Integrating Atmospheric Conditions into the Dynamic-Arrival-Routes Framework

Roghayeh Hajizadeh,
Elina Rönnberg
Department of Mathematics
Linköping University
Linköping, Sweden
roghayeh.hajizadeh@liu.se,
elina.ronnberg@liu.se

Henrik Hardell
Department of Science and Technology
Linköping University
Norrköping, Sweden
Airspace Unit, Luftfartsverket (LFV),
Norrköping, Sweden
henrik.hardell@liu.se

Tatiana Polishchuk,
Christiane Schmidt
Department of Science and Technology
Linköping University
Norrköping, Sweden
tatiana.polishchuk@liu.se,
christiane.schmidt@liu.se

***Abstract*—We consider the computation of dynamic arrival routes (DARs): arrival routes from all entry points of a terminal maneuvering area that merge traffic arriving in a given time interval towards the runway for which we can guarantee separation for all arriving aircraft and enable all aircraft to follow optimal descent profiles. We exploit our recent mathematical results to bring the concept of DARs closer to practical applicability by demonstrating how we can integrate both the impact of atmospheric conditions on the aircraft's descent profiles and the interaction with departing traffic into our DARs framework. With experimental results for Stockholm Arlanda airport, we give a proof of concept of this integration—showing that we can quickly obtain optimal solutions—and highlight the importance of these aircraft-specific and traffic-situation-dependent factors.**

***Keywords*—dynamic arrival routes, environmentally friendly descents, automated aircraft arrival routing and scheduling, atmospheric conditions**

I. Introduction

While air traffic volumes dropped significantly during the Covid-19 pandemic, volumes have recovered and reached 3.8% above pre-pandemic levels in 2024 [1]. This trend is expected to continue, where the International Air Transport Association (IATA) [2] predicts that air travel demand will double by 2040, growing at an average annual rate of 4.3%. This ever increasing traffic volume puts more and more stress and workload on air traffic controllers (ATCOs) and increases the complexity of their task. Additionally, it results in elevated environmental impact. While the aim is to curtail these effects in general, a specific focus is on terminal maneuvering areas (TMAs), as all traffic is gathered there.

A potential mitigation approach for the environmental impact in TMAs are environmentally friendly, fuel-optimized descents. However, to allow such optimal aircraft-specific 4D trajectories with an airport's strategic standard arrival routes may result in undercut separation minima and generally reduced predictability for the ATCOs. The controllers may then either increase separation buffers—decreasing throughput (and airport capacity) despite increasing traffic volumes—or issue instructions to deviate from the optimal trajectories—not fully exploiting the environmental benefits. Thus, it is crucial to support ATCOs with tools that provide arrival routes with automated separation, both to employ the fuel-optimized descents effectively and to reduce the ATCO workload. To answer to the increasing traffic volumes, any such tool should deliver an optimal flight sequence to achieve a high throughput. Additionally, as the characteristics of the traffic situation (number of aircraft, entry-point distribution of arrivals) change significantly during the day, the arrival routes (which should enable the optimal flight sequence) have to change over a day as well—leading to dynamic arrival routes (DARs).

If we want to demonstrate that such a tool can in the future go beyond a proof of concept, we need to ensure that all factors that affect the aircraft-specific optimal trajectories (e.g., the aircraft type of the approaching aircraft, the wind direction, and other atmospheric conditions) and the traffic situation (e.g., the wake-turbulence categories of consecutively arriving aircraft) are taken into account and that we quickly compute the arrival routes and the associated 4D trajectories, such that regular recomputation and with that truely dynamic routes are possible. This is crucial as the DARs (determining all but the temporal component) for all aircraft are fixed in any computation horizon (the dynamics stem from changes from time period to time period) and if long computation times enforce long time periods of fixed routes, an aircraft/aircraft pair that was present in the TMA a long time ago may enforce unnecessarily long arrival routes for all aircraft now.

The entry-point–runway paths of the DARs should form an (arrival) tree: exactly one path leaves each entry point, and once two paths merged, they will never diverge—as we aim for low complexity— where we require all paths to merge before the runway. Hence, all paths together form a (graph-theoretical) tree (with the runway as root and entry points as leaves).

A team of authors [3] developed a mixed-integer-programming-based (MIP-based) graph-based model to compute DARs, that is, optimal aircraft arrival routes with fully automated scheduling of fuel-optimized descents that ensure aircraft separation and proposed an operational concept that facilitates the use of these routes. For this paper, optimal descent profiles for all aircraft and different lengths of entry-point–runway routes were computed in advance. The MIP-

based model picks the correct descent profile for each aircraft depending on the length of the arrival route from the entry point to the runway, which is also selected by the model. Additionally, in the model the advancement of each aircraft along its route in the underlying grid graph is tracked (and the arrival of each aircraft to the TMA entry point can be adapted within a given time interval). Consequently, the model is an arc- and flow-based model, where we assign aircraft with a specific descent profile to a node in the underlying grid graph at a specific point in time. Within the paper, the author team demonstrated the general feasibility of the approach, but the specific MIP-based model is seriously affected by long runtimes: computing the arrival trees and sequence for a half-hour scenario for Stockholm Arlanda airport took up to 40.9 hours for high-traffic cases with artificially inserted light aircraft; one-hour scenarios with high traffic could not be solved at all. This is clearly an infeasible runtime in real-world operations, where we would aim to recompute the DARs about every 30 minutes, that is, within the magnitude of the time an aircraft typically spends in the TMA.

The framework also struggles to include additional, practically critical features, particularly, the influence of wind and other atmospheric conditions: in reality, the descent profile depends not only on the length of the descent but also on, for example, the wind direction relative to the aircraft's trajectory. However, in the model, we track how many edges in an entry-point–runway path an aircraft has covered at a specific time to determine how much time it will need to cover the next edge, where both the path length and the descent profile are chosen in the model. Thus, the model features only a dependence on how many grid-graph edges have been covered, not in which direction each of these runs—all arrival paths of equal length are treataed equally. This aircraft-progress tracking yields a very large number of variables[1]. Making the descent profiles—and with that the time an aircraft spends to traverse the k-th edge in the entry-point–runway path—dependent not only on the length of the entry-point–runway path, but also on the direction of the edges is not feasible—already without this feature the model is clearly at the edge of what can be computed with commercial solvers.

