

Automation for Separation with Continuous Descent Operations: Dynamic Aircraft Arrival Routes

Raúl Sáez^{*} and Xavier Prats[†]

Technical University of Catalonia (UPC), Castelldefels, Spain, 08860

Tatiana Polishchuk[‡], Valentin Polishchuk[§] and Christiane Schmidt[¶]

Linköping University, Norrköping, Sweden, 60174

We present a mixed-integer programming approach to compute aircraft arrival routes in a terminal maneuvering area (TMA) that guarantee temporal separation of all aircraft arriving within a given time period, where the aircraft are flying according to their optimal continuous descent operation speed profile with idle thrust and no speed-brakes usage. The arrival routes form a merge tree that satisfies several operational constraints, e.g., all merge points are spatially separated. We detail how the continuous descent operation speed profiles for different route lengths are computed. Experimental results are presented for calculation of fully automated continuous descent operation-enabled arrival routes during one hour of operation on a busy day at Stockholm terminal maneuvering area. Our framework successfully generates an arrival tree, merging the traffic from all TMA entry points to the runway, for 30-minute intervals. For a complete one-hour period, we compute two trees (per 30 minutes) that differ as little as possible.

Nomenclature

\mathcal{A}_b	=	set of all aircraft arriving at entry point b
a	=	aircraft
b	=	TMA entry point
c_e	=	$ \Gamma_e $
D	=	aerodynamic drag
E	=	set of edges
e	=	edge
f	=	nominal fuel flow

^{*}PhD Candidate, Department of Physics - Aerospace Division, raul.saez.garcia@upc.edu

[†]Serra i Hünter fellow (Associate Professor), Department of Physics - Aerospace Division, xavier.prats@upc.edu

[‡]Assistant Professor, Communications and Transport Systems, ITN, tatiana.polishchuk@liu.se

[§]Associate Professor, Communications and Transport Systems, ITN, valentin.polishchuk@liu.se

[¶]Assistant Professor, Communications and Transport Systems, ITN, christiane.schmidt@liu.se

f_a	=	dynamic constraints
f_{idle}	=	idle fuel flow
f_e	=	flow on edge e
G	=	bidirectional graph; (V, E)
g	=	gravity acceleration
h	=	altitude
h_a	=	path constraints
h_f	=	altitude at the metering fix
J	=	cost function
ℓ_{ij}	=	length of edge (i, j)
L	=	distance threshold
L_a	=	Lagrange term of the cost function
\mathcal{L}	=	set of vertices in a path
m	=	mass
$N(i)$	=	neighbors of i in graph G
n	=	number of columns in the grid
\mathcal{P}	=	set of entry points
p	=	speed profile
p_a	=	time-independent parameters
Q	=	large number
q	=	number of rows in the grid
r	=	runway
$\mathcal{S}(a)$	=	set of aircraft speed profiles
s	=	distance to go
s_f	=	distance to go at the metering fix
T	=	time interval
T_{idle}	=	idle thrust
TMA	=	Terminal maneuvering area
t	=	time
U	=	limit on differing edges
u	=	control vector
u_a	=	time unit

$u_{a,p,k}$	=	time that aircraft a using speed profile p needs to cover segment number k
V	=	set of grid nodes
v	=	true airspeed
v_{CAS}	=	calibrated airspeed
$v_{CAS,min}$	=	minimum operative calibrated airspeed
v_{cruise}	=	cruise true airspeed
v_f	=	true airspeed at the metering fix
w	=	wind
w_b	=	number of aircraft entering the TMA via b
x	=	state vector
x_0	=	initial conditions
x_e	=	binary indicator for edge e participating in the arrival tree
$x_{e,b}$	=	binary indicator for edge e participating in the route from entry point b
x_f	=	state vector at the metering fix
$y_{a,j,t}$	=	binary indicator for aircraft a occupying vertex j at time t
$y_{a,j,p,n,t}$	=	binary indicator for aircraft a using speed profile p occupies n -th vertex (on a route from b) at time t
$z_{a,j,k,b,t}$	=	product of x_e and $y_{a,j,t}$
β	=	used (as β and $(1 - \beta)$) to obtain a convex combination of the two objectives: paths length and tree weight
Γ_e	=	the set of all outgoing edges from j that form an angle $\leq \theta$ with e
γ	=	flight path angle
γ_{min}	=	minimum descent gradient
θ	=	angle threshold
λ	=	upper bound on the number of vertices
λ_L	=	first vector of Lagrange multipliers
μ_L	=	second vector of Lagrange multipliers
σ	=	time separation between any pair of aircraft at all vertices
ϕ	=	Mayer term of the cost function
ψ_a	=	terminal constraints

I. Introduction

Over the last decades, air transportation experienced significant growth, and the International Air Transport Association (IATA) projected that the number of passengers will reach 7 billion/year by 2034 [1]. On the one hand, this

is desirable for a growing global economy, on the other hand, the higher air traffic volume comes with a drastically heightened environmental impact and a dramatically increased complexity for air traffic control officers (ATCOs). Particularly, the area surrounding one or more neighboring aerodromes, the terminal maneuvering area (TMA), is affected by congestion. Hence, it becomes critically important to design arrival and departure procedures in the TMA that mitigate the environmental impact and lessen the ATCO task load, while allowing for a high throughput.

Today, more sophisticated satellite-based navigation systems become available, which enable aircraft to follow routes with a high level of accuracy while following an optimized flight profile. The International Civil Aviation Organization (ICAO) [2] identified continuous descent operations (CDOs) as a promising solution to mitigate the environmental impact by executing optimal engine-idle descents. Eurocontrol [3] states that CDOs “allow aircraft to follow a flexible, optimum flight path that delivers major environmental and economic benefits—reduced fuel burn, gaseous emissions, noise and fuel costs—without any adverse effect on safety”, cf. also [4, 5].

Furthermore, ICAO has published guidance material [2] to support air navigation service providers (ANSPs) in the design of CDO approach procedures, suggesting vertical corridors within which all the descending aircraft trajectories must be contained. This approach provides separation from conflicting air traffic flows in the vicinity. However, as reported in [6], the suggested criteria have been established without explicitly considering the aircraft type, assuming international standard atmosphere (ISA) conditions and with coarse assumptions regarding the aircraft gross mass and performance data. This results in very restrictive flight corridors, which limit the fuel-saving capabilities of CDOs in real operations.

At most airports standard routes for departure and arrival, standard instrument departures (SIDs) and standard terminal arrival routes (STARs), are designed manually. ATCOs are responsible for guaranteeing safe separation of aircraft along the suggested static routes. Separation is provided at merge points (typically for arriving traffic) and also can be provided by crossing flows at different altitudes (typically for arriving traffic conflicting with departure traffic). These crossings are typically done at constant altitude and require level-offs, impeding CDOs and Continuous Climb Operations (CCOs). Separation of the aircraft arriving from different directions require special attention of the ATCO, especially at the merge points. It is a demanding task for a controller even with a moderate amount of traffic, and when air traffic volume increases, ATCOs often face situations with very high, or even unmanageable, task load. Hence, automation tools that help them to secure the necessary separation along the routes and, especially, at the merge points, are required.

In this work we design an optimization framework for computing aircraft arrival routes at a pre-tactical level that guarantee temporal separation of all aircraft arriving to TMA within a given time period, incorporating realistic CDO speed profiles. In particular, for each aircraft, based on its type, we compute optimized continuous descent speed profiles for different feasible route lengths. These are then input to a mixed integer programming (MIP) formulation. We output arrival routes that ensure safe separation from the entry point to the runway along the entire routes, with each aircraft

following an optimized continuous descent profile. It is expected that this both will reduce ATCO task load (by enabling automated separation tools) and environmental impact (by ensuring CDOs). The separation guarantee is given for aircraft arriving at the TMA at the expected time and all flights adhering to the computed arrival trajectories.

In the next section we review related work on this topic. In Section III we formally define our problem, given an overview on our approach and our assumptions in Section IV, and present our grid-based MIP formulation in Section V. In Section VI we detail how we use the aircraft arrival time and aircraft type to compute the optimized continuous descent speed profile for different route lengths. We present an experimental study of our concept in Section VII, before we conclude in Section VIII.

II. Background

Various authors considered automatic design of arrival routes. However, to the best of our knowledge, prior to our work the computation of optimal trees that take into account both the temporal separation and the turn constraints has not been considered. Prior approaches either constructed the routes one-by-one instead of simultaneously, did not consider all our constraints, or did not plan arrival routes at which aircraft are fully (temporally) separated at all times.

Turn-constrained route planning for just a single path (that is, no merging of paths, and construction of trees) was considered by Krozel et al. [7]. Similarly, Zhou et al. [8] constructed single, individual routes through weather-impacted TMA; and Visser and Wijnen [9] constructed single routes that minimize noise impact. The temporal component, however, was not considered by any of these works.

