 | 182.83 | 79.91 | 89.83 | 3600.58 | 96375 | 56.00% |
| grtopM_61_4 | 2 | 159.76 | 112.63 | 73.76 | 3600.43 | 149115 | 30.00% |
| grtopM_61_5 | 2 | 121.34 | 80.89 | 43.34 | 3600.54 | 105711 | 33.00% |
| grtopM_62_1 | 2 | 141.84 | 81.54 | 50.84 | 3600.46 | 82085 | 43.00% |
| grtopM_62_2 | 2 | 167.52 | 78.92 | 88.52 | 3600.47 | 84507 | 53.00% |
| grtopM_62_3 | 3 | 193.56 | 154.27 | 78.56 | 3600.44 | 40665 | 20.00% |
| grtopM_62_4 | 3 | 163.16 | 78.92 | 40.16 | 3600.65 | 108157 | 52.00% |
| grtopM_62_5 | 2 | 153.53 | 91.17 | 50.53 | 3600.61 | 79393 | 41.00% |

waypoints and landing sites has been chosen in order to match the dimensions of the real locations. Figure 6 depicts the maps indicating the selection of waypoints for two flood-prone areas of two cities in the UK, namely, London and Highbridge.



(a) Waypoints selection for `grtopS_lond1`.

(b) Waypoints selection for `grtopS_highb`.

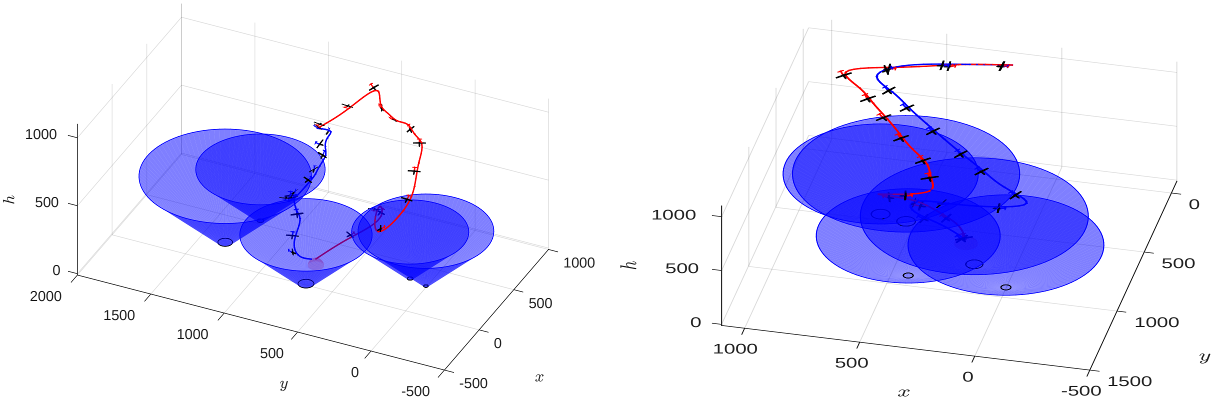
Figure 6: Maps of flood-prone areas in London and Highbridge. Source: Government Digital Service (2017).

Table 9 shows the results of our experiments with the generated UK instances. In this table, column $\#W$ represents the number of waypoints in the instance. The remaining columns keep the same meaning as in Table 8. The number of landing sites for all instances was set to 1. The number of collocation points has been set to $N = 30$. Figure 7 depicts the solutions for 2 UK instances, namely, `grtopS_lond1` (London) and `grtopS_highb` (Highbridge).

Table 9: Results of the GRTOP formulation for the UK instances.

| Name | #W | Fleet | UB | LB | Error | CPU(s) | Tree | Gap(%) |
|---|---|---|---|---|---|---|---|---|
| grtop_bost1 | 7 | 3 | 73.03 | 73.03 | 34.03 | 263.95 | 9973 | 0.00% |
| grtop_bost2 | 7 | 2 | 75.38 | 75.15 | 31.38 | 3600.16 | 391381 | 0.00% |
| grtop_bost3 | 7 | 3 | 69.07 | 69.07 | 31.07 | 255.58 | 24063 | 0.00% |
| grtop_highb | 5 | 2 | 57.08 | 57.08 | 23.08 | 34.23 | 830 | 0.00% |
| grtop_lond1 | 5 | 2 | 59.44 | 59.44 | 27.44 | 54.46 | 4923 | 0.00% |
| grtop_lond2 | 7 | 3 | 73.86 | 73.86 | 34.86 | 106.89 | 3059 | 0.00% |
| grtop_lond3 | 10 | 1 | 77.41 | 75.09 | 34.41 | 3600.42 | 241168 | 3.00% |
| grtop_lond4 | 7 | 3 | 75.48 | 75.48 | 30.48 | 142.17 | 5207 | 0.00% |
| grtop_moore | 7 | 1 | 80.39 | 80.39 | 25.40 | 51.51 | 751 | 0.00% |
| grtop_prtsm | 5 | 1 | 65.59 | 65.59 | 28.59 | 103.49 | 16355 | 0.00% |

From the results in Table 9, one can observe that all instances are solved to optimality, except `grtop_lond3`. The results of numerical experiments with the UK instances are similar to the results achieved for the small-ranged random instances.



