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 that guarantee temporal separation of all aircraft arriving within a given time period, where the aircraft are flying according to the optimal continuous descent operation speed profile with idle thrust. 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.

Nomenclature

$\mathcal{A}_b$	=	set of all aircraft arriving at entry point b
a	=	aircraft
b	=	TMA 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_{idle}	=	idle fuel flow
f_e	=	flow on edge e

[*] PhD Candidate, Department of Physics - Aerospace Division, raul.saez.garcia@upc.edu

[†] Serra i Húnter 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

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
$\mathcal{P}$	=	set of entry points
p	=	speed profile
p_a	=	time-independent parameters
Q	=	large number
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	=	maximum number of 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 = weighting coefficient used in the convex combination of the paths length and tree weight objectives

Γ_e = the set of all outgoing edges from j that form an angle $\leq \gamma$ with e

γ = angle threshold

γ_a = flight path angle

$\gamma_{a,min}$ = minimum descent gradient

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 workload, 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, workload. 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 workload (by enabling automated separation tools) and environmental impact (by ensuring CDOs).

In the next section we review related work on this topic. In Section III we formally define our problem, and present our grid-based MIP formulation in Section IV. In Section V 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 VI, before we conclude in Section VII.

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, or did not consider all our constraints, or they 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 consider the impact of different merge typologies on the aircraft scheduling. The location of merge points, or constraints like the limit on a turn angle were not considered by Choi et al.

A lot of attention has been given to sequencing close to the runway, for example, re-categorization projects on both sides of the Atlantic (see, e.g., [11]) 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.

A MIP-based approach for automation of aircraft separation along the optimized arrival routes was presented in [12]. 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—i.e., a tree that 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)—such that all aircraft flying according to their optimal continuous descent speed profile for the route length of the entry point-runway path along the tree are separated at all points of the arrival routes. 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 [13]. We can formulate this requirement as a maximum in-degree of 2 for all vertices of the tree (vertices with in-degree 1 lie simply on a path, we need to merge routes towards the runway, but we minimize the number of routes that may merge at any vertex).
3) *Merge point separation:* Operational constraint 2) could be circumvent 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 [13].
4) *No sharp turns:* Because of aircraft dynamics, an aircraft cannot turn at an arbitrarily acute angle: there is a 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 γ. Again, if we would allow to use arbitrarily short edges, a sequence of many short edges with pairwise angles larger than γ could still lead to a too sharp turn for the aircraft. So, it is the combination of the threshold γ 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 γ—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. Grid-based MIP Formulation

Our MIP formulation is based on the MIP we presented in [12]: to simplify, we assumed there that covering a single edge takes u_a time units for all aircraft (independent of aircraft type and distance to runway). In this paper, we integrate aircraft with different speed, and in particular the speed given by a continuous descent profile for the specific aircraft type. Any other speed profile could be used.

We start with a review of our prior MIP-formulation for optimal STAR merge trees with temporal separation [12] (which is based on our formulation for static routes from [14]) in Subsection IV.A, we then describe how we integrate different speed profiles in Subsection IV.B, in particular, we will use the speed profiles that stem from CDOs for the different aircraft, see Section V for details of their computation.

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

We use a dicretization: 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 $(i, j) \in E$.

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 [(14]) is based on the flow MIP formulation for Steiner trees [15, 16]. We use 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):

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

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

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

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

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 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 $b \in \mathcal{P}$:

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

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

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

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

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

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

1. Degree constraints

Equations (1)-(4) are standard MIP-constraints for a 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 in-degree is 2:

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

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

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

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

Equation (11) ensures that the runway r has one in-going 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 γ with e. Let Γ_e be the set of all outgoing edges from j that form an angle $\leq \gamma$ with e, i.e., $\Gamma_e = \{(j, k) : \measuredangle ijk \leq \gamma, (j, k) \in E\}$, and let $c_e = |\Gamma_e|$. 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 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 IV.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, $V' = V \setminus \{\text{last row}\} \setminus \{\text{last column}\}$.

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

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 Equation (14) does not exist. Thus, we add Equations (15)-(18).

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

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

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

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

4. Integration of Temporal Separation

So far, our MIP can compute an optimal (static) arrival tree (according to objectve function (8)). In this subsection we review the temporal separation we described in [12]. 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):

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

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

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

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

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, 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 in-going 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.