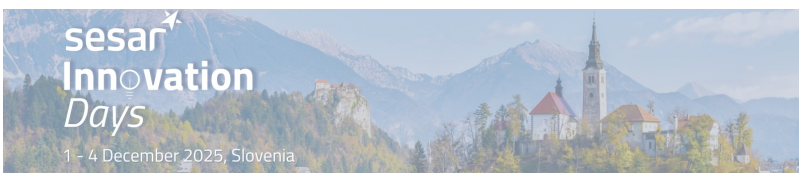
The goal of this paper is to show that these problems are not inherent to the concept of DARs—but stem from the specific mathematical formulation—and that we can in fact integrate more aircraft-specific and traffic-situation-dependent factors while achieving short runtimes that allow for the frequent re-computation that is needed in an operational setting–altogether narrowing the gap to practical applicability of DARs. To resolve the issues on runtime, a subset of the authors of this paper [4], [5] presented a Dantzig-Wolfe reformulation, based on discretisation, of the complete previous model [3] and demonstrated that this can significantly reduce runtimes—allowing us to compute solutions within 5 seconds to 12.65 minutes for the half-hour high-traffic scenarios from Stockholm Arlanda airport, for which the previous model needed up to 40.9 hours; additionally, it enables us to solve one-hour high-traffic scenarios for the same airport within 22.22 to 58.57 minutes—instances we could not solve at all with the original model (and which cover a longer time horizon than what we would expect in operations). The reformulation implies that we moved from the arc- and flow-based model, where we assign aircraft with a specific descent profile to a node in an overlayed grid at a specific point in time, to a path-based model, where we decide on which path from each TMA entry point to the runway should be used—which then directly fixes the descent profile for each aircraft's trajectory.

In this paper, we exploit these mathematical—rather theoretical—results to bring the concept of DARs closer to practical applicability: we show how the reformulation enables even the integration of the impact of atmospheric conditions and the interaction with departure routes. Our main focus is on the impact of atmospheric conditions. As described, previously, descent profiles for DARs were computed in international standard atmosphere (ISA) conditions and no wind—out of necessity because of the described complexity issues in the first case [3] and because the sake of the later work [4], [5] was to show the improvement that the Dantzig-Wolfe reformulation yields over the previous model. With the new model, we have a one-to-one correspondence between each entry-point–runway path and the descent profile for each aircraft's trajectory. Hence, a single descent profile does not have to depend only on the length of the path, and we detail how it can depend on the atmospheric conditions, for example, the wind direction w.r.t. different segments on the path. Consequently, the computation of the descent profiles in the real-world scenario can take the forecasted conditions into account, which then yield a separation guarantee for these projected conditions.

Moreover, we cover another aspect of narrowing the gap for our tool to be applicable in operations: so far, we focused solely on the arrival routes (and the scheduling of the aircraft along these routes), of course, these do not exist isolatedly in the TMA, but departures are scheduled as well. As a first approach to integrate this interaction, we exclude grid nodes covered by departure routes from the feasible paths. A (less conservative) alternative would be to only exclude those nodes covered by departure routes that we expect to be used in the considered time period. At a later stage, we aim to handle the departures similar to how we treat arriving aircraft that are still present in the TMA from the last arrival-route-computation period in [5], where we consider the 4D positions of these aircraft as fixed and enforce separation from them for all aircraft in the current computation period—an even less conservative approach.

We plan the arrival paths in a two-dimensional grid graph, but aim for 4D positions. The aircraft trajectories (given by the arrival path plus the descent profile) determine the spatial and temporal position of the aircraft and, thus, we do in fact plan 4D arrivals.

[1] We have Ω(# entry points $\times$ #arriving aircraft $\times$ # of grid-graph edges $\times$ upper bound on the # edges in any entry-point–runway path $\times$ # time steps in the time horizon) many variables.



A. Roadmap

In the remainder of this section, we present related work. We state the formal problem in Section II. In Section III, we detail our method: we provide the path-based model from [4], [5] in Subsection III-A, we detail how we compute the atmospheric-conditions-dependent descent profiles in Subsection III-B, and we present how we integrate the interaction with departures in Subsection III-C. We give experimental results for Stockholm Arlanda Airport in Section IV and conclude in Section V.

B. Related Work

In [3], a detailed concept of operations (CONOPS) is given that could technically enable the solutions provided in this paper. In short, the proposed strategy aims to manage the aircraft arriving to an airport from the cruise phase until the runway in three regions: a pre-sequencing area, a dynamic-trajectories area, and the final merging fix. In the pre-sequencing area, communication between ATCOs and aircraft is established and the required time of arrival (RTA) at the dynamic-trajectories horizon and a specific route in the dynamic-trajectories area are negotiated. Aircraft follow published routes in this area, but may adapt speed, vertical profile and, to some extent, their route to meet the negotiated RTA. The routes in the dynamic-trajectories area link each entry point with the runway and these are the DARs computed in this paper, which in the CONOPS will be dynamically generated by the ATCO ground-support system. Finally, all traffic flows will merge at the final merging fix, from where they continue to the runway.

Both the optimization of aircraft trajectories (e.g., [6]), routing (e.g., [7]) and aircraft sequencing/separation (e.g., [8]) have been studied by many authors, however, we are interested in the integration of these problems and limit our review of related work on combined approaches. For a more detailed review of this related work, we refer to [5].

Choi et al. [9] and Kamo et al. [10] presented aprroaches to compute aircraft arrival routes and the arrival sequence. While Kamo et al. [10] showed a robust-optimization approach, they presented only a very limited case study. Ma et al. [11] integrated various scheduling problems in the TMA and at the airport and compute speed, arrival and departure times, the runway assignment, and the pushback time at the airport.

Several groups of authors considered point-merge arrivals: Hardell et al. [12] presented a (simple) MIP to sequence and merge the arriving traffic (all followiong continuous descent operations) and the departing traffic. Liu et al. [13] used a mathematical model to set flight routes, aircraft entry times and speeds, while ruling out conflicts at merge points and on route segments. The authors aimed to compare point-merge and trombone arrival routes with the model.

In this paper, we intgegrate the interaction of arrival and departure routes by excluding the graph vertices close to at least one departure route in the considered time interval. A similar approach was presented in [7], though in that work, the authors compute static arrival routes.

II. Problem Formulation

We are given the locations of TMA entry points and the runway, a set of aircraft planned for arrival within a given time interval, a time window in which each aircraft can arrive to the entry point (which we need to circumnavigate infeasibility), path-dependent descent profiles for these aircraft, and separation requirements. We aim to compute an optimal arrival tree, such that all aircraft follow the path in the tree from their entry point to the final merging fix, from where they will proceed to the runway (for simplicity, we usually refer to these as entry-point–runway paths). The tree must fulfill various conditions:

1) No more than two routes merge at a point (to limit complexity).
2) Merge points are separated by a minimum distance $\mathcal{D}$, [14] (to prevent many merge points in close vicinity).
3) Routes do not make sharp turns in order to respect aircraft dynamics.
4) Obstacles, like no-fly zones, are avoided.
5) Temporal separation depending on the wake-turbulence categories of leading and trailing arriving aircraft is ensured. In the second part of our paper, we require additionally that separation between arriving and departing aircraft is guaranteed during the considered time interval.
6) All aircraft arrive to their entry points within a given time interval around the planned time.
7) All aircraft follow a set of descent profiles that take atmospheric conditions into account; in this paper, all aircraft follow optimal descent profiles.