(a) Solution of `grtopS_lond1`.

(b) Solution of `grtopS_highb`.

Figure 7: Depiction of the optimal solutions of two UK instances.

### *5.5.1. Alternative objective functions*

In this section, we present the results of computational experiments involving the alternative objective functions proposed in Section 3 for each group of small range instances. These results are summarised in Tables 10 and 11.

Table 10, presents the average values of the upper bound (**UB**), lower bound (**LB**) and error (**Error**) associated to each objective considered in Section 3, namely, minimising the total flight time, minimising the makespan and minimising the total latency, corresponding to columns **Routing Cost**, **Makespan** and **Latency**, respectively. In this table, one can observe, on average, the range of values for each objective function. In general, these ranges overlap for the smaller instances and gradually become distinct as the instance size increases. This table also highlights the influence of the error term on each objective. The error values are significantly smaller for the makespan when compared to the other objectives. This can be explained by the fact that the makespan accounts for the duration of a single route, therefore its magnitude is comparatively smaller than the magnitude of the routing cost and latency objectives. We recall that the objective functions considered in our formulation are composed by two terms, a time measure and the error term. During the optimisation, one term might be prioritised over the other depending on their relative values in order to achieve the overall minimum. Therefore, the error term has a higher relative importance on the overall objective function value when the makespan is optimised. The last row of Table 10 shows the contribution of the error term to the overall objective function value, on average. Our experiments confirm that the error term has a higher impact on the makespan objective in comparison with the other ones.

Table 10: Average upper bound, lower bound and error values for each different objective.

| | Routing Cost | | | Makespan | | | Latency | | |
|---|---|---|---|---|---|---|---|---|---|
| Group | UB | LB | Error | UB | LB | Error | UB | LB | Error |
| GRTOP-S2 | 32.5 | 32.5 | 13.8 | 32.5 | 32.5 | 13.8 | 32.6 | 32.6 | 14.4 |
| GRTOP-S3 | 35.8 | 35.8 | 14.2 | 35.8 | 35.8 | 14.2 | 46.3 | 46.3 | 19.2 |
| GRTOP-S4 | 50.1 | 50.1 | 20.3 | 35.8 | 35.4 | 14.6 | 53.1 | 53.1 | 22.4 |
| GRTOP-S5 | 55.7 | 55.6 | 22.9 | 38.5 | 38.5 | 15.1 | 62.7 | 62.7 | 24.2 |
| GRTOP-S6 | 60.1 | 58.7 | 24.8 | 35.4 | 31.8 | 15.1 | 60.6 | 60.6 | 23.4 |
| GRTOP-S7 | 62.8 | 62.4 | 25.3 | 36.3 | 32.5 | 15.6 | 73.8 | 73.8 | 27.1 |
| GRTOP-S8 | 59.3 | 57.1 | 23.5 | 35.0 | 30.4 | 14.3 | 74.7 | 71.9 | 27.7 |
| GRTOP-S9 | 60.7 | 57.1 | 24.2 | 35.5 | 31.1 | 15.3 | 76.8 | 76.2 | 29.4 |
| GRTOP-S10 | 64.0 | 55.9 | 25.4 | 39.0 | 28.6 | 16.5 | 86.4 | 85.6 | 30.0 |
| Error÷LB(%) | | | 41.8% | | | 45.3% | | | 38.7% |

Table 11 compares the values of the routing cost, makespan and latency when each objective is optimised in turn. More specifically, the first row of this table denotes the objective function being optimised, while the second row represents the averages of the routing cost, makespan and latency for the corresponding solutions. In this table, column **Fleet size** shows the number of available gliders for each group of instances. Table 11 has been further divided to reflect each run of the different objectives. The average total flight time (a.k.a. routing cost) is shown in column **RC**, this value can be computed as

$$\sum_{g\in G}\sum_{i\in L}\sum_{t\in T} tb_{git}\eta,$$

by summing up the landing times of each glider. The makespan is shown in column **MKS** according to the formula:

$$\max_{g\in G}\left\{\sum_{i\in L}\sum_{t\in T} tb_{git}\eta\right\}.$$

Finally, the latency objective is presented in column **LTC**, this value can be computed as

$$\sum_{g\in G}\sum_{i\in L}\sum_{t\in T} ta_{git}\eta.$$

For each subgroup, column **Used** show the average number of deployed gliders. Results from Table 11 show that solutions regarding the makespan and latency objectives lead to a higher fleet utilisation (all gliders are used for the latency objective). It is possible to reduce the duration of trajectories and waiting times by using the maximum allowed number of gliders.