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

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

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

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

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 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

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

However, 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):

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

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

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

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

With this we can reformulate Equation (27) as:

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

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:

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

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 V for the computation of these speed profiles. The speed profile determines the time to cover the first, second.... segment of the route.

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, $\mathcal{L} = \{1, \ldots, \lambda\}$. We substitute Equations (23)-(26) by (and extend the set of constraints that set variables to zero):

$$\begin{aligned}
&\sum_{p \in \mathcal{S}(a)} y_{a,b,p,1,t_a^b} &&= 1 && \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b && (34)\\
&y_{a,b,p,k,t_a^b} &&= 0 && \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall k \neq 1 \in \mathcal{L} && (35)\\
&y_{a,b,p,1,t} &&= 0 && \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall t \in T \setminus \{t_a^b\} && (36)\\
&y_{a,b',p,k,t} &&= 0 && \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a), \forall k \in \mathcal{L}, \forall t \in T && (37)\\
&y_{a',b,p,1,t_a^b} &&= 0 && \forall b \in \mathcal{P}, \forall a' \neq a \in \mathcal{A}_b, \forall p \in \mathcal{S}(a) && (38)\\
&y_{a,j,p,k,t} &&\leq \sum_{\substack{i \in V:\\ (i,j) \in E}} x_{(i,j)} && \forall j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in \mathcal{S}(a), \forall k \in \mathcal{L}, \forall t \in T && (39)\\
&y_{a,j,p,1,t} &&= 0 && j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in \mathcal{S}(a), \forall t \in T && (40)\\
&\sum_{p \in \mathcal{S}(a)} y_{a,j,p,k,t} &&\leq 1 && j \in V, \forall a \in \mathcal{A}, \forall k \in \mathcal{L}, \forall t \in T && (41)
\end{aligned}$$

We add a constraint, which, for each entry point, computes the length of the path from entry point to runway in the arrival tree:

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

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:

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

Then we set:

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

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:

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

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

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

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

Finally, we need to reformulate the constraints for forwarding the arrival time information along the arrival path from 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.

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

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

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

The new version of Equation (32) is:

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

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

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

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 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:

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

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

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

D. The Complete MIP

To enhance readability, we give the list of all constraints in the complete MIP. We use objective function (8). As constraints we use: (2), (5), (9), (10), (11), (12), (13), (14), (15)-(18), (19)-(22), (34)-(41), (42), (44), (45)-(48), (49)-(51), (52), and (53). Additionally, we of course set the range of binary and non-negative variables (like presented in Equations (3), (4)).

V. 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 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 [17].

Further in Subsection V.A, we present the generic optimal control problem and in Subsection V.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 [18]:

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

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; $L_a : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}$ and $\phi : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}$ 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 $\boldsymbol{f_a} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}^{n_x}$; $\boldsymbol{h_a} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}^{n_h}$ and $\boldsymbol{\psi_a} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}^{n_{\psi a}}$ 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:

$$H = L_a + \boldsymbol{\lambda_L}^T \boldsymbol{f_a} + \boldsymbol{\mu_L}^T \boldsymbol{h_a} \tag{58}$$

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 [19]:

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

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 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 ($\boldsymbol{u} = [\gamma_a]$), 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 (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.,$\sin\gamma_a \simeq \gamma_a$ and $\cos\gamma_a \simeq 1$):

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

where $T_{idle} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}$ is the idle thrust; $D : \mathbb{R}^{n_x \times n_u} \longrightarrow \mathbb{R}$ 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 [20].

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:

$$J = \frac{f}{v_{cruise}} + \int_{t_0}^{t_f} (f_{idle} + \mathrm{CI})\, dt \tag{61}$$

where $f : \mathbb{R}^{n_x \times n_u} \longrightarrow \mathbb{R}$ and $f_{idle} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}$ are the nominal and idle fuel flow, respectively; and 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 [21].

In addition to the dynamic constraints $\boldsymbol{f}$, 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:

$$\boldsymbol{h_a} = \begin{bmatrix} v_{CAS,min} - v_{CAS} \\ v_{CAS} - \text{VMO} \\ M - \text{MMO} \\ \gamma_a \\ \gamma_{a,min} - \gamma_a \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{62}$$

where $v_{CAS} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}$ is the calibrated airspeed (CAS) and $M : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}$ 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 $\gamma_{a,min}$ 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 [22].