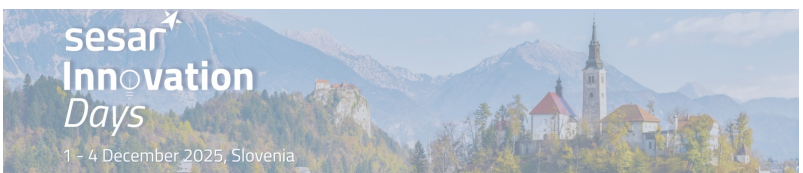
Our tree should be good w.r.t. two criteria: the total distance flown in the TMA (with the motivation of obtaining short flight routes for the aircraft)—the paths length—and the weight of the tree (with the motivation of low complexity for the ATCOs and a small area impacted by noise etc.)—the tree weight.

III. Dynamic-Arrival-Routes Framework

In this section, we describe the components of our optimization framework. For clarity, we introduce the path-based model from [4], [5] in Subsection III-A; in Subsection III-B, we detail how we compute the descent profiles that serve as input to the mathematical model; and in Subsection III-C, we present how we handle the interaction with the departure routes in the form of standard instrument departure routes (SIDs) to ensure that arriving and departing traffic are separated.

A. The Path-Based Model

1) Setup: We discretize the TMA into a square grid of size $q \times n$ with node set V, where the side length of each grid pixel is equal to the minimum separation distance $\mathcal{D}$. This also ensures that (consecutive) merge points are separated by a minimum distance. Entry points and runway are snapped to the grid. We consider a bi-directed grid graph on V, where each node is connected to its eight neighbors with edges from the edge set E, the exception are the entry points (with only



outgoing edges) and the runway (with only incoming edges), otherwise for any adjacent $i, j \in V$ both (i, j) and (j, i) are included in E. We denote the length of edge $(i, j) \in E$ by l_{ij}, the set of entry points by $\mathcal{P} \subseteq V$, and the runway by $r \in V$.

Let $\mathcal{A}_b$ denote the set of aircraft arriving to entry point $b \in \mathcal{P}$ and $\mathcal{A} = \bigcup_{b\in\mathcal{P}} \mathcal{A}_b$ the set of all aircraft. Let $\bar{t}_a$ denote the planned arrival time of aircraft a at its entry point. Enforcing this time for all aircraft may already yield undercut separation minima at the entry points, hence, we allow an aircraft to arrive at any time within $[\bar{t}_a - \mu, \bar{t}_a + \mu]$. Additionally, let $\overline{T}$ be an upper bound on the time all considered aircraft spend in the TMA until landing. We discretize the time and denote the set of time stamps by $T = \{0, \ldots, \overline{T}\}$.

For the temporal separation, we need to consider wake-turbulence categories: let $\mathcal{C} = \{1, 2, \ldots, n_c\}$ be the set of indices of all different disjoint wake-turbulence categories where C_κ for $\kappa \in \mathcal{C}$ is a set of aircraft from the same category and $\mathcal{A} = \bigcup_{\kappa\in\mathcal{C}} C_\kappa$. Let $\sigma_{\kappa_1,\kappa_2}$ denote the required temporal separation if the leading aircraft is from C_{κ_1} and the trailing aircraft is from C_{κ_2}. We let $\Omega = \max_{\kappa_1,\kappa_2\in\mathcal{C}} \sigma_{\kappa_1,\kappa_2}$.

2) Generation of Paths and Trajectories: We select exactly one path from each entry point to the runway, and with constraints that limit which combinations of these paths from the different entry points are allowed, we enforce Conditions 1–7. This selection means that we have one variable per path, and for that we need to create all possible paths a priori.

For us, the criteria to determine whether a path is possible (or, better, feasible) are the length (we have an upper bound λ on the number of grid-graph edges in any path) and the angle between consecutive path edges (to ensure Condition 3). We essentially perform an angle-limited depth-first search (DFS) from each entry point in the grid-graph; we improve the runtime by precomputing shortest paths (with minimum number of edges) from each entry point to the runway, and when we reach a node v with the DFS, we check whether the edges already covered until v plus the shortest path exceeds λ to cut infeasible branches off early.

Let Π_b denote the set of paths from entry point b to the runway and $\mathbf{\Pi} = \bigcup_{b\in\mathcal{P}} \Pi_b$ be the set of all paths. For any $\pi \in \mathbf{\Pi}$, we define θ_π as the set of edges that path π passes through. Let $\xi_{a,\pi,i}$ be the time when aircraft a with entry time $\bar{t}_a$ occupies node i on path π based on our pre-computed, atmospheric-conditions-dependent descent profiles. Moreover, let $\Xi_{a,i,t} = \{\pi \in \overline{\Pi}_i \cap \Pi_b : \quad \xi_{a,\pi,i} = t\}$, that is, $\Xi_{a,i,t}$ is the set of all paths on which aircraft a with entry time $\bar{t}_a$ occupies node i at time t. Finally, let $\Upsilon_{a,i,t,\sigma} = \{\pi \in \overline{\Pi}_i \cap \Pi_b : \quad t \leq \xi_{a,\pi,i} \leq t + \sigma - 1\}$, that is, $\Upsilon_{a,i,t,\sigma}$ denotes the set of all paths on which aircraft a with entry time $\bar{t}_a$ occupies node i sometime between time t and $t+\sigma-1$. We use these two sets of paths to ensure our separation minima (Condition 6).

3) MIP: Given all the sets and parameters defined in Subsubsections III-A1 and III-A2, we present the path-based formulation. We use binary variables:

- ρ_π indicates whether path π is used in the arrival tree.
- $x_{i,j}$ indicates whether edge $(i, j) \in E$ participates in the arrival tree.
- $\tau_{a,\pi,t}$ indicates whether aircraft a uses path π and arrives at its entry point at time t. We use these variables to select the appropriate trajectory on the chosen entry-point–runway path for each aircraft.

