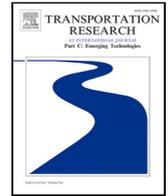


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Automated sequencing and merging with dynamic aircraft arrival routes and speed management for continuous descent operations

Raúl Sáez^{a,*}, Tatiana Polishchuk^b, Christiane Schmidt^b, Henrik Hardell^{b,c},
Lucie Smetanová^b, Valentin Polishchuk^b, Xavier Prats^a

^a Department of Physics - Aerospace Engineering Division, Technical University of Catalonia (UPC) - BarcelonaTech, Castelldefels, Barcelona, Spain

^b Communications and Transport Systems, ITN Linköping University (LiU), Norrköping, Sweden

^c Procedure Design Unit, Luftfartsverket (LFV), Norrköping, Sweden

ARTICLE INFO

Keywords:

Air traffic management
Traffic synchronization
Continuous descent operations
Required time of arrival
Trajectory optimization
Trajectory based operations

ABSTRACT

In this paper, we present a novel methodology to manage arrival traffic in terminal airspace. We define two areas around the airport, aiming to efficiently schedule incoming traffic. A four-dimensional (4D) trajectory negotiation/synchronization process between the air traffic control officer (ATCO) and the aircraft is performed in the pre-sequencing area, while the aircraft are still in the en-route phase of flight. On the other hand, in the dynamic-trajectories area, the ATCO, with the help of a ground support tool, generates dynamic arrival routes that automatically adapt to the current traffic demand. These arrival routes allow the aircraft to fly neutral continuous descent operations (CDOs, descents with idle thrust and no speed-brakes usage) and to ensure a separation throughout the arrival procedure. We choose a mixed-integer-programming approach to generate the arrival routes, while we formulate and solve an optimal control problem to generate a set of candidate CDOs per aircraft. Results show that, with a sufficient look-ahead time, it is possible to assign a required time of arrival (RTA) within each aircraft-arrival time window that would allow to efficiently schedule traffic even in the most challenging and dense scenarios. Besides improving efficiency of current operations in terminal airspace, the methodology presented in this paper could become a technical enabler towards an extended arrival manager (E-AMAN) with extended capabilities and, ultimately, to a fully deployed trajectory based operations (TBO) environment.

1. Introduction

Air transportation has been experiencing a continuous growth over the last decades and, although the 2020 setback might slow down this upward trend (Iacus et al., 2020), high levels of air traffic are still projected for the future. Despite the desirable growth of the global economy, the higher volume of traffic has also increased the environmental impact of aviation and the workload faced by air traffic control officers (ATCOs). This is especially evident in terminal maneuvering areas (TMAs), which are areas of controlled traffic surrounding one or more aerodromes. Specifically in large airports, TMAs are very congested and experience large levels of noise and emissions produced by aircraft. This leads to the need for research into methods for achieving a greener air transportation and for lessening the ATCO workload, while allowing for an increase in capacity.

* Corresponding author.

E-mail address: raul.saez.garcia@upc.edu (R. Sáez).

<https://doi.org/10.1016/j.trc.2021.103402>

Received 2 March 2021; Received in revised form 13 September 2021; Accepted 13 September 2021

Available online 7 October 2021

0968-090X/© 2021 The Authors.

Published by Elsevier Ltd.

This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

One of the solutions to reduce aircraft environmental impact in TMAs is the use of continuous descent operations (CDOs), which allow to reduce aircraft fuel consumption, pollutant emissions and noise nuisance, while bringing major economic benefits and without any adverse effect in safety (Erkelens, 2000; Warren and Tong, 2002; Clarke et al., 2004). CDOs consist of engine-idle descents optimized to the operating capability of the aircraft, resulting in different optimal trajectories depending on the aircraft model. Even more, each aircraft is *unique*, and presents variations in performance data with respect to the same aircraft type when issued from the assembly line. In this context, aircraft performance, as discussed in Airbus (2002), changes with aging, since the aerodynamic and engine characteristics typically deteriorate over time (and with maintenance actions). Furthermore, weather conditions also have an impact on the descent trajectory, since different wind, air temperature and/or pressure conditions will lead to different idle descent trajectories. Consequently, in practical terms, each individual flight performs a different CDO, which leads to a significant decrease of the vertical and temporal predictability of incoming traffic flows, with the resulting increase of ATCO workload. Thus, ATCOs usually apply larger separation buffers to handle CDOs, leading to airspace and runway-capacity losses that are not desirable in major TMAs, especially during peak hours.

For these reasons, CDOs are barely applied in busy TMAs. At a strategic level, altitude and speed constraints published in standard terminal arrival route (STAR) charts do not take into account the particular operating capability of each aircraft, thus, limiting the possibility of performing optimal descent trajectories. Furthermore, at a tactical level, ATCOs tend to issue *open-loop* instructions (i.e., radar vectoring) so as to maintain safe separation between aircraft and to maximize the throughput in these static routes (Sáez et al., 2020a). However, the duration of these instructions, as well as how and when the aircraft is going to rejoin its planned route, is not known by the flight crew. Thus, it is impossible for state-of-the-art flight management systems (FMSs) to predict the remaining distance to go and, therefore, to optimize the trajectory to achieve the most environmentally friendly descent profile. Finally, ensuring separation in STARs is also a very demanding task for the ATCOs, who could face situations with very high, or even unmanageable, workload. Thus, automation tools are often needed to help them ensure the necessary separation along the arrival routes and, especially, at the merge points.

New air traffic management (ATM) paradigms, such as the trajectory-based operations (TBO) defined by SESAR (SESAR Joint Undertaking, 2015) or NextGen (Federal Aviation Administration (FAA), 2019), aim to completely remove open-loop vectoring and strategic constraints on the trajectories by efficiently implementing a four-dimensional (4D) trajectory negotiation process to synchronize airborne and ground equipment with the aim of maximizing both flight efficiency and throughput. In this scenario, there would be no need for vectoring except for unforeseen or contingency situations. In this context, in a previous work developed by some of the authors of this paper (Sáez et al., 2020a), the authors applied an interim solution towards a full TBO concept. The current tromboning sequencing and merging paradigm was enhanced with a 4D trajectory negotiation and synchronization process, with the aim of maximizing CDOs. However, tromboning procedures have a limited number of pre-defined routes, and depending on the traffic demand the available routes might be sub-optimal.

In this work, we propose an optimization framework to automatically generate aircraft arrival routes in terminal airspace at a tactical level; these routes are computed for periods of time from 30 min to 1 h and assigned to incoming traffic. Furthermore, the generated arrival routes guarantee temporal separation, automatically adapt to the current traffic demand and enable the use of CDOs for all the arriving aircraft. The framework serves as a ground-support tool for ATCOs, which will both reduce their workload and the environmental impact of the operations.

A Mixed-Integer-Programming-based (MIP-based) approach for the generation of optimized arrival routes to enable CDOs for all arriving aircraft was first presented in Sáez et al. (2020b). However, scenarios with a high traffic demand or a challenging arrival traffic distribution led to infeasible solutions. In our most recent work (Polishchuk et al., 2020), we assumed aircraft could arrive within a time window at the TMA entry point, which could be achieved by adjusting the speed during the en-route phase. We developed a small proof of concept to test this new feature, where we were able to schedule all aircraft in a high-traffic load scenario. In this paper, we extend this work by, first, formulating and providing a detailed explanation of the concept of operations (CONOPS) to show the applicability of our framework in the current air traffic operations system. Furthermore, we apply our methodology to different scenarios with different traffic levels/distributions, we assess the sensitivity of our solution to the size of the entry-point time window and we analyze the effect that different traffic mixes have on the optimized arrival routes. By testing this variety of scenarios under several conditions, the aim is to capture the benefits of our concept in terminal airspace operations by comparing it with the current static arrival routes paradigm. Finally, this paper also contains the whole MIP formulation, which was not present in previous publications.

Besides improving the efficiency of current operations in terminal airspace, the methodology presented in this work could become a technical enabler towards an extended arrival manager (E-AMAN) with extended capabilities, in which the sequencing horizon would be extended into the en-route sector and airport's arrival management information will be shared with upstream sectors in real time by using a system-wide information management (SWIM) service. Similarly, our solution could also become a technical enabler towards an advanced 4D trajectory synchronization in line with the TBO paradigm.

2. Background and state of the art

Turn-constrained route planning for just a single path was considered in Krozel et al. (2006). Similarly, in Zhou et al. (2014), individual routes were generated in a TMA by taking into account the current weather. Then, in Visser and Wijnen (2001), single routes—where the focus was on the optimization of a single trajectory, but without considering the interactions with other aircraft in the vicinity—were generated in order to minimize the noise impact. However, in none of these works any temporal component

was taken into account: the routes were generated without taking into account the effect that real traffic—which involves having constraints on the separation between aircraft—could have on the generation of the arrival routes.

In [Chevalier \(2020\)](#), a simulated annealing approach was used to generate arrival trees in a TMA. However, while MIP methods (the one used in this work) are an exact approach, meta-heuristic methods like simulated annealing are not. As a result, in general, better—probably optimal—results will be obtained with MIP, but at the expense of higher computational times.

The scheduling of traffic along arrival routes was considered in [Choi et al. \(2010\)](#), where it was shown how throughput can be increased in congested terminal airspace by using scheduling algorithms different than the traditional first-come-first-served (FCFS) approach. In addition, other authors, see e.g. [Adacher et al. \(2003\)](#), studied sequencing models based on alternative graph formulations. Similarly, in [Le Ny and Pappas \(2011\)](#), mixed integer geometric programming was used to determine the optimal arrival order of aircraft at a metered fix in a stream-merging scenario. The merging of traffic flows was also studied in [Michelin et al. \(2011\)](#), where the optimization of the aircraft trajectories merging at a given fix was done in two steps in order to ensure a sufficient separation between the arrival flows. More recently, in [Samà et al. \(2019\)](#), various approaches were presented to integrate several modeling features in the aircraft scheduling problem—both for take-off and landing operations—with the aim of minimizing, for instance, the total travel time or the maximum delay. However, the location of merge points or constraints like the limit on a turn angle were not considered by any of these works, neither was the use of CDOs for the descents.

3D path arrival management (3D-PAM) ([Scharl et al., 2007](#)) is a concept that also focuses on arrival scheduling. One of its core tools is the en-route descent advisor (EDA) ([Coppenger et al., 2004](#)), which ensures an automatic time-based arrival metering to help ATC to deliver aircraft to a metering fix subject to a scheduled time constraint. Furthermore, the EDA tool can be used to assess the concept of tailored arrivals; in [Coppenger et al. \(2009\)](#), the feasibility of CDOs in congested terminal airspace was studied by using an advanced air and ground automation over data-link. However, these studies do not focus on the generation of arrival routes optimized to the current traffic demand, as they are more interested in having an accurate trajectory prediction in order to solve potential conflicts.

Differently from [Dalmau and Prats \(2015, 2016\)](#), whose authors studied speed variation for airborne delay to adjust the controlled time of arrival (CTA) of one flight, we use paths adjustment to coordinate the merge of multiple aircraft trajectories. Furthermore, the idea of assigning not only routes, but also times, is applied similarly in a non-aviation context, in dynamic vehicle routing ([Bertsimas and van Ryzin, 1991](#)) and dynamic scheduling ([Ouelhadj and Petrovic, 2008](#)).

Last but not least, noise also plays an important role in the generation of both arrival and departure routes in TMA, and extensive research has been made in recent years about this topic ([Visser and Wijnen, 2001](#); [Ho-Huu et al., 2020](#)). For instance, in [Ho-Huu et al. \(2020\)](#) the authors proposed a multilevel optimization framework to design departure routes. The approach was divided in two steps: first, both noise impact and fuel consumption were taken into account in order to obtain a set of optimal routes; then, routes from this set were selected and the departing flights were distributed among these routes.

3. Concept of operations

In this section, we focus on the concept of operations (CONOPS) proposed in this work. In Section 3.1, we describe the CONOPS, while in Section 3.2, we provide a discussion regarding the applicability of this CONOPS as a technical enabler for several air traffic operations solutions. Furthermore, we enumerate several factors directly affecting this CONOPS.

3.1. Concept of operations: General description

[Fig. 1](#) depicts a simplified scenario used to illustrate the problem addressed in this work. Let us assume that several aircraft have planned to land in a given airport. The strategy proposed to safely bring these aircraft from the cruise phase until the runway by following a CDO is managed, in a general way, in the following regions:

- (1) **Pre-sequencing area:** the (data-link) communication between ATCOs and aircraft is established in the pre-sequencing horizon, where the sequencing requirements are negotiated: a required time of arrival (RTA) at the dynamic-trajectories horizon and a specific route in the dynamic-trajectories area. Aircraft follow the published routes in this pre-sequencing area, but in order to meet the negotiated RTA, can adapt their speed, vertical profile and, to some extent, even the route—by stretching or shortening, within some bounds, the route followed in the pre-sequencing area.
- (2) **Dynamic-trajectories area:** in this area, arrival routes linking each of the entry points with the runway will be dynamically generated by the ATCO ground-support system. Aircraft are requested to follow these routes, which may merge in some intermediate merging fixes.
- (3) **Final merging fix:** all traffic flows will eventually merge at the final merging fix, from where they will proceed to the landing runway.

A more detailed explanation of this CONOPS is given in the remainder of this subsection. The aim is to better describe the interactions and roles of both aircraft and ATCOs in the pre-sequencing and dynamic-trajectories areas, as well as the function of these areas.

As it has been aforementioned, the arriving traffic flows enter the pre-sequencing area through several entry points, located in the pre-sequencing horizon (E_a , E_b and E_c in [Fig. 1\(a\)](#)). Once the aircraft enter the pre-sequencing area, they follow the published route until the entry point of the dynamic-trajectories area, located in the dynamic-trajectories horizon (T_a , T_b and T_c in [Fig. 1\(a\)](#)). At

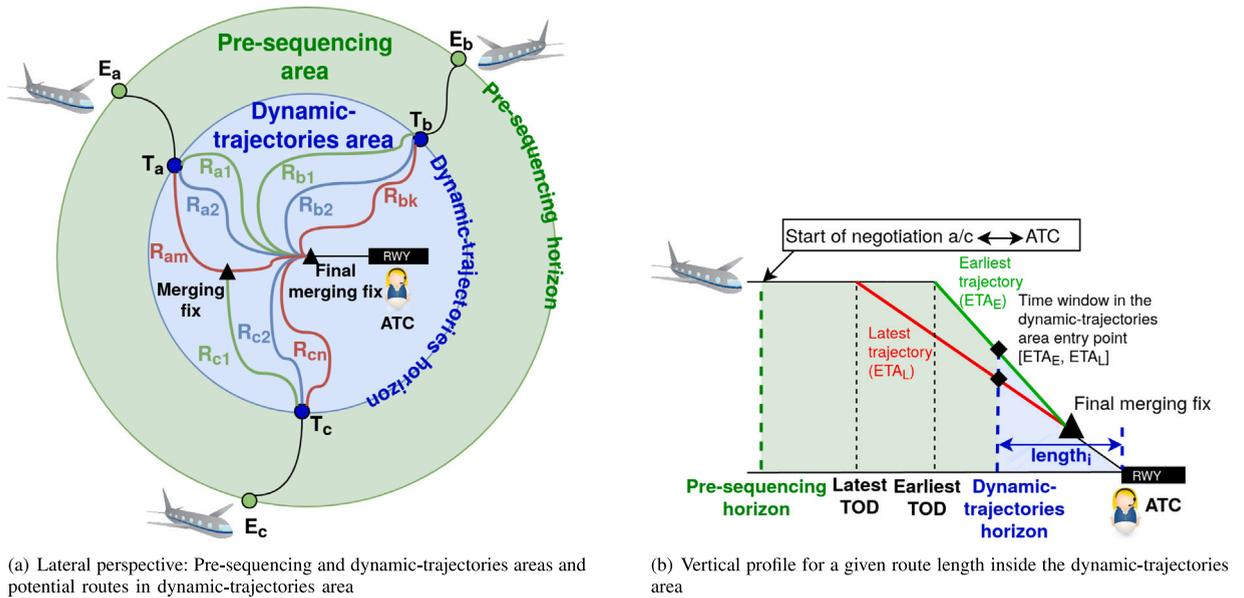


Fig. 1. Concept of operations: trajectories.

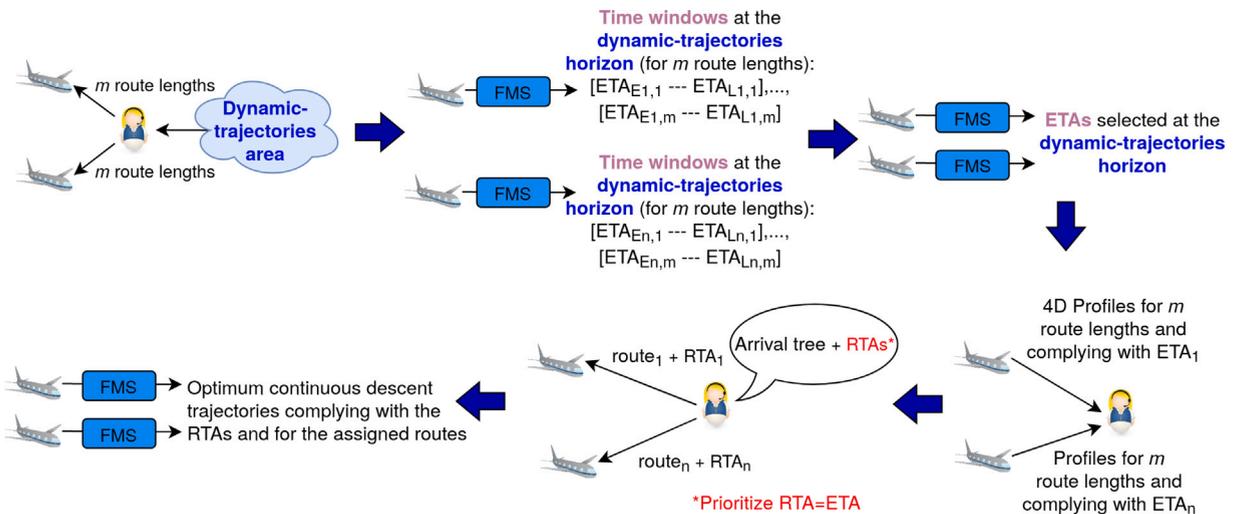


Fig. 2. Concept of operations: Required time of arrival (RTA) and route negotiation process for n aircraft and m routes.

the same time, they initiate an RTA-and-route-negotiation/synchronization process with the ATCO (or ground-automated system), as shown in Fig. 2.

This RTA-negotiation/synchronization process starts with the aircraft being requested to compute an estimated time of arrival (ETA)—as well as the 4D trajectory complying with this ETA—at the entry point of the dynamic-trajectories area for a given set of pre-defined route lengths inside this area. In order to do that, aircraft have to compute first the time window at the dynamic-trajectories horizon. This time window is the difference between the earliest and the latest time of arrival at a given fix (in this case, the entry point of the dynamic-trajectories area), corresponding to ETA_E and ETA_L in Figs. 1(b).¹ and 2. These times of arrival depend on aircraft performance, flight envelope and weather conditions, and are assumed to be computed by the FMS of the different incoming aircraft by considering the earliest and latest trajectories (while following the published routes joining the two horizons).

¹ The top of descent for the earliest trajectory is further (or closer to the final merging fix) than the top of descent for the latest trajectory, as aircraft can fly at higher speeds during the cruise phase (Fig. 5).

Once all time windows—of aircraft arriving within a limited time interval—are computed, the aircraft FMS selects a feasible ETA at the dynamic-trajectories horizon. The method we propose to do this selection is explained in Section 4. Once the ETA is selected, aircraft compute and downlink a set of 4D profiles (i.e., descent trajectories complying with a specific ETA at the dynamic-trajectories horizon and flying a given route length within the dynamic-trajectories area) to the ATCO.