On the other hand, the total time in which gliders remain airborne is larger for latency and makespan objectives when compared to the routing cost objective. Our solutions show that the makespan objective usually produces routes with similar durations, however results from the computational experiments presented in Appendix A demonstrate that the GRTOP formulation with this objective is harder to solve. One may notice that the values in column MKS regarding the routing cost objective are actually smaller than the values for the makespan objective. Two main factors cause such behaviour. The first one is related to the influence of the error term in the minimisation of the makespan, as discussed above. The second ones relates to the difficulty of solving the GRTOP with the makespan objective.

Table 11: Comparison between the total flight time, makespan and latency objectives.

| | | Routing Cost | | | | Makespan | | | | Latency | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Group | Fleet size | Used | RC | MKS | LTC | Used | RC | MKS | LTC | Used | RC | MKS | LTC |
| GRTOP-S2 | 1 | 1.0 | 38.8 | 38.8 | 45.4 | 1.0 | 38.8 | 38.8 | 45.4 | 1.0 | 52.5 | 52.5 | 37.8 |
| GRTOP-S3 | 1 | 1.0 | 48.1 | 48.1 | 95.1 | 1.0 | 48.1 | 48.1 | 95.1 | 1.0 | 53.2 | 53.2 | 60.3 |
| GRTOP-S4 | 2 | 1.8 | 70.8 | 41.4 | 100.6 | 2.0 | 100.3 | 50.3 | 130.6 | 2.0 | 102.1 | 56.7 | 72.7 |
| GRTOP-S5 | 2 | 1.7 | 77.3 | 42.0 | 121.6 | 2.0 | 109.8 | 55.0 | 177.2 | 2.0 | 110.1 | 60.7 | 90.0 |
| GRTOP-S6 | 3 | 2.3 | 86.7 | 36.1 | 113.2 | 3.0 | 141.9 | 49.8 | 168.8 | 3.0 | 158.9 | 64.8 | 91.5 |
| GRTOP-S7 | 3 | 2.2 | 96.0 | 39.5 | 151.7 | 3.0 | 155.9 | 53.3 | 253.1 | 3.0 | 168.9 | 65.8 | 120.1 |
| GRTOP-S8 | 3 | 2.7 | 92.1 | 39.0 | 168.4 | 2.9 | 149.3 | 53.5 | 260.1 | 3.0 | 163.1 | 70.9 | 120.9 |
| GRTOP-S9 | 3 | 2.3 | 92.8 | 38.8 | 200.9 | 2.9 | 145.0 | 51.4 | 277.0 | 3.0 | 148.2 | 61.3 | 120.3 |
| GRTOP-S10 | 3 | 2.7 | 99.5 | 46.6 | 219.6 | 3.0 | 162.6 | 58.2 | 336.5 | 3.0 | 178.9 | 70.3 | 145.0 |

### *5.5.2. Testing different launching altitudes*

The initial launching altitude of the gliders can influence the success of an aerial survey operation from a disaster assessment perspective. For example, if the gliders are released from a very high altitude more time is going to be spent descending to the level of the waypoints. On the other hand, not providing the gliders with enough initial height might cause them to land before visiting all the required waypoints. The launching altitude of the gliders can be incorporated into our model as a variable (as well as the initial $xy$ position), however preliminary experiments showed that this would significantly increase computing times.

In order to assess the effect of the gliders' initial altitude from the application and modelling point of view, we performed computational experiments by varying the initial height for the UK instances. A summary of results from these experiments can be found in Table 12. In this table, results are grouped into four different initial altitudes, namely, $600, 800, 1000$ and $1200$ metres. Two columns in each group show the optimal total flight times (**Time**) and computing times (**CPU(s)**).

From the results in Table 12 one can observe that increasing the initial height leads to higher flight times as expected. It means that gliders have to spend more time soaring in order to reach a proper altitude to take pictures. On the other hand, lower initial altitudes, such as 600 metres, usually lead to models that are more difficult to solve (requiring longer computing times).

### *5.5.3. The impact of the wind strength parameter*

In order to assess the influence of the wind strength parameter ($\beta$) on the performance of the gliders, computational experiments were carried out by setting $\beta$ to six different values. Since $\beta$ is used in the computation of the wind speed and its derivative ($U(h)$ and $\dot{U}$ in Equations (11) and (12), respectively),

Table 12: Results of the GRTOP formulation for different launching altitudes.