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 $\boldsymbol{x}_f = \left[v_f, h_f, s_f\right]$ 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 [23].

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 sates 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 [23].

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}. It can be observed 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.

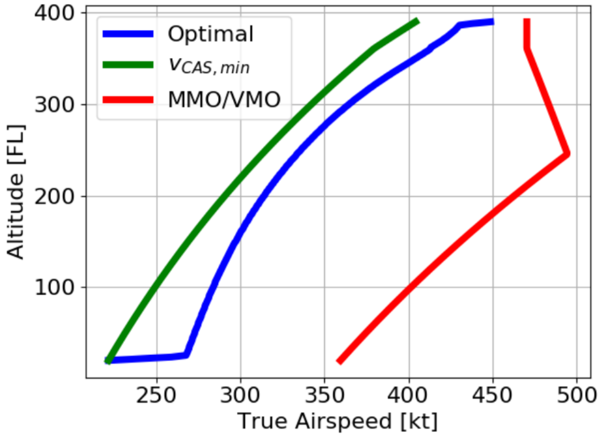


Fig. 1 Example of an Airbus A320 optimal speed profile

2. Input data

We obtained the flight traffic data needed to generate the trajectories from EUROCONTROL's data demand repository (DDR2) [24], which contains information about the trajectories flown every day. To generate the optimum trajectories, we use five parameters: aircraft model; cruise altitude; distance to go, i.e., the distance remaining to the metering fix by following a given route; speed, i.e., the true airspeed of the aircraft in cruise, obtained from the segment length and time from DDR2; and the cost index.

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

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 to 90 NM were considered, with segment lengths of 6 NM. This means that for every flight 10 trajectories were computed. Furthermore, all those trajectories were generated to ensure the same time at the TMA entry point, which meant that different cost index values were used for each trajectory. Finally, we also obtained the entry point for each flight form DDR2.

VI. 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 [25], 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 yield 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.

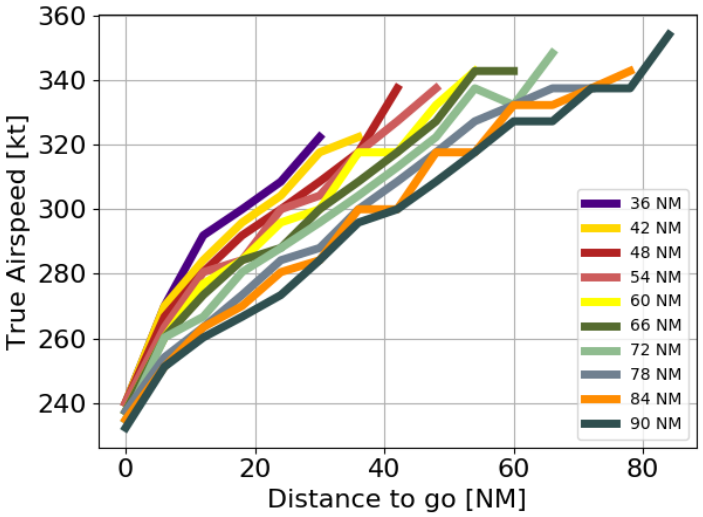
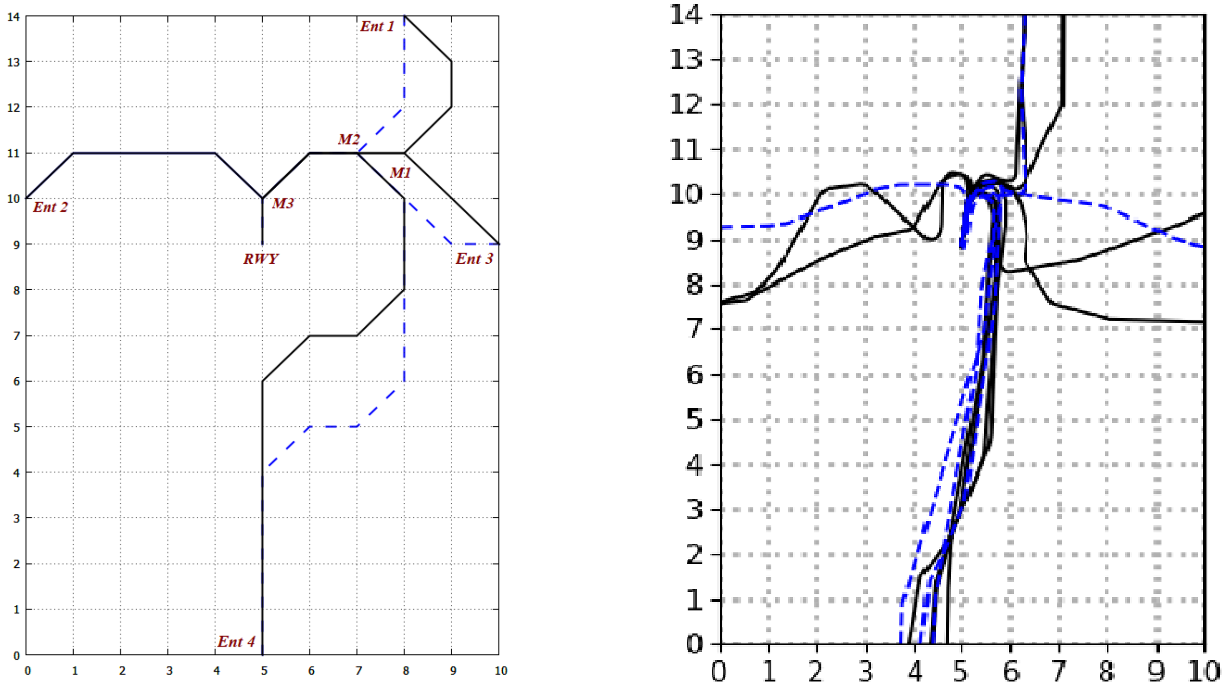


