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

Raúl Sáez* and Xavier Prats†
Technical University of Catalonia, 08860 Castelldefels, Spain
and
Tatiana Polishchuk,‡ Valentin Polishchuk,§ and Christiane Schmidt¶
Linköping University, 60174 Norrköping, Sweden

https://doi.org/10.2514/1.D0176

This paper presents a mixed-integer programming approach to compute aircraft arrival routes in a terminal maneuvering area (TMA) with guaranteed temporal separation of all aircraft arriving within a given time period. Furthermore, the aircraft are assumed to be flying according to their optimal continous descent operation speed profile with idle thrust and no speed brakes usage. The arrival routes form a merge tree that satisfies several operational constraints; for example, all merge points are spatially separated. How the continuous descent operation speed profiles for different route lengths are computed is detailed. Experimental results are presented for calculation of fully automated continuous descent operation-enabled arrival routes during 1 h of operation on a busy day at a Stockholm terminal maneuvering area. The current framework successfully generates an arrival tree, merging the traffic from all TMA entry points to the runway for 30 min intervals. For a complete 1 h period, two trees (per 30 min) are computed that differ as little as possible.

Nomenclature

$\mathcal{A}_b$	=	set of all aircraft arriving at entry point b
a	=	aircraft
b	=	terminal maneuvering area entry point
c_e	=	$\lvert\Gamma_e\rvert$
D	=	aerodynamic drag
E	=	set of edges
e	=	edge
f	=	nominal fuel flow
f_a	=	dynamic constraints
f_e	=	flow on edge e
f_{idle}	=	idle fuel flow
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	=	distance threshold
$\mathcal{L}$	=	set of vertices in a path
L_a	=	Lagrange term of the cost function
ℓ_{ij}	=	length of edge (i, j)
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
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
$v_{\text{CAS,min}}$	=	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 terminal maneuvering area via b
x	=	state vector
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
x_0	=	initial conditions
$y_{a,j,p,n,t}$	=	binary indicator for aircraft a using speed profile p occupying nth vertex (on a route from b) at time t
$y_{a,j,t}$	=	binary indicator for aircraft a occupying vertex j 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: path 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

Received 14 August 2019; revision received 12 March 2020; accepted for publication 17 April 2020; published online 28 May 2020.

*Ph.D. Candidate, Department of Physics–Aerospace Division; raul.saez.garcia@upc.edu.
†Serra I Hunter Fellow, 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.

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 projected that the number of passengers will reach seven 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, known as 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 Refs. [4,5].

Furthermore, the ICAO has published guidance material [2] to support air navigation service providers 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 Ref. [6], the suggested criteria have been established without explicitly considering the aircraft type, assuming international standard atmosphere 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 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 leveloffs, impeding CDOs and continuous climb operations. Separation of the aircraft arriving from different directions requires 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 loads. 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 pretactical level that guarantees temporal separation of all aircraft arriving to the 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 will reduce both the ATCO task load (by enabling automated separation tools) and the 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 Sec. III, we formally define our problem; we give an overview on our approach and our assumptions in Sec. IV, and we present our grid-based MIP formulation in Sec. V. In Sec. 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 Sec. VII before we conclude in Sec. VIII.

II. Background

Various authors considered automatic design of arrival routes. However, to the best of our knowledge, before 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 TMAs; 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 approach can increase throughput in congested terminal airspace. They studied routes and scheduling together, and they considered the impact of different merge topologies on the aircraft scheduling. Moreover, Adacher et al. [11], studied sequencing models based on alternative graph formulation. Furthermore, Isaacson et al. [12], presented the precision air traffic operations concept, where new scheduling constraints were considered. However, the locations of merge points or constraints like the limit on a turn angle were not considered by either of these works.

Other research focusing on arrival scheduling includes the concept of three-dimensional (3-D) path arrival management (PAM), studied by Scharl et al. [13], which represents an efficient and predictable method to deliver aircraft from the top of descent to a metering fix, where air traffic control (ATC) automation support tools and modern Flight Management System (FMS) are needed. One of the core components of 3-D PAM is the en route descent advisor (EDA), as studied by Coppenbarger et al. [14]. 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. [15] also investigated 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 through digital datalink communications. 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, recategorization projects on both sides of the Atlantic (see, e.g., Ref. [16]) 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. [17] provided more information concerning this recategorization, known as wake turbulence recategorization (known as RECAT in the United States and RECAT-EU in Europe). They described the overall three-phased approach of RECAT, as well as the different data sources and severity metrics used in the development of this concept. Sadovsky et al. [18] 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 operation in high-density TMAs, as shown by Robinson and Kamgarpour [19]; in some cases,

powered CDOs (where additional thrust and the use of speed brakes are allowed) or conventional descents with leveloffs 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 Ref. [20]. The authors took the arrival times at entry points of all aircraft during a day as an input, and they 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 CDOs 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 (as 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 the 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 because 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 they are 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:

Operational Constraint 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.