| | 600 | | 800 | | 1000 | | 1200 | |
|---|---|---|---|---|---|---|---|---|
| Name | Time | CPU(s) | Time | CPU(s) | Time | CPU(s) | Time | CPU(s) |
| grtopS_bost1 | 135.9 | 277.0 | 158.6 | 272.1 | 176.7 | 264.0 | 190.3 | 311.3 |
| grtopS_bost2 | 156.1 | 2840.9 | 183.6 | 3600.7 | 197.4 | 3600.2 | 202.0 | 519.8 |
| grtopS_bost3 | 118.8 | 289.5 | 161.5 | 217.2 | 180.5 | 267.5 | 209.0 | 258.5 |
| grtopS_highb | 62.1 | 51.9 | 75.6 | 37.1 | 89.1 | 34.1 | 99.9 | 40.0 |
| grtopS_lond1 | 107.9 | 111.4 | 116.2 | 58.4 | 132.8 | 54.8 | 141.1 | 79.1 |
| grtopS_lond2 | 101.4 | 142.4 | 118.3 | 77.5 | 128.4 | 107.4 | 148.7 | 94.0 |
| grtopS_lond3 | 100.2 | 3600.2 | 120.2 | 1737.1 | 143.6 | 3600.6 | 167.0 | 1383.2 |
| grtopS_lond4 | 93.5 | 174.1 | 101.8 | 114.8 | 121.0 | 144.5 | 140.3 | 145.2 |
| grtopS_moore | 73.5 | 94.6 | 90.3 | 41.0 | 113.4 | 51.5 | 119.7 | 53.5 |
| grtopS_prtsm | 99.3 | 287.3 | 102.6 | 195.5 | 122.5 | 106.5 | 125.8 | 88.4 |

a different linear dynamical system was generated for each value of $\beta$. Table 13 summarises the results of our experiments. In this table, **length** represents the average trajectory length of a solution calculated as $length = \frac{\sum_{i=1}^{q} l_i}{q}$, where $l_i$ represents the length of the $i$-th trajectory and $q$ denotes the number of gliders in a solution. Analogously, **speed** represents the average speed of a solution computed as $speed = \frac{\sum_{i=1}^{q} \frac{l_i}{\Delta_i}}{q}$, where $\Delta_i$ represents the duration of trajectory $i$.

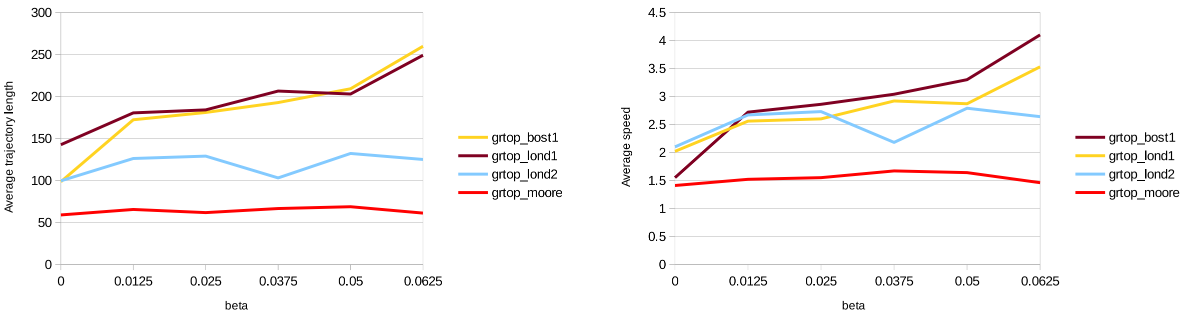
Table 13: Results of the GRTOP formulation for different values of $\beta$.

| Instance | 0 | | 0.0125 | | 0.025 | | 0.0375 | | 0.05 | | 0.0625 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *length* | speed | *length* | speed | *length* | speed | *length* | speed | *length* | speed | *length* | speed |
| grtop_bost1 | 98.5 | 1.6 | 172.3 | 2.7 | 181.0 | 2.9 | 192.8 | 3.0 | 209.2 | 3.3 | 259.8 | 4.1 |
| grtop_bost2 | 158.6 | 2.2 | 178.3 | 2.7 | 247.1 | 3.4 | 268.7 | 3.6 | 269.7 | 3.6 | 247.0 | 3.5 |
| grtop_bost3 | 256.6 | 4.0 | 275.2 | 4.3 | 286.2 | 4.5 | 281.4 | 4.4 | 284.8 | 4.4 | 283.4 | 4.1 |
| grtop_highb | 80.9 | 1.7 | 102.2 | 2.1 | 113.4 | 2.3 | 180.5 | 3.9 | 196.6 | 4.3 | 169.3 | 3.7 |
| grtop_lond1 | 142.7 | 2.0 | 180.5 | 2.6 | 184.1 | 2.6 | 206.5 | 2.9 | 203.0 | 2.9 | 249.1 | 3.5 |
| grtop_lond2 | 99.6 | 2.1 | 126.2 | 2.7 | 129.0 | 2.7 | 103.1 | 2.2 | 132.2 | 2.8 | 125.1 | 2.6 |
| grtop_lond3 | 186.8 | 3.1 | 205.8 | 3.4 | 134.4 | 2.2 | 128.7 | 2.1 | 157.7 | 2.6 | 144.1 | 2.4 |
| grtop_lond4 | 119.4 | 2.7 | 165.5 | 3.8 | 207.4 | 4.7 | 172.9 | 3.9 | 164.2 | 3.7 | 223.0 | 4.8 |
| grtop_moore | 59.0 | 1.4 | 65.5 | 1.5 | 61.8 | 1.6 | 66.6 | 1.7 | 68.8 | 1.6 | 61.2 | 1.5 |
| grtop_prtsm | 80.5 | 1.1 | 242.7 | 3.3 | 245.3 | 3.4 | 235.5 | 3.2 | 196.6 | 2.7 | 171.7 | 2.4 |