The automated system on-ground, after receiving all the 4D profiles, assigns, first, an RTA² at the dynamic-trajectories horizon. This RTA, if possible, will be equal to the ETA. Then, the automated system on-ground generates an optimal arrival tree in the dynamic-trajectories area, which is composed of a single arrival route from each of the entry points of this area to the final merging fix. As a result, each aircraft is assigned a single 4D profile to execute (with a final route length chosen from the pre-defined set of lengths inside the dynamic-trajectories area).

While an RTA equal to the ETA might work for low to average-traffic scenarios, it might not be enough for more congested airspaces surrounding larger airports (Sáez et al., 2020b). As the traffic increases, the separation between aircraft at the dynamic-trajectories horizon is reduced, and the generation of the arrival tree becomes more challenging. Not only the amount of traffic could be a problem, but also the arrival distribution at the dynamic-trajectories horizon. In general, it is not desirable that large aircraft clusters arrive in a short period of time, as it might be impossible to schedule them effectively and safely. Furthermore, some aircraft arrival distributions, which *a priori* might seem good and easy to schedule, in the end might lead to an infeasible scenario where it is impossible to schedule all arrival traffic. Even if the entry times at the dynamic-trajectories horizon for each aircraft are safely separated, when taking the dynamic-trajectories area as a whole (i.e., by taking into account all its entry points), it might be impossible to generate an arrival tree that safely schedules all arriving aircraft. This *goodness* in the arrival distribution is indeed a very interesting topic and will be investigated in future work.

To be able to schedule all arriving aircraft even in the most challenging scenarios, we propose the following: ATCOs, when communicating with the aircraft in the pre-sequencing horizon, could request them to gain or lose a given amount of time in the pre-sequencing area. Thus, as depicted in Fig. 2, ATCOs, when generating the arrival tree, may eventually assign an RTA different from each aircraft's ETA. As it will be shown in Section 6, by requesting aircraft to arrive a few minutes earlier or later at the dynamic-trajectories horizon, the scheduling of the arrival traffic could be enhanced and applied in dense traffic scenarios. This time could be gained/lost in the pre-sequencing area.

The generated arrival tree will change during the day (new time intervals will be considered), as it will automatically adapt to the actual traffic demand. The objective is that the aircraft know the RTA at the dynamic-trajectories horizon and the current arrival-tree configuration (and, thus, the distance to go) while still in cruise, somewhere in the pre-sequencing area. Therefore, the FMS on-board (with RTA capabilities) can compute a neutral descent trajectory (i.e., idle thrust and no speedbrakes usage) that complies with the RTA and follows the corresponding route of the optimal arrival tree requested by the ATCO. Consequently, the flight sequence should be generated fast enough within the distance between the pre-sequencing horizon and the dynamic-trajectories horizon, with the aim of ensuring aircraft know the profiles they have been assigned before reaching the top of descent (TOD).

Furthermore, the distance between horizons is also affected by the amount of time to be gained/lost in the pre-sequencing area. In Delgado and Prats (2012), it is shown that for typical cost-index values between 30 kg/min and 100 kg/min, aircraft can gain or lose between 1.2 s/NM and 4 s/NM by managing their speed. For instance, if we consider the worst-case scenario—1.2 s/NM—and a pre-sequencing horizon located at 250NM from the dynamic-trajectories horizon, aircraft would be able to gain/lose up to 5 min in the pre-sequencing area. As it will be shown in Section 5, these are the conditions considered in the case-studies defined in this paper.

The amount of time that can be gained/lost in the pre-sequencing area depends on various factors, such as the aircraft model or the cruise flight level. In addition, the effects of wind should also be considered (Delgado and Prats, 2013). Finally, another option to even gain/lose a higher amount of time would be to shorten/stretch the path in the pre-sequencing area. However, this strategy has been left out of the scope of this paper.

Similarly, the location of the dynamic-trajectories horizon can be also modified depending on several factors. The farther the horizon is located, the larger the dynamic-trajectories area, which would allow the ATCOs to generate a set of dynamic arrival routes in a larger region. Consequently, arriving aircraft would be sequenced in optimal arrival routes from an earlier stage. However, a larger dynamic-trajectories area comes at the expense of a higher computational load. As a result, the time needed for the RTA-and-route-negotiation/synchronization process would increase, which will ultimately have an effect on the location of the pre-sequencing horizon too.

3.2. Concept of operations: Technical discussion

In this work, we focus only on the trajectory plan generated by the FMS. Therefore, we do not deal with how this plan will be flown by the aircraft, which is the task of the guidance system of the FMS and which is out of the scope of this paper. Once the trajectory plan is generated, this system will be responsible of adapting the aircraft trajectory in order to meet the assigned RTA and minimize uncertainty. For instance, a guidance system like the one proposed in Garrido-López et al. (2009) could be used to minimize wind uncertainties by combining elevator and throttle actuations to control ground speed and efficiently match a prescribed ground speed law, thus improving the predictability of the operation.

² If the aircraft FMS is not equipped with the RTA capability, speed advisories will be given instead.

In this CONOPS, aircraft trajectories are computed by on-board systems (such as an FMS), and not by ground-based systems. This improves the accuracy of the generated trajectories as aircraft systems have access to much more accurate aircraft performance data than ground-based systems. This aligns with the initial 4D (i4D) concept, which will be integrated as a new feature in current FMSs. With i4D, the aircraft FMS will be able to downlink the whole 4D trajectory flight plan with the associated predictions—e.g. predicted aircraft weight, as well as the predicted horizontal and vertical speed—of up to 128 waypoints, so called extended projected profile (EPP) (Bronsvooort et al., 2016), via automatic dependent surveillance-on contrast (ADS-C). Similarly, the FMS will also be able to receive a direct up-link of ATC clearances upon crew confirmation. Such data-link enhancements will provide seamless exchanges that are key to the success of i4D operations. Furthermore, another key element of i4D operations is also to improve trajectory predictions using time constraints, being the assignment of RTAs a key criterion for pilots to follow.

The trajectories computed by the FMS are used to obtain the arrival time window at the dynamic-trajectories horizon. However, this window is not something fixed and depends on several factors, such as aircraft performance, weather conditions, etc. Specifically, there are two main factors that could be changed in order to modify the width of the time window. The first one is related to the type of CDO performed. Neutral CDOs are the ones considered in this work, which involve descents with idle thrust and no speedbrakes. On the other hand, non-neutral or powered CDOs could also be considered, allowing both the use of thrust and speedbrakes, which leads to wider time windows, but at the expense of increasing the fuel burn and/or noise nuisance.

The other factor affecting the width of the time window is related to the moment the available route lengths are notified to the aircraft. Since the RTA negotiation will begin (at the earliest) at the pre-sequencing horizon, the arrival time window will be wider the further this horizon is from the dynamic-trajectories horizon. Besides this distance to the RTA metering point, also the window width will be much wider if the aircraft has not yet initiated the descent: while in cruise, not only the aircraft could easily accelerate or decelerate to achieve earlier or later times of arrival, but it could also adapt the position of the TOD. Several studies have dealt with the assignment of RTAs (and the quantification of the feasible time window) when the aircraft is still in cruise, well before the TOD. For instance, in Takeichi and Inami (2010), the authors quantified the feasible time window that could be achieved by either adding or omitting waypoints in order to stretch or reduce the flight-path length. Once the descent has already started, ATCOs could still assign (or update) RTAs, but the possibilities to fly neutral CDOs are reduced. This situation was addressed in Dalmau and Prats (2017), where it was shown that noticeable time windows are still possible while aircraft are already flying a neutral CDO.

The solution presented in this paper is expected to be a technical enabler towards an E-AMAN (SESAR Joint Undertaking, 2019) with extended capabilities. In short, the aim of E-AMAN is to alleviate pressure in the terminal airspace by allowing to sequence arrival traffic much earlier than is currently the case. The current AMAN horizon is extended from the airspace close to the airport to further upstream, thus, allowing a smoother traffic management. Controllers in the upstream sectors (e.g., en-route sectors), which may be in a different control center or even in a different functional airspace block (FAB), obtain system advisories to support an earlier pre-sequencing of aircraft. E-AMAN is supported by sharing the airport's arrival management information with upstream sectors in real time. All parties share the same information using a SWIM service. In this work, the pre-sequencing horizon would correspond to the E-AMAN horizon. Furthermore, we extend the capabilities of E-AMAN by introducing an additional region surrounding the airport (i.e., the dynamic-trajectories area), which would be similar to the current TMAs—where arrival routes are static—but with additional functions and a variable size. In the CONOPS that we propose, as discussed in Section 3.1, routes inside this region would be in constant evolution, automatically adapting to the current traffic demand (which would be divided into several time intervals). Previous works considering aircraft scheduling and E-AMAN algorithms under the TBO concept include Toratani et al. (2018), Toratani (2019), focusing on optimizing simultaneously the aircraft arrival sequence and the trajectory. However, to the best of the authors' knowledge, no current work considers the generation of dynamic arrival routes together with E-AMAN algorithms.

As detailed in this section, our solution aligns with some of the key concepts and technologies defined by SESAR and it could be applied with existing aircraft equipment. However, there are some concepts which are still under development or deployment (e.g. EPP functionality), which may mean that still not all aircraft (and airports) have the required functionalities to make our solution work. Furthermore, aircraft with old systems on-board—for instance, an aircraft FMS without the RTA functionality—may also cause a problem for the application of our solution.

4. Methodology

In this section, we describe the workflow of our system (depicted in Fig. 3) and we give an overview of the methodology followed. Our system consists of the following steps:

- (1) **Entry points identification:** from each arriving aircraft's trajectory, the entry points to the pre-sequencing and dynamic-trajectories area are identified—which should be static and published in the corresponding Aeronautical Information Publication (AIP). As soon as aircraft enter the pre-sequencing area, the RTA-and-route-negotiation process starts.
- (2) **Definition of route lengths in the dynamic-trajectories area:** in this area, no static routes are published, so it is the task of the ATCO (with the help of a ground support tool) to dynamically generate the arrival routes. In order to ensure the traffic can be properly managed by the ATCO and/or ground support tool, and in order to ensure the routes are flyable by the aircraft (and the pilot), it can be considered that the number of potential routes in the dynamic-trajectories area is finite. Furthermore, the departing traffic, potential obstacles and other kind of limitations, such as separation constraints, could also limit the number of potential arrival routes. Thus, as depicted in Fig. 1(a), an aircraft entering the dynamic-trajectories area through point T_a could follow any single route out of m (finite) potential routes until the final merging fix. As a result, the ground support tool defines a finite number of route lengths. Then, each route is divided into several fixed distance segments.

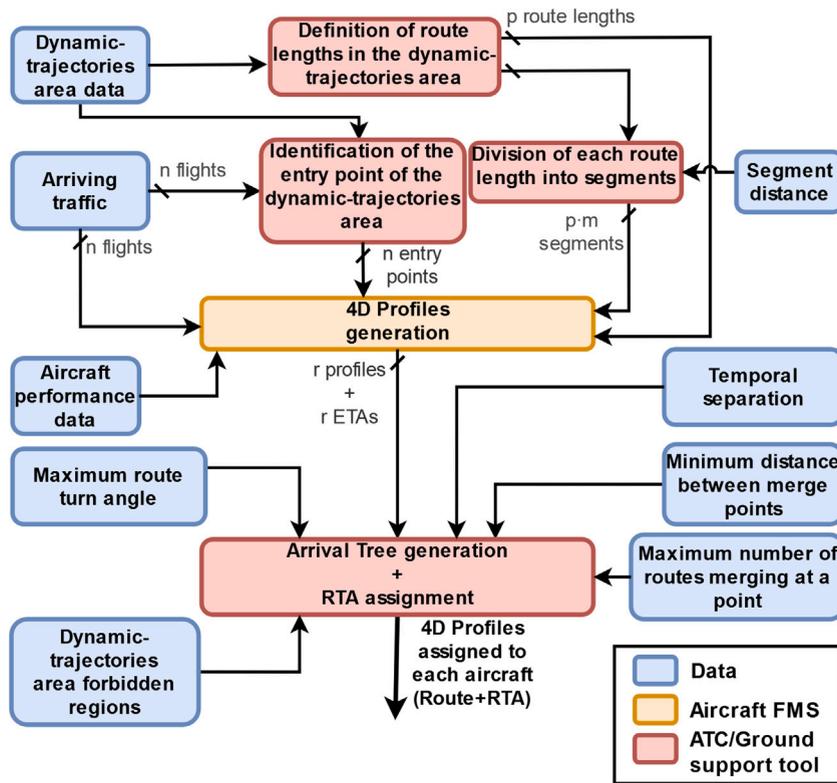


Fig. 3. Dynamic scheduling workflow.

- (3) **Generation of earliest and latest trajectories + ETA computation at the dynamic-trajectories horizon:** the ATCO requests from each arriving aircraft an ETA at the dynamic-trajectories horizon, as well as the 4D profile complying with this ETA. Aircraft compute, first, their corresponding neutral time window at the entry point of the dynamic-trajectories area for each route length inside this area. The aircraft FMS accesses the current aircraft performance data and computes the earliest and latest neutral trajectories (i.e., neutral CDOs) for each route length. Then, the FMS selects an ETA within the neutral time window and downlinks it to the ATCO, together with the 4D profile for each available route length and complying with this ETA. Furthermore, the time required to cover each segment of the corresponding route is also downlinked to the ATCO. The ETA selection is done as follows: let us assume m route lengths are defined inside the dynamic-trajectories area, depicted as $D_1, D_2, D_3,$ and D_m in Fig. 4. For each route length, the earliest and latest trajectories are computed in order to obtain a time window at the dynamic-trajectories horizon. We can observe that some of the possible times of arrival are shared by trajectories corresponding to different route lengths, corresponding to the purple time interval in Fig. 4. The idea is that each aircraft FMS selects an ETA at the center of this interval, which will be attainable by the aircraft no matter which route is chosen. In case there are no feasible (i.e., overlapping) sets of arrival times common to all route lengths, longer lengths are discarded first. Thus, longer route options are removed from the set of potential routes until there is an overlap where a feasible ETA can be selected.
- (4) **Generation of the arrival tree and RTA assignment:** the ATCO, supported by a ground tool, generates an arrival tree in the dynamic-trajectories area—which takes into account a set of operational constraints—and then assigns to each aircraft an RTA at the dynamic-trajectories horizon. While preferably the assigned RTA would be equal to the ETA, for scenarios with a larger amount of traffic, aircraft might be requested to arrive a few minutes earlier or later at the dynamic-trajectories horizon. Consequently, ATCOs will consider an offset in the time window, thus, shifting the time windows depicted in Fig. 4 to earlier or later times. In such a case, an RTA different from the ETA would be assigned, which would involve aircraft adjusting both the speed and the vertical profile in order to meet the RTA. As discussed in Section 3.1, this time will be gained/lost by the aircraft in the pre-sequencing area.

In Appendix A, we present the methodology used to generate the (optimal) trajectories for the validation scenarios of this paper (simulated traffic), based on the semi-analytic trajectory optimizer proposed in Park et al. (2017). The underlying computations and theory behind step 3 of this section are explained (corresponding to the yellow box of Fig. 3—4D profiles generation). In real life, these trajectories would be generated by an advanced functionality of the FMS of each aircraft, equipped with the RTA functionality.

Two examples of trajectories that can be obtained with the semi-analytic trajectory optimizer are shown in Fig. 5, computed for an Airbus A320 in international standard atmosphere (ISA) conditions and no wind. Fig. 5(a) corresponds to the earliest trajectory

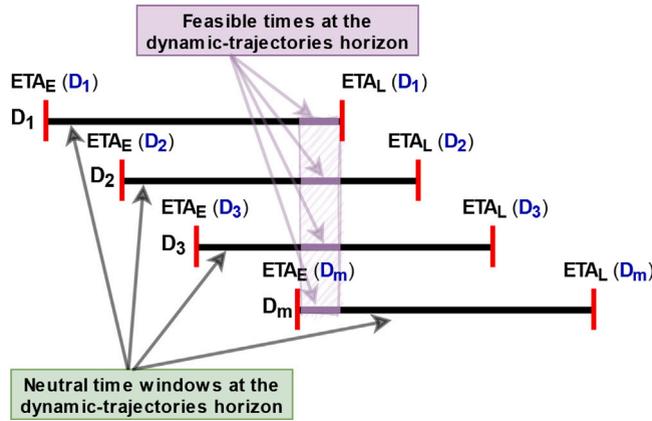


Fig. 4. Feasible times at the dynamic-trajectories horizon for profiles corresponding to different route lengths inside the dynamic-trajectories area.

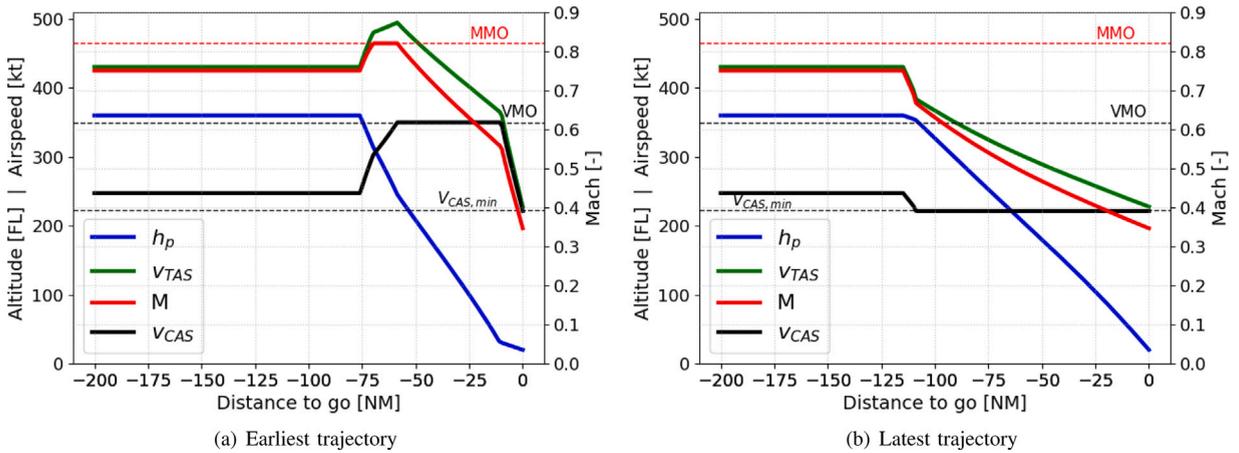


Fig. 5. Illustrative earliest and latest trajectories for an A320 in international standard atmosphere and no wind conditions.

and Fig. 5(b) to the latest trajectory; both trajectories are computed until the final merging fix. In both plots, we show the evolution of true airspeed (V_{TAS}), pressure altitude (h_p), Mach (M) and calibrated airspeed (V_{CAS}) as a function of the distance to go.

Fig. 5(a) depicts a trajectory flown with a high cost index³ (999), since the objective is to arrive as early as possible to the final merging fix. On the other hand, Fig. 5(b) depicts the same aircraft with cost index set to 0, thus arriving as late as possible to the same point. In both cases, the cruise speed is not modified; however, the TOD differs. For the earliest trajectory, the TOD is closer to the end of the trajectory, while for the latest trajectory it is the opposite, the TOD is further from the end of the trajectory. In the earliest case, the aircraft follows a maximum operative Mach/maximum operative calibrated airspeed (MMO/VMO) profile after starting the descent (with a fast deceleration segment at the end to fulfill the final point constraints), while in the latest case the aircraft decelerates to the minimum calibrated airspeed ($v_{CAS,min}$) when the descent starts. For both cases, neutral CDOs are considered, so no additional thrust nor speed-brakes are allowed during the descent.

In reality, aircraft could fly with a slightly negative cost index, thus, leading to later times of arrival than those obtained when flying with a cost index equal to 0 (as considered in this work). In such a case, aircraft would be flying at a range of speed between the lowest selectable speed and the maximum lift-to-drag ratio speed. However, these speeds are hardly ever chosen by aircraft operators in normal operations. In such a case, aircraft would be flying at unstable speeds (in the 2nd regime of the power curve or region of reversed command) and, although this is manageable by the auto-pilot, it usually involves undesired continuous actions on the aircraft throttle in order to keep the requested speed.