With these, we can present our model:

$$\min \quad \beta \sum_{(i,j)\in E} l_{i,j} x_{i,j} + (1-\beta) \sum_{b\in\mathcal{P}} \sum_{\pi\in\Pi_b} \sum_{(i,j)\in\theta_\pi} |\mathcal{A}_b| l_{i,j} \rho_\pi \tag{1}$$

$$\text{s.t.} \quad \sum_{j:(j,i)\in E} x_{j,i} \leq 2, \qquad \forall i \in V \setminus \{\mathcal{P} \cup r\}, \tag{2}$$

$$\sum_{j:(i,j)\in E} x_{i,j} \leq 1, \qquad \forall i \in V \setminus \{\mathcal{P} \cup r\}, \tag{3}$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \qquad \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\}, \tag{4}$$

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1, \qquad \forall i \in \mathcal{P} \cap V', \tag{5}$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1, \qquad \forall i : i+1 \in \mathcal{P}, \tag{6}$$

$$x_{i,i+1+n,} + x_{i+n+1,i} + x_{i+n,i+1} \leq 1, \qquad \forall i : i+n \in \mathcal{P}, \tag{7}$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1, \qquad \forall i : i+n+1 \in \mathcal{P}, \tag{8}$$

$$\sum_{\pi\in\Pi_b} \rho_\pi = 1, \qquad \forall b \in \mathcal{P}, \tag{9}$$

$$\rho_\pi \leq x_{i,j}, \qquad \forall \pi \in \mathbf{\Pi}, \forall (i,j) \in \theta_\pi, \tag{10}$$

$$\sum_{\pi\in\Pi_b} \sum_{t=\bar{t}_a-\mu}^{\bar{t}_a+\mu} \tau_{a,\pi,t} = 1, \qquad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \tag{11}$$

$$\sum_{t=\bar{t}_a-\mu}^{\bar{t}_a+\mu} \tau_{a,\pi,t} = \rho_\pi, \qquad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall \pi \in \Pi_b, \tag{12}$$

$$\sum_{b\in\mathcal{P}} \sum_{a\in\mathcal{A}_b\cap C_{\kappa_2}} \sum_{t'=-\mu}^{\mu} \sum_{\pi\in\Upsilon_{a,i,t-t',\sigma_{\kappa_1,\kappa_2}}} \tau_{a,\pi,\bar{t}_a+t'} \leq \Omega\Big(1 - \sum_{b\in\mathcal{P}} \sum_{a'\in\mathcal{A}_b\cap C_{\kappa_1}} \sum_{t'=-\mu}^{\mu} \sum_{\pi'\in\Xi_{a',i,t-t'}} \tau_{a',\pi',\bar{t}_{a'}+t'}\Big),$$
$$\forall \kappa_1, \kappa_2 \in \mathcal{C} : \kappa_1 \neq \kappa_2, \ \forall i \in V, \ \forall t \in \{0, \ldots, \overline{T} - \sigma_{\kappa_1,\kappa_2} + 1\}, \tag{13}$$

$$\sum_{b\in\mathcal{P}}\sum_{a\in\mathcal{A}_b\cap C_\kappa: a\neq a'}\sum_{t'=-\mu}^{\mu}\sum_{\pi\in\Upsilon_{a,i,t-t',\sigma_{\kappa,\kappa}}}\tau_{a,\pi,\bar{t}_a+t'} \leq \Omega(1-\sum_{t'=-\mu}^{\mu}\sum_{\pi'\in\Xi_{a',i,t-t'}}\tau_{a',\pi',\bar{t}_{a'}+t'}),$$

$$\forall\kappa\in\mathcal{C},\forall a'\in C_\kappa,\forall i\in V,\ \forall t\in\{0,\dots,\overline{T}-\sigma_{\kappa,\kappa}+1\}, \quad (14)$$

$x_{i,j}$ binary, $\forall(i,j)\in E$, (15)

ρ_π binary, $\forall\pi\in\mathbf{\Pi}$, (16)

$\tau_{a,\pi,t}$ binary, $\forall b\in\mathcal{P},\forall a\in\mathcal{A}_b,\forall t\in[\bar{t}_a-\mu,\bar{t}_a+\mu]$. (17)

Our objective function (1) is a convex combination of the two criteria given in Section II: the paths length and the tree weight. Constraints (2) and (3) enforce Condition 1, and we use Constraints (4)–(8) to prevent crossings of tree edges (which we need to do as we are not solely aiming for shortest paths). With Constraint (9), we ensure that exactly one path from each entry point (to the runway) is used for the arrival tree, and we connect variables (those for the used paths and those for the grid-graph edges forming the underlying tree) with Constraint (10). We enforce that for each aircraft, exactly one trajectory is chosen with Constraint (11). Constraint (12) forces the number of used trajectories on a path to one if the path is used and to zero otherwise. (Constraints (12) and (9) yield Constraint (11), but explicitly including it improved computational performance). Finally, we enforce temporal separation according to the wake-turbulence categories of leading and trailing aircraft with Constraints (13)–(14), where (13) and (14) handle separation of two aircraft from two or one wake-turbulence category, respectively. Finally, Constraints (15)–(17) define the domains of variables.

B. Computation of Descent Profiles

To compute the dynamic arrival routes with the path-based model in Subsection III-A, we need to model the descent profiles of all arriving aircraft for all feasible paths as input. As the purpose of this paper is to showcase that we can handle the impact of atmospheric conditions, we briefly detail how we compute these descent profiles. For each aircraft a arriving to entry point b, we need to compute the descent profile for all feasible entry-point–runways paths $\pi\in\Pi_b$, that is, we need to compute $\sum_{b\in\mathcal{P}}|\mathcal{A}_b||\Pi_b|$ distinct descent profiles. In this paper, we use fuel-optimized descent profiles for all aircraft—this solely stems from our motivation to reduce environmental impact and is not required by the model.

$$(Th-D)\cdot V_{\text{TAS}} = m\cdot g_0\cdot\frac{dh}{dt}+m\cdot V_{\text{TAS}}\cdot\frac{dV_{\text{TAS}}}{dt} \quad (18)$$

We use EUROCONTROL Base of Aircraft Data (BADA) v.4 to model the performance of each arriving aircraft. By continuously solving the Total Energy Model (Equation (18)) for dh/dt at each timestamp, starting at the final point of the trajectory and going backwards, we obtain the rate of descent along the entire arrival of the flight. At each timestamp, we update and input the idle thrust (as we are assuming a CDO), drag D, true airspeed V_{TAS}, and mass m (accounting for the decrease due to burning fuel) of the aircraft. Additionally, in Equation (18), Th indicates the thrust force and g_0 the gravitational acceleration. For the descent profile, we match the initial calibrated airspeed (CAS) or Mach number of the actual flight, estimated from the ADS-B data (obtained either from the cruise or the descent, depending on whether the computed descent profile contains only the descent or both the cruise and the descent), and assume for the rest of the descent that the aircraft follows the aircraft-type-specific descent speed schedule proposed by BADA. The speed schedule is a function of the altitude of the aircraft and is based either on fixed speeds provided for the aircraft in the BADA performance data files, or on the stall speed. This speed schedule provides the necessary input for dv/dt. While computing the descent profile backwards, we do not allow a top of descent altitude higher than the cruise altitude of the corresponding actual flight. That is, a path may contain a flat cruise segment depending on the length of it. As a results of computing the descent profile backwards, the resulting CDO trajectory may start at a higher altitude at TMA entry, compared to that of the actual flight.