A few authors considered the scheduling problem along the arrival routes: Choi et al. [10] claimed that using scheduling algorithms that are more efficient than the traditional first-come-first-served (FCFS) approach can increase throughput in congested terminal airspace. They studied routes and scheduling together and considered the impact of different merge topologies on the aircraft scheduling. Moreover, Adacher et al. [11], studied sequencing models based on alternative graph formulation. However, the location of merge points or constraints like the limit on a turn angle were not considered by neither of these works.

Other research focusing on arrival scheduling includes the concept of 3D path arrival management, studied by Scharl et al. [12], which represents an efficient and predictable method to deliver aircraft from the top of descent (TOD) to a metering fix, where ATC automation support tools and modern FMS are needed. One of the core components of 3D PAM is the en route descent advisor (EDA), as studied by Coppenbarger et al. [13]. This tool provides an automatic time-based arrival metering, which helps ATC to deliver aircraft to a metering fix subject to a scheduled time constraint. A great emphasis is given to an accurate prediction of the trajectory. Coppenbarger et al. [14] investigated also the concept of tailored arrivals by using the EDA tool, studying the feasibility of CDOs in high density terminal airspace thanks to advanced air and ground automation over datalink. However, these works do not deal with the problem of generating the optimal arrival route for the aircraft and do not adapt to the current traffic demand; instead, they focus

more on trajectory prediction and conflict resolution.

A lot of attention has been given to spacing close to the runway. For example, re-categorization projects on both sides of the Atlantic (see, e.g., [15]) aim to replace the current standard of using only a few aircraft categories, where separation is determined by the category of leading and trailing aircraft, by a per-aircraft-type separation standard. Cheng et al. [16], provided more information concerning this re-categorization, known as wake turbulence re-categorization (known as RECAT in the United States and RECAT-EU in Europe). They described the overall three-phased approach of RECAT, and the different data sources and severity metrics used in the development of this concept. Sadosky et al. [17], investigated another approach to routing that considers flight separation, where paths within the TMA and control strategies are chosen so as to meet the constraints and reach the destination at the required time.

In our approach, CDOs have been considered for all arriving aircraft. However, it is difficult to fly this kind of operations in high-density TMAs, as shown by Robinson et al. [18], and in some cases powered CDOs (where additional thrust and the use of speed-brakes are allowed) or conventional descents with level-offs might be needed. In such cases, the benefits in fuel consumption would be lower.

A MIP-based approach for automation of aircraft separation along the optimized arrival routes was presented in [19]. The authors took the arrival times at entry points of all aircraft during a day as an input, and computed arrival routes that ensure safe separation from the entry point to the runway along the entire routes. They use the simplifying assumptions that all aircraft (independently of the type) flew with the same speed. We extend this work by incorporating actual speed profiles for different types of aircraft, using CDO to provide environmentally-friendly fuel-efficient arrivals.

III. Problem Description

In this paper, we present a mathematical programming framework for finding optimal dynamic aircraft arrival trees that guarantee separation for all aircraft that fly according to their optimal CDO speed profiles. The arrival trees are dynamic—compared to the standard, static STARS—, because our arrival route trees are recomputed for different time periods. Thus, they change during the day to reflect the actual incoming traffic demand.

Our input consists of the position of entry points to the TMA; the location and direction of the runway; and arrival time (within a given time interval T), arrival entry point, and aircraft type for all aircraft. We aim to output an arrival tree that merges traffic from the entries to the runway such that all aircraft are separated at all points of the arrival routes. The tree has the entries as leaves and the runway as the root (here we slightly abuse notation twice, as directed trees are usually called arborescences, and these are usually directed from leaves to root). Aircraft are assumed to fly according to their optimal continuous descent speed profile for the route length of the entry point-runway path along the tree. These speed profiles are computed separately, and provided as input to the MIP in the form of different speed profiles for different arrival route lengths.

Our arrival tree must fulfill a set of operational constraints:

- 1) *Temporal separation of all aircraft along the routes:* Each pair of aircraft that—while moving along the entry point-runway paths along the computed arrival tree—shares a subroute, is separated by a temporal distance of at least σ along this subroute. Hence, if all aircraft arrive to the TMA entry points at the planned point in time, all aircraft are guaranteed to be safely separated along the arrival routes.
- 2) *No more than two routes merge at a point:* ATCOs need to give heightened attention to merge points of routes. Thus, traffic complexity around the merges should be as low as possible [20]. We can formulate this requirement as a maximum indegree of 2 for all vertices of the tree. This allows an indegree of 1 or 2. Several vertices with indegree 1 simply constitute a path (we also ask for a maximum outdegree of 1). However, we need to merge the routes from the entry points to the runway, so we allow the minimum number of merging routes in any point: 2.
- 3) *Merge point separation:* Operational constraint 2) could be circumvented by placing merge points arbitrarily close to another—in practice, this would still result in a very small zone with high traffic complexity in terms of many merging routes. Thus, we add the requirement of a minimum distance between any two merge points: it needs to be larger than a given distance threshold L [20].
- 4) *No sharp turns:* Because of aircraft dynamics, an aircraft cannot turn at an arbitrarily acute angle: there is a limit on the angle at which the routes can turn (bank angle). Consequently, any turn from a route segment to the consecutive route segment must be larger than a given angle threshold θ . Again, if we would allow the use of arbitrarily short edges, a sequence of many short edges with pairwise angles larger than θ could still lead to a too sharp turn for the aircraft. So, it is the combination of the threshold θ with the lower bound on edge length, L , which enforces the limited turning angle [7]. We assume that the runway is the last segment of every route: this way, the turn onto the runway must also be larger than θ —the aircraft must align with the runway before the touchdown.
- 5) *Obstacle avoidance:* We can mark a set of regions (e.g. no-fly zones, noise-sensitive areas, etc.) over which routes may not pass.

IV. Overview of Our Approach

Our approach consists of two main steps:

- I. Computation of CDO speed profiles for different lengths of the entry-point-runway path for all aircraft in the considered time interval.
- II. Computation of the arrival trees, that allow for temporal separation of all considered aircraft flying along the computed arrival paths using CDO speed profile, for the considered time interval.

For II we use a discretization: we overlay the TMA with a square grid, and use both axis-aligned and diagonal edges as possible components of the arrival tree. This yields a discrete set of possible lengths of any entry-point-runway path (from the shortest grid-path from entry point to runway to the longest edge-disjoint grid path from entry point to

runway). In particular, we impose an upper bound on these paths. For these discrete paths length, we compute the CDO speed profiles (I.), see Section VI for details: based on the given route length, we optimize the vertical profile, where we assume neutral CDOs for all descents.

Our computation of arrival trees (II) uses a MIP formulation, see Section V for the MIP construction. We build a tree that has the runway as its root, and the entry points as leaves. By using further constraints we enforce the operational requirements 1)-4). The speed profiles from I. are input to our MIP. We enforce that aircraft following the computed entry-point-runway paths along the arrival tree are temporally separated, and that each aircraft uses the speed profile that corresponds to the route length of the entry-point-runway path of the computed tree. Some further constraints are integrated in the MIP.

Assumptions and Limitations In this work, we assume an idealistic scenario with deterministic aircraft arrival times to the TMA and no significant weather events disturbing normal operations, to get an idea on the upper bound on efficiency. Hence, the arrival times of aircraft at the TMA entry points are given, that is, for this first paper on this framework, we assume deterministic and not stochastic arrival times. Additionally, we do not take weather or changes in the wind direction into account. In particular, for this paper, the CDO speed profiles are computed for a route length, where the different segments of that route could run in different directions, the changes of wind impact that entails are not taken into account.

Furthermore, in this work, we focus on the pre-tactical planning phase, i.e., we assume that we compute the arrival routes several hours (or at least 30 minutes) before operation. Thus, the computation may take several minutes. The trees will then be available for the ATCOs and pilots during the operational phase. This also enables speed alterations for the en-route phase.

Moreover, we focus on the arrival routes, however, in prior work [21] we also enforced that arrival trees (without considering the specific aircraft using these routes) and the existing SIDs are vertical separated: we enforced that STARs are not allowed to intersect a given SID closer than d from the runway. Hence, while this is not part of this work, integrating these constraints in future work is straightforward. Also, while our grid has both axis-aligned and diagonal edges, as a simplification, we do not distinguish these edge types for computing the time it takes to traverse an edge.

Additionally, we assume a general temporal separation value, that is, between each pair of aircraft a temporal separation of σ is enforced. Variable separation criteria based on wake turbulence, that is, the temporal separation between two aircraft depending on the category of both the leading and the following aircraft, is left to future work. Note that it is straightforward to integrate this for the case of a single arrival path. As the arrival sequence is given (via the arrival times to the TMA) we could reformulate our constraint for temporal separation pairwise for consecutive aircraft with different separation values depending on the leading aircraft. This will result in a constraint for all times in the time horizon, for all possible speed profiles used by the two aircraft, and for all vertices along the grid path flown.