In Appendix B, we present the grid-based MIP formulation used to generate the optimal arrival tree that complies with all the operational constraints that will be described hereafter. For computing the arrival trees, we overlay the dynamic-trajectories area with a square grid, and use both axis-aligned and diagonal edges as possible components of the arrival tree. This yields a discrete

³ The cost index, which is a parameter chosen by the airspace user, reflects the relative importance of the cost of time with respect to fuel costs (Airbus, 1998).

set of possible lengths of any entry-point-runway path (from the shortest grid-path from entry point to runway to the longest edge-disjoint grid path from entry point to runway). In particular, we impose an upper bound on these paths. For these discrete paths length, we compute the corresponding neutral CDO profiles by solving an optimal control problem.

Our computation of arrival trees uses a MIP formulation. We build a tree that has the runway as its root, and the entry points as leaves. Then, the speed profiles obtained from the CDO trajectories are an input to our MIP. We enforce that aircraft following the computed path along the arrival tree from the entry point to the runway are temporally separated. Furthermore, we also ensure that each aircraft uses the speed profile that corresponds to the route length of the entry-point-runway path of the computed tree. By using further constraints, we enforce operational constraints (1)–(4):

- (1) *Temporal separation of all aircraft along the routes:* each pair of aircraft (with the leading and trailing aircraft being of wake-turbulence category A and B, respectively) that shares a given subroute is separated by a temporal distance of at least $\sigma_{A,B}$ along this subroute. Thus, if all aircraft comply with the planned time of arrival at the entry point of the dynamic-trajectories area, they will be safely separated along the arrival routes.
- (2) *Limitation of the number of routes merging 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 (Prete et al., 2008). We assume that at most 2 routes can merge at a point in the dynamic-trajectories area.
- (3) *Merge points separation:* operational constraint (2) could be circumvented by placing merge points arbitrarily close to each other. As a result, this would still result in a very small zone with a very 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 (Prete et al., 2008).
- (4) *No sharp turns:* because of aircraft dynamics and limitations in the bank angle—which usually does not exceed 30 degrees for most aircraft under nominal conditions for reasons such as safety and passenger comfort—there is a limit on the turn angle for each arrival route. Consequently, any turn from a route segment to the consecutive route segment must be larger than a given minimum track change angle θ .
- (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.

5. Experimental setup

In this section, we present the experimental setup used to illustrate the CONOPS and methodology proposed in this paper. Stockholm-Arlanda-airport TMA has been chosen for this study so, from now on, and for the sake of simplicity, we will refer to the dynamic-trajectories area as the TMA. A hypothetical pre-sequencing area with a radius of 250NM is defined around the airport. As soon as aircraft enter the pre-sequencing area, and after a pre-negotiation with the ATCO, RTAs at the TMA entry points and routes along the optimal arrival tree are assigned. In this scenario, aircraft are assumed to fly straight to the TMA entry point, after which they follow the assigned route inside the TMA.

In Section 5.1, we describe Arlanda airport and its available runway configurations, as well as its current TMA. In Section 5.2, we detail the input parameters used for the MIP, while in Section 5.3, we focus on the trajectory generation: the required input parameters and the set of profiles generated. Then, in Sections 5.4 and 5.5, we focus on the theory behind the computation of a series of key performance indicators, used to evaluate the efficiency gain of our solution. Finally, in Section 5.6, we detail the set of experiments carried out in this work. Furthermore, we detail the analysis performed to assess the sensitivity of our solution to several parameters.

5.1. Arlanda airport and stockholm TMA

Arlanda airport has three runways: Runway 1 (01L/19R), Runway 2 (08/26) and Runway 3 (01R/19L). Runways 1 and 3 are parallel runways that can be operated independently of one another and, when used at the same time, allow the airport to handle simultaneous take offs and landings.

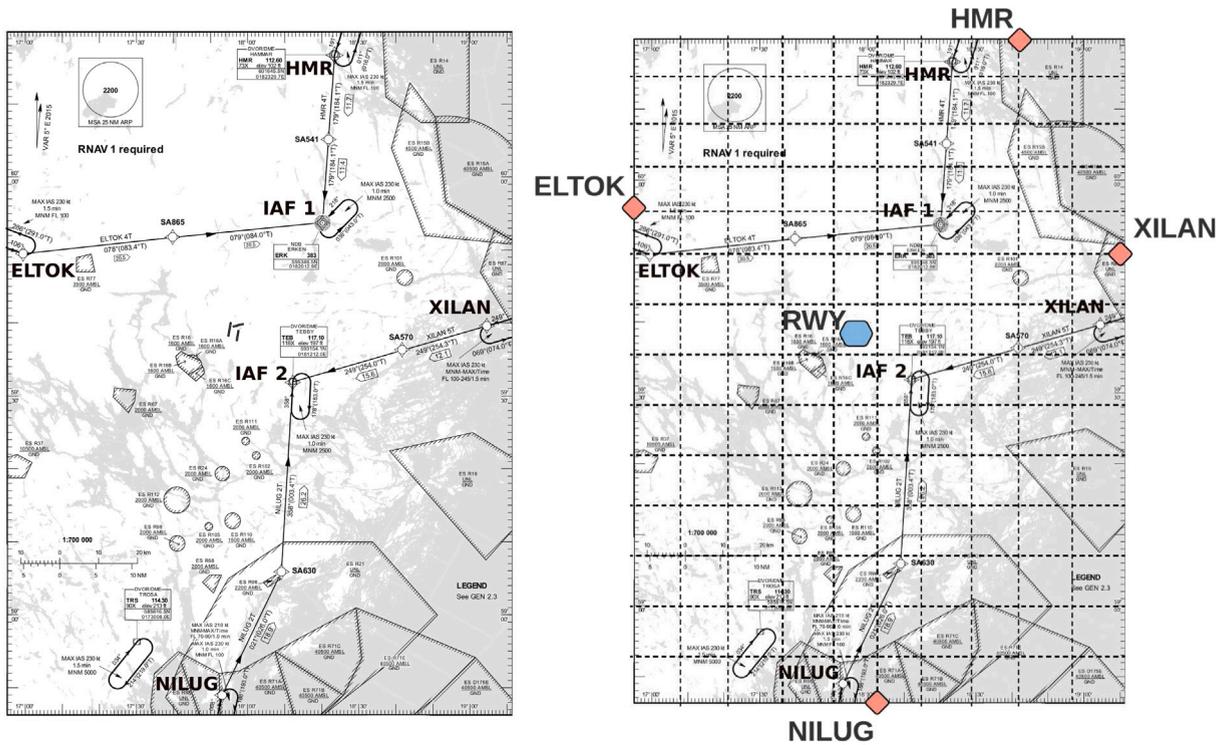
Arlanda arriving and departing traffic is managed in the Stockholm TMA. Currently, there are 4 TMA entry points in Arlanda, corresponding to the 4 STAR starting points: HMR (north), XILAN (east), NILUG (south) and ELTOK (west). These points are shown in Fig. 6(a), which corresponds to the RNAV (area navigation) STAR for runway 26.

5.2. MIP configuration parameters

In this work, the objective is to generate an optimal arrival tree that adapts to the current aircraft demand. Therefore, none of the STARs designed for Arlanda (such as the one shown in Fig. 6(a)) will be considered. However, we keep the location of the TMA entry points, from where each of the routes that make up the arrival tree will start.

We overlay the TMA of Stockholm Arlanda airport with a 15×11 grid (Fig. 6(b)), which automatically ensures a separation of about 6NM (corresponding to parameter L , described in the MIP formulation in Appendix B). Using finer grids would make the problem computationally too expensive.

Apart from the location of the TMA entry points and the grid used to overlay Arlanda's TMA, the following input parameters needed by the framework have been set:



(a) Arlanda RNAV(DME/DME or GNSS) STAR for RWY 26 and TMA entry points (source: Swedish AIP)

(b) 15x11 grid over Arlanda TMA

Fig. 6. Overlaying Arlanda TMA with a 15 × 11 grid.

- **Runway orientation and location:** in this work, we assume a single runway. Using parallel runways is supported by our framework immediately, as we snap the runway location to the closest grid point. If an airport has several, non-parallel runways, we may compute a merge tree per runway and impose constraints on how they may interact (e.g., on intersections). However, this functionality is not yet implemented in the current version of our framework.
- **Aircraft time at the TMA entry point:** for each arriving aircraft, an entry time at the TMA entry point is required, corresponding to the ETA selected by each aircraft FMS.
- **Aircraft speed profiles:** for each arriving aircraft, the corresponding speed profile for different route lengths is required. These speed profiles determine the time each aircraft spends to cover each TMA segment.
- **Temporal separation (as a function of aircraft category):** a separation between aircraft has to be ensured in order to maintain the safety of the operation. The separation will be defined by taking into account the wake turbulence categorization proposed by ICAO (SKYbrary, 2020), which classifies aircraft into three categories depending on their maximum take-off mass. In this work, the minimum time separation between aircraft that ICAO proposes according to that categorization (ICAO, 2016) will be used: 3 min of time separation will be considered for light aircraft following medium or heavy aircraft, while 2 min of time separation will be considered otherwise. As reported in a recent project (Flache, 2020), wake turbulence effects could even be noticeable during the cruise phase of flight, so it is important to ensure this separation in the whole TMA in order to avoid potential wake turbulence hazards.
- **Number of differing edges (U):** as defined in Appendix B, the parameter U is used in the MIP formulation to ensure a consistency in terms of number of differing edges between arrival trees during consecutive time periods (a small sensitivity study of the resulting trees for several U parameter values was conducted in Hardell et al., 2020a). Depending on the experiment, and as shown in Table 1, the value for U has been set to either 20 or 30.
- **Minimum angle of outgoing edge (θ):** as detailed in Section 4 and Appendix B, in order to avoid sharp turns in the arrival routes, a minimum angle θ for each outgoing edge is needed. For all the experiments, the value for θ has been set to 135 degrees.
- **Time-window offset at the TMA entry point:** we define a time-window offset at the TMA entry point, which will be different for each experiment, as detailed in Section 5.6.
- **Parameter for objective function (β):** the objective function is a convex combination of the paths lengths (β) and the tree weight ($1 - \beta$). We set this parameter to 0.1 in all experiments in order to prioritize the minimization of the sum of paths lengths.

Table 1
Design matrix for the experiments.

Case study	Day & Time	Runway	Number of aircraft	Time-window offset (minutes)	Traffic mix (% light a/c)	Number of differing edges (U)
Low-traffic	29/12/2018 - 11h to 12 h	01L	8	$\pm 0, \pm 1$	0	–
Average-traffic	04/10/2017 - 15 h to 16 h	19R	22	$\pm 0, \pm 1$	0	20
				± 2	0 20	20
				± 3	0, 20	20
				± 4	20	20
High-traffic	16/05/2018 - 05 h to 06 h	01L	30	$\pm 0, \pm 2$	0	20
				$\pm 1, \pm 3, \pm 4$	0	30
				± 5	0, 20	20

5.3. Trajectory input data and profiles computation

The flight-traffic data needed to carry out the experiments was mainly obtained from the OpenSky Network ([OpenSky Network, 2019](#)). This network contains raw-trajectory data stored in a large historical database, which is obtained thanks to the automatic dependent surveillance-broadcast (ADS-B) technology. Two types of data-sets are available in OpenSky: OpenSky states vectors, which provide a very-high-resolution data of the actual flown trajectory, and OpenSky tracks, which contain a subset of the recordings from the states data. We mainly used OpenSky states for our experiments, due to its higher resolution. However, there were two flights used in the experiments which lacked an ADS-B transponder. Thus, there was no data available in OpenSky. For those specific cases, we used data from Eurocontrol's demand data repository (DDR) ([Eurocontrol, 2017](#)), which is less accurate than OpenSky, especially in areas close to the airport.

In order to generate optimal CDOs, we need to model the different aircraft performance functions such as idle thrust (T_{idle}), drag (D), cruise fuel flow (f_{cr}), idle fuel flow (f_{idle}) and their respective parameters ([Appendix A](#)). In this work, we adopt the aircraft performance model defined in EUROCONTROL's base of aircraft data (BADA) v4 ([Poles et al., 2010](#)).

The upper bounds on the speed limits (VMO and MMO) and the minimum speed $v_{CAS,min}$, which are aircraft-dependent, are also obtained from the BADA v4.2 aircraft performance files. The maximum descent gradient (γ_{min}) is set to -7° for this paper. In the case the aircraft model does not correspond to any of the BADA models, we use a similar aircraft in terms of performance and dimensions.

The initial state of the aircraft, consisting of the cruise altitude and cruise speed, is obtained directly from OpenSky in the position where the aircraft enter the pre-sequencing area. In addition, the corresponding entry point to the TMA is identified from each aircraft's trajectory. The terminal constraints defined in the trajectory-optimization formulation in [Appendix A](#) are the following: the final speed, v_f , corresponds to the V_{F1} speed, which is the speed at which flaps/slats configuration 1 is typically deployed; the final altitude, h_f , is equal to the altitude of a hypothetical FAP (final approach point), which would correspond to roughly 2,000 ft; and the distance to go, s_f , corresponds to the distance to the FAP, which depends on the route length. The final merging fix in our CONOPS ([Section 3.1](#)) would be located at a given distance before the FAP.

For any given flight, the number of trajectories generated depends on the number of possible routes the aircraft can fly. In this case, route lengths within the TMA from 30 NM (corresponding to the minimum route length within the grid) to 108 NM are considered, with each route split into several segments of constant length equal to 6 NM. For instance, a 30 NM route inside the TMA would be split into 5 segments (i.e. [0–6, 6–12, ..., 24–30]). We chose the set of route lengths in this experiment according to the grid size (as described in [Section 5.2](#)). The lower bound stems directly from the grid (it is not possible to have route lengths lower than 30 NM from any of the entry points to the runway). We could also derive the upper bound from the grid (no path from any entry point to the runway can use any edge twice). However, this is far from a realistic upper bound and we impose a smaller upper bound, which still allows for feasible solutions. That is, we chose an upper bound, such that we can compute an arrival tree fulfilling all constraints using route lengths in the interval [lower bound, upper bound] (which would not be possible for an upper bound equal to the lower bound), but that is significantly smaller than the upper bound we can directly identify from the grid.

First, as detailed in [Section 4](#), earliest and latest trajectories are computed in order to obtain the time window at the TMA entry point. In total, we computed 28 trajectories per flight (i.e. 14 possible route lengths ranging from 30 NM to 108NM for both earliest and latest cases). Then, once an ETA is selected for each flight, we generated 14 trajectories per aircraft and for each route length ensuring the same arrival time (i.e., the ETA) at the TMA entry point. As indicated in [Section 4](#), in case the same ETA was not achievable for all route lengths, longer routes were discarded first.

5.4. Arrival sequencing key performance indicators

Our optimized solutions guarantee separation at any point of the TMA with the chosen separation parameter $\sigma_{A,B}$. We evaluate the benefits provided by this approach using a set of the Key Performance Indicators (KPIs) recently proposed by Eurocontrol Experimental Centre in [Christien et al. \(2019\)](#), with several adjustments to the proposed methodology, which are detailed in [Smetanová \(2020\)](#):

- **Minimum time to final (ttf):** the minimum ttf is defined as the minimum time it takes an aircraft to fly from its current position to the FAP. We calculate the minimum ttf as the minimum time needed from any point within a grid cell to the FAP along any of the aircraft trajectories passing through the cell.
- **Spacing deviation (sd):** the spacing of an arriving aircraft pair at time t is defined as the difference between the respective minimum times to final. The spacing deviation (sd) is computed for pairs of leading and trailing aircraft at time t ; it captures the aircraft's mutual position in time. It is calculated using Eq. (1):

$$sd(t) = \text{ttf}(\text{trailer}(t)) - \text{ttf}(\text{leader}(t - s_{rwy})), \quad (1)$$

where s_{rwy} is the temporal separation at the runway.

The spacing deviation reflects information about the control error, which is the accuracy of spacing around the airport.

- **Sequence pressure (sp):** the sequence pressure for an aircraft at time t corresponds to the number of aircraft sharing the same minimum time to final within a given time window; it reflects the density of aircraft. It is calculated for each aircraft at any time of its presence within the TMA in discrete time steps. We choose a window of 2 min (the minimum separation requirement in our optimization framework).

5.5. Vertical and lateral efficiency

In order to assess the benefits of our solution, we do a study of the vertical efficiency for the different case studies, where we compare the vertical profiles of the actual flown trajectories with the corresponding flights flying neutral CDOs and following the assigned route of the arrival tree. As mentioned in Section 5.3, we took most of the data for the actual trajectories from the OpenSky Network, with the exception of two aircraft whose data was obtained from Eurocontrol's DDR.

Because DDR data is sparser, the information about the vertical profiles is not as detailed as in OpenSky states data. On the other hand, OpenSky states data is very dense. Thus, in order to speed up the computations, we sparsify the selected dataset by using every third recording, which reflects aircraft position every three seconds and results in less fluctuations. The data also contains erroneous position records, as well as false altitude records, which we filter out. Data from DDR is used unfiltered.

In order to compare the vertical profiles of the actual flown trajectories with the corresponding flights flying neutral CDOs, we cut the actual trajectories at the same altitude as the end of the CDO trajectories. Since we are only interested in comparing the vertical profiles inside the TMA, we define the start of the trajectory as the point where the aircraft enters Stockholm TMA.

We also study the difference in distance flown inside the TMA between the actual trajectories and the trajectories following a CDO, as well as the time spent in the TMA. The definition of the start and the end of the trajectory is identical to the one used for the vertical profiles. We calculate the distance between each recording based on the latitude and longitude coordinates provided in OpenSky and DDR data.

5.6. Set of experiments

Table 1 details all the experiments carried out in this work. The case studies are named after the corresponding amount of arriving traffic (i.e. low, average and high) during the day and time chosen. We mainly study the sensitivity of our solution to three parameters:

- **Number of arriving aircraft:** three different case studies are analyzed with different amounts of traffic.
- **Entry-time-window offset:** for each case study, different entry-time-window offsets are assumed. As detailed in Section 3.1, gaining/losing time in the pre-sequencing area to achieve different entry times at the TMA entry point could help to schedule aircraft even in the most challenging scenarios. Discrete time-window offsets are used, so, for instance, an offset of ± 3 min would represent that the arriving traffic could arrive to the entry point of the dynamic-trajectories area 3, 2 or 1 min earlier or later than the original time of arrival.
- **Traffic mix:** for the average and high-traffic case studies, we replace a given number of randomly chosen aircraft with a light aircraft type. The model chosen was an Embraer EMB-500 Phenom 100 (E50P).

As it can be seen in Table 1, not all parameters are tested for all case studies. For instance, in the low-traffic case study, only an entry point time-window offset of ± 1 min is tested (apart from the case where no deviation in the entry time is considered). As it will be shown in the results (Section 6), we performed experiments until we obtained the minimum-length tree. Therefore, in the particular case of the low-traffic case study, it did not make sense to simulate the results for larger time-window offsets as the minimum length tree was already obtained when considering an offset of ± 1 min. Larger offsets will lead to the same results.

Regarding the analysis of different traffic mixes, we decided to only study the effects of replacing 20 percent of the arriving aircraft with light aircraft. The aim is to study how our framework deals with different kinds of aircraft types and the effect they have on the resulting trees. However, we have decided not to study further cases (with different numbers of light aircraft) due to the low applicability that such kind of study would have in real operations. In major airports such as Arlanda, the ratio of light aircraft per day is usually between 1% and 3%. Even in airports with a less uniform traffic mix, like Palma de Mallorca airport, the ratio of light aircraft during a day hardly surpasses 5%. On the other hand, airports with a higher ratio of light aircraft are usually small aerodromes with a low amount of traffic, where our solution would not be worth to implement. Therefore, in this work, the aim is just to study the theoretical limits of our framework in a scenario with a significant number of light aircraft.