Operational Constraint 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 [21]. We can formulate this requirement as a maximum indegree of two for all vertices of the tree. This allows an indegree of one or two. Several vertices with an indegree of one simply constitute a path (we also ask for a maximum outdegree of one). However, we need to merge the routes from the entry points to the runway, and so we allow the minimum number of merging routes in any point: two.

Operational Constraint 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 [21].

Operational Constraint 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 that 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.

Operational Constraint 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:

1) Step 1 involves computation of CDO speed profiles for different lengths of the entry-point–runway path for all aircraft in the considered time interval.

2) Step 2 involves computation of the arrival trees, which allow for temporal separation of all considered aircraft flying along the computed arrival paths using the CDO speed profile, for the considered time interval.

For step 2, we use a discretization: we overlay the TMA with a square grid, and we 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 the entry point to the runway to the longest edge-disjoint grid path from the entry point to the runway). In particular, we impose an upper bound on these paths. For these discrete path lengths, we compute the CDO speed profiles (step 1); see Sec. 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 (step 2) uses a MIP formulation; see Sec. 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 Operational Requirements 1–4. The speed profiles from step 1 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.

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 pretactical planning phase; i.e., we assume that we compute the arrival routes several hours (or at least 30 min) 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 [22], Andersson et al. 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, nonparallel 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-brake 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 [23]. More information regarding the computation and assumptions when generating CDOs can be found in Sec. VI.

V. Grid-Based MIP Formulation

Our MIP formulation is based on the MIP presented in Ref. [20]: to simplify, it was assumed in that paper 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 Sec. V.A, we start with a review of our prior MIP formulation for optimal STAR merge trees with temporal separation [20] (which is based on the formulation for static routes from Ref. [22]). Then, we describe how we integrate different speed profiles in Sec. V.B. In particular, we will use the speed profiles that stem from CDOs for the different aircraft; see Sec. VI for details of their computation.

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

We use a discretization: we overlay the TMA with a square grid, and we snap the location of both entry points and the 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 eight neighbors [where $N(i) =$ denotes the 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 [22] is based on the flow MIP formulation for Steiner trees [24,25]. 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 Eqs. (1–4):

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

$$x_e \geq \frac{f_e}{Q} \qquad \forall\, e \in E \tag{2}$$

$$f_e \geq 0 \qquad \forall\, e \in E \tag{3}$$

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

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 one 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 nonnegative [Eq. (3)], and the edge variables are binary [Eq. (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 Eq. (1) (and increase Q accordingly). Let w_b be the number of aircraft entering the TMA via entry point $b \in \mathcal{P}$:

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

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

$$\min \sum_{e\in E} \ell_e f_e \tag{6}$$

$$\min \sum_{e\in E} \ell_e x_e \tag{7}$$

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

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

1. Degree Constraints

Equations (1–4) are standard MIP constraints for a minimum cost flow Steiner tree formulation, and Eq. (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 Sec. III. For Operational Constraint 2, we require that the outdegree of every node is at most one and that the maximum indegree is two:

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

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

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

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

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

2. Turn Angle Constraints

The next operational constraint from Sec. III is Operational 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., $\Gamma_e = \{(j, k)\colon \measuredangle ijk \leq \theta, (j, k) \in E\}$], and let $c_e = |\Gamma_e|$, $k \neq i$, and $e = (i, j) \notin \Gamma_e$. We add the following constraint:

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

By Eq. (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 Sec. 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 that belong to the last column and last row of the grid; that is, $V' = V \setminus \{\text{last row}\} \setminus \{\text{last column}\}$:

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall\, i \in V' \setminus \{\mathcal{P} \cup r\}\colon i+1+n, i+n, i+1 \in \{\mathcal{P} \cup r\} \tag{14}$$

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 Eq. (14) does not exist. Thus, we add Eqs. (15–18):