Fig. 2 Example of an Airbus A320 speed profiles for several path lengths within Stockholm TMA

In figure 3(a) we show arrival trees computed for two half hours within a one-hour period of operation. An arrival tree for 10 aircraft entering Stockholm TMA between 3:00 and 3:30 pm is shown in solid black, 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. 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 [26]. 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.



(a)

(b)

Fig. 3 a) Arrival route trees calculated using optimization framework for one hour of Stockholm Arlanda operation in October 3, 2017, for aircraft arriving during the periods: *black solid line* 3:00-3:30 pm, *blue dashed line* 3:30-4:00 pm. Entry points and merge points are reference points for the time schedule presented in Table 1. b) Real aircraft trajectories during the same time period obtained from Historical Database of the 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 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.

Table 1 Time schedule example for 17 aircraft arriving between 3 pm and 4 pm at Stockholm TMA on October 3, 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	22
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	-	-	-

VII. Conclusions and Future Work

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 CDO speed profiles for the entry point–runway path length in the tree. Our approach has the potential to both reduce the environmental impact (CDOs) and the workload 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 feasbile.

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 at minimum will provide consistency between the trees, preventing extra workload because of switching.

Finally, in the future we also plan to integrate dynamic re-planning for weather avoidance into our framework.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Goemans, M. X., and Myung, Y., "A catalog of Steiner tree formulations," *Networks*, Vol. 23, No. 1, 1993, pp. 19–28.

[17] 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.

[18] Soler, M., A, O., , and Staffetti, E., "Multiphase Optimal Control Framework for Commercial Aircraft Four-Dimensional Flight-Planning Problems," *Journal of Aircraft*, 2014. doi:10.2514/1.C032697.

[19] Bryson, A. E., and Ho, Y.-C., *Applied optimal control : optimization, estimation, and control*, Taylor and Francis Group, New York, 1975.

[20] 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.

[21] Airbus, "Getting to grips with the cost index - Issue II," Tech. Rep. 2, 1998.

[22] 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.

[23] 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.

[24] Eurocontrol, *DDR2 Reference Manual for General Users 2.9.4*, 2017.

[25] “Tetralith server, NSC, Linköping University,” https://www.nsc.liu.se/systems/tetralith/, 2019. Accessed on 28/01/2019.

[26] OpenSky, “OpenSky Network,” https://opensky-network.org, July 2019. Accessed on December 2, 2018.