However, in case of a tree, as we have merge points, the separation depends on the arrival sequence to the merge point, which in turn depends on the lengths of the paths to the merge points.

Finally, we assume a single runway. Using parallel runways is supported by our framework immediately, as we snap the runway location to the closest grid point. If an airport has several, non-parallel runways, we may compute a merge tree per runway and impose constraints on how they may interact (e.g., on intersections).

Operationally, we assume that all aircraft follow the assigned route and speed profile. Neutral CDOs (i.e. descents with idle thrust and no speed-brakes usage) are considered for all arriving aircraft, and the same time at the TMA entry point is enforced for several path lengths. This involves changing the aircraft cost index when solving the optimal control problem, which is a parameter chosen by the airspace user that reflects the relative importance of the cost of time with respect to fuel costs [22]. More information regarding the computation and assumptions when generating CDOs can be found in Section VI.

V. Grid-based MIP Formulation

Our MIP formulation is based on the MIP we presented in [19]: to simplify, we assumed there that traversing a single edge takes u_a time units for all aircraft (independent of aircraft type and distance to runway). In this paper, we consider aircraft with different speeds, and in particular the speed profile given by a continuous descent operation for the specific aircraft type. Any other speed profile could be used.

In Subsection V.A, we start with a review of our prior MIP-formulation for optimal STAR merge trees with temporal separation [19] (which is based on our formulation for static routes from [21]). Then, we describe how we integrate different speed profiles in Subsection V.B. In particular, we will use the speed profiles that stem from CDOs for the different aircraft, see Section VI for details of their computation.

A. Review of our Grid-based MIP for Dynamic Arrival Routes with Guaranteed Temporal Separation

We use a discretization: we overlay the TMA with a square grid, and snap the location of both entry points and runway to the grid. Let \mathcal{P} denote the set of (snapped) entry points, and r the (snapped) runway. We use the threshold L as side length of grid pixels, hence, we fulfill operational Constraint 3) with any path in the grid. Every grid node is connected to its 8 neighbors (where $N(i)$ = denotes set of neighbors of i , including i), resulting in a bidirectional graph $G = (V, E)$. That is, for any two neighbors i and j , both edges (i, j) and (j, i) are included in E ; exceptions are the entry points (they do not have incoming edges) and r (it does not have outgoing edges). Let ℓ_{ij} denote the length of an edge $(i, j) \in E$. Let G be a grid of size $q \times n$.

In case we use operational Constraint 5), we delete the edges from the region that our routes may not pass from the edge set E (as we build our arrival tree from grid edges no route will then cross any obstacle).

Our underlying STAR MIP formulation [21] is based on the flow MIP formulation for Steiner trees [23, 24]. We use

binary decision variables x_e that indicate whether the edge e participates in the arrival tree. Moreover, we have flow variables: f_e gives the flow on edge $e = (i, j)$ (i.e., the flow from i to j). The constraints are given in Equations (1)-(4):

$$\sum_{k:(k,i) \in E} f_{ki} - \sum_{j:(i,j) \in E} f_{ij} = \begin{cases} |\mathcal{P}| & i = r \\ -1 & i \in \mathcal{P} \\ 0 & i \in V \setminus \{\mathcal{P} \cup r\} \end{cases} \quad (1)$$

$$x_e \geq \frac{f_e}{Q} \quad \forall e \in E \quad (2)$$

$$f_e \geq 0 \quad \forall e \in E \quad (3)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (4)$$

where Q is a large number (e.g., $Q = |\mathcal{P}|$).

Equation (1) ensures that a flow of $|\mathcal{P}|$ reaches the runway r , a flow of 1 leaves every entry point, and in all other vertices of the graph the flow is conserved. Equation (2) enforces edges with a positive flow to participate in the STAR. The flow variables are non-negative (Equation (3)), the edge variables are binary (Equation (4)).

In case we aim to minimize the sum of trajectory lengths flown by all arriving aircraft, where each path is counted as many times as it is used by aircraft (instead of minimizing the length of paths from entry points to the runway), we can easily integrate this by changing the right-hand side of Equation (1) (and increase Q accordingly). Let w_b be the number of aircraft entering the TMA via entry point $b \in \mathcal{P}$:

$$\sum_{k:(k,i) \in E} f_{ki} - \sum_{j:(i,j) \in E} f_{ij} = \begin{cases} \sum_{b \in \mathcal{P}} w_b & i = r \\ -w_i & i \in \mathcal{P} \\ 0 & i \in V \setminus \{\mathcal{P} \cup r\} \end{cases} \quad (5)$$

We consider two objective functions: paths length and tree weight. These are given in Equations (6) and (7), respectively:

$$\min \sum_{e \in E} \ell_e f_e \quad (6)$$

$$\min \sum_{e \in E} \ell_e x_e \quad (7)$$

For this paper, we will consider convex combinations of these objective functions, that is:

$$\min \quad \beta \sum_{e \in E} \ell_e x_e + (1 - \beta) \sum_{e \in E} \ell_e f_e \quad (8)$$

1. Degree constraints

Equations (1)-(4) are standard MIP-constraints for a MinCostFlow Steiner tree formulation, (5) allows us to weigh different paths in the resulting tree differently. However, we still need to add further equations to enforce the operational constraints presented in Section III. For operational Constraint 2) we require that the out-degree of every node is at most 1 and that the maximum indegree is 2:

$$\sum_{k:(k,i) \in E} x_{ki} \leq 2 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \quad (9)$$

$$\sum_{j:(i,j) \in E} x_{ij} \leq 1 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \quad (10)$$

$$\sum_{k:(k,r) \in E} x_{kr} = 1 \quad (11)$$

$$\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \forall i \in \mathcal{P} \quad (12)$$

Equation (11) ensures that the runway r has one ingoing edge, Equation (12) makes sure that each entry point has one outgoing edge, the maximum indegree of 2 for all other vertices is given by Equation (9), the maximum outdegree of 1 by Equation (10).

2. Turn angle constraints

The next operational constraint from Section III is Constraint 4): We require that for each edge $e = (i, j)$ used in the arrival tree, all outgoing edges at j must form an angle of at least θ with e . Let Γ_e be the set of all outgoing edges from j that form an angle $\leq \theta$ with e , i.e., $\Gamma_e = \{(j, k) : \angle ijk \leq \theta, (j, k) \in E\}$, and let $c_e = |\Gamma_e|$, $k \neq i$ and $e = (i, j) \notin \Gamma_e$. We add the following constraint:

$$c_e x_e + \sum_{f \in \Gamma_e} x_f \leq c_e \quad \forall e \in E \quad (13)$$

By Equation (13) we can either use edge x_e (which sets the left-hand side to c_e , the upper bound provided by the right-hand side, and prohibits the use of any other edge in Γ_e), or we may use any subset of the edges in Γ_e .

3. Auxilliary Constraints to Prevent Crossings

While route crossings at vertices are prevented by the degree constraints in Subsubsection V.A.1, we may still encounter routes crossing within a grid square, and we add auxiliary constraints to prevent this behavior. (Note that in the trees that minimize the length without adding the temporal separation constraints such a crossing would never occur, because uncrossing the routes would shorten them.)

We define V' as the set of all grid nodes without those which belong to the last column and last row of the grid, that is, $V' = V \setminus \{\text{last row}\} \setminus \{\text{last column}\}$.

$$\begin{aligned} x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} &\leq 1 \\ \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 &\notin \{\mathcal{P} \cup r\} \end{aligned} \quad (14)$$

Remember that entry points have no incoming edges. Hence, if one of the grid points in the considered grid square is an entry point, one of the four edges considered in Equation (14) does not exist. Thus, we add Equations (15)-(18).

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in \mathcal{P} \quad (15)$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \quad \forall i : i+1 \in \mathcal{P} \quad (16)$$

$$x_{i,i+1+n} + x_{i+n+1,i} + x_{i+n,i+1} \leq 1 \quad \forall i : i+n \in \mathcal{P} \quad (17)$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i : i+n+1 \in \mathcal{P} \quad (18)$$

4. Integration of Temporal Separation

So far, our MIP can compute an optimal (static) arrival tree (according to objective function (8)). In this subsection we review the temporal separation we described in [19]. The assumption there was that traversing any edge for any aircraft takes a unit time of u_a . We introduce new, binary variables $y_{a,j,t}$ that indicate whether aircraft a occupies vertex j at time t . Additionally, apart from the indicator x_e for an edge e participating in the routes, we introduce indicators $x_{e,b}$ for the edge e participating in the route from entry point b to the runway (for all entry points $b \in \mathcal{P}$). We set the

variables $x_{e,b}$ using Equations (19)-(22):

$$x_{e,b} \leq x_e \quad \forall b \in \mathcal{P}, \forall e \in E \quad (19)$$

$$\sum_{j:(b,j) \in E} x_{(b,j),b} = 1 \quad \forall b \in \mathcal{P} \quad (20)$$

$$\sum_{j:(j,r) \in E} x_{(j,r),b} = 1 \quad \forall b \in \mathcal{P} \quad (21)$$

$$\sum_{i:(i,j) \in E} x_{(i,j),b} - \sum_{k:(j,k) \in E} x_{(j,k),b} = 0 \quad \forall j \in V \setminus \{\mathcal{P} \cup r\},$$

$$\forall b \in \mathcal{P} \quad (22)$$

We still need to set the new variables $y_{a,j,t}$. We will set $y_{a,b,t_a^b} = 1$, because we know that aircraft a arrives at entry point b at time t_a^b , Equation (23). Additionally, we set various $y_{a,j,t}$ to zero: whenever we know that an aircraft cannot occupy the vertex at all or certain points in time. Equation (24) ensures that an aircraft that does not arrive at entry point b will occupy b at no point in time, Equation (25) yields that an aircraft arriving at b occupies this vertex at no time apart from t_a^b . Finally, Equation (26) ensures that any aircraft a can occupy a vertex j at any time t only if there exists an ingoing edge for j , that is, if j is located on a route. That is, if j is a grid vertex, but not a vertex on any route, no aircraft a will occupy it at any time.