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \;\forall\; i \in \mathcal{P} \tag{15}$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \;\forall\; i\colon i+1 \in \mathcal{P} \tag{16}$$

$$x_{i,i+1+n} + x_{i+n+1,i} + x_{i+n,i+1} \leq 1 \;\forall\; i\colon i+n \in \mathcal{P} \tag{17}$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \;\forall\; i\colon i+n+1 \in \mathcal{P} \tag{18}$$

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 described in Ref. [20]. 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 Eqs. (19–22):

$$x_{e,b} \leq x_e \;\forall\; b \in \mathcal{P},\; \forall\; e \in E \tag{19}$$

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

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

$$\sum_{i\colon (i,j)\in E} x_{(i,j),b} - \sum_{k\colon (j,k)\in E} x_{(j,k),b} = 0 \quad \forall\; j \in V \setminus \{\mathcal{P} \cup r\},\; \forall\; b \in \mathcal{P} \tag{22}$$

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 [Eq. (23)]. Additionally, whenever we know an aircraft cannot occupy a given vertex at all or certain points in time the corresponding $y_{a,j,t}$ variables are set to 0. Equation (24) ensures that an aircraft that does not arrive at entry point b will occupy b at no point in time, and Eq. (25) yields that an aircraft arriving at b occupies this vertex at no time apart from t_a^b. Finally, Eq. (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 $T = \{0, \ldots, \overline{T}\}$ be the considered time interval:

$$y_{a,b,t_a^b} = 1 \;\forall\; b \in \mathcal{P},\; \forall\; a \in \mathcal{A}_b \tag{23}$$

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

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

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

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 Eqs. (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

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

However, Eq. (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 Eqs. (28–31):

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

$$z_{a,j,k,b,t} \leq y_{a,j,t} \quad \forall\, a \in \mathcal{A},\quad \forall\, j,k \in V,\quad \forall\, b \in \mathcal{P},\quad \forall\, t \in \{0,\ldots,\overline{T}-u\} \tag{29}$$

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

$$z_{a,j,k,b,t} \geq 0 \quad \forall\, a \in \mathcal{A},\quad \forall\, j,k \in V,\quad \forall\, b \in \mathcal{P},\quad \forall\, t \in \{0,\ldots,\overline{T}-u\} \tag{31}$$

With this, we can reformulate Eq. (27) as

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

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:

$$\sum_{\tau=t}^{t+\sigma-1} \sum_{a\in\mathcal{A}} y_{a,j,\tau} \leq 1 \quad \forall\; j \in V, \quad \forall\; t \in \{0,\ldots,\overline{T}-\sigma+1\} \tag{33}$$

B. Integration of Different Speed Profiles for Aircraft

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 Sec. VI for the computation of these speed profiles. The speed profile determines the time spent to cover each segment of the route. So, for each aircraft a, $S(a)$ 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 en entry-point–runway path) $\times L$. We ensure [Eqs. (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 nth vertex (on a route from b) j at time t. Why do we need to add two new indices, and why do we 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, $\mathcal{L} = \{1, \ldots, \lambda\}$. We substitute Eqs. (23–26) by (and extend the set of constraints that set variables to zero)

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

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

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

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

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

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

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

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

We add a constraint that, 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:

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

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

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

Then, we set

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

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

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

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

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

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

Finally, we need to reformulate the constraints for forwarding the arrival time information along the arrival path from the entry point to the runway [Eqs. (28–30) and (27); Eq. (31) is not necessary for a binary variable] and the separation constraint [Eq. (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:

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \le x_{(j,i),b} \qquad \forall\, b \in \mathcal{P},\ \forall\, a \in \mathcal{A}_b,\ \forall\, (j,i) \in E, \forall\, k \in \mathcal{L},\ \forall\, t \in \{u_{a,p,k}+1,\ldots,\overline{T}+1\} \tag{49}$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \le y_{a,j,p,k,t-u_{a,p,k}} \qquad \forall\, b \in \mathcal{P},\ \forall\, a \in \mathcal{A}_b, \forall\, (j,i) \in E,\ \forall\, k \in \mathcal{L},\ \forall\, t \in \{u_{a,p,k}+1,\ldots,\overline{T}+1\} \tag{50}$$

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

The new version of Eq. (32) is

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

For the separation (we again require a minimum temporal separation of σ time units between all aircraft at all vertices), we add