Finally, for the average and high-traffic case studies, we divided the corresponding 1-h-period into 2 half-an-hour-periods. As a result, two arrival trees were generated for each case study, which adapt better to the current traffic situation. On the other hand, generating just one tree for a high-traffic scenario during a 1-h period may lead to unnecessarily long paths from the entry points to the runway. Moreover, as discussed in Section 6.1, it would be computationally too expensive. In order to ensure a smooth transition between trees, different values of the parameter U were used.

6. Results

In this section, we present and analyze the results of the experiments described in Table 1 (Section 5.6). For the sake of simplicity, in this section, whenever we refer to the entry point, we are actually referring to the entry point of the dynamic-trajectories area, which in this case corresponds to the Stockholm TMA.

6.1. Computational effort

We solve our MIP using the Gurobi optimization solver installed on a powerful Tetralith server (Tetralith server, 2019), utilizing Intel HNS2600BPB computer nodes with 32 CPU cores, 384 GB, provided by the Swedish National Infrastructure for Computing (SNIC). The computational time required to generate the arrival trees varies from 1.58 h for the low-traffic case study to 40.9 h for the high-traffic case study with light traffic. One of the main issues when running the simulation was the memory usage, specially for the case-studies with a higher amount of traffic, in which the memory usage reached values of 89% and sometimes even went up to a 99%.

Currently, the computational times obtained with our framework may limit its application in real operations, except for low traffic loads. The arrival management problem is very dynamic, and fast computational times are usually required in order to ensure the success of the operations. There are several factors affecting the computational load of the route-optimization framework. Firstly, the size of the dynamic-trajectories area and, consequently, the size of the grid, has an important impact on it. In this case, we decided to define an area that fitted the Stockholm TMA. However, smaller areas could be chosen in order to speed up the process. In addition, the computational time is also affected by the fact that only neutral CDOs are considered—which involve a smaller time window—and by the amount and distribution of the arrival traffic. In both cases, ATCOs are eventually forced to request the aircraft to gain/lose a specific amount of time in the pre-sequencing area in order to find feasible solutions. As a result, the route-optimization framework has to deal with a larger number of potential solutions (i.e., more times of arrival for each arriving aircraft at the entry point), which slows down the computation of the arrival tree.

For the high and medium-traffic case studies, two trees were generated for the 1-h period, in order to obtain solutions that adapt better to the current traffic situation and in order to speed-up the process. In the future, it may be worth trying to generate arrival trees for even smaller periods of time, so that the framework has to work with a lower amount of arrival traffic. In addition, working with smaller time horizons could help to generate trees that adapt better to the current traffic situation. However, continuous switches between the trees are undesirable from an ATCO point of view. Therefore, in future work, we are planning to do a trade-off analysis to find out which would be the optimum time horizon to generate the arrival trees depending on the current traffic situation. While there are several factors dramatically increasing the computational load of our framework, in this work, the results were generated with a very simple prototype. A more efficient code could be written to speed up the process, or more powerful solvers could be used. Furthermore, it would be worth investigating other approaches, like the one proposed in Chevalier (2020), which provides a sub-optimal solution but could potentially accelerate the generation of the arrival tree while working with the same CONOPs.

In general, the solution proposed in this paper needs further improvements in order to work properly in high-traffic scenarios. In this section, we have identified some future lines of research that could potentially speed-up the route-generation process. Among them, the most promising ones might be related with changing the model used to generate the arrival tree. Although the MIP model proposed always yields optimal solutions, the computational time is too high. For that reason, it might be a good idea to just use the MIP model for low-traffic scenarios. Then, as the traffic increases, other methodologies might be a better option, like the simulated annealing approach presented in Chevalier (2020). This simplification in the model would yield sub-optimal arrival trees—involving, for instance, longer arrival routes—but at the same it would allow to apply our solution in real operations due to the reduction in the computational time. By using this approach, our solution could be applied in more dense traffic scenarios, while using the same CONOPs.

6.2. Arrival trees and summary of results

Table 2 shows the number of aircraft scheduled and the sum of paths lengths of the resulting arrival tree for each case study. Note that in the low-traffic case study only one arrival tree was generated for the whole-hour period.

The lower the traffic, the easier it is to schedule all the arriving aircraft. However, as the traffic increases, the number of potential solutions is reduced and, ultimately, it is necessary to ask the aircraft to gain or lose a given amount of time in the pre-sequencing area in order to be able to succeed in the scheduling. Fig. 7 shows the effect of the entry-time-window offset on the percentage of aircraft scheduled and on the difference, for each case study, between the obtained sum of paths lengths and the minimum sum of paths lengths (in NM).

Table 2
Results summary.

Case study	Time-window offset (minutes)	Traffic mix (% light a/c)	Number of aircraft scheduled (#a/c 1st 30'/#a/c 2nd 30')	Sum of paths lengths [NM] (Length 1st 30'/Length 2nd 30')
Low-traffic (8 aircraft)	± 0	0	8	220.22
	± 1	0	8	216.68
Average-traffic (22 aircraft)	± 0	0	10/5	224.92/216.68
	± 1	0	13/6	213.14/216.68
	± 2	0	14/8	220.22/213.14
	± 2	20	12/8	216.68/213.14
	± 3	0	14/8	213.14/213.14
	± 3	20	13/8	220.22/213.14
	± 4	20	14/8	220.22/213.14
High-traffic (30 aircraft)	± 0	0	8/10	246.16/242.62
	± 1	0	11/11	243.78/227.3
	± 2	0	13/13	227.3/220.22
	± 3	0	14/13	230.84/220.22
	± 4	0	14/14	230.84/214.33
	± 5	0	15/15	230.84/220.22
	± 5	20	13/12	230.84/220.22

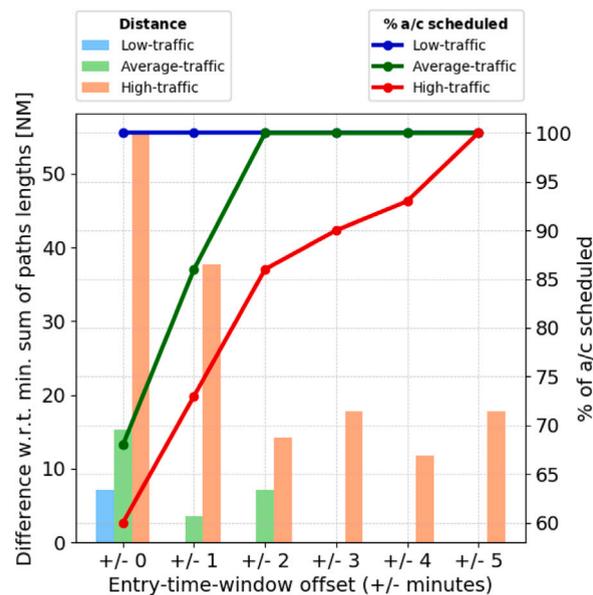


Fig. 7. Entry-time-window offset effect on the number of aircraft scheduled and on the sum of paths lengths for each case study (0% light aircraft and distance differences expressed in NM).

In the average-traffic case study, a time-window offset of at least ± 2 min is needed to schedule all aircraft, while in the high-traffic case study the required offset is ± 5 min. On the other hand, in the low-traffic case study, there is no need for the arriving aircraft to gain or lose time in the pre-sequencing area, as all aircraft can be scheduled by taking into account the original times of arrival. Once all aircraft are scheduled for a specific case study, it is possible to further increase the time-window offset to reduce the total sum of paths lengths. For instance, in the average-traffic case study, if the time-window offset is increased from ± 2 min to ± 3 min, the sum of paths lengths is reduced. Similarly, the sum of paths lengths is reduced by approximately 8NM in the low-traffic case study if an offset of ± 1 minute is considered. The route-optimization framework generates a shorter arrival tree because the aircraft arrival time at the TMA entry point is flexible, which results in a larger set of feasible solutions (and the minimum objective value over a superset cannot be larger than that over the original set). However, it is not evident whether it would be better for the aircraft to consider larger time-window offsets—which would involve gaining or losing more time in the pre-sequencing area, with the consequence of flying further from the optimum speed—or to fly a longer route but with a lower deviation with respect to the original time of arrival at the entry point. Both strategies have the potential to increase or reduce the fuel consumption of the flights and the final decision might depend on the specific scenario.

While in the average and low-traffic case studies it is possible—with a sufficiently large time-window offset—to schedule all aircraft and to obtain an arrival tree with the shortest possible paths, in the high-traffic case study this is impossible. As it can be

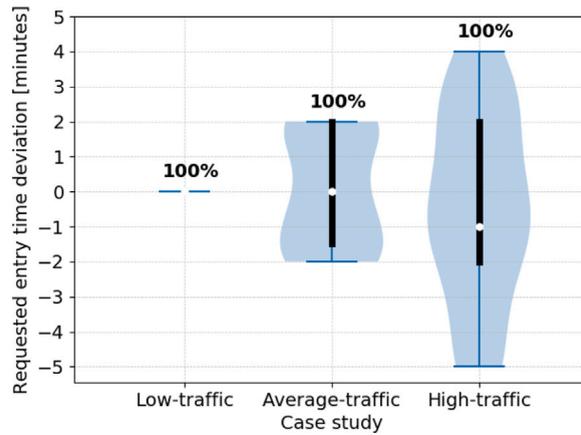


Fig. 8. Entry time deviation per aircraft with an entry-time-window offset of +/- 5 min for each case study.

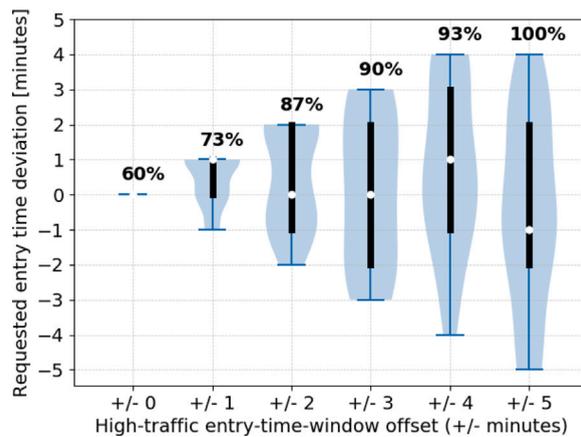


Fig. 9. High-traffic case study: requested entry time deviation distribution + percentage of aircraft scheduled (as a function of the entry-time-window offset).

observed in Fig. 7, although the difference with respect to the minimum sum of paths lengths is reduced as the time-window offset increases, even with an offset of ± 5 min the resulting arrival tree is around 18NM longer than the shortest arrival tree.

Another interesting matter is the entry time deviation requested for each aircraft. As discussed in Section 5.6, this value depends on the considered time-window offset. For instance, for an offset of ± 3 min, the route-optimization framework considers that the arriving aircraft can gain or lose 1, 2 or 3 min in the pre-sequencing area. Figs. 8 and 9 depict the distribution of the requested entry time deviation per aircraft. Fig. 8 focuses on the effect of the amount of traffic, depicting the results for the low, average and high-traffic case studies when an time-window offset of ± 5 min is considered. On the other hand, Fig. 9 focuses on the high-traffic case study, but by showing the results for different offsets. For the low-traffic case study all aircraft can be scheduled by considering no deviation at the entry point. Conversely, in the average-traffic case study this is not possible, and a time-window offset of ± 2 min is necessary to schedule all traffic. As it can be observed in Fig. 8, most of the aircraft are requested to gain or lose 1 or 2 min in the pre-sequencing area in that case, while a few of them can still arrive at the entry point at their original time of arrival. On the other hand, in the high-traffic case study, where a time-window offset of ± 5 min is needed to schedule all traffic, the majority of aircraft are requested to gain or lose a time in the ± 2 min range. Still, some aircraft are also requested to gain a larger amount of time (i.e., 3 and 4 min) and some of them are even requested to lose 5 min. As a result, all aircraft can be safely scheduled.

If we focus on the high-traffic case study (Fig. 9), we can observe the different entry time deviations requested depending on the considered time-window offset.

In general, and whenever it is allowed, most of the aircraft are requested to gain/lose a time in the ± 2 min range. While only a time-window offset of ± 5 min allows to automatically schedule all aircraft, it is also true that the largest variation in the number of aircraft scheduled is observed from an offset of ± 0 to ± 2 min, where 26 out of 30 can be scheduled (i.e., 87% of the total aircraft). Then, considering larger offsets slightly increases the number of aircraft scheduled. In this particular case it is possible to find a reasonable time-window offset to schedule all aircraft (i.e., ± 5 min); however, if that was not possible, or if a very big offset was needed, those aircraft that were impossible to schedule could be managed manually by the ATCO via the assignment of vector instructions.

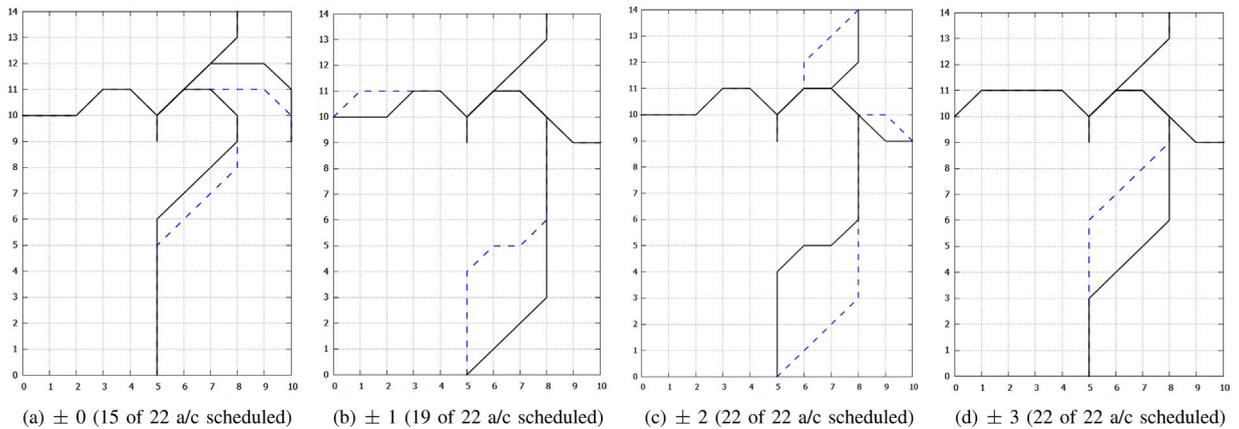


Fig. 10. Arrival trees as a function of the entry-time-window offset for the average-traffic case study (0% light aircraft). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Minimum time to final (ttf), spacing deviation (sd) and sequence pressure (sp): Min/Max/Average/Standard deviation.

Case study	Actual ttf [sec.]	Optimal ttf [sec.]	Actual sd [sec.]	Optimal sd [sec.]	Actual sp [# a/c]	Optimal sp [# a/c]
Low-traffic	0/836/456/215	0/660/332/162	-203/152/0.62/73.18	0/240/36.73/57.21	1/2/1.15/0.36	1/1/1/0
Average-traffic	0/1073/807/400	0/1140/607/292	-236/249/2.89/64.72	-300/360/11.25/117.22	1/4/1.33/0.55	1/2/1.04/0.19
High-traffic	0/986/494/228	0/660/331/161	-328/338/-2.86/86.25	-300/300/16.42/69.45	1/4/1.38/0.65	1/1/1/0

Furthermore, powered descents could be considered in order to ease the scheduling of the most conflicting aircraft, as they offer more flexibility than neutral CDOs. In such a situation, there would be an efficiency loss, but, still, all of the aircraft could be scheduled.

Regarding the results for different traffic mixes, it can be observed that, as expected, having a significant ratio of light aircraft in the arrival traffic complicates the task of the route-optimization framework. Larger time-windows offsets have to be considered in order to be able to schedule all aircraft. In the average-traffic case study, an offset of ± 4 min is required, while only an offset of ± 2 min was needed to schedule all aircraft when considering the original arrival distribution without light aircraft. On the other hand, in the high-traffic case study, it was not possible to schedule all aircraft even with a time-window offset of ± 5 min. Still, as discussed in Section 5.6, the aim when testing different traffic mixes was to assess how our framework handled such a situation; however, from an operational point of view, this is not representative enough, as in major airports the ratio of light aircraft per day is usually between 1% and 3%.

An example of the resulting arrival trees generated by the route-optimization framework is depicted in Fig. 10, corresponding to the average-traffic case study. Arrival trees for the given 1-h-period and for several entry-time-window offsets are presented. As discussed in Section 5.6, two trees are generated for each 1-h-period; in Fig. 10, black paths correspond to the first arrival tree and blue paths correspond to the second arrival tree, while the paths that are shared by both trees are black. In this particular case, arrival trees from no time-window offset to an offset of ± 3 min (where minimum route lengths are achieved) are presented. For the high and low-traffic case studies, the arrival trees can be found in Appendix C. It is important to highlight the fact that the trees depicted in Fig. 10 (as well as the ones in Appendix C) correspond to the output of the route-optimization framework. Thus, in some situations, it could happen that the trees obtained are not optimal from an operational point of view, even if the cost function defined in Appendix B.1 is minimized. For instance, for the average-traffic case study with a time-window offset of ± 3 min (Fig. 10(d)), the route-optimization framework proposes to change the shape of the tree for the second half-hour period. However, the second tree (blue path) has the same length as the first tree (black path). Furthermore, only the route from the southern entry point is affected. Therefore, from an ATCO point of view (in order to reduce the workload), it would be better to have the same arrival tree for the whole 1-h period. Thus, a post-processing would be needed in this kind of situations in order to obtain such a solution. Another option would be to alter the value of U , which is the parameter of the MIP formulation used to ensure a consistency in terms of number of differing edges between arrival trees during consecutive time periods. Smaller values than the ones used in the experiments could be used in order to obtain the same tree. The “small” value of U could be found with a binary search, however, this would be a slow process. On the other hand, noise could also have an effect on the final shape of the arrival tree. For instance, we could suppose a scenario where the route coming from the south in the first tree (in Fig. 10(d)) overflies a populated area. In such a case, maybe it would be better to keep the tree as it is now in order to minimize the noise effects over populated regions (assuming the southern route of the second tree avoids these regions). Similarly, it might be also better to keep two trees in order to increase noise dispersion and, thus, minimize aircraft noise impact on communities around an airport (Ho-Huu et al., 2020). The effect of noise is indeed a very interesting topic and will be investigated in future work.