Let \mathcal{A}_b be the set of all aircraft arriving at entry point $b \in \mathcal{P}$, and $\mathcal{A} = \cup_{b \in \mathcal{P}} \mathcal{A}_b$. Moreover let $T = \{0, \dots, \bar{T}\}$ be the considered time interval.

$$y_{a,b,t_a^b} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (23)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A} \setminus \mathcal{A}_b, \forall t \in T \quad (24)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall t \in T \setminus \{t_a^b\} \quad (25)$$

$$y_{a,j,t} \leq \sum_{\substack{k \in V: \\ (k,j) \in E}} x_{(k,j)} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}, \forall j \in V \setminus \mathcal{P}, \forall t \in T \quad (26)$$

We will then forward the information on the times at which a arrives at nodes along the route from b to the runway. That is, while Equations (23)-(26) set the variable $y_{a,j,t}$ for entry points, we need to set the variable for other vertices along the arrival tree. An aircraft a can reach vertex k at time $t + u$ ($y_{a,k,t+u} = 1$) only by traversing an edge from another vertex j to vertex k (which takes u_a time units by assumption). Hence, a must have occupied some vertex j at time t ($y_{a,j,t} = 1$), such that the edge (j, k) exists in the path from b . If no such edge (j, k) exists, or if a did not

occupy any such vertex at time t ($y_{a,j,t} = 0 \forall j$ for which edge (j, k) exists in the path from b), a cannot reach k at $t + u$, and we set $y_{a,k,t+u} = 0$.

We could achieve this by formulating

$$\sum_{j:(j,k) \in E} x_{(j,k),b} \times y_{a,j,t} = y_{a,k,t+u} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (27)$$

However, Equation (27) is not a linear constraint (we multiply two binary variables), which we cannot add to our MIP. Hence, we define a new binary variable $z_{a,j,k,b,t}$ as the product of $x_{(j,k),b}$ and $y_{a,j,t}$ using Equations (28)-(31):

$$z_{a,j,k,b,t} \leq x_{(j,k),b} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (28)$$

$$z_{a,j,k,b,t} \leq y_{a,j,t} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (29)$$

$$z_{a,j,k,b,t} \geq x_{(j,k),b} - (1 - y_{a,j,t}) \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (30)$$

$$z_{a,j,k,b,t} \geq 0 \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (31)$$

With this we can reformulate Equation (27) as:

$$\sum_{j:(j,k) \in E} z_{a,j,k,b,t} - y_{a,k,t+u} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (32)$$

Finally, we ensure that temporal separation between any pair of aircraft along the routes is kept: we require a minimum temporal separation of σ time units between all aircraft at all vertices:

$$\sum_{\tau=t}^{t+\sigma-1} \sum_{a \in \mathcal{A}} y_{a,j,\tau} \leq 1 \quad \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma + 1\} \quad (33)$$

B. Integration of Different Speed Profiles for Aircraft

In order to transfer our approach to a real-world scenario we distinguish different aircraft types and consider their optimal continuous descent speed profile for different route lengths.

For each aircraft a we are given a set of speed profiles, $\mathcal{S}(a)$, which contains speed profiles, p , of different lengths (that is, the speed profile is optimized for different route lengths), see Section VI for the computation of these speed profiles. The speed profile determines the time to cover the first, second... segment of the route. So, for each aircraft a , $\mathcal{S}(a)$ has speed profiles for routes of length (minimum number of edges in an entry-point-runway path) $\times L$ to (maximum number of edges in an entry-point-runway path) $\times L$. We ensure (equations (43)-(48)) that on an entry-point-runway path with X edges a uses the speed profile p for a route of length $X \times L$. Again, we assume that this time is constant independent of the grid-edge that constitutes this segment.

We need to substitute the binary variables $y_{a,j,t}$ (that indicate whether aircraft a occupies vertex j at time t) by binary variables $y_{a,j,p,n,t}$ that indicate whether aircraft a using speed profile p occupies the n -th vertex (on a route from b) j at time t . Why do we need to add two new indices, and do not simply add p ? For each speed profile the time it takes to cover an edge depends on which edge it is, that is, a profile p has a time for covering the first edge, another for covering the second edge etc.. Hence, we need to know what number of edge we cover (which we will deduct from the number of the vertex along the route). Let λ be an upper bound on the number of vertices in any path, $\mathcal{L} = \{1, \dots, \lambda\}$. We substitute Equations (23)-(26) by (and extend the set of constraints that set variables to zero):

$$\sum_{p \in \mathcal{S}(a)} y_{a,b,p,1,t_a^b} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (34)$$

$$y_{a,b,p,k,t_a^b} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall k \neq 1 \in \mathcal{L} \quad (35)$$

$$y_{a,b,p,1,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall t \in T \setminus \{t_a^b\} \quad (36)$$

$$y_{a,b',p,k,t} = 0 \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall k \in \mathcal{L}, \forall t \in T \quad (37)$$

$$y_{a',b,p,1,t_a^b} = 0 \quad \forall b \in \mathcal{P}, \forall a' \neq a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (38)$$

$$y_{a,j,p,k,t} \leq \sum_{\substack{i \in V: \\ (i,j) \in E}} x_{(i,j)} \quad \forall j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in \mathcal{S}(a), \forall k \in \mathcal{L}, \forall t \in T \quad (39)$$

$$y_{a,j,p,1,t} = 0 \quad j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in \mathcal{S}(a), \forall t \in T \quad (40)$$

$$\sum_{p \in \mathcal{S}(a)} y_{a,j,p,k,t} \leq 1 \quad j \in V, \forall a \in \mathcal{A}, \forall k \in \mathcal{L}, \forall t \in T \quad (41)$$

We add a constraint, which, for each entry point, computes the length of the path from entry point to runway in the arrival tree. This length is not the physical length of the prescribed route, instead, it corresponds to the number of grid

edges in the path:

$$\ell(b) = \sum_{(i,j) \in E} x_{(i,j),b} \quad (42)$$

Now, we need to make sure that for each aircraft arriving at entry point b , we pick the speed profile from $\mathcal{S}(a)$ that has length $\ell(b)$. That is, we want $y_{a,b,\ell(b),1,t_a^b} = 1$ and $y_{a,b,p,1,t_a^b} = 0 \forall p \neq \ell(b)$. However, as $\ell(b)$ is a variable, we cannot use it as an index, and we use two auxiliary binary variables $\psi_{b,a,p}$ and $\phi_{b,a,p}$ to achieve the desired result. We will make sure that:

$$\psi_{b,a,p} = \begin{cases} 0 & p = \ell(b) \\ 1 & p \neq \ell(b) \end{cases} \quad (43)$$

Then we set:

$$y_{a,b,p,1,t_a^b} = 1 - \psi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (44)$$

Together with Equation (34) this ensures that for at least one profile (the one of correct length) the variable y is set to one. We ensure Equation (43) by adding Equations (45)-(48) to the MIP:

$$-\psi_{b,a,p} \leq (\ell(b) - p + \frac{1}{2})/\lambda + \lambda \cdot \phi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (45)$$

$$\psi_{b,a,p} \leq -\ell(b) + p + \frac{1}{2} + \lambda \cdot \phi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (46)$$

$$\psi_{b,a,p} \leq (\ell(b) - p) + \frac{1}{2} + \lambda(1 - \phi_{b,a,p}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (47)$$

$$-\psi_{b,a,p} \leq (-\ell(b) + p + \frac{1}{2})/\lambda + \lambda(1 - \phi_{b,a,p}) \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) \quad (48)$$

Finally, we need to reformulate the constraints for forwarding the arrival time information along the arrival path from entry point to runway (Equations (28)-(30) and (27); (31) is not necessary for a binary variable), and the separation constraint (Equation (33)). We add indices to our auxiliary binary variable $z_{a,j,i,b,t}$ to $z_{a,j,i,b,p,k,t}$, which will now represent the product of $x_{(j,i),b}$ and $y_{a,j,p,k,t}$. Let $u_{a,p,k}$ be the time that aircraft a using speed profile p needs to cover segment number k on its route from entry point to runway.