During the aircraft-performance computations, we consider the actual wind and temperature at the specific time and point in space that our experiment covers, obtained via the European Centre for MediumRange Weather Forecasts ERA5 Reanalysis dataset. We use linear interpolation to estimate the value at a specific time and position. Taking these atmospheric properties into consideration, affects, for example, the conversion between different speed formats. Access to temperature data enables the conversion from Mach number to V_{TAS}, in case the speed of the schedule is expressed in Mach (at high altitudes). In case the speed of the schedule is expressed in CAS, we use the temperature for obtaining the correct air density, which goes into the conversion formula from CAS to V_{TAS}. Once we have computed the V_{TAS}, we are able to compute the ground speed (GS), by using the flight track angle and the u- and v-components of the wind data. This conversion process is employed during every iteration of a descent profile computation, with small variations depending on the altitude of the flight, which controls whether Mach number or CAS acts as the input speed format. Several other formulas within the descent profile computation are affected by the temperature, including the ones for idle thrust force and idle thrust fuel flow. More details on our descent performance computations can be found in [15]. For the descent profiles without incorporating atmospheric conditions (or, more accurately, for international standard atmosphere conditions and no wind), we employ the method from [3].

C. Taking Departures into Account

As discussed in Section I, in this paper, we take the interaction of the dynamic arrival routes with departures into account by excluding all grid nodes that are close to SIDs within a specified along-route distance from the runway. Formally, let d be the distance threshold. For a point p on a SID, let $s(p)$ be the distance measured along that SID from the runway, and let $P^{\mathbf{SID}}$ be the set of all points on the SIDs. Define the subset of SID points within along-route distance d as $\bar{P}_d^{\mathbf{SID}} = \{\, p \in P^{\mathbf{SID}} :\ s(p) \leq d \,\}$.

Moreover, let $\Delta(u, v)$ denote the distance between points u and v on the Earth's surface, and let $\mathcal{D}$ be the minimum separation distance. We define the excluded node set as $V_T^{\mathbf{SID}}(d, \mathcal{D}) = \{v \in V : \exists p \in \bar{P}_d^{\mathbf{SID}}$ with $\Delta(p, v) \leq \mathcal{D}\}$, and consider $V' = V \setminus V_T^{\mathbf{SID}}(d, \mathcal{D})$ and the induced subgraph on V' for the computation in time period T.

IV. Experiments: Stockholm Arlanda Airport

The implementation is done in Python version 3.13.7, and we solve our model using Gurobi version 12.0.3 as the solver with gurobipy as the interface to Python. We run the experiments on a MacBook Pro, M1 2020.

We use real-world arriving traffic data at Stockholm Arlanda TMA for a high-traffic scenario on May 16, 2018, and we obtain the data from the open-source database of the Opensky Network [16]. Arlanda airport has three runways 01L/19R, 01R/19L, and 08/26 and four main entry points HMR (north), XILAN (east), NILUG (south), and ELTOK (west). Moreover, we use an 11×15 grid, which ensures a separation of 6 NM.

We generate all possible entry-point–runway path with at most $\lambda = 12$ (for the experiments in Subsection IV-A) and $\lambda = 14$ (for the experiments in Subsection IV-B) edges. The angle between consecutive path edges is set to be at least 135°.

We employ ICAO aircraft categories [17]: LIGHT (L), MEDIUM (M) and HEAVY (H), and group them as $C_1 = \{H, M\}$ and $C_2 = \{L\}$. Following ICAO separation minima [18], we set $\sigma_{1,2} = 3$ minutes and $\sigma_{1,1} = \sigma_{2,2} = \sigma_{2,1} = 2$ minutes, that is, we deviate from the 2-minute separation only when a light aircraft trails a medium or heavy one. For only a single aircraft category, this yields a temporal separation of $\sigma = 2$ minutes. Finally, we choose $\beta = 0.1$ in the objective, which places emphasis on the weighted shortest paths.

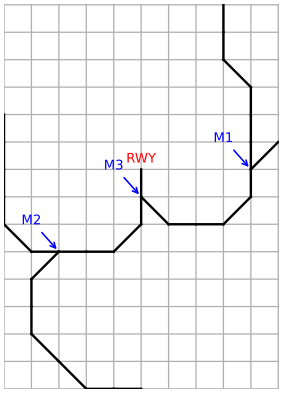
A. Arrival Routes with Impact of Atmospheric Conditions

We consider arrival traffic at Stockholm Arlanda on May 16, 2018, from 5:00 to 5:09 AM, including seven aircraft arriving on runway 01R. We generate all possible entry-point–runway paths with at most $\lambda = 12$ edges, yielding a total of 843 paths. This computation takes only 0.17 seconds. We choose a quite short time interval and a relatively small value for λ, because we need to compute $\sum_{b \in \mathcal{P}} |\mathcal{A}_b||\Pi_b|$ distinct descent profiles —for each aircraft, one for each of the feasible paths from its entry point—and our main goal here is to give a proof of concept. We have $\sum_{b \in \mathcal{P}} |\mathcal{A}_b||\Pi_b| = 1210$. In contrast, we need only 56 distinct descent profiles if we omit the impact of atmospheric conditions (for each of the seven aircraft, we compute descent profiles of length $5, \ldots, 12$)—these profiles depend only on the length, thus, we need fewer descent profiles for each instance. We generate the 1210 and 56 descent profiles with the method described in Subsection III-B (including the MET conditions for the considered time period) and the method presented in [3], respectively. Computing a single descent profile with atmospheric conditions takes approximately 25 seconds, and since each aircraft must generate profiles for all feasible paths from its entry point, the total runtime would exceed eight hours if executed sequentially. This makes onboard computation impractical. However, as the generation of individual profiles is entirely independent across both paths and aircraft, the process is highly parallelizable. With sufficient computational resources on the ground, all profiles could be produced simultaneously—enabling near real-time solution generation during operations. On the other hand, the descent profiles that depend only on the path length (without atmospheric conditions) can be computed within a few seconds each.

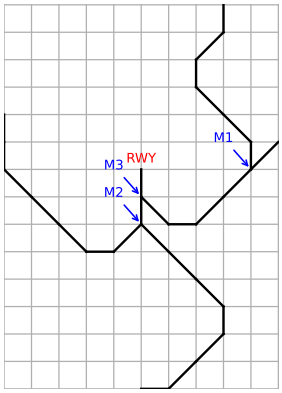
We run the model both with and without atmospheric conditions, under two levels of time discretization: 1 minute and 30 seconds.With that, we aim to test which granularity is feasible and what the impact on the DARs is. In the 1-minute discretization, we use fixed entry times, so the planned and scheduled entry times coincide. In the 30-second discretization, we do not obtain a feasible solution under fixed entry times. Hence, we allow an entry point time-window offset of $\mu = 1$, corresponding to one discretization step (30 seconds), with which an aircraft's scheduled entry time can deviate from the planned entry time by at most that offset.