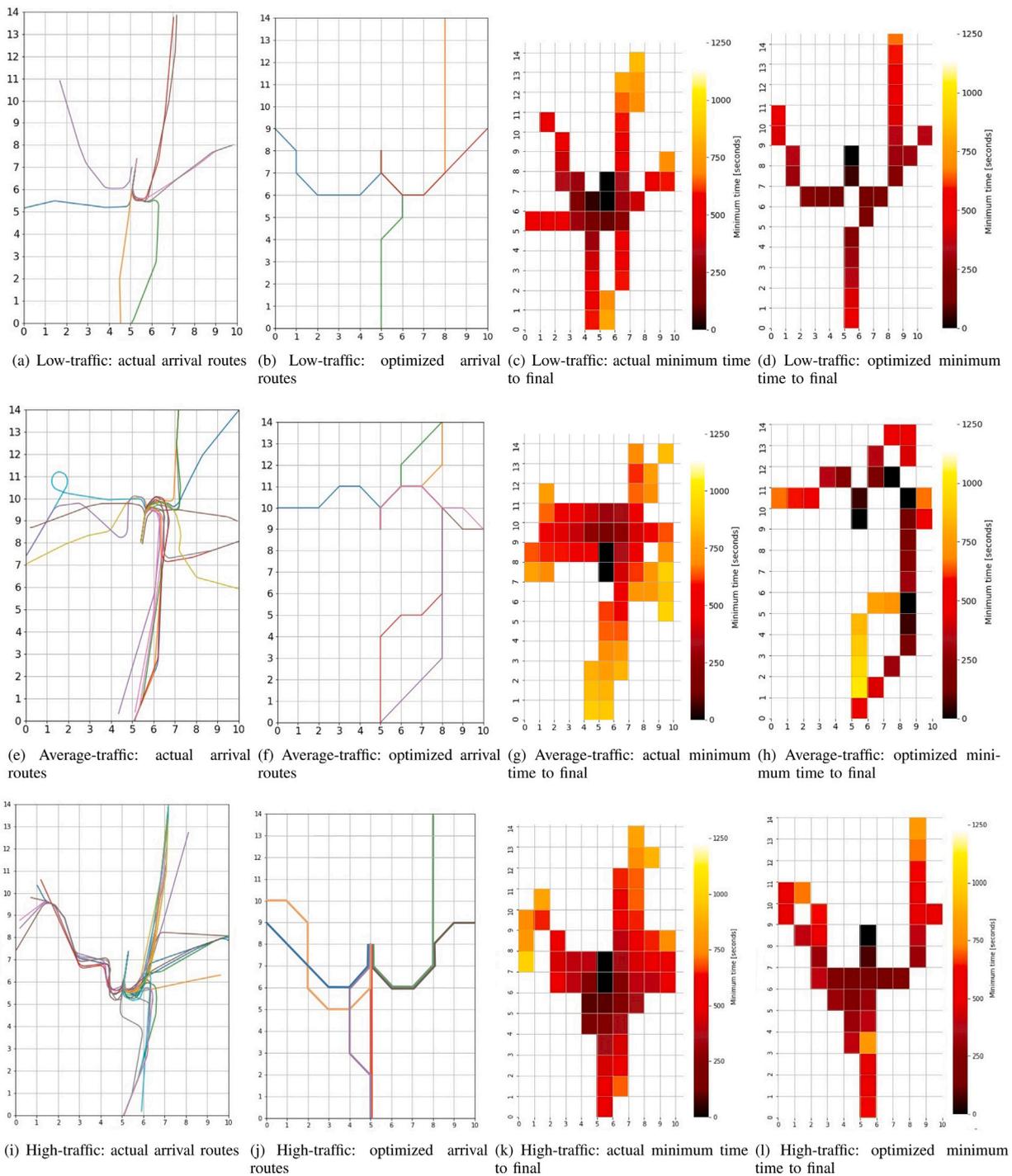


Fig. 11. Actual flown trajectories vs. optimized trajectories for low, average and high-traffic case studies: arrival routes and minimum time to final.

6.3. Evaluation of arrival sequencing

Fig. 11 depicts the actual and optimized routes (i.e., optimal arrival tree) for each case study. Additionally, several heat maps are generated in order to depict the minimum ttf for each case. Table 3 shows the minimum, maximum, average and standard deviation values for the minimum ttf, as well as for the spacing deviation and sequence pressure key performance indicators.

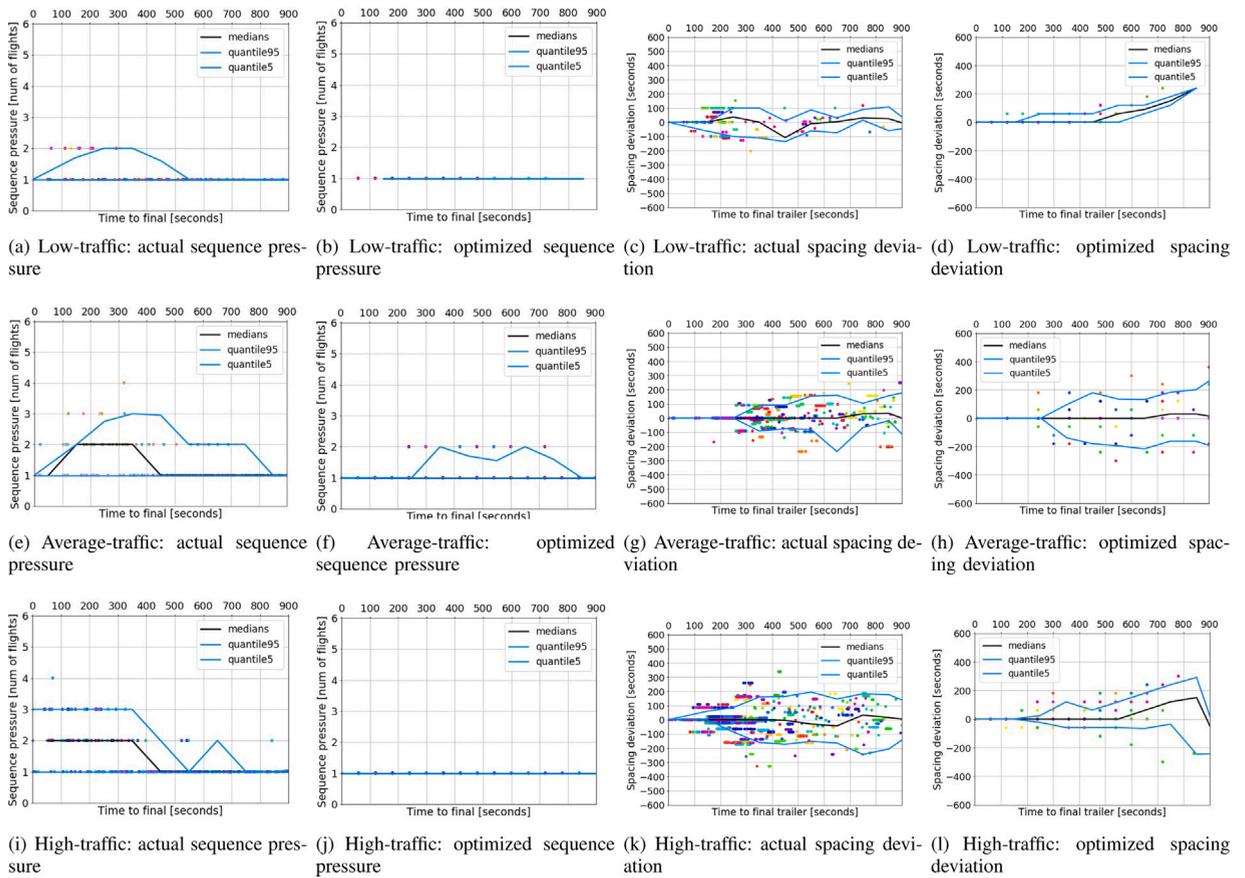


Fig. 12. Actual trajectories vs. optimized trajectories for low, average and high-traffic case studies: sequence pressure and spacing deviation.

In general, the shape of the optimal arrival tree is similar to that of the actual arrival routes. However, in the optimal arrival tree, there is only one arrival route per entry point, while in the actual case aircraft follow several routes from each of the entry points; actually, it seems that there are more than only 4 entry points to the TMA in the actual case. This is due to the fact that in reality ATCOs might need to give vector instructions in order to be able to schedule all aircraft. On the other hand, in the optimal arrival tree, the arrival routes automatically ensure that a separation is already maintained between the arriving aircraft. Therefore, there is no need for vectoring and, consequently, there is a lower route dispersion leading to a lower volume of airspace used.

In the optimal case, there is a higher situational awareness by both the ATCO and the flight crew than in the actual case. ATCOs, when aircraft are in the pre-sequencing horizon, up-link an RTA at the TMA entry point and an arrival route from the optimal arrival tree. Thus, the flight crew already knows which route they should follow in the TMA (Figs. 11(b), 11(f), 11(j)), and the FMS on board can optimize the descent trajectory accordingly. This is quite different from the current situation, where neither the flight crew nor the ATCO know the arrival route that will be followed inside the TMA. This is specially remarkable in very congested scenarios, where many vector instructions might be needed in order to safely schedule all traffic (Figs. 11(a), 11(e), 11(i)).

Regarding the minimum time to final, in all three case studies lower average values are obtained in the optimal case, with a reduction in comparison to the actual case of between 2 and 3 min, depending on the case study. However, in the average-traffic case study, the maximum value is higher than that of the actual case, around 300 s more. Still, our solution provides a lower average value. Furthermore, it is important to remark that in this work we are scheduling all aircraft assuming neutral CDOs so, even if in some cases longer routes are generated—which may lead to higher ttF values—the overall efficiency of the operations is improved.

The results for the spacing deviation and sequence pressure analysis are shown in Fig. 12. For the low and high-traffic case studies, shorter intervals (i.e., difference between the maximum and minimum values) for the spacing deviation are obtained with our solution, which means that, on average, for an arriving aircraft pair at time t , the difference between their respective minimum times to final is lower. On the other hand, in the average-traffic case study, longer intervals are obtained in the optimized case. Another important metric to analyze is the maximum width of the 90th quantile, which quantifies the spread of the 90% confidence interval and is shown in turquoise in Figs. 12(c), 12(d), 12(g), 12(h), 12(k) and 12(l). For both the low and average-traffic case studies a lower width of the 90th quantile is observed for the optimized case in comparison with the actual case, 211/120 and 387/363 s respectively. On the other hand, in the high-traffic case study, the value obtained in the real case is 427 s, which is lower than the width of the 90th quantile for the optimized case, which is 537 s.

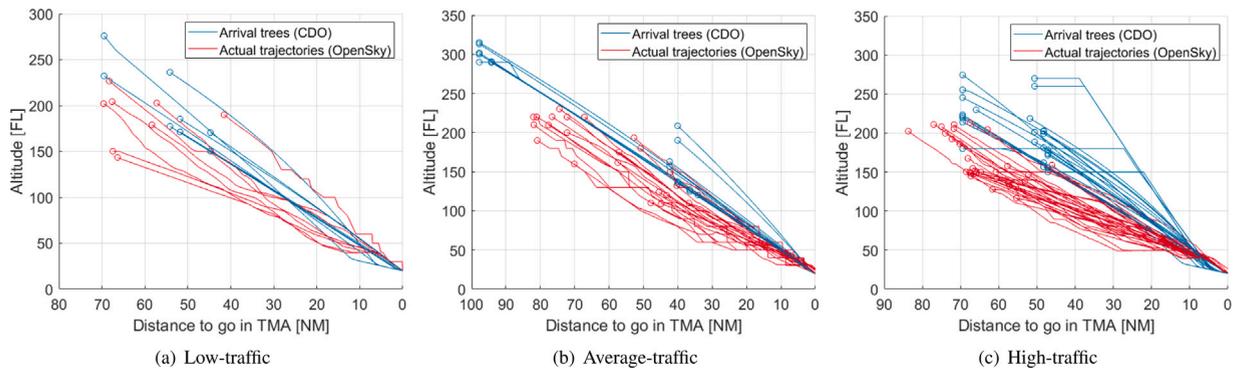


Fig. 13. Vertical profiles for the different traffic case studies.

Table 4

CDOs vs actual flown trajectories: Distance and time in TMA.

Case study	CDOs average distance in TMA [NM]	Actual trajectories average distance in TMA [NM]	CDOs average time in TMA [minutes]	Actual trajectories average time in TMA [minutes]
Low-traffic	55.06	56.66	8.5	12.13
Average-traffic	62.63	61.68	17.14	13.76
High-traffic	57.32	65.3	9.33	15.12

Regarding the sequence pressure, in all case studies our solution provides lower values than the actual case. For a window size of 120 s, in both the low and high-traffic case studies the minimum, maximum and average sequence pressure obtained with our solution is equal to 1, while in the average-traffic case study the maximum value is equal to 2. In that case, that means that up to two aircraft intended to arrive at the same time (± 2 min), which might indicate a potential separation problem. However, as shown in [Hardell et al. \(2020b\)](#), this is not always the case and the fact of having sequence pressure values equal to 3 does not always necessarily indicate a separation violation problem. Regarding the actual case, sequence pressure maximum values up to 4 are obtained. This metric clearly uncovers the benefits of our approach, which noticeably lowers the sequence pressure not only at the runway, but also at all points in the TMA. As a result, ATCOs would need to spend less time resolving potential separation problems to smooth the scheduling process of the arrival traffic.

6.4. Vertical and lateral efficiency results

In this section, we present the results of the vertical and lateral efficiency analysis of our solution, which are obtained by comparing the actual flown trajectories—mainly obtained from the OpenSky network, which follow the standard arrival procedure inside the TMA—with the trajectories flying CDOs and following the assigned routes along the arrival tree.

A comparison of the vertical profiles inside the TMA is shown in [Fig. 13](#). As expected, some of the actual flown trajectories follow very inefficient descent profiles with several steps at constant altitude; some of them are even performed at very low altitudes, which has an adverse effect on fuel consumption. On the other hand, CDOs allow the aircraft to follow a more continuous descent path, without intermediate steps at constant altitude when following the arrival route. Furthermore, when flying CDOs, aircraft tend to start the descent later, meaning that their top of descent is usually closer to the final point of the trajectory. As a result, aircraft flying CDOs usually enter the TMA at a higher altitude, thus, improving the flight efficiency. Finally, in the high-traffic case study ([Fig. 13\(c\)](#)), we can observe that there are several optimized trajectories that enter the TMA at very low cruise altitudes, at FL270 or even at FL150. These trajectories correspond either to short flights, where there is no time for the aircraft to reach a higher cruise altitude, or to turboprop aircraft, which tend to fly at lower cruise altitudes.

[Table 4](#) shows the average distance and time spent by the arriving aircraft in TMA for both actual trajectories and CDOs. Then, [Fig. 14](#) depicts the individual values for each arriving aircraft. While better results are obtained for the low and high-traffic case studies—with a reduction in the average distance and time spent in TMA of 1.6 NM/3.6 min and 8 NM/5.8 min, respectively—worse results are obtained for the average-traffic case study—where, on average, an increase of approximately 1 NM and 3.5 min spent in TMA are obtained with our solution.

While our solution does not ensure a reduction in the distance and time spent in TMA in all cases, it is also true that only neutral CDOs are considered in all case studies. In such a situation, the route-optimization framework—whose task is to optimize the scheduling of the arrival traffic by providing an automated separation in time and space—needs to generate, in some cases, arrival trees with longer routes than those flown in the actual case. For the actual flown trajectories, it can be observed that CDOs are not flown and trajectories follow in some cases step-down approaches. Shorter paths might be considered in some cases, but at the expense of flying less efficient vertical profiles.

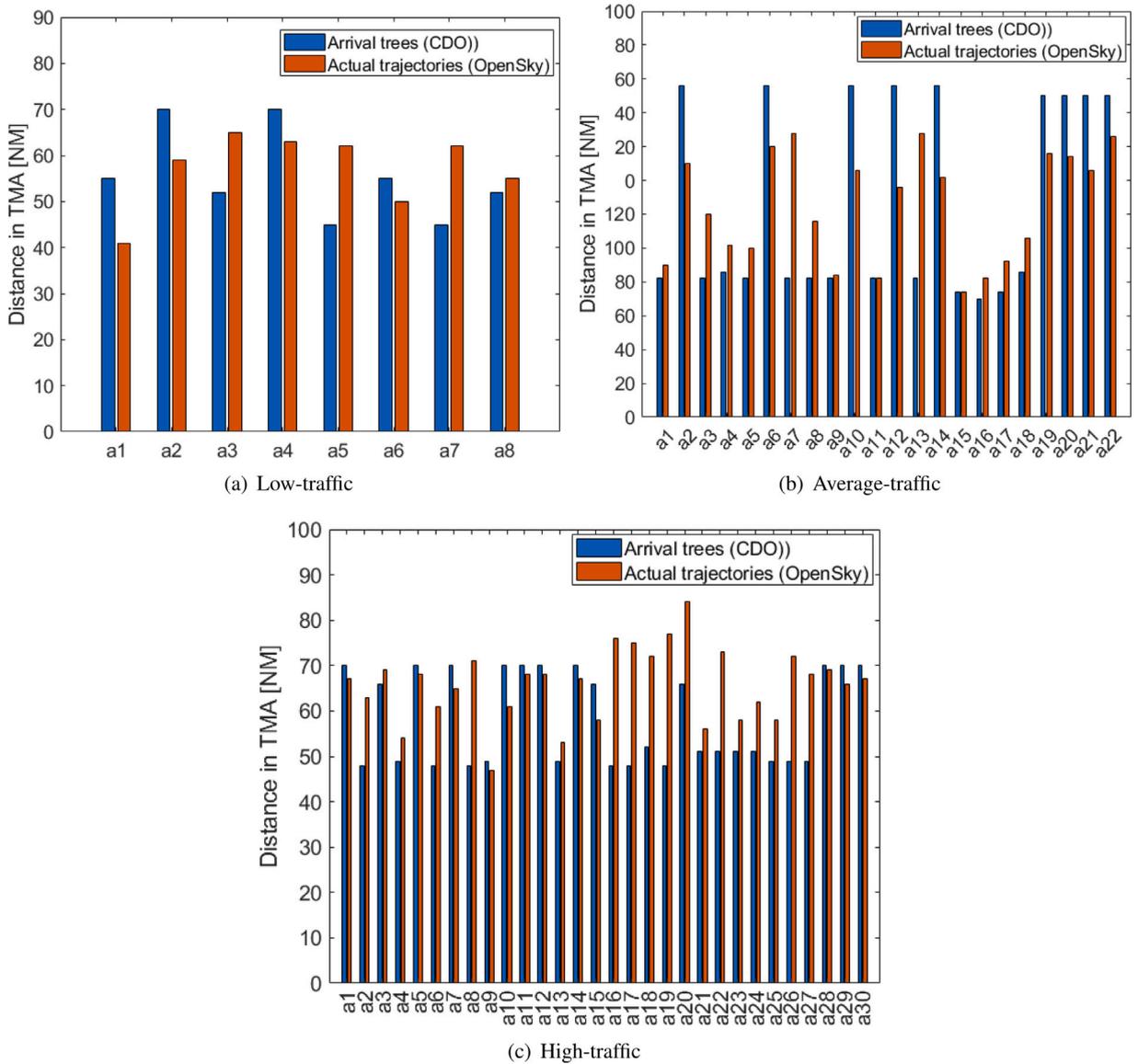


Fig. 14. Distance in TMA for the different traffic case studies.

7. Conclusions

In this paper, we proposed to schedule arrival traffic by means of two different areas surrounding the airport. In the pre-sequencing area, a 4D-traffic-negotiation/synchronization process is performed between the ATCO and the arriving aircraft. The outcome of this process, which is managed by a ground-based tool, is a set of optimal arrival routes (i.e., arrival tree) inside the dynamic-trajectories area, as well as the assignment of an RTA at the entry point of this area for each arriving aircraft. Our solution adapts to the current traffic demand, allows the aircraft to fly CDOs and ensures a safe separation between aircraft throughout the arrival procedure.

Our approach has the potential to both reduce the environmental impact of aircraft operations and the workload of ATCOs. However, there are several factors to be considered that may limit the benefits of our solution. The arrival-traffic amount and distribution greatly affect the outcome of our route-optimization framework. Although we have shown that a solution can always be found, even in the scenarios with a high amount of traffic, it is also true that in such cases aircraft are requested to gain/lose more time in the pre-sequencing area. As a result, the computational time needed for generating the optimal arrival tree drastically increases. In addition, we enforced the use of neutral CDOs (idle thrust and no speed-brakes usage allowed) for all aircraft, with the consequent reduction in each aircraft's arrival time window when compared with conventional descends or powered CDOs (where both thrust and speed-brakes usage is allowed). Hence, it takes a longer time for the route-optimization framework to find a solution.

Finally, the size of the dynamic-trajectories area, where the optimal arrival tree is generated, could also affect the computational load of the framework. In this work, we decided to define this area in such a way that it fitted the current Stockholm TMA. A smaller area would potentially speed up the generation of the arrival tree, while larger areas would have the opposite effect. The size of the dynamic-trajectories area (and the pre-sequencing area) is indeed a very interesting topic and will be investigated in future work.

While a solution is always found by our optimization framework, the computational time in some cases is too high. A more efficient code could be written or other approaches—like the one proposed in [Chevalier \(2020\)](#), which provides a sub-optimal solution—could be investigated to speed up the computations. In future work, we are also planning to integrate wind to our solution, which might involve computing a higher number of trajectories (where wind is considered when generating the CDO).