From the results in Table 13, one can observe that the average trajectory lengths and speeds tend to increase for some of the UK instances as the wind strength parameter increases. However, significant changes in the objective function value for the same instances were not observed. Figure 8 shows the changes in the average trajectory length and average speed for a number of different instances. One can observe that for the `grtop_bost1` and `grtop_lond1` instances the average lengths and speeds tend to increase with the value of $\beta$, while for the `grtop_lond2` and `grtop_moore` instances this trend cannot be observed. Figure 9 depicts the behaviour of $\gamma$, $\varphi$ and $\mu$ for the instance `grtop_lond3` for $\beta = 0$ and $\beta = 0.0625$.

## 6. Conclusion

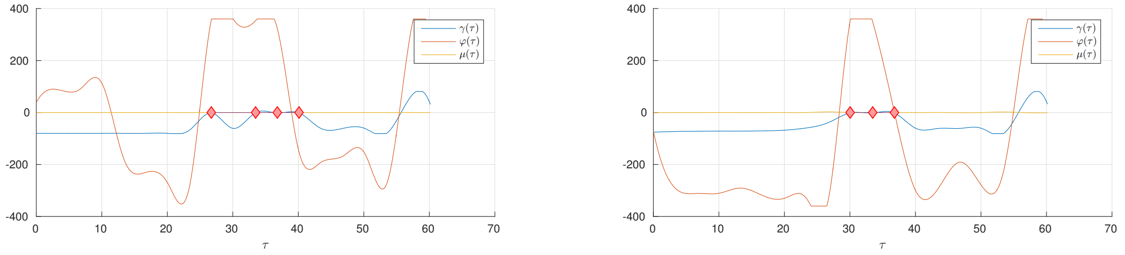
In this paper, we have considered the GRTOP. This problem has been motivated by a disaster assessment application. In the GRTOP, we are asked to simultaneously find optimal routes and trajectories for a fleet of unmanned aerial gliders. The fleet of gliders is modelled by their EOMs, which consist of



(a) Average trajectory length for different values of $\beta$.

(b) Average speed for different values of $\beta$.

Figure 8: Influence of the wind speed parameter on the average trajectory length and speed.



(a) Solution for $\beta = 0$.

(b) Solution for $\beta = 0.0625$.

Figure 9: Aircraft angles for different wind strengths. Red markers highlight the time instants were waypoints are visited.

a set of ordinary differential equations. We propose a novel MINLP formulation for the GRTOP, which accounts for routing and trajectory optimisation in an integrated framework. In order to avoid a non-convex formulation, we linearise the gliders' EOM using a set of steady-state conditions. This reduces the MINLP to a MISOCP problem. To allow for a more tractable problem while keeping the quality of our solutions, we relax the linear dynamical system and penalise the corresponding error term in the objective function. We present several discretisation methods for the resulting linear dynamical system.

In order to test our model, we have generated 180 random instances. We compared different commercial solvers on a subset of instances, namely, CPLEX, Gurobi and Xpress. Based on the results, Xpress was chosen for the next experiments. The second set of experiments was concerned about the discretisation methods and errors. Higher order integration methods were able of achieving smaller error magnitudes, but at the expense of CPU times. On the other hand, lower order methods typically reduced computation times, at the expense of solution accuracy. A detailed analysis has been carried out regarding the magnitude of the error. It was shown that the errors represent a small fraction of the magnitudes of the state variables and therefore are considered acceptable. Experiments on a subset of instances showed that the error is mostly due to the position variables, which have higher magnitude than the variables regarding angular orientation and airspeed. Finally, we studied the effect of increasing the number of collocation points on the magnitudes of error and CPU times. In general, the trade-off between error magnitudes and CPU times becomes clear on the choice of $N$.

The results for medium range instances showed that acceptable accuracy can also be achieved for long range flights even by employing lower-order discretisation methods. In order to guarantee the convergence of integration methods, the number of collocation points has to be increased. This has a direct influence on the number of medium range instances solved to optimality. Computational experiments were performed with the so-called UK instances. These instances were created for disaster assessment in UK cities with high flooding risk. Results showed that the model presents a good performance over instances from this group. Motivated by our disaster assessment application, we performed computational experiments involving different objective functions. In addition, we analysed the effects of the wind strength parameter and initial launching altitude on the quality of our solutions in terms of response time.