We provide the resulting arrival trees for our DARs (both with and without taking the atmospheric conditions into account) for the 1-minute discretization in Figure 1 and for the 30-second discretization in Figure 2. The corresponding optimized time schedules are given in Tables I and II for the 1-minute discretization, and in Tables III and IV for the 30-second discretization. For each aircraft, the tables list the planned entry time $\bar{t}_a$ (and the scheduled entry time when entry times are flexible), the time at which it passes each merge point denoted on the corresponding tree, and its final arrival time at the runway. Note that we have one aircraft category and the temporal separation for that is $\sigma = 2$. In all cases, the results confirm that the temporal separation constraints are satisfied along every path.

Without atmospheric conditions, the objective value, tree weight, total paths lengths (the total aircraft distances along the paths), and total run time in seconds (generating and solving the MIP model) are 80.13, 33.73, 85.28, and 1.28, respectively, for a 1-minute discretization, and 70.14, 31.31, 74.45, and 3.40, respectively, for a 30-second discretization. With atmospheric conditions, these values are 75.81, 33.63, 80.50, and 0.43 for a 1-minute discretization, and 69.96, 32.56, 74.11, and 0.67 for a 30-second discretization. The total time to generate the trajectories for each experiment is less than one second. The total time to generate all feasible paths and the time to generate all descent profiles is described in the beginning of

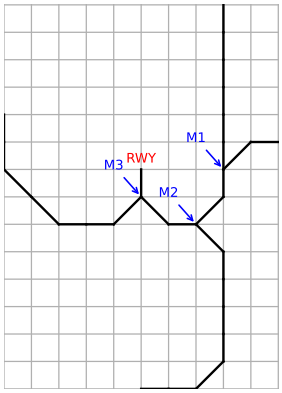


(a) Without atmospheric conditions taken into account

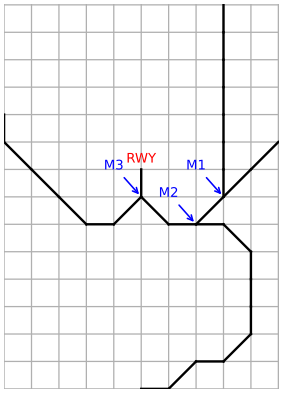


(b) With atmospheric conditions taken into account

Figure 1. Arrival trees for the 1-minute discretization



(a) Without atmospheric conditions taken into account



(b) With atmospheric conditions taken into account

Figure 2. Arrival trees for the 30-second discretization

TABLE I. OPTIMIZED TIME SCHEDULE WITHOUT ATMOSPHERIC CONDITIONS AND 1-MINUTE DISCRETIZATION (WITH SEPARATION $\sigma = 2$ MINS)

Entry point	Aircraft	$\bar{t}_a$	M1	M2	M3	RWY
XILAN	1	5:05	5:06	-	5:11	5:12
HMR	2	5:04	5:10	-	5:15	5:16
HMR	3	5:07	5:13	-	5:18	5:19
HMR	4	5:10	5:16	-	5:21	5:23
ELTOK	5	5:03	-	5:09	5:13	5:14
ELTOK	6	5:08	-	5:26	5:38	5:41
NILUG	7	5:07	-	5:28	5:40	5:43

TABLE II. OPTIMIZED TIME SCHEDULE WITH ATMOSPHERIC CONDITIONS AND 1-MINUTE DISCRETIZATION (WITH SEPARATION $\sigma = 2$ MINS)

Entry point	Aircraft	$\bar{t}_a$	M1	M2	M3	RWY
XILAN	1	5:05	5:06	-	5:14	5:16
HMR	2	5:04	5:10	-	5:18	5:20
HMR	3	5:07	5:16	-	5:24	5:26
HMR	4	5:10	5:20	-	5:28	5:30
ELTOK	5	5:03	-	5:14	5:16	5:18
ELTOK	6	5:08	-	5:20	5:22	5:24
NILUG	7	5:07	-	5:18	5:20	5:22

TABLE III. OPTIMIZED TIME SCHEDULE WITHOUT ATMOSPHERIC CONDITIONS AND 30-SECOND DISCRETIZATION (WITH SEPARATION $\sigma = 2$ MINS)

Entry point	Aircraft	$\bar{t}_a$	Scheduled time	M1	M2	M3	RWY
XILAN	1	5:05:00	5:05:30	5:07:30	5:09:30	5:11:30	5:13:00
HMR	2	5:04:00	5:03:30	5:09:30	5:11:30	5:13:30	5:15:00
HMR	3	5:07:00	5:07:30	5:13:30	5:15:30	5:17:30	5:19:00
HMR	4	5:09:30	5:10:00	5:16:30	5:19:30	5:22:30	5:24:30
ELTOK	5	5:02:30	5:02:00	-	-	5:09:00	5:10:30
ELTOK	6	5:08:00	5:08:30	-	-	5:15:30	5:17:00
NILUG	7	5:07:00	5:07:30	-	5:31:30	5:37:30	5:41:00

TABLE IV. OPTIMIZED TIME SCHEDULE WITH ATMOSPHERIC CONDI-TIONS AND 30-SECOND DISCRETIZATION (WITH SEPARATION σ = 2 MINS)

Entry point	Aircraft	$\bar{t}_a$	Scheduled time	M1	M2	M3	RWY
XILAN	1	5:05:00	5:05:00	5:08:00	5:10:00	5:14:00	5:16:00
HMR	2	5:04:00	5:03:30	5:10:30	5:12:30	5:16:30	5:18:30
HMR	3	5:07:00	5:07:30	5:14:30	5:16:30	5:20:30	5:22:30
HMR	4	5:09:30	5:10:00	5:20:30	5:22:30	5:26:30	5:28:30
ELTOK	5	5:02:30	5:02:00	-	-	5:12:00	5:14:00
ELTOK	6	5:08:00	5:08:30	-	-	5:18:30	5:20:30
NILUG	7	5:07:00	5:07:30	-	5:18:30	5:22:30	5:24:30

this subsection. Without atmospheric conditions, each aircraft spends on average 13.64 minutes in the TMA with the 30-second discretization and 17.71 minutes with the 1-minute discretization. With atmospheric conditions, the average times are 14.36 minutes in the 30-second discretization and 16.00 minutes in the 1-minute discretization.