The CONOPS proposed in this paper replaces the current STAR paradigm—where routes are static and not designed to meet the requirements of the arriving traffic—with a dynamic generation of arrival routes that increases the efficiency and safety of operations while keeping the same levels of capacity. Our solution aligns with the TBO concept, and is expected to become a technical enabler towards an E-AMAN with extended capabilities, where the sequencing horizon would be extended into the en-route sector, thus improving the situational awareness of both the flight crew and the ATCO. Furthermore, airport's arrival management information would be shared with upstream sectors in real time by using a SWIM service.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is a part of ODESTA project supported by the Swedish Transport Administration (Trafikverket) and in-kind participation of LfV.

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Appendix A. Trajectory optimization problem

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

Given a known route, 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 ([Sáez et al., 2020a](#)). Several approaches can be found in the literature addressing this problem, like the ones found in [Prats et al. \(2017\)](#) and [Dalmau et al. \(2016\)](#), which focused on the development of advanced concepts and technologies in order to satisfy RTAs with high accuracy. In this work, the methodology chosen is the semi-analytic trajectory optimizer proposed in [Park et al. \(2017\)](#), which proved to be very robust and fast.

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

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

$$\mathbf{f}_a = \begin{bmatrix} \dot{v} \\ \dot{h} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \frac{T_{idle}-D}{m} - g\gamma \\ v\gamma \\ v + w \end{bmatrix} \quad (2)$$

where $T_{idle} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the idle thrust; $D : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}$ is the aerodynamic drag; g is the gravity acceleration; w is the along track wind as a function of the altitude 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 ([Clarke et al., 2004](#)).

The generic form of the cost function to be minimized is presented in Eq. (3):

$$J_a = \int_{t_0}^{t_f} [f + CI] dt \quad (3)$$

where f is the fuel flow and CI—equal to c_t/c_f —is the cost index, which is a parameter chosen by the airspace user that reflects the relative importance of the cost of time (c_t) with respect to fuel costs (c_f) ([Airbus, 1998](#)). For Airbus aircraft models, depending on the FMS vendor, this value could be scaled in the range 0–99 or 0–999, where units are usually given in kg/min.

In this paper, a variation of the cost function presented in Eq. (3) is used. The trajectory is divided in two phases: the last 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, yielding to the following cost function:

$$J_a = \int_{t_0}^{t_f} - \left(\frac{f_{cr} + CI}{v_{cr}} \right) (v + w) + f_{idle}(v, h) + CI dt \quad (4)$$

where v_{cr} is the cruise speed (which is assumed to be constant), f_{cr} is the cruise fuel flow (which is assumed to be constant, since the cruise altitude and speed are also constant), and $f_{idle} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the idle fuel flow in the descent (which depends on the speed and altitude).

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

$$h_a = \begin{bmatrix} v_{CAS,min} - v_{CAS} \\ v_{CAS} - VMO \\ M - MMO \\ \gamma \\ \gamma_{min} - \gamma \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5)$$

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

Finally, terminal constraints fix the final states vector:

$$\psi_a = \begin{bmatrix} v - v_f \\ h - h_f \\ s - s_f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

where $x_f = [v_f, h_f, s_f]$ is the state vector at the metering fix.

In the formulation presented herein, there is only one control variable, which appears linearly in the equations describing the dynamics of the system as well in the cost function to be minimized. This ultimately leads to a singular optimal control problem which can be solved semi-analytically from the implicit formulation of optimal singular arcs (Park et al., 2017). Since the initial and final states of the trajectory are fixed, the optimal trajectory will be of a ‘‘bang-singular-bang’’ type. These solutions consist of three arcs: one non-singular arc with the control variable at its maximum or minimum value to go from x_0 to the singular arc; a singular arc where the optimal control is given as a function of the states vector; and a final non-singular arc to go from the singular arc to the final state x_f . The control input of the non-singular arcs is determined by the airspeed difference between the singular arc and the boundary points at the initial and final altitudes. At the initial altitude, if airspeed on the singular arc is larger than cruise speed, the aircraft must accelerate to meet the singular arc. In such a case, a maximum given flight path angle is selected for the non-singular arc generation. On the other hand, if the singular arc speed is less than the cruise speed, a minimum given flight path angle is selected. The first non-singular arc (i.e. from the initial state to the singular arc) is computed by forward integration and the second non-singular arc (i.e. from the singular arc to the final state) by backward integration. The integration was performed by using a Runge–Kutta numerical method of 4th order (RK-4).

Finally, it is important to detail some of the limitations of the trajectory-optimization technique used in this work. First of all, only vertical-profile optimization for climbs and descents is considered. The optimization of the lateral profile is not performed by this method, which is not a problem for the work presented in this paper. The optimal routes will be computed by the route-optimization framework (as described in Appendix B), so, in the end, this task is not part of the trajectory optimizer. Another limitation of this method is the fact that RTAs can only be achieved by brute force, iterating over several cost indexes.

Although the method chosen for optimizing the trajectory could have some drawbacks for other applications, it is very robust and fast and fits well given the requirements of our application. The optimization is not based on a non-linear solver, so there are no convergence problems and a deterministic solution is always found. In all the optimization runs performed for this paper, a single trajectory was always computed in less than 5 s.

Appendix B. Grid-based MIP formulation

Our MIP formulation is based on the MIP we presented in Dahlberg et al. (2018) and Sáez et al. (2020b). For better understanding, we first present our MIP from Dahlberg et al. (2018), then we introduce changes from Sáez et al. (2020b). Finally, we introduce our new constraints.

In Dahlberg et al. (2018), we made the simplified assumption that traversing a single edge takes u time units for all aircraft (independent of aircraft type and distance to runway). In Sáez et al. (2020b), we considered aircraft with different speeds, and, in particular, the speed profile given by a CDO for the specific aircraft type.

In both papers, we used a uniform separation value, that is, any pair of consecutive aircraft needs to be separated by σ time units. In this paper, we take the wake-turbulence categories of the leading and trailing into account, that is, we no longer use a uniform separation of σ , but a temporal separation $\sigma_{A,B}$ that depends on the categories A and B of the leading and trailing aircraft, respectively. We also introduce flexibility of the arrival time at the entry point of the dynamic-trajectories area.

In [Appendices B.1–B.3](#), we start with a review of our prior MIP-formulation for static optimal STAR merge trees from [Andersson et al. \(2016\)](#). In [Appendix B.4](#), we present auxiliary constraints to prevent crossings which become necessary with the temporal separation using a uniform temporal separation of σ from [Dahlberg et al. \(2018\)](#), which we detail in [B.5](#). Then, we describe how we integrate different speed profiles in [Appendix B.6](#). In particular, we will use the speed profiles that stem from CDOs for the different aircraft, see [Appendix A](#) for details of their computation. In [Appendix B.7](#), we conclude the presentation of prior constraints and show how we can enforce largest possible consistency of trees from two consecutive time periods. In [Appendix B.8](#), we allow that aircraft arrive at their entry point of the dynamic-trajectories area within a time window (from which we may pick any arrival time) instead of forcing them to arrive at a specific time. This flexibility enables us to allow for scheduling more aircraft with CDOs. In [Appendix B.9](#), we introduce the temporal separation based on the wake turbulent categories of the aircraft. Finally, in [Appendix B.10](#), we present the complete MIP for better readability.

Our input consists of the position of entry points to the dynamic-trajectories area; the location and direction of the runway; arrival time (within a given time window T), arrival entry point, and aircraft type for all aircraft. We decided to use the position of the runway for the experiments of this work, although in general, as described in the CONOPS ([Section 3.1](#)), the final merging fix position should be considered instead. We aim to output an arrival tree that merges traffic from the entries to the runway such that all aircraft are separated at all points of the arrival routes. The tree has the entries as leaves and the runway as the root (here we slightly abuse notation twice, as directed trees are usually called arborescences, and these are usually directed from leaves to root). Aircraft are assumed to fly according to their optimal continuous descent speed profile for the route length of the entry point-runway path along the tree. These speed profiles are computed separately, and provided as input to the MIP in the form of different speed profiles for different arrival route lengths.

B.1. Tree constraints and objective function

We use a discretization: we overlay the dynamic-trajectories area with a square grid, and snap the location of both entry points and runway to the grid. We denote the set of (snapped) entry points by \mathcal{P} , and the (snapped) runway by r . 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 $G = (V, E)$: for any two adjacent vertices i and j , both edges (i, j) and (j, i) are included in E ; exceptions are the entry points (they do not have incoming edges) and r (it does not have outgoing edges). Let ℓ_{ij} denote the length of an edge $(i, j) \in E$. Let G be a grid of size $q \times n$.

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

Our underlying MIP formulation for static arrival routes ([Andersson et al., 2016](#)) is based on the flow MIP formulation for Steiner trees ([Wong, 1984](#); [Goemans and Myung, 1993](#)). We use binary decision variables x_e that indicate whether the edge e participates in the arrival tree. Moreover, we have flow variables: f_e gives the flow on edge $e = (i, j)$ (i.e., the flow from i to j). The constraints are given as follows:

$$\sum_{k:(k,i) \in E} f_{ki} - \sum_{j:(i,j) \in E} f_{ij} = \begin{cases} |\mathcal{P}| & i = r \\ -1 & i \in \mathcal{P} \\ 0 & i \in V \setminus \{\mathcal{P} \cup r\} \end{cases} \quad (7)$$

$$x_e \geq \frac{f_e}{Q} \quad \forall e \in E \quad (8)$$

$$f_e \geq 0 \quad \forall e \in E \quad (9)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (10)$$

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

Eq. [\(7\)](#) 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. Eq. [\(8\)](#) enforces edges with a positive flow to participate in the arrival route. The flow variables are non-negative (Eq. [\(9\)](#)), the edge variables are binary (Eq. [\(10\)](#)).

We can easily adapt Eq. [\(7\)](#) to allow minimization of the sum of trajectory lengths flown by all arriving aircraft. In that case, 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 simply change the right-hand side of Eq. [\(7\)](#) (and increase Q accordingly). Let w_b be the number of aircraft entering the dynamic-trajectories area via entry point $b \in \mathcal{P}$. Then the new constraint is written as:

$$\sum_{k:(k,i) \in E} f_{ki} - \sum_{j:(i,j) \in E} f_{ij} = \begin{cases} \sum_{b \in \mathcal{P}} w_b & i = r \\ -w_i & i \in \mathcal{P} \\ 0 & i \in V \setminus \{\mathcal{P} \cup r\} \end{cases} \quad (11)$$

We consider two objective functions: paths length and tree weight. These are given in Eqs. [\(12\)](#) and [\(13\)](#), respectively:

$$\min \sum_{e \in E} \ell_e f_e \quad (12)$$

$$\min \sum_{e \in E} \ell_e x_e \quad (13)$$

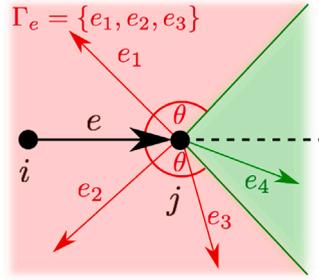


Fig. 15. Limited turn: if edge $e = (i, j)$ is used, only edges within the green region are allowed, that is, edges with an angle of at least θ with e . If edges in the light red region, Γ_e , are used, x_e must be set to zero. Here: $\Gamma_e = \{e_1, e_2, e_3\}, e_4 \notin \Gamma_e$.

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

$$\min \beta \sum_{e \in E} \ell_e x_e + (1 - \beta) \sum_{e \in E} \ell_e f_e \tag{14}$$

B.2. Degree constraints

Eqs. (7)–(10) are standard MIP-constraints for a MinCostFlow Steiner tree formulation, (11) 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 4. For operational Constraint (2) we require that no more than two routes merge at any point, that is, we restrict the in- and out-degree of every vertex: the out-degree is at most 1 and the maximum indegree is 2:

$$\sum_{k:(k,i) \in E} x_{ki} \leq 2 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{15}$$

$$\sum_{j:(i,j) \in E} x_{ij} \leq 1 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{16}$$

$$\sum_{k:(k,r) \in E} x_{kr} = 1 \tag{17}$$

$$\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \forall i \in \mathcal{P} \tag{18}$$

Eq. (17) enforces one ingoing edge for the runway r , Eq. (18) guarantees that each entry point has one outgoing edge. For all other vertices, Eq. (15) yields the maximum indegree of 2 and Eq. (16) results in the maximum out-degree of 1.

B.3. Turn angle constraints

The next operational constraint from Section 4 is Constraint (4): we must avoid sharp turns. Thus, we enforce that if edge $e = (i, j)$ is used in the arrival tree, all outgoing edges at j must form an angle of at least θ with e . Let Γ_e be the set of all outgoing edges from j that form an angle $\leq \theta$ with e , i.e., $\Gamma_e = \{(j, k) : \angle ijk \leq \theta, (j, k) \in E\}$, and let $c_e = |\Gamma_e|$.

$$c_e x_e + \sum_{f \in \Gamma_e} x_f \leq c_e \quad \forall e \in E \tag{19}$$

By Eq. (19) 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 . See Fig. 15 for an example.

B.4. Auxiliary constraints to prevent crossings

When we only minimize the length of the arrival tree, but do not integrate any constraint on temporal separation, no route crossings arise: uncrossing the routes would always shorten them by triangle inequality. However, we aim to integrate temporal separation constraints, hence, we need to take care of possible route crossings. While such route crossings at vertices are prevented by the degree constraints in Appendix B.2 (we would need at least indegree and out-degree of 2 each), we may still encounter routes crossing within a grid square, and we add auxiliary constraints to prevent this behavior.

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, $V' = V \setminus \{\text{last row}\} \setminus \{\text{last column}\}$. Then, we use Eq. (20) to prevent crossings.

$$\begin{aligned} x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} &\leq 1 \\ \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\} \end{aligned} \tag{20}$$

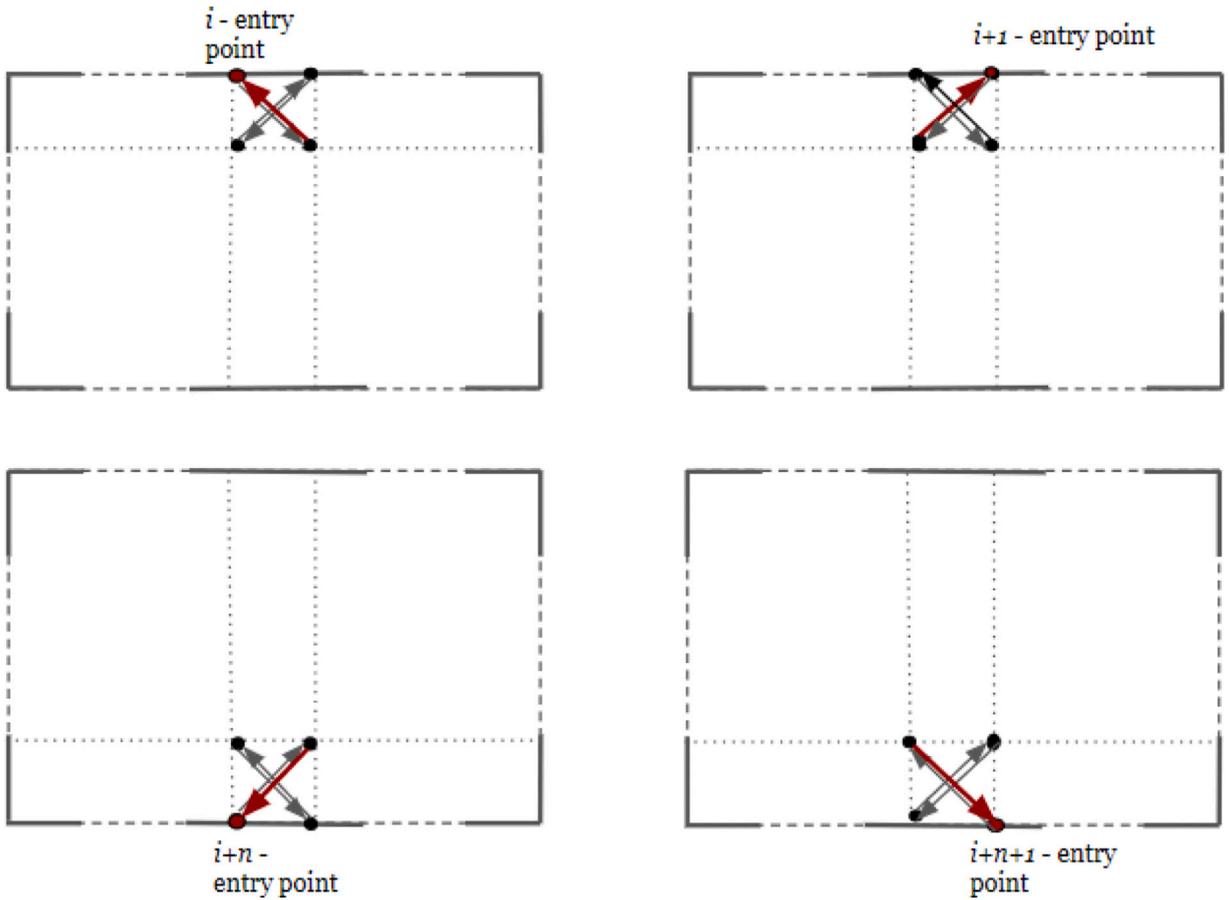


Fig. 16. Four cases with entry point (marked in red) in a grid square: missing edges are shown in red. The figures in order refer to Eqs. (21), (22), (23) and (24), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 Eqs. (20) does not exist. Thus, we add Eqs. (21)–(24). Fig. 16 illustrates the four cases depending on the location of the entry point.

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in \mathcal{P} \tag{21}$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \quad \forall i : i + 1 \in \mathcal{P} \tag{22}$$

$$x_{i,i+1+n} + x_{i+n+1,i} + x_{i+n,i+1} \leq 1 \quad \forall i : i + n \in \mathcal{P} \tag{23}$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i : i + n + 1 \in \mathcal{P} \tag{24}$$

B.5. Integration of temporal separation

So far, our MIP can compute an optimal (static) arrival tree (according to objective function (14)). In this subsection, we recapitulate the temporal separation we described in Dahlberg et al. (2018). The assumption there was that traversing any edge for any aircraft takes a unit time of u . We also assume that the time of arrival at the entry point of the dynamic-trajectories area is fixed for all aircraft: aircraft a arrives at entry point $b \in \mathcal{P}$ at time t_a^b . We introduce new variables that keep track of where aircraft are located at what time: binary variables $y_{a,j,t}$ that indicate whether aircraft a occupies vertex j at time t .