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

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 the 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 periods, respectively. We define a variable ax_{ij} that determines $|x_{ij} - x_{ij}^{\text{old}}|$ in Eqs. (54) and (55), and then we limit the number of differing edges in Eq. (56) by parameter U:

$$ax_{ij} \le x_{ij} - x_{ij}^{\text{old}}\ \forall\ (j,i) \in E \tag{54}$$

$$ax_{ij} \le x_{ij}^{\text{old}} - x_{ij}\ \forall\ (j,i) \in E \tag{55}$$

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

D. 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 toward 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 edge and the entry points one outgoing edge each, constraints (9) and (10) ensure that all other vertices have a maximum indegree of two and a maximum outdegree of one. 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 [Eq. (34)]. Constraint (42) computes the number of grid edges in the path from entry point to runway in the arrival tree. Constraints (44) and (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 nonnegative variables [like presented in Eqs. (3) and (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) or speed-brake 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 [26].

Further in Sec. VI.A, we present the generic optimal control problem and, in Sec. 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 [27]:

$$\begin{aligned} \min_{\boldsymbol{u}(t)} \quad & J_{\text{ocp}} \coloneqq \phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} L_a(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p_a})\, \mathrm{d}t \\ \text{s.t.} \quad & \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ & \dot{\boldsymbol{x}}(t) = \boldsymbol{f_a}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p_a}) \\ & \boldsymbol{h_a}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p_a}) \le 0 \\ & \boldsymbol{\psi_a}(\boldsymbol{x}(t_f)) = 0 \end{aligned} \tag{57}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector, with fixed initial conditions $\boldsymbol{x}_0$; $\boldsymbol{u} \in \mathbb{R}^{n_u}$ is the control vector; and the vector $\boldsymbol{p_a} \in \mathbb{R}^{n_p}$ includes all the time-independent parameters of the model. $L_a\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}$ and $\phi\colon \mathbb{R}^{n_x} \to \mathbb{R}$ are the Lagrange and Mayer terms of the cost function, respectively. The dynamics of the state vector are expressed by a set of nonlinear equations $\boldsymbol{f_a}\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$; and $\boldsymbol{h_a}\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_h}$ and $\boldsymbol{\psi_a}\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_{\psi a}}$ 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 [Eq. (57)] is

$$\boldsymbol{H} = L_a + \boldsymbol{\lambda}_L^T \boldsymbol{f}_a + \boldsymbol{\mu}_L^T \boldsymbol{h}_a \tag{58}$$

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 [28]

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

B. Optimal Control Problem for Aircraft Descents

In this paper, the state vector has been chosen as $\boldsymbol{x} = [v, h, s]$, where v is the true airspeed (TAS), h is the altitude of the aircraft, and s is the distance to go. To obtain environmentally friendly trajectories, idle thrust is assumed and speed-brake use is not allowed throughout the descent. In such conditions, the flight-path angle is the only control variable in this problem ($\boldsymbol{u} = [\gamma]$), 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 $\boldsymbol{x}$ are expressed by the following set of ordinary differential equations, 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 $\cos\gamma \simeq 1$):

$$\boldsymbol{f}_a = \begin{bmatrix} \dot{v} \\ \dot{h} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \frac{T_{\text{idle}} - D}{m} - g\gamma \\ v\gamma \\ v + w \end{bmatrix} \tag{60}$$

where $T_{\text{idle}}: \mathbb{R}^{n_x} \to \mathbb{R}$ is the idle thrust; $D: \mathbb{R}^{n_x \times n_u} \to \mathbb{R}$ is the aerodynamic drag; g is the gravity acceleration; w is the wind; and m is the mass, which is assumed to be constant because the fuel consumption during an idle descent is a small fraction of the total m [29].

In this paper, the trajectory is divided in two phases: the latter part of the cruise phase before the top of descent, 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-phase optimal control problem can be converted into a single-phase optimal control problem as follows:

$$J_a = \int_{t_0}^{t_f} -\left(\frac{f + \text{CI}}{v_{\text{cr}}}\right)(v + w) + f_{\text{idle}} + \text{CI}\, \mathrm{d}t \tag{61}$$

where v_{cr} is the cruise speed (which is assumed to be constant), $f: \mathbb{R}^{n_x \times n_u} \to \mathbb{R}$ is the cruise fuel flow (which is assumed to be constant since the cruise altitude and speed are also constant), and $f_{\text{idle}}: \mathbb{R}^{n_x} \to \mathbb{R}$ 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 [23].