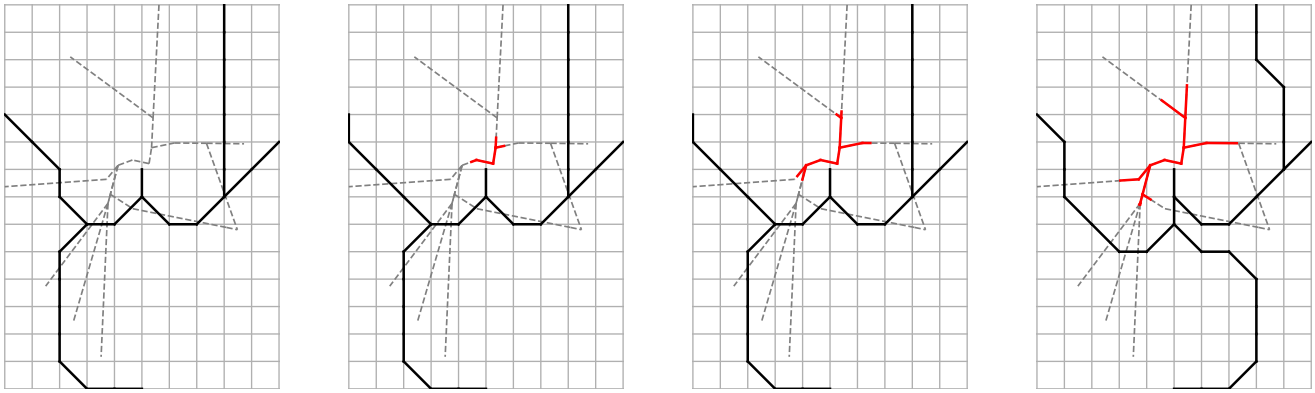
Aircraft 6 and 7 spent a longer time in the TMA than the other aircraft in the results presented in Table I. The main reason lies in the structure of the descent profiles: for aircraft of certain types, such as Aircraft 6, the descent profile yields a (rounded) time of one minute per edge traversal for paths of up to ten edges, but for longer paths, the descent profile yields a rounded time of three minutes per edge traversal. Hence—depending on the length of the arrival path and the chosen time discretization—the time to traverse arrival paths of 10 or 11 edges may differ significantly (in the case of Aircraft 6, we obtain traversal times of 10 and 33 minutes, respectively), and with that may yield a long time spent in the TMA.

The wind during the time period of our experimental evaluation is very light, hence, the main contributor (except different path length) to the significant difference in time in TMA for the scenarios with and without MET conditions may be the different methods we used for modeling the descent of an aircraft. This difference may have an effect on the speed profile of the aircraft.

The finer resolution of 30 seconds clearly both reflects the real-world situation better and yields significantly lower objective values and lower time in TMA. Our model can generate feasible DARs while fully considering atmospheric conditions, a necessary condition for real-world usage. Hence, our approach is capable of incorporating realistic operational conditions, providing a reliable foundation for practical use in air traffic management.

B. *Arrival Routes Integrated with Departure Routes*

For this experiment, we consider arrival traffic at Stockholm Arlanda on May 16, 2018, from 5:00 to 5:29 AM, including 16 aircraft (arriving on runway 01R). To introduce multiple wake-turbulence categories, six of these 16 aircraft were chosen randomly to be (artificially) modeled as light aircraft, resulting in two categories for the experiments. We compute the descent profiles according to [3] (without taking atmospheric conditions into account). We set the entry point time-window offset to $\mu = 2$ (for a discretization of one minute) and consider



(a) Without consideration of SIDs (b) For $d = 6$ NM (c) For $d = 12$ NM (d) For $d = 18$ NM

Figure 3. Optimal arrival trees with and without consideration of SIDs, and for different values of d

all SIDs originating from runway 01L. Runways 01R and 01L are parallel and can be operated independently, allowing simultaneous take-offs and landings.

Based on our approach from Subsection III-C, we require that arrival routes must maintain separation from any SID within a distance d from the runway. We run the model for several values of d. For each case, we identify all points on the SIDs that lie within along-route distance d of the runway and remove the corresponding grid nodes, as well as all nodes within 6 nautical miles of those points (that is, we use $\mathcal{D} = 6$ NM to determine $V_T^{\text{SID}}(d, \mathcal{D})$).

We provide the resulting optimal arrival trees for our DARs in Figure 3: we show the arrival tree without integration with the SIDs in Figure 3(a); we give the arrival trees for $d = 6$, $d = 12$, and $d = 18$ NM in Figure 3(b), Figure 3(c), and Figure 3(d), respectively. All SIDs are shown as dashed gray lines, the parts of the SIDs to which we want to guarantee separation (i.e., all points in $\bar{P}_d^{\text{SID}}$) are marked in red, and the optimal arrival tree is shown in black. We can clearly see that for an increasing value of d the DARs are "pushed out" towards the TMA boundary. For $d \leq 12$ we obtain the same DARs, and the objective value is 149.03, the tree length is 31.14, the total paths lengths is 162.13, and each aircraft spends on average 13.50 minutes in the TMA. For $d = 18$, the objective value is 162.04, the tree length is 33.97, the total paths lengths is 176.27, and each aircraft spends on average 14.37 minutes in the TMA. The runtime to obtain the DARs without integration with SIDs is 64.17 seconds, while with integration for $d = 6$, 12, and 18 the runtimes are 60.36, 56.53, and 26.77 seconds, respectively. The time to generate all trajectories without integration is 18.13 seconds, and with integration for $d = 6$, 12, and 18 the corresponding times are 15.79, 14.34, and 8.00 seconds, respectively. The runtime decreases as d increases because the total number of feasible paths decreases. The time to generate all the feasible paths in each case in less than one second. We allow a maximum entry-time deviation of 2 minutes, the average entry-time deviation in our DARs is 1.19, 1.13, 1.44, and 1.44 minutes for the SID-independent DARs, and for $d = 6$, 12, and 18, respectively.

For $d = 24$ NM no feasible solution could be found (as this excludes too many grid nodes around the runway). Hence, the complete-SIDs-based exclusions, as applied here, can make the problem infeasible for large values of d—an indication that this approach is indeed too conservative.

V. Conclusion

In this paper, we showed that our novel framework for computing DARs allows for the integration of the impact of atmospheric conditions on descent profiles (and, here, specifically, on environmentally friendly, fuel-optimized descent profiles) and the interaction with departing traffic—primarilly the first is a direct exploitation of our recent theoratical results and the implied one-to-one correspondence of paths and descent profiles. That is, we showed that further aircraft-specific and traffic-situation-dependent aspects can be integrated and allow for short computation times, bringing the concept of DARs closer to practical applicability.

We provided an experimental evaluation of our method for Stockholm Arlanda Airport. Comparing the resulting DARs when atmospheric conditions are taken into account with those without, we clearly confirm our expectation: they are crucial, and by ignoring them, we may obtain DARs that do not provide the guaranteed separation when aircraft follow the determined routes under real-world conditions. We can compute DARs for seven arriving aircraft entering the TMA within a 10-minute interval and with arrival paths of up to 12 edges within less than a second, providing a proof of concept—and as simply employing a different set of descent profiles does not inherently change the problem, we expect similar runtimes as, for example, detailed in [5], for (re)computing DARs for time intervals of ca. 30 minutes and larger values of λ, yielding operational feasible computation times for atmospheric-conditions-dependent DARs. In future work, we aim to provide a more detailed analysis.