Additionally, apart from the indicator x_e for an edge e participating in the routes, we introduce indicators $x_{e,b}$ for the edge e participating in the route from entry point b to the runway (for all entry points $b \in \mathcal{P}$). We set the variables $x_{e,b}$ using Eqs. (25)–(28):

$$x_{e,b} \leq x_e \quad \forall b \in \mathcal{P}, \forall e \in E \tag{25}$$

$$\sum_{j:(b,j) \in E} x_{(b,j),b} = 1 \quad \forall b \in \mathcal{P} \tag{26}$$

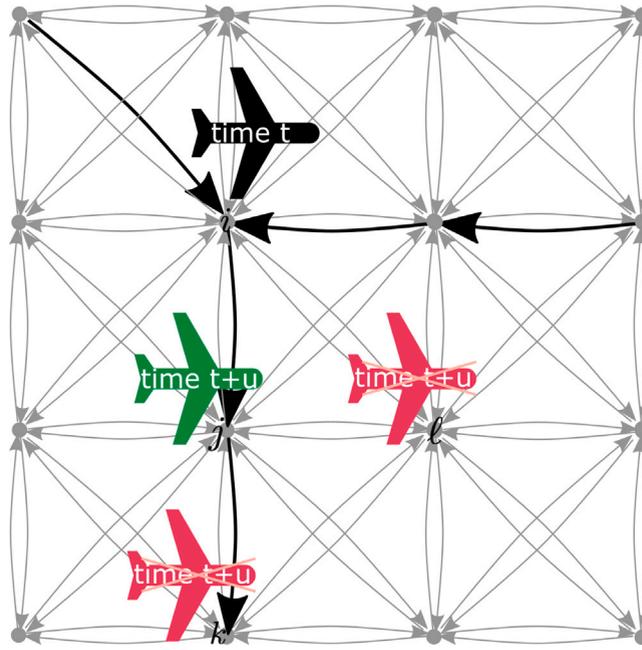


Fig. 17. Grid edges are shown in gray, edges of the arrival tree are shown in black. From its location at i at time t , the black aircraft a can reach the green position at j at time $t + u$, the red positions cannot be reached: l is not on any path, and for k we have $x_{j,k} = 1$, but a is not at j at time t .

$$\sum_{j:(j,r) \in E} x_{(j,r),b} = 1 \quad \forall b \in \mathcal{P} \tag{27}$$

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

$$\forall b \in \mathcal{P} \tag{28}$$

With this, we can set the new variables $y_{a,j,t}$. Let \mathcal{A}_b be the set of all aircraft arriving at entry point $b \in \mathcal{P}$, and $\mathcal{A} = \cup_{b \in \mathcal{P}} \mathcal{A}_b$. Moreover let $T = \{0, \dots, \bar{T}\}$ be the considered time window. Aircraft a arrives at entry point b at time t_a^b , hence, with Eq. (29), we set $y_{a,b,t_a^b} = 1$. Moreover, we can set several of the y -variables to zero: whenever we know that an aircraft cannot occupy the vertex at all or not at certain points in time. We ensure that an aircraft that does not arrive at entry point b will occupy b at no point in time with Eq. (30), and we enforce that an aircraft a arriving at b occupies b only at t_a^b using Eq. (31). Finally, in Eq. (32), we combine the y -variables and the $x_{e,b}$; any aircraft a can occupy a vertex j at any time t only if there exists an ingoing edge for j , that is, if j is located on a route. Thus, if j is a grid vertex but does not lie on any route, no aircraft a will occupy it at any time.

$$y_{a,b,t_a^b} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \tag{29}$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A} \setminus \mathcal{A}_b, \forall t \in T \tag{30}$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall t \in T \setminus \{t_a^b\} \tag{31}$$

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

$$\forall j \in V \setminus \mathcal{P}, \forall t \in T \tag{32}$$

We then forward the information on the times at which a arrives at nodes along the route from b to the runway. Eqs. (29)–(31) only set the variable $y_{a,j,t}$ for entry points, we still need to set the variable for all the other vertices along the arrival tree (Eq. (32) only enforces that for vertices that are not located on the arrival tree the y -variable will be equal to zero at all times and for all aircraft). 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 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 ($y_{a,j,t} = 0 \quad \forall j$ for which edge (j, k) exists in the path from b), a cannot reach k at $t + u$, and we set $y_{a,k,t+u} = 0$; see Fig. 17.

The forwarding of the temporal information on the aircraft along each arrival route can be formulated as:

$$\sum_{j:(j,k) \in E} x_{(j,k),b} \times y_{a,j,t} = y_{a,k,t+u}$$

$$\forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (33)$$

However, Eq. (33) is not a linear constraint (we multiply two binary variables), which we cannot add to our MIP. Hence, we define a new binary variable $z_{a,j,k,b,t}$ as the product of $x_{(j,k),b}$ and $y_{a,j,t}$ using Eqs. (34)–(36):

$$\begin{aligned} z_{a,j,k,b,t} &\leq x_{(j,k),b} && \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \\ &&& \forall t \in \{0, \dots, \bar{T} - u\} \end{aligned} \quad (34)$$

$$\begin{aligned} z_{a,j,k,b,t} &\leq y_{a,j,t} && \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \\ &&& \forall t \in \{0, \dots, \bar{T} - u\} \end{aligned} \quad (35)$$

$$\begin{aligned} z_{a,j,k,b,t} &\geq x_{(j,k),b} - (1 - y_{a,j,t}) && \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \\ &&& \forall t \in \{0, \dots, \bar{T} - u\} \end{aligned} \quad (36)$$

Using the auxiliary variables $z_{a,j,k,b,t}$, we can reformulate Eq. (33) as a linear constraint:

$$\begin{aligned} \sum_{j:(j,k) \in E} z_{a,j,k,b,t} - y_{a,k,t+u} &= 0 \\ \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \end{aligned} \quad (37)$$

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

$$\sum_{\tau=t}^{t+\sigma-1} \sum_{a \in \mathcal{A}} y_{a,j,\tau} \leq 1 \quad \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma + 1\} \quad (38)$$

B.6. 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. Thus, the assumption that traversing any edge takes u time units for all aircraft is no longer a valid assumption.

For each aircraft a , a set of speed profiles is given, $S(a)$, which contains speed profiles, p , of different lengths (that is, the speed profile is optimized for different route lengths). See Appendix A for the computation of these speed profiles. The speed profile determines the time to cover the first, second..., and subsequent segments of the route. So, for each aircraft a , $S(a)$ has speed profiles for routes of length (minimum number of edges in an entry-point-runway path) $\times L$ to (maximum number of edges in an edge-disjoint entry-point-runway path) $\times L$. We then need to ensure that on an entry-point-runway path with X edges a uses the speed profile p for a route of length $X \times L$. Here, we assume that the time to cover the n th segment of the route is constant, that is, the time is independent of the grid-edge that actually constitutes that segment in the arrival tree. This assumption is valid due to the fact that no wind is considered in the case-studies analyzed in this paper.

So far, we used binary variables $y_{a,j,t}$ to indicate whether aircraft a occupies vertex j at time t . This is no longer enough to keep track of the aircraft's location: the evolution of the time to cover a segment depends on the number of segments covered in the route until the current vertex—a profile p has a time for covering the first edge, another for covering the second edge, etc. Thus, we do not simply have to know which profile p is used, but what is the edge number along the route (which we will deduce by counting how many vertices lie before the vertex along the route). Hence, we substitute $y_{a,j,t}$ 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 . Let λ be an upper bound on the number of vertices in any path, $\mathcal{L} = \{1, \dots, \lambda\}$. We substitute Eqs. (29)–(32) by (and extend the set of constraints that set variables to zero):

$$\sum_{p \in S(a)} y_{a,b,p,1,t_a^b} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (39)$$

$$\begin{aligned} y_{a,b,p,k,t_a^b} &= 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \\ &\quad \forall p \in S(a), \forall k \neq 1 \in \mathcal{L} \end{aligned} \quad (40)$$

$$\begin{aligned} y_{a,b,p,1,t} &= 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \\ &\quad \forall p \in S(a), \forall t \in T \setminus \{t_a^b\} \end{aligned} \quad (41)$$

$$\begin{aligned} y_{a,b',p,k,t} &= 0 \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \\ &\quad \forall p \in S(a), \forall k \in \mathcal{L}, \forall t \in T \end{aligned} \quad (42)$$

$$\begin{aligned} y_{a',b,p,1,t_a^b} &= 0 \quad \forall b \in \mathcal{P}, \forall a' \neq a \in \mathcal{A}_b, \\ &\quad \forall p \in S(a) \end{aligned} \quad (43)$$

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

$$\forall p \in S(a), \forall k \in \mathcal{L}, \forall t \in T \tag{44}$$

$$y_{a,j,p,1,t} = 0 \quad j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \quad \forall p \in S(a), \forall t \in T \tag{45}$$

$$\sum_{p \in S(a)} y_{a,j,p,k,t} \leq 1 \quad j \in V, \forall a \in \mathcal{A}, \quad \forall k \in \mathcal{L}, \forall t \in T \tag{46}$$

Finally, we want to ensure that on an entry-point-runway path with X edges a uses the speed profile p for a route of length $X \times L$. First, we introduce a variable $\ell(b)$ that gives the length of the path from entry point to runway in the arrival tree—where we count the number of grid edges and not the physical length of the path:

$$\ell(b) = \sum_{(i,j) \in E} x_{(i,j),b} \quad \forall b \in \mathcal{P} \tag{47}$$

Then, we aim that for each aircraft a arriving at entry point b , we pick the speed profile of length $\ell(b)$ from $S(a)$. If $\ell(b)$ would be a parameter, we could simply state $y_{a,b,\ell(b),1,t_a^b} = 1$ and $y_{a,b,p,1,t_a^b} = 0 \forall p \neq \ell(b)$. However, $\ell(b)$ is a variable and we cannot use it as an index of another variable. Consequently, we introduce two auxiliary binary variables $\psi_{b,a,p}$ and $\phi_{b,a,p}$ to achieve the desired result. We will make sure that:

$$\psi_{b,a,p} = \begin{cases} 0 & p = \ell(b) \\ 1 & p \neq \ell(b) \end{cases} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{48}$$

Then we set:

$$y_{a,b,p,1,t_a^b} = 1 - \psi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{49}$$

Together with Eq. (39) this ensures that for at least one profile (and exactly for the one of the correct length) the variable y is set to one.

We enforce Eq. (48) by adding Eqs. (50)–(53) to the MIP:

$$-\psi_{b,a,p} \leq (\ell(b) - p + \frac{1}{2}) / \lambda + \lambda \cdot \phi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{50}$$

$$\psi_{b,a,p} \leq -\ell(b) + p + \frac{1}{2} + \lambda \cdot \phi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{51}$$

$$\psi_{b,a,p} \leq (\ell(b) - p) + \frac{1}{2} + \lambda(1 - \phi_{b,a,p}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{52}$$

$$-\psi_{b,a,p} \leq (-\ell(b) + p + \frac{1}{2}) / \lambda + \lambda(1 - \phi_{b,a,p}) \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \tag{53}$$

Finally, we need to reformulate the constraints we use to forward the information about when an aircraft arrives at which vertex along the arrival route: Eqs. (34)–(36) and (37). Instead of the binary variable $z_{a,j,k,b,t}$ being the product of $x_{(j,k),b}$ and $y_{a,j,t}$, we introduce two new indexes and use the binary variables $z_{a,j,i,b,p,k,t}$: representing 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. We substitute Eqs. (34)–(36) by Eqs. (54)–(56).

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq x_{(j,i),b} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \tag{54}$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq y_{a,j,p,k,t-u_{a,p,k}} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \tag{55}$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \geq x_{(j,k),b} - (1 - y_{a,j,p,k,t-u_{a,p,k}}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \tag{56}$$

The new version of Eq. (37) is:

$$\sum_{j:(j,i) \in E} z_{a,j,i,b,k,t-u_{a,p,k}} - y_{a,i,p,k+1,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), \forall k \in \mathcal{L}, \quad \forall i \in V \setminus \mathcal{P}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T}\} \tag{57}$$

We still use a uniform minimum temporal separation of σ time units between all pairs of consecutive aircraft at all vertices. To enforce this separation we add Eq. (58).

$$\sum_{\tau=t}^{t+\sigma-1} \sum_{a \in \mathcal{A}} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} y_{a,j,p,k,\tau} \leq 1$$

$$\forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma + 1\} \quad (58)$$

B.7. Consistency between trees of different time periods

The arrival trees are optimized for the currently arriving aircraft. Certain traffic might require longer arrival paths than traffic at other times. Hence, the arrival tree should be recomputed frequently to adapt to the aircraft currently moving in the dynamic-trajectories area. However, trees for consecutive time periods should potentially not differ too much. 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 Eqs. (59)–(60), and then limit the number of differing edges in Eq. (61) by parameter U :

$$ax_{ij} \leq x_{ij} - x_{ij}^{old} \quad \forall (j, i) \in E \quad (59)$$

$$ax_{ij} \leq x_{ij}^{old} - x_{ij} \quad \forall (j, i) \in E \quad (60)$$

$$\sum_{(i,j) \in E} ax_{ij} \leq U \quad (61)$$

B.8. Flexibility at the entry points of the dynamic-trajectories area

In this Subsection and in Appendix B.9, we introduce our new constraints. We start with allowing flexibility at the entry points of the dynamic-trajectories area. When two aircraft arriving consecutively at the same entry point undercut the required temporal separation at the entry point, this yields infeasibility already at the entry point, and they cannot be scheduled. On the other hand, we aim to schedule as many arriving aircraft with a CDO profile as possible, to ensure that, we allow deviation from the planned time at the entry points of the dynamic-trajectories area. That is, to schedule even such aircraft, instead of requiring that aircraft a arrives at its entry point b at the given time t_a^b (Eq. (39)), we allow for it to arrive in the time window $[t_a^{b,1}, t_a^{b,2}]$. Thus, we need to substitute Eqs. (39)–(41) and Eq. (48) by Eqs. (62), (63), (64) and (65), and add Eq. (66), which ensures that aircraft a cannot be at any node before $t_a^{b,1}$. Eq. (62) ensures that for each aircraft a and its entry point b , exactly one speed profile p is chosen and the aircraft arrives at exactly one time in $[t_a^{b,1}, t_a^{b,2}]$. Eqs. (63), (64) set several of the y -variables to zero, and Eq. (65) ensures that for the speed profile of the correct length (the length $\ell(b)$ of the arrival path from b) y is set to 1.

$$\sum_{t=t_a^{b,1}}^{t_a^{b,2}} \sum_{p \in S(a)} y_{a,b,p,1,t} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (62)$$

$$y_{a,b,p,k,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a),$$

$$\forall t \leq t_a^{b,1}, \forall k \neq 1 \in \mathcal{L} \quad (63)$$

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

$$\forall t \in T : t_a^{b,1} < t < t_a^{b,2} \quad (64)$$

$$\sum_{t=t_a^{b,1}}^{t_a^{b,2}} y_{a,b,p,1,t} = 1 - \psi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \quad (65)$$

$$y_{a,j,p,k,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a),$$

$$\forall t < t_a^{b,1}, \forall k \in \mathcal{L} \quad (66)$$

B.9. Separation with different wake turbulence categories

Finally, we include separation criteria based on the wake turbulence categories of the leading and trailing aircraft (that is, we deviate from a uniform σ). We use ICAO's aircraft categories: LIGHT (L), MEDIUM (M), HEAVY (H) (SUPER can easily be included in our concept as well). We define $C_1 = \{H,M\}$, and $C_2 = \{L\}$. Let $\sigma_{A,B}$ be the temporal separation if the leading aircraft is of category A and the trailing aircraft is of category B .

Each aircraft a is an element of either set A or B . We choose $\Omega = \max \sigma_{A,B}$. If the leading and trailing aircraft are of two different ($A \neq B$) or the same category, we enforce a temporal separation of $\sigma_{A,B}$ and $\sigma_{A,A}$ using a constraint of type (67) (and a constraint of

type (68) if $\sigma_{A,B} \geq 2$) for all categories A, B and a constraint of type (69) (and a constraint of type (70) if $\sigma_{A,A} \geq 2$) for all categories A , respectively.

$$\sum_{a \in B} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{t+\sigma_{A,B}-1} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot \sum_{a' \in A} \sum_{p' \in S(a')} \sum_{k' \in \mathcal{L}} y_{a',j,p',k',t}$$

$$\forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma_{A,B} + 1\} \tag{67}$$

$$\sum_{a \in B} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{\bar{T}} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot \sum_{a' \in A} \sum_{p' \in S(a')} \sum_{k' \in \mathcal{L}} y_{a',j,p',k',t}$$

$$\forall j \in V, \forall t \in \{\bar{T} - \sigma_{A,B} + 2, \dots, \bar{T}\} \tag{68}$$

$$\sum_{a \neq a' \in A} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{t+\sigma_{A,A}-1} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot y_{a',j,p',k',t}$$

$$\forall a' \in A, \forall p' \in S(a'), \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma_{A,A} + 1\} \tag{69}$$

$$\sum_{a \neq a' \in A} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{\bar{T}} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot y_{a',j,p',k',t}$$

$$\forall a' \in A, \forall p' \in S(a'), \forall j \in V, \forall t \in \{\bar{T} - \sigma_{A,A} + 2, \dots, \bar{T}\} \tag{70}$$

B.10. The complete MIP

To enhance readability, we present the complete MIP in the following set of equations. We do not state the variable ranges in the equations and we use the constraints for consistency between trees for consecutive time periods.

$$\min \beta \sum_{e \in E} \ell_e x_e + (1 - \beta) \sum_{e \in E} \ell_e f_e \tag{12}$$

$$s.t. \sum_{k:(k,i) \in E} f_{ki} - \sum_{j:(i,j) \in E} f_{ij} = \begin{cases} \sum_{b \in \mathcal{P}} w_b & i = r \\ -w_i & i \in \mathcal{P} \\ 0 & i \in V \setminus \{\mathcal{P} \cup r\} \end{cases} \tag{9}$$

$$x_e \geq \frac{f_e}{Q} \quad \forall e \in E \tag{9}$$

$$\sum_{k:(k,i) \in E} x_{ki} \leq 2 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{13}$$

$$\sum_{j:(i,j) \in E} x_{ij} \leq 1 \quad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{14}$$

$$\sum_{k:(k,r) \in E} x_{kr} = 1; \tag{15}$$

$$\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \forall i \in \mathcal{P} \tag{16}$$

$$c_e x_e + \sum_{f \in I_e} x_f \leq c_e \quad \forall e \in E \tag{17}$$

$$x_{i,j+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\} \tag{18}$$

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in \mathcal{P} \tag{19}$$

$$x_{i,j+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \quad \forall i : i+1 \in \mathcal{P} \tag{20}$$

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i : i+n \in \mathcal{P} \tag{21}$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i : i+n+1 \in \mathcal{P} \tag{22}$$

$$x_{e,b} \leq x_e \quad \forall b \in \mathcal{P}, \forall e \in E \tag{23}$$

$$\sum_{j:(b,j) \in E} x_{(b,j),b} = 1 \quad \forall b \in \mathcal{P} \tag{24}$$

$$\sum_{j:(j,r) \in E} x_{(j,r),b} = 1 \quad \forall b \in \mathcal{P} \tag{25}$$

$$\sum_{i:(i,j) \in E} x_{(i,j),b} - \sum_{k:(j,k) \in E} x_{(j,k),b} = 0 \quad \forall j \in V \setminus \{\mathcal{P} \cup r\}, \forall b \in \mathcal{P} \tag{26}$$

$$\sum_{t=t_a^{b,2}}^{t_a^{b,1}} \sum_{p \in S(a)} y_{a,b,p,1,t} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (60)$$

$$y_{a,b,p,k,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), \forall t \leq t_a^{b,1}, \forall k \neq 1 \in \mathcal{L} \quad (61)$$

$$y_{a,b,p,1,t_a^{b,1}} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), t \in T : t_a^{b,1} < t < t_a^{b,2} \quad (62)$$

$$y_{a,b',p,k,t} = 0 \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), \forall k \in \mathcal{L}, \forall t \in T \quad (40)$$

$$y_{a',b,p,1,t_a^{b,1}} = 0 \quad \forall b \in \mathcal{P}, \forall a' \neq a \in \mathcal{A}_b, \forall p \in S(a) \quad (41)$$

$$y_{a,j,p,k,t} \leq \sum_{\substack{i \in V: \\ (i,j) \in E}} x_{(i,j)} \quad \forall j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in S(a), \forall k \in \mathcal{L}, \forall t \in T \quad (42)$$

$$y_{a,j,p,1,t} = 0 \quad j \in V \setminus \mathcal{P}, \forall a \in \mathcal{A}, \forall p \in S(a), \forall t \in T \quad (43)$$

$$\sum_{p \in S(a)} y_{a,j,p,k,t} \leq 1 \quad j \in V, \forall a \in \mathcal{A}, \forall k \in \mathcal{L}, \forall t \in T \quad (44)$$

$$\ell(b) = \sum_{(i,j) \in E} x_{(i,j),b} \quad \forall b \in \mathcal{P} \quad (45)$$

$$\sum_{t=t_a^{b,1}}^{t_a^{b,2}} y_{a,b,p,1,t} = 1 - \psi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \quad (63)$$

$$y_{a,j,p,k,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), \forall t < t_a^{b,1}, \forall k \in \mathcal{L} \quad (64)$$