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq x_{(j,i),b} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (49)$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq y_{a,j,p,k,t-u_{a,p,k}} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (50)$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \geq x_{(j,k),b} - (1 - y_{a,j,p,k,t-u_{a,p,k}}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (51)$$

The new version of Equation (32) is:

$$\sum_{j:(j,i) \in E} z_{a,j,i,b,k,t-u_{a,p,k}} - y_{a,i,p,k+1,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall k \in \mathcal{L}, \forall i \in V \setminus \mathcal{P}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T}\} \quad (52)$$

For the separation—we again require a minimum temporal separation of σ time units between all aircraft at all vertices—we add:

$$\sum_{\tau=t}^{t+\sigma-1} \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{S}(a)} \sum_{k \in \mathcal{L}} y_{a,j,p,k,\tau} \leq 1 \quad \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma + 1\} \quad (53)$$

C. Consistency between Trees of Different Time Periods

Additionally, we may aim for trees for consecutive time periods that do not differ a lot. Here, we measure consistency in terms of number of different edges used for the routes in the trees. Let x_{ij} and x_{ij}^{old} denote the edge indicator variable for the current and previous period, respectively. We define a variable, ax_{ij} , that determines $|x_{ij} - x_{ij}^{old}|$ in Equations (54)-(55), and then limit the number of differing edges in Equation (56) by parameter U :

$$ax_{ij} \leq x_{ij} - x_{ij}^{old} \quad \forall (j,i) \in E \quad (54)$$

$$ax_{ij} \leq x_{ij}^{old} - x_{ij} \quad \forall (j,i) \in E \quad (55)$$

$$\sum_{(i,j) \in E} ax_{ij} \leq U \quad (56)$$

D. The Complete MIP

To enhance readability, we list all constraints in the complete MIP. We use objective function (8), which is a convex combination of the objectives paths length and tree weight. Constraint (5) ensures that a flow of the number of aircraft entering the TMA via an entry point leaves the entry point towards the TMA, that the total number of aircraft arrives at the runway, and that for all other grid points the same number of aircraft arrives and leaves (in particular, it allows us to weigh the different entry-point-runway paths in the tree by the number of aircraft using them). Constraint (2) enforces edges with a positive flow to participate in the arrival tree. Constraints (11) and (12) ensure that the runway has one ingoing and the entry points one outgoing edge each, constraints (9) and (10) ensure that all other vertices have a

maximum indegree of 2 and a maximum outdegree of 1. Constraint (13) ensures a minimum turn angle. Constraints (14) and (15)-(18) prevent route crossings. Constraints (19)-(22) set variables $x_{e,b}$ for an edge e participating in the route from entry point b to the runway. Constraints (34)-(41) set initial values of the binary variables $y_{a,j,p,n,t}$, e.g., for each entry point and each aircraft arriving at that entry point, exactly one speed profile is used with the entry point being the first vertex on the path ((34)). Constraint (42) computes the number of grid edges in the path from entry point to runway in the arrival tree. Constraints (44), (45)-(48) ensure that for each aircraft the speed profile with the right length is chosen. Constraints (49)-(52) forward the arrival time information along the arrival path from entry point to runway. Constraint (53) ensures minimum time separation. Constraints (54)-(56) ensure consistency between trees of different time periods. Additionally, we of course set the range of binary and non-negative variables (like presented in Equations (3), (4)).

VI. Computation of CDO Speed Profiles

In this paper, we compute several descent trajectories for each aircraft arriving at the studied airport and for each possible route length within the TMA. We assumed neutral CDOs for all the descents, with no additional thrust (only idle thrust) nor speed-brakes usage allowed.

Given a known route (route length), and consequently a fixed distance to go, the optimization of the vertical profile (altitude and speed) can be formulated as an optimal control problem, which aims at computing the control time history of a system, here the aircraft, such that a cost function is minimized while satisfying some dynamic and operational constraints [25].

Further in Subsection VI.A, we present the generic optimal control problem and in Subsection VI.B we focus on the trajectory optimization itself—applied to aircraft descents.

A. Trajectory Optimization: Generic Optimal Control Problem

A generic optimal control problem is defined as [26]:

$$\begin{aligned}
\min_{\mathbf{u}(t)} \quad & J_{ocp} := \phi(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} L_a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_a) dt \\
\text{s.t.} \quad & \mathbf{x}(t_0) = \mathbf{x}_0 \\
& \dot{\mathbf{x}}(t) = \mathbf{f}_a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_a) \\
& \mathbf{h}_a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}_a) \leq 0 \\
& \psi_a(\mathbf{x}(t_f)) = 0
\end{aligned} \tag{57}$$

Where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the state vector, with fixed initial conditions \mathbf{x}_0 ; $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control vector; and the vector $\mathbf{p}_a \in \mathbb{R}^{n_p}$ includes all the time-independent parameters of the model; $L_a : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}$ and $\phi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ are

the Lagrange and Mayer terms of the cost function, respectively. The dynamics of the state vector are expressed by a set of non-linear equations $\mathbf{f}_a : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$; $\mathbf{h}_a : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_h}$ and $\boldsymbol{\psi}_a : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_{\psi_a}}$ represent applicable path and terminal constraints, respectively. Note that if the time interval is not fixed, t_f becomes a new decision variable.

The Hamiltonian of the optimal control problem (57) is:

$$H = L_a + \boldsymbol{\lambda}_L^T \mathbf{f}_a + \boldsymbol{\mu}_L^T \mathbf{h}_a \quad (58)$$

where $\boldsymbol{\lambda}_L$ and $\boldsymbol{\mu}_L$ are vectors of Lagrange multipliers. The set of necessary conditions for J to be stationary optimum is [27]:

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= - \left(\frac{\partial H}{\partial \mathbf{x}} \right) \\ 0 &= \left(\frac{\partial H}{\partial \mathbf{u}} \right) \\ \boldsymbol{\lambda}(t_f) &= \left(\frac{\partial \phi}{\partial \mathbf{x}} + \mathbf{v}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}} \right)_{t=t_f}^T \\ 0 &= \left[\left(\frac{\partial \phi}{\partial \mathbf{x}} + \mathbf{v}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}} \right) \mathbf{f} + L_a \right]_{t=t_f} \\ \boldsymbol{\mu} &= \begin{cases} \geq 0 & \text{if } \mathbf{h} = 0 \\ = 0 & \text{if } \mathbf{h} < 0 \end{cases} \end{aligned} \quad (59)$$

B. Optimal Control Problem for Aircraft Descents

In this paper, the state vector has been chosen as $\mathbf{x} = [v, h, s]$, where v is the true airspeed (TAS), h the altitude of the aircraft, and s the distance to go. In order to obtain environmentally friendly trajectories, idle thrust is assumed and speed-brakes use is not allowed throughout the descent. In such conditions, the flight path angle is the only control variable in this problem ($\mathbf{u} = [\gamma]$), which is used to manage the energy of the aircraft and achieve different times of arrival at the metering fix with minimum fuel consumption and noise nuisance.

The dynamics of \mathbf{x} are expressed by the following set of ordinary differential equations (ODE), considering a point-mass representation of the aircraft reduced to a "gamma-command" model, where vertical equilibrium is assumed (lift balances weight). In addition, the cross and vertical components of the wind are neglected, and the aerodynamic flight path angle is assumed to be small (i.e., $\sin \gamma \simeq \gamma$ and $\cos \gamma \simeq 1$):

$$\mathbf{f}_a = \begin{bmatrix} \dot{v} \\ \dot{h} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \frac{T_{idle}-D}{m} - g\gamma \\ v\gamma \\ v + w \end{bmatrix} \quad (60)$$

where $T_{idle} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the idle thrust; $D : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}$ is the aerodynamic drag; g is the gravity acceleration; w is the wind and m the mass, which is assumed to be constant because the fuel consumption during an idle descent is a small fraction of the total m [28].

In this paper, the trajectory is divided in two phases: the latter part of the cruise phase prior the top of descent (TOD), and the idle descent down to the metering fix. Assuming that the original cruise speed will not be modified after the optimization process, the two-phases optimal control problem can be converted into a single-phase optimal control problem as follows:

$$J_a = \int_{t_0}^{t_f} - \left(\frac{f + CI}{v_{cr}} \right) (v + w) + f_{idle} + CI \, dt \quad (61)$$

where v_{cr} is the cruise speed (which is assumed to be constant), $f : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}$ is the cruise fuel flow (which is assumed to be constant, since the cruise altitude and speed are also constant) and $f_{idle} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the idle fuel flow in the descent (which depends on the speed and altitude). Finally, CI is the cost index, which is a parameter chosen by the airspace user that reflects the relative importance of the cost of time with respect to fuel costs [22].