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

$$\boldsymbol{h}_a = \begin{bmatrix} v_{\text{CAS,min}} - v_{\text{CAS}} \\ v_{\text{CAS}} - \text{VMO} \\ M - \text{MMO} \\ \gamma \\ \gamma_{\text{min}} - \gamma \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{62}$$

where $v_{\text{CAS}}: \mathbb{R}^{n_x} \to \mathbb{R}$ is the calibrated airspeed (CAS) and $M: \mathbb{R}^{n_x} \to \mathbb{R}$ is the Mach number (both are functions of the state vector); $v_{\text{CAS,min}}$ and VMO are the minimum and maximum operative CAS, respectively; MMO is the 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}, as well as their respective parameters. Here, we adopt EUROCONTROL's Base Of Aircraft Data (BADA) Version 4 model [30].

Finally, terminal constraints fix the final states vector:

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

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 [Eq. (58)] is also linear with respect to the control, leading to a singular optimal control problem that can be solved semianalytically from the implicit formulation of optimal singular arcs [31].

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 goes 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 aforementioned model and the steps to generate an optimal trajectory semianalytically can be found in Ref. [31].

1. *Speed Profile*

As an example, Fig. 1 shows the optimal speed profile for an Airbus A320, in international standard atmospheric 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 $v_{\text{CAS,min}}$ and MMO/VMO. Also note that, since these maximum and minimum speeds are given in terms of the CAS and Mach, the corresponding TAS changes with

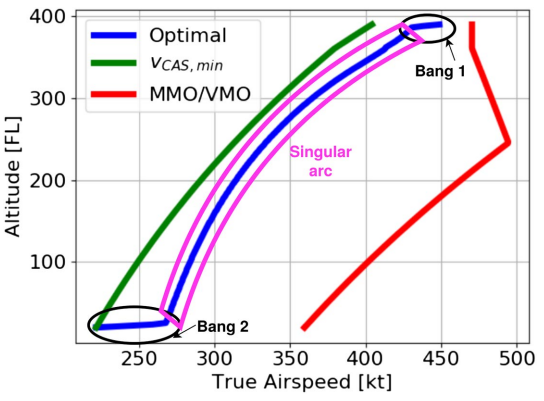


Fig. 1 Example of an Airbus A320 optimal speed profile (FL = flight level, RWY = runway).

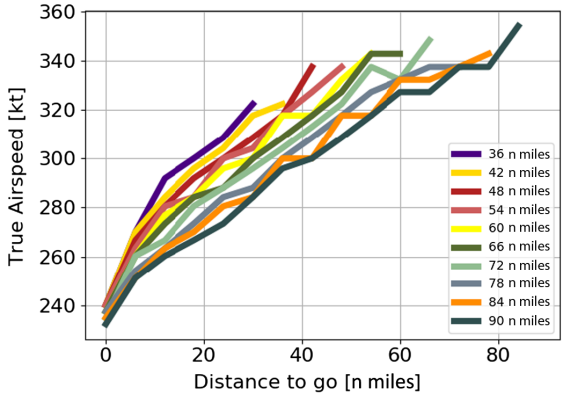


Fig. 2 Example of Airbus A320 speed profiles for several path lengths within Stockholm TMA.

altitude. The earliest and latest trajectories would correspond to the $v_{\mathrm{CAS,min}}$ and MMO/VMO speed profiles, respectively.

2. *Input Data*

We obtained the flight traffic data needed to generate the trajectories from EUROCONTROL's Demand Data Repository 2 (DDR2) [32], 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 Sec. VI.B). In the case in which the aircraft model did not correspond to any of the BADA models, we used a comparable aircraft in terms of performance and dimensions.