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

$$\psi_{b,a,p} \leq -\ell(b) + p + \frac{1}{2} + \lambda \cdot \phi_{b,a,p} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \quad (49)$$

$$\psi_{b,a,p} \leq (\ell(b) - p) + \frac{1}{2} + \lambda(1 - \phi_{b,a,p}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \quad (50)$$

$$- \psi_{b,a,p} \leq (-\ell(b) + p + \frac{1}{2}) / \lambda + \lambda(1 - \phi_{b,a,p}) \quad \forall b' \neq b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a) \quad (51)$$

$$\sum_{a \in \mathcal{B}} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{t+\sigma_{AB}-1} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot \sum_{a' \in \mathcal{A}} \sum_{\substack{p' \in \\ S(a')}} \sum_{k' \in \mathcal{L}} y_{a',j,p',k',t} \quad \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma_{A,B} + 1\} \quad (65)$$

$$\sum_{a \in \mathcal{B}} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{\bar{T}} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot \sum_{a' \in \mathcal{A}} \sum_{\substack{p' \in \\ S(a')}} \sum_{k' \in \mathcal{L}} y_{a',j,p',k',t} \quad \forall j \in V, \forall t \in \{\bar{T} - \sigma_{A,B} + 2, \dots, \bar{T}\} \quad (66)$$

$$\sum_{a \neq a' \in \mathcal{A}} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{t+\sigma_{AA}-1} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot y_{a',j,p',k',t} \quad \forall a' \in \mathcal{A}, \forall p' \in S(a'), \forall j \in V, \forall t \in \{0, \dots, \bar{T} - \sigma_{A,A} + 1\} \quad (67)$$

$$\sum_{a \neq a' \in \mathcal{A}} \sum_{p \in S(a)} \sum_{k \in \mathcal{L}} \sum_{\tau=t}^{\bar{T}} y_{a,j,p,k,\tau} \leq \Omega - \Omega \cdot y_{a',j,p',k',t} \quad \forall a' \in \mathcal{A}, \forall p' \in S(a'), \forall j \in V, \forall t \in \{\bar{T} - \sigma_{A,A} + 2, \dots, \bar{T}\} \quad (68)$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq x_{(j,i),b} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (52)$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \leq y_{a,j,p,k,t-u_{a,p,k}} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (53)$$

$$z_{a,j,i,b,p,k,t-u_{a,p,k}} \geq x_{(j,k),b} - (1 - y_{a,j,p,k,t-u_{a,p,k}}) \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall (j,i) \in E, \forall k \in \mathcal{L}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T} + 1\} \quad (54)$$

$$\sum_{j:(j,i) \in E} z_{a,j,i,b,p,k,t-u_{a,p,k}} - y_{a,i,p,k,t+1} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall p \in S(a), \forall k \in \mathcal{L}, \forall i \in V \setminus \mathcal{P}, \forall t \in \{u_{a,p,k} + 1, \dots, \bar{T}\} \quad (55)$$

Appendix C. Arrival trees: High and low-traffic case studies

Figs. 18 and 19 show the resulting arrival trees as a function of the entry-time-window offset for the high and low-traffic case studies respectively.

For the high-traffic case study, two trees are generated for the 1-h-period, where the black paths correspond to the first arrival tree and the blue paths correspond to the second arrival tree (the paths that are common to both trees are black).

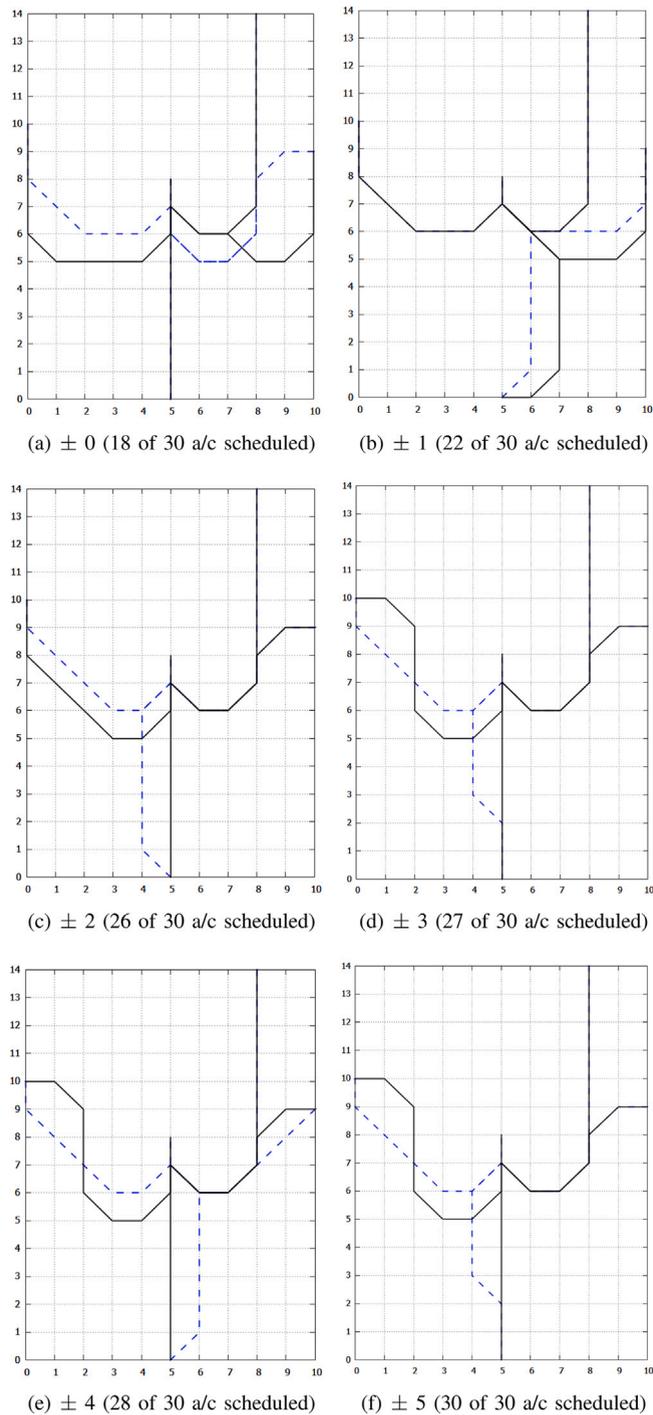


Fig. 18. Arrival trees as a function of the entry-time-window offset for the high-traffic case study (0% light aircraft). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

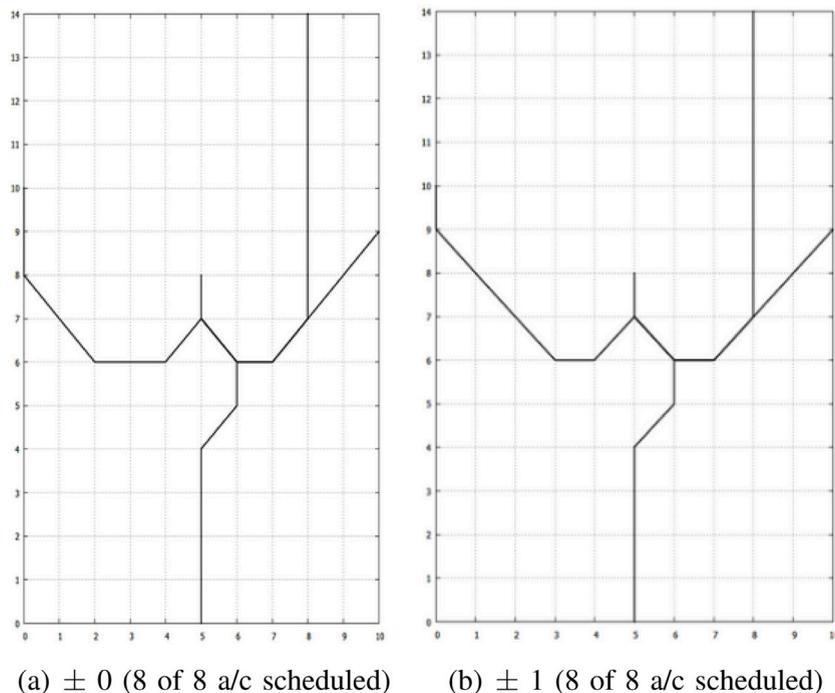


Fig. 19. Arrival trees as a function of the entry-time-window offset for the low-traffic case study (0% light aircraft). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

- Adacher, L., Pacciarelli, D., Pranzo, M., 2003. A Graph-Theoretical Model for the Aircraft Sequencing Problem. Università degli Studi Roma Tre, Dipartimento di Informatica e Automazione, Report n. RT-DIA-79-2003.
- Airbus, 1998. Getting to Grips with the Cost Index - Issue II. Tech. Rep., Flight Operations Support & Line Assistance.
- Airbus, 2002. Getting to grips with Aircraft Performance Monitoring. Flight Operations Support and Line Assistance, Blagnac Cedex, France.
- Andersson, T., Polishchuk, T., Polishchuk, V., Schmidt, C., 2016. Automatic design of aircraft arrival routes with limited turning angle. In: Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS).
- Bertsimas, D.J., van Ryzin, G., 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Oper. Res.* 39 (4), 601–615.
- Bronsvort, J., McDonald, G., Torres, S., Paglione, M., Young, C., Hochwarth, J., Boucquey, J., Vilaplana, M., 2016. Role of extended projected profile down-link to achieve trajectory synchronisation in support of trajectory based operations. In: 16th AIAA Aviation Technology, Integration, and Operations Conference (ATIO). Washington, D. C., <http://dx.doi.org/10.2514/6.2016-4356>.
- Chevalier, J., 2020. Departure and Arrival Route Optimisation Approaching Big Airports (Ph.D. thesis). Université Paul Sabatier.
- Choi, S., Robinson, J.E., Mulfinger, D.G., Capozzi, B.J., 2010. Design of an optimal route structure using heuristics-based stochastic schedulers. In: IEEE/AIAA 29th Digital Avionics Systems Conference (DASC).
- Christien, R., Hoffman, E., Zeghal, K., 2019. Spacing and pressure to characterise arrival sequencing. In: 13th ATM Seminar.
- 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., Wat, J., 2004. Continuous descent approach: Design and flight test for Louisville international airport. *J. Aircr.* (ISSN: 0021-8669) 41 (5), 1054–1066. <http://dx.doi.org/10.2514/1.5572>.
- Coppenbarger, R.A., Lanier, R., Sweet, D., Dorsky, S., 2004. Design and development of the en route descent advisor (EDA) for conflict-free arrival metering. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island. <http://dx.doi.org/10.2514/6.2004-4875>.
- Coppenbrager, R.A., Mead, R.W., Sweet, D.N., 2009. Field evaluation of the tailored arrivals concept for datalink-enabled continuous descent approach. *J. Aircr.* 46, 1200–1209. <http://dx.doi.org/10.2514/1.39795>.
- Dahlberg, J., Granberg, T.A., Polishchuk, T., Schmidt, C., Sedov, L., 2018. Capacity-driven automatic design of dynamic aircraft arrival routes. In: IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). (ISSN: 2155-7209) <http://dx.doi.org/10.1109/DASC.2018.8569646>.
- Dalmau, R., Prats, X., 2015. Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations. *Transp. Res. D* 62–71.
- Dalmau, R., Prats, X., 2016. Assessment of the feasible CTA windows for efficient spacing with energy-neutral CDO. In: International Conference on Research in Air Transportation (ICRAT).
- Dalmau, R., Prats, X., 2017. Controlled time of arrival windows for already initiated energy-neutral continuous descent operations. *Transp. Res. C* 85, 334–347. <http://dx.doi.org/10.1016/j.trc.2017.09.024>.
- Dalmau, R., Verhoeven, R., Gelder, N.D., Prats, X., 2016. Performance comparison between TEMO and a typical FMS in presence of CTA and wind uncertainties. In: IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, USA.
- Delgado, L., Prats, X., 2012. En route speed reduction concept for absorbing air traffic flow management delays. *J. Aircr.* 49 (1), 214–224. <http://dx.doi.org/10.2514/1.C031484>.
- Delgado, L., Prats, X., 2013. Effect of wind on operating-cost-based cruise speed reduction for delay absorption. *IEEE Trans. Intell. Transp. Syst.* 14 (2), 918–927.
- Erkelens, L., 2000. Research into new noise abatement procedures for the 21st century. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. Denver, CO, <http://dx.doi.org/10.2514/6.2000-4474>.
- Eurocontrol, 2017. DDR2 Reference Manual for General Users 2.9.4. Melike Atik.

- Federal Aviation Administration (FAA), 2019. NextGen Implementation Plan 2018-19. Tech. Rep. Washington, D.C.
- Flache, C., 2020. The R-WAKE project. <http://www.rwake-sesar2020.eu/>. (Accessed 20 November 2020).
- Garrido-López, D., D'Alto, L., Ledesma, R.G., 2009. A novel four-dimensional guidance for continuous descent approaches. In: IEEE/AIAA 28th Digital Avionics Systems Conference (DASC), Orlando, FL, USA.
- Goemans, M.X., Myung, Y., 1993. A catalog of Steiner tree formulations. *Networks* 23 (1), 19–28.
- Hardell, H., Lemetti, A., Polishchuk, T., Polishchuk, V., Vishwanath, B., Royo, E., 2020. Morphing STARs vs drones and weather in TMA. In: 9th International Conference for Research in Air Transportation (ICRAT).
- Hardell, H., Polishchuk, T., Smetanová, L., 2020. Fine-grained evaluation of arrival operations. In: 10th SESAR Innovation Days (SIDs), Virtual.
- Ho-Huu, V., Hartjes, S., A. Pérez-Castán, J., G. Visser, H., Curran, R., 2020. A multilevel optimization approach to route design and flight allocation taking aircraft sequence and separation constraints into account. *Transp. Res. C* 117, <http://dx.doi.org/10.1016/j.trc.2020.102684>.
- Iacus, S.M., Natale, F., Santamaria, C., Spyrtos, S., Vespe, M., 2020. Estimating and projecting air passenger traffic during the COVID-19 coronavirus outbreak and its socio-economic impact. *Saf. Sci.* (ISSN: 0925-7535) 129, 104791. <http://dx.doi.org/10.1016/j.ssci.2020.104791>.
- ICAO, 2016. Doc 4444, Procedures for air navigation services - Air Traffic Management, sixteenth ed. (ICAO), International Civil Aviation Organization, Montréal, Quebec, Canada.
- Krozel, J., Lee, C., Mitchell, J.S.B., 2006. Turn-constrained route planning for avoiding hazardous weather. *Air Traff. Control Quart.* 14 (2).
- Le Ny, J., Pappas, G.J., 2011. Joint metering and conflict resolution in air traffic control. *J. Guid. Control Dyn.* 34 (5), <http://dx.doi.org/10.2514/1.53253>.
- Michelin, A., Idan, M., L. Speyer, J., 2011. Mergin of air traffic flows. *J. Guid. Control Dyn.* 34 (1), <http://dx.doi.org/10.2514/1.51134>.
- OpenSky Network, 2019. <https://opensky-network.org>. (Accessed 2 December 2018).
- Ouelhadj, D., Petrovic, S., 2008. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* (ISSN: 1099-1425) 12 (4), 417.
- Park, S.G., Dutta, P., Menon, P.K., 2017. Optimal Trajectory Option Sets for In-Flight Climb-Descend Trajectory Negotiations. In: 17th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Denver, CO.
- Poles, D., Nuic, A., Mouillet, V., 2010. Advanced aircraft performance modelling for ATM: Analysis of bada model capabilities. In: IEEE/AIAA 29th Digital Avionics Systems Conference (DASC), Brétigny-sur-Orge (France). ISBN: 1424403782, <http://dx.doi.org/10.5772/2706>.
- Polishchuk, T., Polishchuk, V., Schmidt, C., Sáez, R., Prats, X., Hardell, H., Smetanová, L., 2020. How to achieve CDOs for all aircraft: Automated separation in TMAs - enabling flexible entry times and accounting for wake turbulence categories. In: 10th SESAR Innovation Days (SIDs).
- Prats, X., Dalmau, R., Verhoeven, R., Bussink, F., 2017. Human-in-the-loop performance assessment of optimized descents with time constraints. Results from full motion flight simulation and a flight testing campaign. In: Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar. Eurocontrol and FAA, Seattle, WA (USA).
- Prete, J., Krozel, J., Mitchell, J., Kim, J., Zou, J., 2008. Flexible, performance-based route planning for super-dense operations. In: AIAA Guidance, Navigation and Control Conference and Exhibit. American Institute of Aeronautics and Astronautics.
- Sáez, R., Prats, X., Polishchuk, T., Polishchuk, V., 2020a. Traffic synchronization in terminal airspace to enable continuous descent operations in trombone sequencing and merging procedures: An implementation study for frankfurt airport. *Transp. Res. C* 121, <http://dx.doi.org/10.1016/j.trc.2020.102875>.
- Sáez, R., Prats, X., Polishchuk, T., Polishchuk, V., Schmidt, C., 2020b. Automation for separation with CDOs: Dynamic aircraft arrival routes. *AIAA J. Air Transp.* 28 (4), 144–154. <http://dx.doi.org/10.2514/1.D0176>.
- Samà, M., D'Ariano, A., Palagachev, K., Gerds, M., 2019. Integration methods for aircraft scheduling and trajectory optimization at a busy terminal manoeuvring area. *OR Spectrum* 41, 641–681. <http://dx.doi.org/10.1007/s00291-019-00560-1>.
- Scharl, J., Berge, M.E., Coats, M.L., Haraldsdottir, A., Schoemig, E.G., 2007. Modeling and analysis of the 3D path arrival management concept. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Hilton Head, South Carolina. <http://dx.doi.org/10.2514/6.2007-6878>.
- SESAR Joint Undertaking, 2015. European ATM Master Plan. The Roadmap for Delivering High Performing Aviation for Europe. Brussels, Tech. Rep..
- SESAR Joint Undertaking, 2019. SESAR Solutions Catalogue 2019. Tech. Rep., SESAR, Luxembourg.
- SKYbrary, 2020. Mitigation of wake turbulence hazard, wake turbulence separation minima. https://www.skybrary.aero/index.php/Mitigation_of_Wake_Turbulence_Hazard. (Accessed 26 September 2020).
- Smetanová, L., 2020. Evaluation of Arrival Sequencing at Arlanda Airport (M. S. thesis). Linköping University and Czech Technical University.
- Takeichi, N., Inami, D., 2010. Arrival-time controllability of tailored arrival subjected to flight-path constraints. *J. Aircr.* 47 (6), 2021–2029. <http://dx.doi.org/10.2514/1.C000272>.
- Tetralith server, 2019. NSC. Linköping University, <https://www.nsc.liu.se/systems/tetralith/>. (Accessed on 28/01/2019).
- Toratani, D., 2019. Application of merging optimization to an arrival manager algorithm considering trajectory-based operations. *Transp. Res. C* 109, 40–59. <http://dx.doi.org/10.1016/j.trc.2019.09.015>.
- Toratani, D., K. Wickramasinghe, N., Itoh, E., 2018. Study on the arrival manager maximizing the benefit of four-dimensional trajectory based operations. In: 31st Congress of the International Council of the Aeronautical Sciences (ICAS), Belo Horizonte, Brazil.
- Visser, H.G., Wijnen, R.A., 2001. Optimization of noise abatement departure trajectories. *J. Aircr.* 38 (4), 620–627.
- Warren, A., Tong, K., 2002. Development of continuous descent approach concepts for noise abatement. In: IEEE/AIAA 21st Digital Avionics Systems Conference (DASC), Irvine, CA, ISBN: 0780373677.
- Wong, R.T., 1984. A dual ascent approach for Steiner tree problems on a directed graph. *Math. Program.* (ISSN: 0025-5610) 28 (3), 271–287.
- Zhou, J., Cafieri, S., Delahaye, D., Sbihi, M., 2014. Optimization of arrival and departure routes in Terminal Maneuvering Area. In: 6th International Conference for Research in Air Transportation (ICRAT), Istanbul, Turkey.