Automated Traffic Scheduling in TMA with Point Merge to Enable Greener Descents

Henrik Hardell
Communications and Transport Systems, ITN,
Linköping University (LiU), Norrköping, Sweden
Procedure Design Unit, Luftfartsverket (LFV),
Norrköping, Sweden, firstname.lastname@liu.se

Tatiana Polishchuk and Lucie Smetanová
Communications and Transport Systems, ITN,
Linköping University (LiU)
Norrköping, Sweden
firstname.lastname@liu.se

***Abstract*—We present a mixed-integer programming approach for fully-automated sequencing and merging of the arriving and departing traffic within the terminal areas implementing point merge. We assume all the arrivals are performing the most fuel-efficient continuous descent operations (descents with idle thrust and no speed-brakes usage), with the exception when the aircraft are flying along the sequencing legs of the point merge system.**

On example of a high-traffic scenario at Dublin airport, we demonstrate that our approach provides significant benefits, including increased vertical performance as well as reduced time and distance spent in the terminal airspace, contributing to fuel savings of up to 22%. The analysis is based on the historical ADS-B traffic data obtained from the Opensky Network.

***Index Terms*—Arrival and Departure Scheduling, Point Merge, Continuous Descent Operations, Integer Programming**

I. Introduction

Point merge procedures were designed to facilitate greener arrivals, including continuous descent operations (CDOs) and noise level reductions. However, suboptimal management does not allow to achieve the desired benefits [1]. The execution of a CDO is aircraft-specific and the optimal descent profile depends on the operating capabilities of the aircraft. As a consequence, predicting the behaviour of the incoming traffic in the terminal area is a difficult task.

In this paper, we demonstrate how automatic scheduling can help to enable more environmentally-friendly approaches in high-traffic scenarios, maintaining the required separation between aircraft at every point in the terminal maneuvering area (TMA). Our testbed is Dublin airport implementing point merge system for the runway 28L.

In previous work [2], [3], we uncovered significant inefficiencies in Dublin TMA, especially in the scenarios with high traffic. With this paper, we prove that a more efficient use of the point merge procedures may improve horizontal and vertical arrival performance, which consequently results in fuel savings. Other benefits include improved predictability of arrivals and departures, as well as an automated sequencing and merging of the aircraft, reducing the workload for air traffic controllers.

II. Related Work

Air traffic scheduling along the arrival routes was considered in [4], 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. The merging of traffic flows was studied in [5], 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 [6], various approaches were presented to integrate several modelling 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.

A Mixed-Integer-Programming-based (MIP-based) approach for the generation of optimized arrival routes to enable CDOs for all arriving aircraft, was presented in [7]. However, scenarios with a high-traffic demand resulted in infeasible solutions. To address the problem, in [8] the authors assumed aircraft could arrive within a given time window at the TMA entry point, which could be achieved by adjusting the speed during the en-route phase. In [1] the work was extended with a detailed explanation of the concept of operations (CONOPS) to show the applicability of the proposed approach in the current air traffic operations. Furthermore, the methodology was successfully tested on several scenarios with different traffic levels/distributions, uncovering the benefits of the proposed concept in terminal airspace operations In this paper, we adopt the aforementioned CONOPS and apply similar approach to enable more environmentally-friendly descents in TMAs with point merge operations.

In [9], the author explored the implementation of new-designed point merge procedures to one runway at Amsterdam-Schiphol airport, accompanied by an arrival management system, using MIP. Another MIP-based arrival scheduling was proposed in [10], applied to the trombone procedures at Frankfurt-Main airport, with the aim to maximise the number of neutral CDOs. The authors of [11] proposed a method for calculating the probability that an arriving aircraft will need to deviate from CDO and revert to the step-down descent approach (SDA).

Supported by the Swedish Transport Administration (Trafikverket) and in-kind participation of LFV within the ODESTA and TMAKPI projects.