In this paper, we did not provide a comparison with a reference scenario of the actually flown trajectories. However, this comparison is provided in detail in [3] and the main purpose of this paper is to highlight the opportunities provided by the new framework for computing DARs.

We also demonstrated an approach to integrate the interaction of DARs with departing traffic on the example of Stockholm Arlanda Airport. We ensure that arrival routes are separated from all departing traffic on SIDs within a along-route distance from the departure runway. The resulting DARs highlight the cost of this integration: the aircraft arrival paths are longer than those when departing traffic is ignored—and this cost increases for larger parts of SIDs to which we guarantee separation. Moreover, for high values of d the problem of computing DARs may become infeasible—particularly when we exclude an area close to the arrival runway. This indicates that our approach for the integration with departing traffic is indeed (as indicated in the introduciton) too conservative. Hence, for future work we aim to investigate two other approaches for integrating the interaction with departing traffic: excluding only those parts for DARs that are covered by departure routes that we expect to be used in the considered computation horizon, or—even less conservative—fixing the 4D trajectories of the departing aircraft and enforcing separation from all those aircraft for the aircraft following our DARs (this is similar to one of our methods [5] for handling interaction with aircraft that remain in the TMA from the previous DARs computation period). The latter approach could also handle vertical separation, which we expect to improve performance.

Additionally, we aim to explore alternative objective functions for computing DARs. While we guarantee that all aircraft can follow fuel-optimized descent profiles, these profiles exist for different lengths of the entry-point–runway paths–leading to different environmental impact–and the arrival routes can occupy different parts of the TMA–leading to noise impact on different regions. Thus, explicitly integrating these factors in the objective may further DARs' environmental benefits.

Finally, in future work we aim to work with both the spatial and temporal resolution: in the presented framework, the number of created paths allows us to include all path variables (with which we select paths for the arrival tree) in the MIP formulation. However, when we aim to use a finer grid, the number of feasible paths will grow accordingly. Additionally, while a finer temporal resolution would already be beneficial for the current grid size, it is certainly necessary for finer grids to properly reflect reality. To enable the finer resolutions, we will not be able to solve the complete MIP directly. Instead, we plan to investigate column generation, which will allow us to include only a subset of all path variables.

Moreover, in reality, we face uncertainty from various factors like weather. Additionally, operationally, we might not be able to enforce that aircraft adhere exactly to 4D trajectories, but that a certain temporal deviation must be allowed. When we optimize our DARs to specific arrival times, deviation from these times could even result in infeasible solutions. Hence, our problem of computing DARs would reach a new level of interest if we can provide robust solutions. Thus, we will consider robust optimization for DARs in the future.

ACKNOWLEDGMENT

This research was supported by grant 2022-03178 (OPT@ATM: Optimization Methods for Large Air Traffic Management Problems) from the Swedish Research Council (Vetenskapsrådet) and by the Swedish Transport Administration (Trafikverket) via the Friendly TMA project.

REFERENCES

[1] IATA, "Global air passenger demand reaches record high in 2024," https://www.iata.org/en/pressroom/2025-releases/2025-01-30-01/, 2025.
[2] ——, "Global outlook for air transport: Highly resilient, less robust," 2023.
[3] R. Sáez, T. Polishchuk, C. Schmidt, H. Hardell, L. Smetanová, V. Polishchuk, and X. Prats, "Automated sequencing and merging with dynamic aircraft arrival routes and speed management for continuous descent operations," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103402, 2021.
[4] R. Hajizadeh, T. Polishchuk, E. Rönnberg, and C. Schmidt, "A Dantzig-Wolfe reformulation for automated aircraft arrival routing and scheduling: Reducing computation times for solving the optimization problem," in *25th ICNS*, 2025, presentation of peer-reviewed extended abstract.
[5] ——, "A Dantzig-Wolfe reformulation for automated aircraft arrival routing and scheduling," 2025. [Online]. Available: https://arxiv.org/abs/2509.13065
[6] R. Dalmau and X. Prats, "Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations," *Transportation Research Part D: Transport and Environment*, vol. 35, pp. 62–71, 2015.
[7] T. Andersson, T. Polishchuk, V. Polishchuk, and C. Schmidt, "Automatic Design of Aircraft Arrival Routes with Limited Turning Angle," in *Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, 2016.
[8] D. Ivanescu, C. Shaw, E. Hoffman, and K. Zeghal, *Towards Performance Requirements for Airborne Spacing—A Sensitivity Analysis of Spacing Accuracy*, 2012.
[9] S. Choi, J. E. Robinson, D. G. Mulfinger, and B. J. Capozzi, "Design of an optimal route structure using heuristics-based stochastic schedulers," in *IEEE/AIAA 29th Digital Avionics Systems Conference*, Oct. 2010.
[10] S. Kamo, J. Rosenow, H. Fricke, and M. Soler, "Robust optimization integrating aircraft trajectory and sequence under weather forecast uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104187, 2023.
[11] J. Ma, D. Delahaye, M. Sbihi, P. Scala, and M. A. Mujica Mota, "Integrated optimization of terminal maneuvering area and airport at the macroscopic level," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 338–357, 2019.
[12] H. Hardell, T. Polishchuk, and L. Smetanova, "Arrival optimization with point merge in a dual-runway environment," in *SIDs 2023*, Sevilla, Spain, 2023.
[13] W. Liu, D. Delahaye, F. A. Cetek, Q. Zhao, and P. Notry, "Comparison of performance between PMS and trombone arrival route topologies in terminal maneuvering area," *Journal of Air Transport Management*, vol. 115, p. 102532, 2024.
[14] J. Prete, J. Krozel, J. Mitchell, J. Kim, and J. Zou, "Flexible, Performance-Based Route Planning for Super-Dense Operations," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2008.
[15] H. Hardell, E. Otero, T. Polishchuk, and L. Smetanová, "Optimizing air traffic management through point merge procedures: Minimizing delays and environmental impact in arrival operations," *Journal of Air Transport Management*, vol. 123, 2025.
[16] "OpenSky Network," https://opensky-network.org, Jul. 2019, accessed on December 2, 2018.
[17] SKYbrary, "Icao wake turbulence categories," https://skybrary.aero/articles/icao-wake-turbulence-category.
[18] ——, "Mitigation of wake turbulence hazard, wake turbulence separation minima," https://skybrary.aero/articles/mitigation-wake-turbulence-hazard, acc. Sept 22 ,2025.