In addition to the dynamic constraints \mathbf{f}_a , the following set of path constraints are enforced to ensure that the aircraft airspeed remains within operational limits, and that the maximum and minimum descent gradients are not exceeded:

$$\mathbf{h}_a = \begin{bmatrix} v_{CAS,min} - v_{CAS} \\ v_{CAS} - VMO \\ M - MMO \\ \gamma \\ \gamma_{min} - \gamma \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (62)$$

where $v_{CAS} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the calibrated airspeed (CAS) and $M : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the Mach number, both functions of the state vector; $v_{CAS,min}$ and VMO are the minimum and maximum operative CAS respectively; MMO is maximum operative Mach; and γ_{min} is the minimum descent gradient.

Different alternatives can be used to model the aircraft performance functions T_{idle} , D , f and f_{idle} and their respective parameters. Here, we adopt EUROCONTROL's base of aircraft data (BADA) v4 model [29].

Finally, terminal constraints fix the final states vector:

$$\boldsymbol{\psi}_a = \begin{bmatrix} v - v_f \\ h - h_f \\ s - s_f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (63)$$

where $\boldsymbol{x}_f = [v_f, h_f, s_f]$ is the state vector at the metering fix.

In the formulation presented herein, there is only one control variable, which appears linearly in the equations describing the dynamics of the system as well in the cost function to be minimized. Consequently, the Hamiltonian of the system (58) is also linear with respect to the control, leading to a singular optimal control problem which can be solved semi-analytically from the implicit formulation of optimal singular arcs [30].

Since the initial and final states of the trajectory are fixed, the optimal trajectory will be of a ‘‘bang-singular-bang’’ type. These solutions are composed by three arcs: one initial bang arc with the control variable at its maximum or minimum value to go from \boldsymbol{x}_0 to the singular arc; a singular arc where the optimal control is given as a function of the states vector; and a final bang arc to go from the singular arc to the final state.

The analytical expression of the optimal control in the singular arc for the above model and the steps to generate an optimal trajectory semi-analytically can be found in [30].

1. Speed profile

As an example, Fig. 1 shows the optimal speed profile for an Airbus A320, in international standard atmospheric (ISA) conditions and no wind. The cost index used to compute the optimal trajectory was 42 kg min^{-1} . We can observe how the optimal speed profile lies in between the boundaries delimited by $v_{CAS,min}$ and MMO/VMO. Also note that since these maximum and minimum speeds are given in terms of CAS and Mach, the corresponding TAS changes with altitude. The earliest and latest trajectories would correspond to the $v_{CAS,min}$ and MMO/VMO speed profiles, respectively.

2. Input data

We obtained the flight traffic data needed to generate the trajectories from EUROCONTROL’s data demand repository (DDR2) [31], which contains information about the trajectories flown in Europe every day. To generate the optimum trajectories, we use five parameters: 1) aircraft model; 2) cruise altitude; 3) distance to go; 4) speed, ; and 5) cost index.

Moreover, we needed the aircraft performance model, which was obtained from EUROCONTROL’s BADA V4 (see Subsection VI.B). In the case the aircraft model did not correspond to any of the BADA models, we used a comparable aircraft in terms of performance and dimensions.

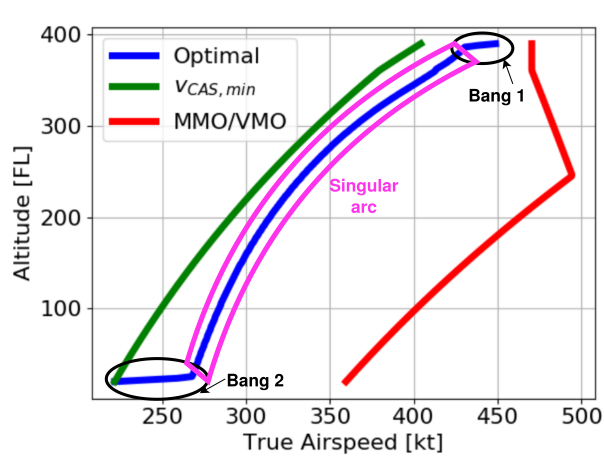


Fig. 1 Example of an Airbus A320 optimal speed profile

For any given flight, the number of trajectories generated corresponds to the number of possible routes the aircraft can fly. In this case, path lengths within the TMA from 36 NM (corresponding to the minimum path length within the grid) to 90 NM were considered, with each path split into several segments of constant length equal to 6 NM. For instance, a 36 NM path inside the TMA would be split into 6 segments (i.e. [0-6, 6-12, ..., 30-36]). In total, we computed 10 trajectories per flight (i.e. 10 possible path lengths ranging from 36 NM to 90NM). In addition, we generated all those trajectories to ensure the same time at the TMA entry point, which meant that different cost index values were used for each trajectory. We chose the distance in this experiment according to the grid size (as described in Section VII). The lower bound stems directly from the grid; additionally, we impose a large enough upper bound to allow for feasible solutions. Finally, we also obtained the entry point for each flight from DDR2.

VII. Experimental Study: Arlanda Airport

In this section, we apply our framework to a real-world instance aiming at showing its feasibility by considering arrival routes in Stockholm TMA. We use the aircraft arrival times at TMA entry points during one hour of airport operation as input (taken from EUROCONTROL’s DDR2) and compute the dynamic arrival routes with guaranteed temporal separation for this time interval. We have chosen a data sample for October 3, 2017, one of the busiest days of that year with in total 432 aircraft arrivals. We use $\beta = 0.1$, that is, we prioritize weighted shortest path in the objective function. We set a temporal separation of at least 2 minutes ($\sigma = 2$ min).

We solve our MIP using Gurobi optimization software installed on a very powerful Tetralith server [32], utilizing the Intel HNS2600BPB computer nodes with 32 CPU cores, 384 GiB, provided by the Swedish National Infrastructure for Computing (SNIC).

We use an 11x15 grid, which automatically guarantees merge point separation of about 6 NM (parameter L). In current operations, a separation of 5 NM is used, that is, we show results in the operational separation range (using a finer

14x19 grid, which would result in a 5 NM separation, makes the problem computationally too expensive). We simulated realistic full CDO speed profiles for all aircraft in the experiment assuming no wind and taking into account aircraft model, current altitude and true airspeed at the top of descent, as well as distance to go (which defines which exact speed profile the aircraft is taking). An example of a set of speed profiles for an aircraft A320 is shown in figure 2. Every curve corresponds to the flight speed profile along the path followed inside the TMA, whose length varies depending on the tree configuration and could take values from 36 NM up to 90 NM.

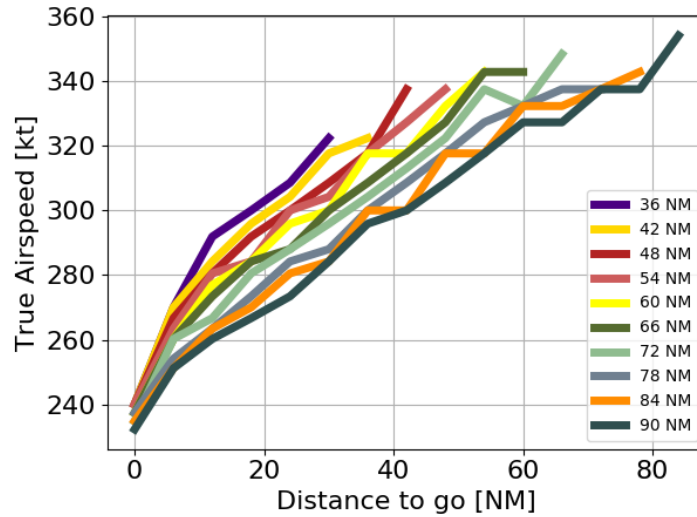


Fig. 2 Example of an Airbus A320 speed profiles for several path lengths within Stockholm TMA

In figure 3(a) we show arrival trees computed for two half hours within a one-hour period of operation in Stockholm Arlanda in October, 3rd, 2017. An arrival tree for 10 aircraft entering Stockholm TMA between 3:00 and 3:30 pm is shown in solid black, and an arrival tree generated for 7 aircraft arriving to the TMA between 3:30 and 4:00 pm is shown in dashed blue. For the second tree we add Equations (54)-(56) and set $U = 23$. This yields consistency between the trees, which we believe is an advantage for actual operation. Entry points and merge points are reference points for the time schedule presented in Table 1. The execution time for computation of such arrival trees was between 3 and 7 hours and memory usage around 55%. Figure 3 (b) illustrates the real trajectories flown by aircraft at Arlanda during the same period (3-4 pm that day), obtained from the Historical Database of the OpenSky Network [33]. As expected, for each entry point aircraft do not follow fixed paths, probably due to path stretching instructions issued by the ATC in order to avoid possible conflicts. With our method, however, there will be no need for such instructions, as the proposed trees ensure enough separation between all the incoming aircraft. This would mean a reduction in ATC task load and the same safety levels. Furthermore, our solution ensures that aircraft fly neutral CDOs, which would represent an increase in flight efficiency in TMA.