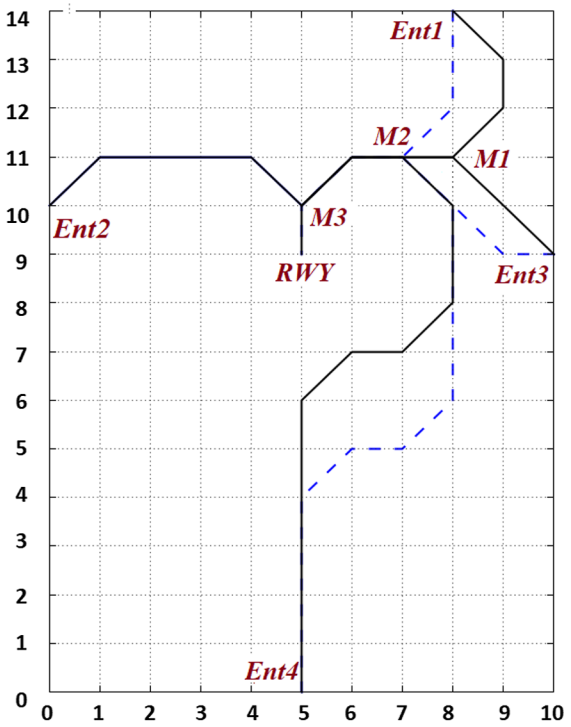
For any given flight, the number of trajectories generated corresponds to the number of possible routes the aircraft can fly. In this case, path lengths within the TMA from 36 n miles (corresponding to the minimum path length within the grid) to 90 n miles were considered, with each path split into several segments of constant length equal to 6 n miles. For instance, a 36-n-mile path inside the TMA would be split into six 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 to 90 n miles). 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 Sec. 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

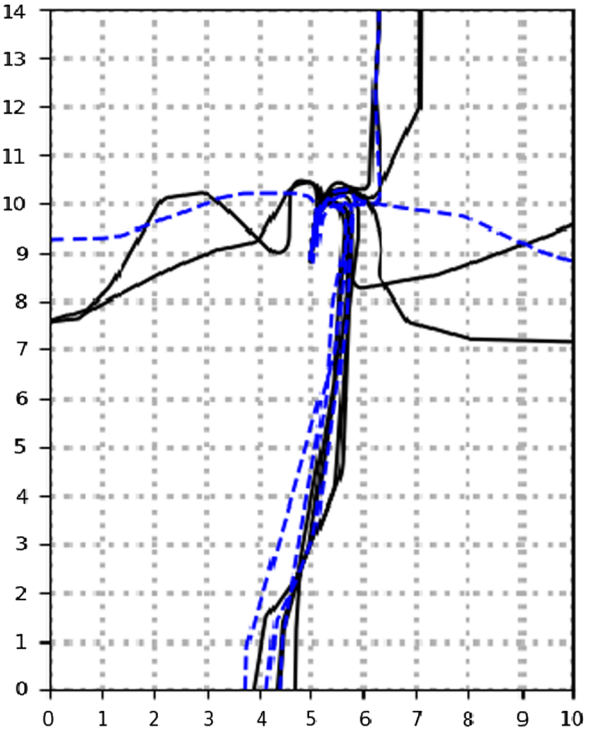
In this section, we apply our framework to a real-world instance, aiming at showing its feasibility by considering arrival routes in a Stockholm TMA. We use the aircraft arrival times at TMA entry points during 1 h of airport operation as input (taken from EUROCONTROL's DDR2) and compute the dynamic arrival routes with guaranteed temporal separation for this time interval. We have chosen a data sample for 3 October 2017, which is one of the busiest days of that year with, in total, 432 aircraft arrivals. We use $\beta = 0.1$; that is, we prioritize the weighted shortest path in the objective function. We set a temporal separation of at least 2 min ($\sigma = 2$ min).

We solve our MIP using Gurobi optimization software installed on a very powerful Tetralith server [33], using the Intel HNS2600BPB computer nodes with 32 CPU cores, 384 GB, provided by the Swedish National Infrastructure for Computing.

We use an 11×15 grid, which automatically guarantees merge point separation of about 6 n miles (parameter L). In current operations, a separation of 5 n miles is used; that is, we show results in the operational separation range (using a finer 14×19 grid, which would result in a 5-n-mile 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 the aircraft model, the current altitude, and the true airspeed at the top of descent, as well as the distance to go (which defines which exact speed profile the aircraft is taking). An example of a set of speed profiles for an A320 aircraft is shown in Fig. 2. Every curve corresponds to the flight speed profile along the path followed inside the TMA, for which the length varies depending on the tree configuration and could take values from 36 up to 90 n miles.



a) Arrival route trees



b) OpenSky network trajectories

Fig. 3 Arrival route trees and real aircraft trajectories (from OpenSky network) during 1 h of Stockholm Arlanda operation ($n_1 = 10$, $n_2 = 14$).