The accuracy of our solutions could be further improved by applying sequential linearisation at each node of the B&B tree, but this would dramatically increase the computation times. Further research could also focus on an adaptive integration method instead of the fixed time grid we have used. However, this improvements would also increase the CPU times.

Our formulation is capable to tackle a good number of test cases. However, for the instances with a larger number of waypoints and for higher discretisation sizes, we were unable to prove optimality. This motivates the development of heuristic and metaheuristic methods that should be able to find good solutions in small CPU times. Furthermore, future research might look at enriching the GRTOP formulation, by incorporating more sophisticated EOMs, considering, for example, more complex wind profiles.

### Acknowledgements

The authors would like to thank Jodie Walshe from RNLI and Dr. Andràs Sóbester from the Engineering Department at the University of Southampton for suggesting this problem. We also would like to thank Dr. Christopher Crispin for his support and Mrs. Fatine Mrabet for creating the UK instances. The first author received grants from CNPq [Grant no. 202241/2041-9].

### Appendix A. Computational performance for different objective functions

The results presented in Table A.14 correspond to the experiments with different objective functions in the GRTOP formulation. From these results, one can observe that the formulation becomes harder if the makespan objective is taken into account. On the other hand, the objective regarding latency is significantly easier according to the values of the average gap and computing times.

Table A.14: Summary of the results for different objective functions.

| Group | Status | Error | Gap(%) | CPU(s) | Tree |
|---|---|---|---|---|---|
| | Total flight time | | | | |
| GRTOP-S2 | (10,0,0) | **1.000** | **0.00%** | 7.124 | **800.60** |
| GRTOP-S3 | (10,0,0) | **1.000** | **0.00%** | 14.024 | **3250.30** |
| GRTOP-S4 | (10,0,0) | 1.393 | **0.00%** | 104.985 | 18475.50 |
| GRTOP-S5 | (9,1,0) | 1.513 | 0.10% | 727.232 | 186538.20 |
| GRTOP-S6 | (7,3,0) | 1.643 | 2.40% | 1294.716 | 140235.50 |
| GRTOP-S7 | (9,1,0) | 1.625 | 0.60% | 1207.613 | 138074.90 |
| GRTOP-S8 | (5,5,0) | 1.640 | 3.40% | 2226.280 | 200746.40 |
| GRTOP-S9 | (7,3,0) | 1.580 | 5.20% | 2157.578 | 173651.90 |
| GRTOP-S10 | (1,9,0) | 1.541 | 12.30% | 3256.597 | **201550.40** |
| avg. | - | 1.44 | 2.67% | 1221.79 | 118147.08 |
| | Makespan | | | | |
| GRTOP-S2 | (10,0,0) | **1.000** | **0.00%** | 7.110 | **800.60** |
| GRTOP-S3 | (10,0,0) | **1.000** | **0.00%** | 13.985 | **3250.30** |
| GRTOP-S4 | (9,1,0) | **1.000** | 1.10% | 908.108 | **188519.30** |
| GRTOP-S5 | (9,1,0) | **1.000** | 0.10% | 845.027 | **211741.00** |
| GRTOP-S6 | (4,6,0) | **1.000** | 9.00% | 2367.894 | **208624.10** |
| GRTOP-S7 | (5,5,0) | **1.000** | 8.50% | 2451.651 | **199228.00** |
| GRTOP-S8 | (2,8,0) | **1.000** | 12.40% | 3030.418 | **213551.90** |
| GRTOP-S9 | (2,8,0) | **1.000** | 10.80% | 3277.839 | **232674.80** |
| GRTOP-S10 | (0,10,0) | **1.000** | 24.80% | 3600.713 | 191401.80 |
| avg. | - | **1.00** | 7.41% | 1833.64 | **161087.98** |
| | Latency | | | | |
| GRTOP-S2 | (10,0,0) | 1.045 | **0.00%** | **4.681** | 97.40 |
| GRTOP-S3 | (10,0,0) | 1.353 | **0.00%** | **7.440** | 571.00 |
| GRTOP-S4 | (10,0,0) | 1.540 | **0.00%** | **22.939** | 1233.40 |
| GRTOP-S5 | (10,0,0) | 1.603 | **0.00%** | **42.374** | 3243.10 |
| GRTOP-S6 | (10,0,0) | 1.550 | **0.00%** | **122.711** | 5858.60 |
| GRTOP-S7 | (10,0,0) | 1.739 | **0.00%** | **627.791** | 52183.00 |
| GRTOP-S8 | (8,2,0) | 1.931 | **3.00%** | **852.686** | 65611.10 |
| GRTOP-S9 | (9,1,0) | 1.917 | **0.60%** | **942.747** | 63686.00 |
| GRTOP-S10 | (8,2,0) | 1.821 | **0.80%** | **1663.378** | 101662.40 |
| avg. | - | 1.61 | **0.49%** | **476.31** | 32682.89 |

## References

Ahmed, E., Hafez, A., Ouda, A., Ahmed, H., & Abd-Elkader, H. (2015). Modelling of a Small Unmanned Aerial Vehicle. *Advances in Robotics & Automation*, *4*. doi:10.4172/2168-9695.1000126.