In Figure 4, we compare the fuel consumption for aircraft following our arrival tree and performing CDOs with the

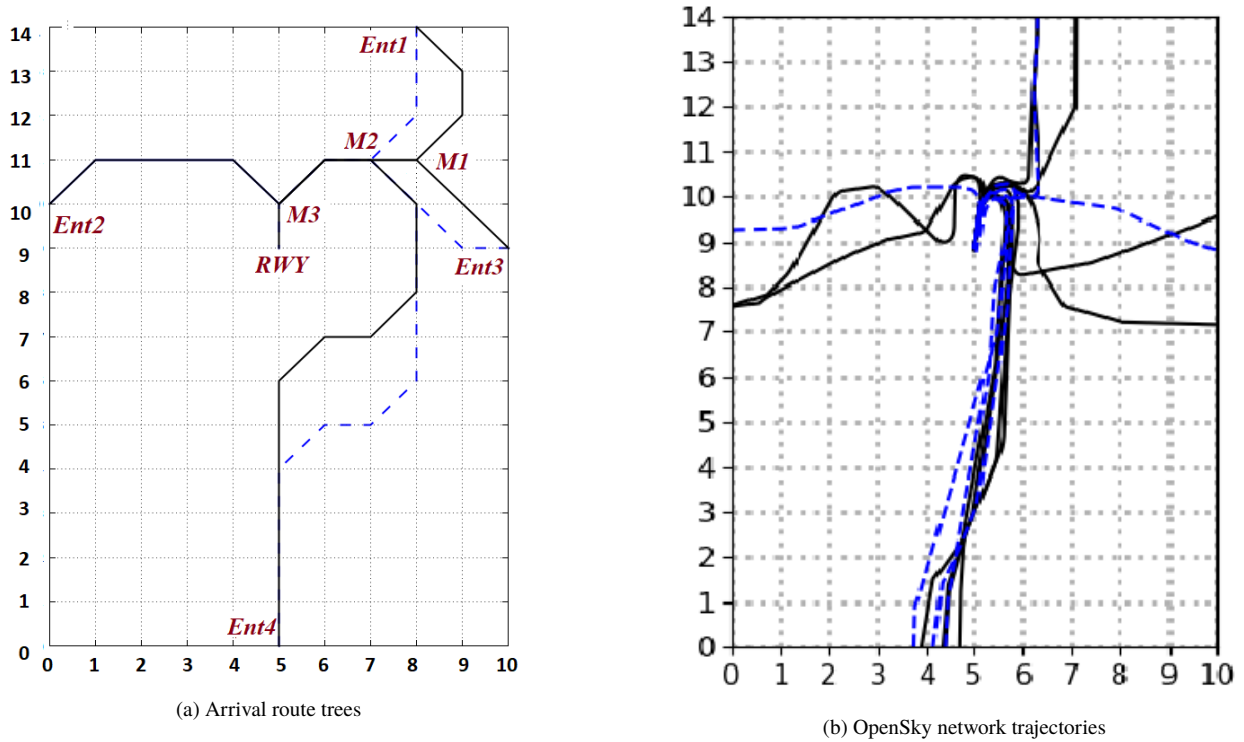


Fig. 3 Arrival route trees and real aircraft trajectories (from OpenSky Network) during one hour of Stockholm Arlanda operation: *black solid line 3:00-3:30 pm, blue dashed line 3:30-4:00 pm* ($n_1 = 10, n_2 = 14$)

same aircraft but flying the trajectories obtained from the OpenSky network. We compute the fuel consumption inside the TMA by using the equations from BADA, and following the same process as the one shown in [34]. We can observe that the fuel consumption increases drastically for the aircraft entering the TMA from the south, as they have to fly a longer distance until the runway. On the other hand, the fuel consumption for aircraft entering from either the east, west or north has similar values, as the major factor affecting the consumption in this case is the aircraft model (which tends to be an A320 or a similar model in terms of dimensions and performances). By flying our arrival tree, aircraft consume around 5% to 25% less fuel. The distance is similar to the one flown by the actual trajectories, but when following the arrival tree aircraft fly neutral CDOs, while in the actual case this is not always true.

Table 1 presents at which time the aircraft reach the marked points (merge points M1-M3 and entry points Ent1-Ent4). More than one aircraft arriving at any point within an interval of 2 minutes would be a conflict, which our MIP excludes. Time separation at M3 provides the separation at the runway. Aircraft *a16* and *a17* did not arrive to any of the merge points by the end of the simulated hour and will reach these in the next half hour.

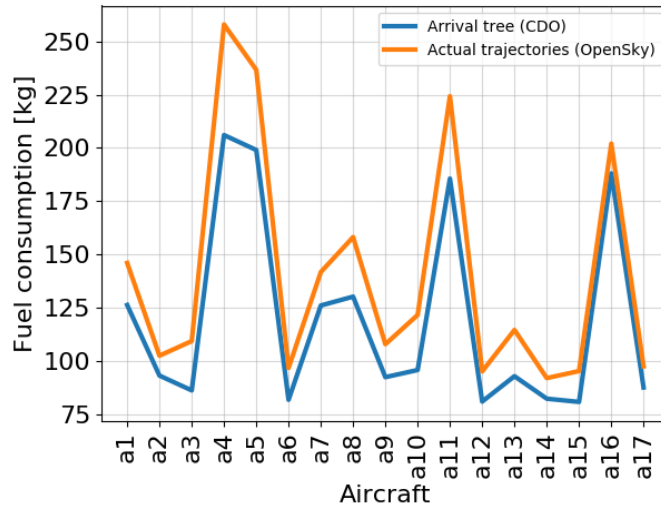


Fig. 4 Fuel consumption for aircraft flying the arrival tree (CDOs) vs. actual trajectories (obtained from OpenSky network)

Table 1 Time schedule example for 17 aircraft arriving between 3 pm and 4 pm at Stockholm TMA on October 3, 2017.

Aircraft	Arrivals Entry point	Simulated time [min]			
		Entry	M1	M2	M3
a1	Ent1 (North)	3	9	11	15
a2	Ent2 (West)	8	-	-	13
a3	Ent3 (East)	13	15	16	18
a4	Ent4 (South)	4	-	18	21
a5	Ent4	18	-	30	32
a6	Ent2	17	-	-	25
a7	Ent1	17	20	21	23
a8	Ent1	21	24	25	27
a9	Ent2	19	-	-	29
a10	Ent3	28	30	32	34
a11	Ent4	34	45	46	48
a12	Ent3	41	43	44	46
a13	Ent2	32	-	-	37
a14	Ent1	39	-	42	44
a15	Ent1	49	-	55	59
a16	Ent4	53	-	-	-
a17	Ent2	57	-	-	-

VIII. Conclusions

We presented a MIP-based framework for the computation of arrival route trees that fulfill several operational constraints and guarantee temporal separation of all aircraft arriving to TMA within the considered time interval. The aircraft fly according to their optimal neutral CDO speed profiles (i.e. descents with idle thrust and no speed-brakes

usage) for the entry point–runway path length in the tree. Our approach has the potential to both reduce the environmental impact (CDOs) and the task load of ATCOs in planned operations. We proved the feasibility of our framework by presenting experiments for calculation of the arrival routes for one hour of operation on a busy day at Stockholm TMA.

Our current implementation is quite sensitive to the number of aircraft. With our setup we could not solve the problem for all 22 scheduled aircraft for the given hour, the current grid only allowed a feasible solution for 17 arrivals. This discrepancy in aircraft number is not due to computational limits of our framework (or server), but is rooted in the input data: If aircraft have less than two minutes separation in the entry point, this would result in an infeasible problem, hence, we filter out aircraft to obtain a feasible input. Moreover, if an aircraft with higher speed follows an aircraft with lower speed—depending on the distance—the optimal speed profile might lead to the faster aircraft overtaking the slower aircraft along the route. This is not feasible, and for this paper, we decided to filter out the problematic aircraft. In future work, we aim to also handle these cases. One approach can be to include en-route traffic and to adapt the speed profile on the en-route segment of the flight, such that the faster aircraft arrives at the entry point with enough temporal distance to the leading aircraft for both aircraft to remain separated along the routes when both apply the CDOs. Alternatively, we can impose a non-optimal speed profile on aircraft in these conflict cases. According to our experiments this would be necessary for about 15-20% of the aircraft, which we deem feasible. In busier airports, more non-optimal speed profiles would be needed in order to schedule all aircraft. Furthermore, a sufficient temporal separation in the entry point, assumed to be achieved in the en-route phase, might be necessary too.