Table 1 Time schedule example for 17 aircraft arriving between 1500 and 1600 hrs at Stockholm TMA on 3 October 2017

Arrivals		Simulated time, min			
Aircraft	Entry point	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	— —	— —	— —

In Fig. 3a, we show arrival trees computed for two half-hours within a 1 h period of operation in Stockholm Arlanda on 3 October 2017. An arrival tree for 10 aircraft entering a Stockholm TMA between 1500 and 1530 hrs is shown in solid black lines, and an arrival tree generated for seven aircraft arriving to the TMA between 1530 and 1600 hrs is shown in dashed blue lines. For the second tree, we add Eqs. (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 h, and memory usage was around 55%. Figure 3b illustrates the real trajectories flown by aircraft at Arlanda during the same period (1500–1600 hrs that day), obtained from the historical database of the OpenSky network [34]. As expected, for each entry point, aircraft do not follow fixed paths; this is probably due to path stretching instructions issued by ATC in order to avoid possible conflicts. With our method, however, there will be no need for such instructions because 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 the TMA.

In Fig. 4, we compare the fuel consumption for aircraft following our arrival tree and performing CDOs with the 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 by following the same process as the one shown in Ref. [35]. We can observe that the fuel consumption increases drastically for the aircraft entering the TMA from the south because 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 because 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 5–25% less fuel. The distance is similar to the one flown by the actual trajectories but, when following the arrival tree, aircraft fly neutral CDOs; whereas in the actual case, this is not always true.

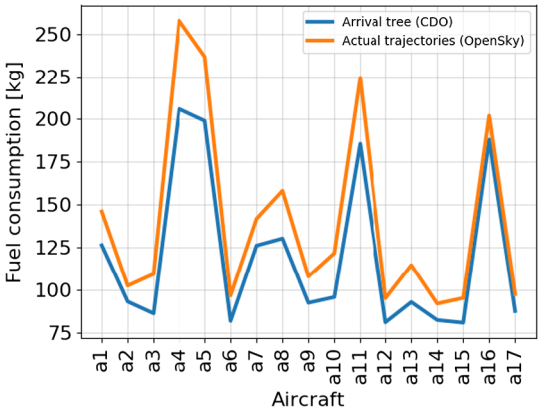


Fig. 4 Fuel consumption for aircraft flying the arrival tree (CDOs) vs actual trajectories (obtained from OpenSky network).

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 min 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.

VIII. Conclusions

A MIP-based framework was presented for the computation of arrival route trees that fulfill several operational constraints and guarantee temporal separation of all aircraft arriving to a 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-brake usage) for the entry-point–runway path length in the tree. The current approach has the potential to both reduce the environmental impact (CDOs) and the task load of ATCOs in planned operations. The feasibility of the current framework was proved by presenting experiments for calculation of the arrival routes for 1 h of operation on a busy day at a Stockholm TMA.

The current implementation is quite sensitive to the number of aircraft. With the current setup, the problem for all 22 scheduled aircraft for the given hour could not be solved; the current grid only allowed a feasible solution for 17 arrivals. This discrepancy in aircraft number is not due to computational limits of the framework (or server) but is rooted in the input data: If aircraft have less than 2 min of separation in the entry point, this would result in an infeasible problem; hence, aircraft are filtered out 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, it was decided to filter out the problematic aircraft. In future work, the aim is 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, a nonoptimal speed profile can be imposed on aircraft in these conflict cases. According to the experiments, this would be necessary for about 15–20% of the aircraft, which is deemed feasible. In busier airports, more nonoptimal 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.

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 min. 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 suboptimal routes. Adjusting the tree configuration every 20–30 min, which is about the time aircraft spend in a 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, the current 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 the solution would be to add some of the air traffic scheduling constraints presented in Ref. [12], which deals with the precision air traffic operations

concept. In the future, it is planned to integrate dynamic replanning for weather avoidance into the framework. It is also planned 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 are covered in one of the eight edge directions. Finally, the concepts discussed in Refs. [13–15] could be integrated in the current tool to improve the precision of the 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," International Air Transport Association Press Release 55, Montreal, Nov. 2015, http://www.iata.org/pressroom/pr/Pages/2015-11-26-01.aspx [retrieved 2 June 2016].

[2] "Continuous Descent Operations (CDO)," International Civil Aviation Organization TR 9931/AN/476, Montreal, 2010.

[3] "Continuous Climb and Descent Operations," European Organisation for the Safety of Air Navigation, Brussels, 2019, https://www.eurocontrol.int/concept/continuous-climb-and-descent-operations [retrieved 15 Dec. 2017].