AMPL (1998). New in ampl: Statuses. http://www.ampl.com/NEW/statuses.html. Accessed: 25/05/2017.

Ariyur, K. B., & Fregene, K. O. (2008). Autonomous tracking of a ground vehicle by a UAV. In *2008 American Control Conference* (pp. 669–671). IEEE. doi:10.1109/ACC.2008.4586569.

Betts, J. T. (2001). *Practical methods for optimal control using nonlinear programming*. Advances in design and control. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Boeing (2000). Operational Use of Angle of Attack on Modern Commercial Jet Airplanes. *Aero Magazine*, (pp. 11–22). URL: http://www.boeing.com/commercial/aeromagazine/aero_12/attack_story.html.

Bower, G. C. (2010). *Boundary Layer Dynamic Soaring for Autonomous Aircraft: Design and Validation*. Ph.D. Thesis Stanford University Stanford, California.

Chakrabarty, A., & Langelaan, J. W. (2011). Energy-Based Long-Range Path Planning for Soaring-Capable Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, *34*, 1002–1015. doi:10.2514/1.52738.

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S. G., & Bian, L. (2017). Drones for disaster response and relief operations: A continuous approximation model. *International Journal of Production Economics*, *188*, 167–184. doi:10.1016/j.ijpe.2017.03.024.

Conway, B. A. (2010). *Spacecraft Trajectory Optimization*. Cambridge University Press.

Coutinho, W. P. (2017). Glider routing. https://www.youtube.com/playlist?list=PL1mldBX67GxrUkuQZSzArTRLbYOowDD45. Accessed: 25/05/2017.

Coutinho, W. P., Battarra, M., & Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, *120*, 116–128. doi:10.1016/j.cie.2018.04.037.

Crispin, C. (2016). *Path Planning Algorithms for Atmospheric Science Applications of Autonomous Aircraft Systems*. Ph.D. Thesis University of Southampton Southampton, UK.

Deittert, M., Richards, A., Toomer, C. A., & Pipe, A. (2009). Engineless Unmanned Aerial Vehicle Propulsion by Dynamic Soaring. *Journal of Guidance, Control, and Dynamics*, *32*, 1446–1457. doi:10.2514/1.43270.

Drew, D. R., Barlow, J. F., & Lane, S. E. (2013). Observations of wind speed profiles over greater london, uk, using a doppler lidar. *Journal of Wind Engineering and Industrial Aerodynamics*, *121*, 98–105. doi:10.1016/j.jweia.2013.07.019.

Flanzer, T. (2012). *Robust Trajectory Optimisation and Control of a Dynamic Soaring Unmanned Aerial Vehicle.*. PhD. Thesis Stanford University Stanford.

Fügenschuh, A., & Müllenstedt, D. (2015). *Flight Planning for Unmanned Aerial Vehicles*. Technical Report AMOS #34(2015) Helmut Schmidt University / University of the Federal Armed Forces Hamburg. Available at `https://www.hsu-hh.de/download-1.5.1.php?brick_id=fyCYI55SQ6oFqe5e`.

Gao, X.-Z., Hou, Z.-X., Guo, Z., Fan, R.-F., & Chen, X.-Q. (2014). Analysis and design of guidance-strategy for dynamic soaring with UAVs. *Control Engineering Practice*, *32*, 218–226. doi:`10.1016/j.conengprac.2013.06.003`.

Golden, B., Kovacs, A., & Wasil, E. (2014). Chapter 14: Vehicle Routing Applications in Disaster Relief. In *Vehicle Routing* MOS-SIAM Series on Optimization (pp. 409–436). Society for Industrial and Applied Mathematics. doi:`10.1137/1.9781611973594.ch14`.

Gordon, R. J. (2006). *Optimal Dynamic Soaring for Full Size Sailplanes*. PhD. Thesis Air Force Institute of Technology Ohio, US.

Government Digital Service (2017). Long term flood risk information. `https://flood-warning-information.service.gov.uk/long-term-flood-risk/map?` Accessed: 20/05/2017.

Hairer, E., Nørsett, S. P., & Wanner, G. (2011). *Solving Ordinary Differential Equations I: Nonstiff Problems* volume 1 of *Springer Series in Computational Mathematics*. Springer.

Hajiyev, C., Soken, H. E., & Vural, S. Y. (2015). Equations of Motion for an Unmanned Aerial Vehicle. In *State Estimation and Control for Low-cost Unmanned Aerial Vehicles* (pp. 9–23). Springer International Publishing.

Harris, M. W., & Acikmese (2013). Maximum Divert for Planetary Landing Using Convex Optimization. *Journal of Optimization Theory and Applications*, *162*, 975–995. doi:`10.1007/s10957-013-0501-7`.