The proposed approach is also sensitive to the length of the time period. The program did not yield feasible solutions for time windows of more than 30 minutes. Yet, computing trees for longer periods may not be needed. Each tree is optimized with respect to the current traffic situation. The routes often get stretched for the purpose of conflict resolution. But when the aircraft in conflict pass the merge point, following aircraft continue flying along sub-optimal routes. Adjusting the tree configuration every 20-30 minutes, which is about the time aircraft spend in TMA on average, will keep them optimized for the actual traffic situation. Keeping parameter U (i.e., the limit on differing edges) at minimum will provide consistency between the trees, preventing extra task load because of switching. Moreover, our routes guarantee separation if the aircraft arrive to the TMA at the assumed times. A future research direction is robust optimization for possible arrival intervals. Other possible improvements to our solution would be to add some of the air traffic scheduling constraints presented in [35], which deals with the precision air traffic operations concept. In the future, we plan to integrate dynamic re-planning for weather avoidance into our framework. We also plan to integrate wind, which would involve computing a higher number of trajectories (where wind is considered when generating the CDO) where the n -th segments is covered in one of the eight edge directions. Finally, the concepts discussed in [12–14] could be integrated in our tool to improve the precision of our framework, to have a better knowledge of the aircraft’s position, and to ensure the safety of the operation by taking into account possible uncertainties.

References

- [1] "IATA Air Passenger Forecast Shows Dip in Long-Term Demand," <http://www.iata.org/pressroom/pr/Pages/2015-11-26-01.aspx>, November 2015. Accessed on June 2, 2016.
- [2] ICAO, "Continuous Descent Operations (CDO) Manual-Doc 9931/AN/476," Tech. rep., ICAO, 2010. Montreal, Quebec, Canada.
- [3] Eurocontrol, "Continuous Climb and Descent Operations," <https://www.eurocontrol.int/articles/continuous-climb-and-descent-operations>, 2019. Accessed December 15, 2017.
- [4] Warren, A., and Tong, K., "Development of continuous descent approach concepts for noise abatement," IEEE/AIAA 21st Digital Avionics Systems Conference (DASC), Irvine, CA, 2002.
- [5] Zou, K. F., and Clarke, J. P., "Adaptive Real-Time Optimization Algorithm for Noise Abatement Approach Procedures," AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum, Aviation Technology, Integration, and Operations (ATIO) Conferences, American Institute of Aeronautics and Astronautics, Denver, Colorado, 2003. doi:10.2514/6.2003-6771.
- [6] Fricke, H., Seiß, C., and Herrmann, R., "Fuel and Energy Benchmark Analysis of Continuous Descent Operations," Air Traffic Control Quarterly, Vol. 23, No. 1, 2015, pp. 83–108. doi:10.2514/atcq.23.1.83.
- [7] Krozel, J., Lee, C., and Mitchell, J. S., "Turn-constrained route planning for avoiding hazardous weather," Air Traffic Control Quarterly, Vol. 14, No. 2, 2006.
- [8] Zhou, J., Cafieri, S., Delahaye, D., and Sbihi, M., "Optimization of Arrival and Departure Routes in Terminal Maneuvering Area," 6th International Conference for Research in Air Transportation (ICRAT), Istanbul, Turkey, 2014.
- [9] Visser, H. G., and Wijnen, R. A., "Optimization of Noise Abatement Departure Trajectories," Journal of Aircraft, Vol. 38, No. 4, 2001, pp. 620–627.
- [10] Choi, S., Robinson, J. E., Mulfinger, D. G., and Capozzi, B. J., "Design of an optimal route structure using heuristics-based stochastic schedulers," IEEE/AIAA 29th Digital Avionics Systems Conference (DASC), 2010.
- [11] Adacher, L., Pacciarelli, D., and Pranzo, M., "A Graph-Theoretical Model for the Aircraft Sequencing Problem," , 2003.
- [12] Scharl, J., Berge, M. E., Coats, M. L., Haraldsdottir, A., and Schoemig, E. G., "Modeling and Analysis of the 3D Path Arrival Management Concept," AIAA Modeling and Simulation Technologies Conference and Exhibit, Hilton Head, South Carolina, 2007. doi:10.2514/6.2007-6878.
- [13] Coppenbarger, R. A., Lanier, R., Sweet, D., and Dorsky, S., "Design and Development of the En Route Descent Advisor (EDA) for Conflict-Free Arrival Metering," AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, 2004. doi:10.2514/6.2004-4875.
- [14] Coppenbrager, R. A., Mead, R. W., and Sweet, D. N., "Field Evaluation of the Tailored Arrivals Concept for Datalink-Enabled Continuous Descent Approach," Journal of Aircraft, Vol. 46, 2009, pp. 1200–1209. doi:10.2514/1.39795.

- [15] NATS, and Eurocontrol, “Operational Service and Environment Definition (OSED) for Time Based Separation for Arrivals (TBS),” , 2013. Project Number 06.08.01.
- [16] Cheng, J., Tittsworth, J., Gallo, W., and Awwad, A., “The Development of Wake Turbulence Re-Categorization in the United States,” 8th AIAA Atmospheric and Space Environments Conference, Washington, D.C., 2016. doi:10.2514/6.2016-3434.
- [17] Sadovsky, A. V., Davis, D., and Isaacson, D. R., “Separation-compliant, optimal routing and control of scheduled arrivals in a terminal airspace,” Transportation Research Part C: Emerging Technologies, Vol. 37, No. 0, 2013, pp. 157 – 176. doi:http://dx.doi.org/10.1016/j.trc.2013.09.017.
- [18] J. E. Robinson III and M. Kamgarpour, “Benefits of Continuous Descent Operations in High-Density Terminal Airspace Under Scheduling Constraints,” 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth, TX, 2010.
- [19] Dahlberg, J., Granberg, T. A., Polishchuk, T., Schmidt, C., and Sedov, L., “Capacity-Driven Automatic Design of Dynamic Aircraft Arrival Routes,” IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), 2018. doi:10.1109/DASC.2018.8569646.
- [20] Prete, J., Krozel, J., Mitchell, J., Kim, J., and Zou, J., “Flexible, Performance-Based Route Planning for Super-Dense Operations,” AIAA Guidance, Navigation and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, 2008.
- [21] Andersson, T., Polishchuk, T., Polishchuk, V., and Schmidt, C., “Automatic Design of Aircraft Arrival Routes with Limited Turning Angle,” Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS), 2016.
- [22] Airbus, “Getting to grips with the cost index - Issue II,” Tech. Rep. 2, 1998.
- [23] Wong, R. T., “A dual ascent approach for Steiner tree problems on a directed graph,” Mathematical Programming, Vol. 28, No. 3, 1984, pp. 271–287.
- [24] Goemans, M. X., and Myung, Y., “A catalog of Steiner tree formulations,” Networks, Vol. 23, No. 1, 1993, pp. 19–28.
- [25] Sáez, R., Dalmau, R., and Prats, X., “Optimal assignment of 4D close-loop instructions to enable CDOs in dense TMAs,” IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, England, UK, 2018.
- [26] Soler, M., A. O., , and Staffètti, E., “Multiphase Optimal Control Framework for Commercial Aircraft Four-Dimensional Flight-Planning Problems,” Journal of Aircraft, 2014. doi:10.2514/1.C032697.
- [27] Bryson, A. E., and Ho, Y.-C., Applied optimal control : optimization, estimation, and control, Taylor and Francis Group, New York, 1975.
- [28] Clarke, J. P., Ho, N. T., Ren, L., Brown, J., Elmer, K., Zou, K. F., Hunting, C., McGregor, D., Shivashankara, B., Tong, K., Warren, A. W., and Wat, J., “Continuous descent approach: Design and flight test for Louisville international airport,” Journal of Aircraft, Vol. 41, No. 5, 2004, pp. 1054–1066. doi:10.2514/1.5572.

- [29] Poles, D., Nuic, A., and Mouillet, V., “Advanced aircraft performance modelling for ATM: Analysis of BADA model capabilities,” IEEE/AIAA 29th Digital Avionics Systems Conference (DASC), Brétigny-sur-Orge (France), 2010. doi:10.5772/2706.
- [30] Park, S. G., Dutta, P., and Menon, P. K., “Optimal Trajectory Option Sets for In-Flight Climb-Descend Trajectory Negotiations,” 17th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Denver, CO, 2017.
- [31] Eurocontrol, DDR2 Reference Manual for General Users 2.9.4, 2017.
- [32] “Tetralith server, NSC, Linköping University,” <https://www.nsc.liu.se/systems/tetralith/>, 2019. Accessed on 28/01/2019.
- [33] OpenSky, “OpenSky Network,” <https://opensky-network.org>, July 2019. Accessed on December 2, 2018.
- [34] Lemetti, A., Polishchuk, T., Sáez, R., and Prats, X., “Analysis of weather impact on flight efficiency for Stockholm Arlanda Airport arrivals,” 6th ENRI International Workshop on ATM/CNS, 2019.
- [35] Isaacson, D. R., Sadosky, A. V., and Davis, D., “Scheduling for Precision Air Traffic Operations: Problem Definition and Review of Prior Research,” Journal of Aerospace Information Systems, Vol. 11, 2014, pp. 234–257.