[4] Warren, A., and Tong, K., "Development of Continuous Descent Approach Concepts for Noise Abatement," *IEEE/AIAA 21st Digital Avionics Systems Conference (DASC)*, IEEE Publ., Piscataway, NJ, 2002, pp. 1E3–1E3. https://doi.org/10.1109/DASC.2002.1067906

[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*, AIAA Paper 2003-6771, 2003. https://doi.org/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. https://doi.org/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, pp. 159–182. https://doi.org/10.2514/atcq.14.2.159

[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)*, EUROCONTROL (European Organisation for the Safety of Air Navigation) & FAA (Federal Aviation Administration), Brussels and Washington, D.C., 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. https://doi.org/10.2514/2.2838

[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)*, IEEE Publ., Piscataway, NJ, 2010, pp. 2.A.5-1–2.A.5-17. https://doi.org/10.1109/DASC.2010.5655500

[11] Adacher, L., Pacciarelli, D., and Pranzo, M., "A Graph-Theoretical Model for the Aircraft Sequencing Problem," Università degli Studi di Roma Tre, Rome, Italy, 2003.

[12] Isaacson, D. R., Sadovsky, 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, No. 4, 2014, pp. 234–257. https://doi.org/10.2514/1.I010119

[13] Scharl, J., Berge, M. E., Coats, M. L., Haraldsdottir, A., and Schoemig, E. G., "Modeling and Analysis of the 3-D Path Arrival Management Concept," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA Paper 2007-6878, 2007. https://doi.org/10.2514/6.2007-6878

[14] 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*, AIAA Paper 2004-4875, 2004. https://doi.org/10.2514/6.2004-4875

[15] 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, No. 4, 2009, pp. 1200–1209. https://doi.org/10.2514/1.39795

[16] "Operational Service and Environment Definition (OSED) for Time Based Separation for Arrivals (TBS)," NATS and EUROCONTROL Project Number 06.08.01, 2013.

[17] 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*, AIAA Paper 2016-3434, 2016. https://doi.org/10.2514/6.2016-3434

[18] 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, Dec. 2013, pp. 157–176. https://doi.org/10.1016/j.trc.2013.09.017

[19] Robinson, J. E., III, and Kamgarpour, M., "Benefits of Continuous Descent Operations in High-Density Terminal Airspace Under Scheduling Constraints," *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, AIAA Paper 2010-9115, 2010.

[20] 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)*, IEEE Publ., Piscataway, NJ, 2018, pp. 1–9. https://doi.org/10.1109/DASC.2018.8569646

[21] 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*, AIAA Paper 2018-6825, 2008.

[22] 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)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2016, pp. 9:1–9:13. https://doi.org/10.4230/OASIcs.ATMOS.2016.9

[23] "Getting to Grips with the Cost Index—Issue II," Airbus TR 2, Blagnac, France, 1998.

[24] 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. https://doi.org/10.1007/BF02612335

[25] Goemans, M. X., and Myung, Y., "A Catalog of Steiner Tree Formulations," *Networks*, Vol. 23, No. 1, 1993, pp. 19–28. https://doi.org/10.1002/(ISSN)1097-0037

[26] 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)*, IEEE Publ., Piscataway, NJ, 2018, pp. 1–10. https://doi.org/10.1109/DASC.2018.8569597

[27] Soler, M., Olivares, A., and Staffetti, E., "Multiphase Optimal Control Framework for Commercial Aircraft Four-Dimensional Flight-Planning Problems," *Journal of Aircraft*, Vol. 52, No. 1, 2014, pp. 274–286. https://doi.org/10.2514/1.C032697

[28] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor and Francis, New York, 1975.

[29] 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. https://doi.org/10.2514/1.5572.

[30] 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)*, IEEE Publ., Piscataway, NJ, 2010, pp. 1.D.1-1–1.D.1-14. https://doi.org/10.1109/DASC.2010.5655518

[31] 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)*, AIAA Paper 2017-3432, 2017.

[32] *DDR2 Reference Manual for General Users 2.9.4*, EUROCONTROL, Brussels, 2017.

[33] "Tetralith Server, NSC, Linköping University," Linköping, Sweden, 2019, https://www.nsc.liu.se/systems/tetralith/ [retrieved 28 Jan. 2019].

[34] "OpenSky Network," OpenSky, Burgdorf, Switzerland, July 2019, https://opensky-network.org [retrieved 2 Dec. 2018].

[35] 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*, Electronic Navigation Research Institute (ENRI), Paper EN-A-31, Tokyo, Japan, 2019.

S. Lozito
Associate Editor