How, J. P., Frazzoli, E., & Chowdhary, G. V. (2015). Linear Flight Control Techniques for Unmanned Aerial Vehicles. In K. P. Valavanis, & G. J. Vachtsevanos (Eds.), *Handbook of Unmanned Aerial Vehicles* (pp. 529–576). Springer Netherlands.

Kara, I., Kara, B. Y., & Kadri, M. (2008). Cumulative Vehicle Routing Problems. In *Vehicle Routing Problem*. Intechopen. (1st ed.).

Keane, A., Scanlan, J., Lock, A., Ferraro, M., Spillane, P., & Breen, J. (2017). Maritime flight trials of the southampton university laser sintered aircraft: Project albatross. *The Aeronautical Journal of the Royal Aeronautical Society*, (p. to appear). Available at `https://eprints.soton.ac.uk/411713/`.

Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., & Balas, G. (2008). Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations. *IEEE Transactions on Control Systems Technology*, *16*, 19–33. doi:`10.1109/TCST.2007.903066`.

Kroo, I. (2001). *Aircraft Design: Synthesis and Analysis*. P.O. Box 20384, Stanford, CA 94309: Desktop Aeronautics. URL: `http://rahauav.com/Library/Design-performance/Aircraft%20Design,%20synthesis%20and%20analysis.pdf` version 0.99.

Langelaan, J. (2007). Long Distance/Duration trajectory optimization for small UAVs. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics.

Laville, S., Ross, A., Topping, A., Gayle, D., & Grierson, J. (2017, June 15). Grenfell tower: firefighters search overnight with toll expected to rise. `https://www.theguardian.com/uk-news/2017/jun/14/fire-24-storey-grenfell-tower-block-white-city-latimer-road-london`.

Maolaaisha, A. (2015). *Free-Flight Trajectory Optimization by Mixed Integer Programming*. Master Thesis University of Hamburg Hamburg.

Matl, P., Hartl, R. F., & Vidal, T. (2018). Workload Equity in Vehicle Routing Problems: A Survey and Analysis. *Transportation Science*, *52*, 239–260. doi:`10.1287/trsc.2017.0744`.

Mittelmann, H. D. (2017). Mixed-integer socp benchmark. `http://plato.asu.edu/ftp/misocp.html`. Accessed: 14/07/2017.

Nedjati, A., Izbirak, G., Vizvari, B., & Arkat, J. (2016). Complete coverage path planning for a multi-uav response system in post-earthquake assessment. *Robotics*, *5*. URL: `http://www.mdpi.com/2218-6581/5/4/26`. doi:`10.3390/robotics5040026`.

Roelofsen, S., Martinoli, A., & Gillet, D. (2016). 3d collision avoidance algorithm for Unmanned Aerial Vehicles with limited field of view constraints. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 2555–2560). doi:`10.1109/CDC.2016.7798647`.

Russell, J. (1996). *Performance and Stability of Aircraft*. Butterworth-Heinemann.

Sartorius, S. (2013). Oswald efficiency estimation function. `https://uk.mathworks.com/matlabcentral/fileexchange/38800-oswald-efficiency-estimation-function?focused=3795877&tab=function`.

Shaw-Cortez, W. E., & Frew, E. (2015). Efficient Trajectory Development for Small Unmanned Aircraft Dynamic Soaring Applications. *Journal of Guidance, Control, and Dynamics*, *38*, 519–523. doi:`10.2514/1.G000543`.

Sobester, A., Castro, I. P., Czerski, H., & Zapponi, N. (2013). Notes on Meteorological Balloon Mission Planning. In *AIAA Balloon Systems (BAL) Conference* (pp. 1–15). American Institute of Aeronautics and Astronautics.

Soler, M., Zou, B., & Hansen, M. (2014). Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach. *Transportation Research Part C: Emerging Technologies*, *48*, 172–194. doi:`10.1016/j.trc.2014.08.009`.

Stengel, R. F. (2004). *Flight Dynamics*. Princeton University Press.

Sukumar, P. P., & Selig, M. S. (2013). Dynamic Soaring of Sailplanes over Open Fields. *Journal of Aircraft*, *50*, 1420–1430. doi:10.2514/1.C031940.

Yuan, C., Zhang, Y., & Liu, Z. (2015). A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, *45*, 783–792. doi:10.1139/cjfr-2014-0347.

Zapponi, N. (2013). Astra high altitude balloon flight planner. http://astra-planner.soton.ac.uk/. Accessed: 21/02/2017.

Zhao, Y. J. (2004). Optimal patterns of glider dynamic soaring. *Optimal Control Applications and Methods*, *25*, 67–89. doi:10.1002/oca.739.

Zhao, Y. J., & Qi, Y. C. (2004). Minimum fuel powered dynamic soaring of unmanned aerial vehicles utilizing wind gradients. *Optimal Control Applications and Methods*, *25*, 211–233. doi:10.1002/oca.744.