III. Methodology

In this work, we assume the implementation of the CONOPS described in [1], which enables four-dimensional trajectory negotiation/synchronization process between the Air Traffic Control Officer (ATCO) and the aircraft, performed in the pre-sequencing area, while the aircraft are still in the en-route flight phase. ATCO up-streams the arrival route generated with the help of a ground support tool, which the Flight Management System (FMS) in the cockpit can use for constructing the optimal descent profiles. The optimized arrival routes allow the aircraft to fly neutral CDOs (descents with idle thrust and no speed-brakes usage) and to ensure a separation of the arrivals throughout the arrival procedure. In addition, our framework ensures separation between the arrivals and the departures for optimized runway utilization.

A. Arrival Routes

In the proposed framework, the arrival routes are generated based on the point merge STARs of Dublin runway 28L (Figure 1). The proposed approach is suitable for any other airport operating point merge sequencing and merging.

We start from constructing a set of feasible routes connecting the TMA entry points to the final merge point (LAPMO). We use the coordinates where the flight (obtained from the Opensky Network database [12]) enters the 50 NM boundary circle as the first point (since Dublin airport is not centered in the TMA, we use a 50 NM circle, centered at the runway, in order to capture the descent parts of the trajectories from the east), and then take the published waypoint along the TMA border, where the aircraft enters TMA, as the second waypoint. In case the actual aircraft does not pass over a pusblished waypoint at TMA entry, we use the nearest published waypoint for that. For the rest of the route, until reaching the point merge system, we use the published waypoints along the corresponding STAR.

Each arriving aircraft can be assigned one of the feasible routes, which differ in how long the aircraft stays on the sequencing leg of the point merge system. We are interested in finding the optimal point where the aircraft should leave the sequencing legs and initiate a turn towards the merge point. In order to define the set of route options, we create four equally-spaced waypoints between the five published waypoints along the sequencing legs, which in total results in 21 different points. Since the published waypoints along the sequencing legs are spaced by 7 NM for aircraft going in the northern direction, and 6 NM for aircraft going in the southern direction, the new waypoints are spaced by 1.4 and 1.2 NM, respectively.

We set KOGAX as the earliest waypoint where aircraft arriving from the north can turn towards the merge point, and SIVNA as the earliest point for aircraft arriving from the south. In combination with the rest of the 20 waypoints along the sequencing legs (including the published and new points), it creates 21 feasible arrival routes for each aircraft. The resulting set of arrival routes from the southern entry point (SUTEX) is illustrated in Figure 2.

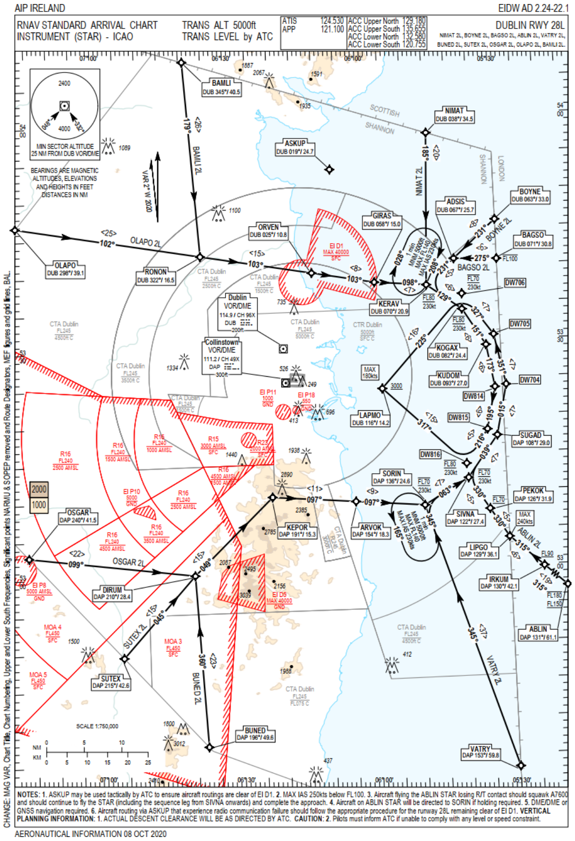


Fig. 1: Published point merge procedures at Dublin runway 28L (Source: Irish AIP [13]).

We do not consider realistic turns when changing track. Hence, aircraft turn to the new track instantly. We also ignore the final part of the flight, from LAPMO to the runway threshold.

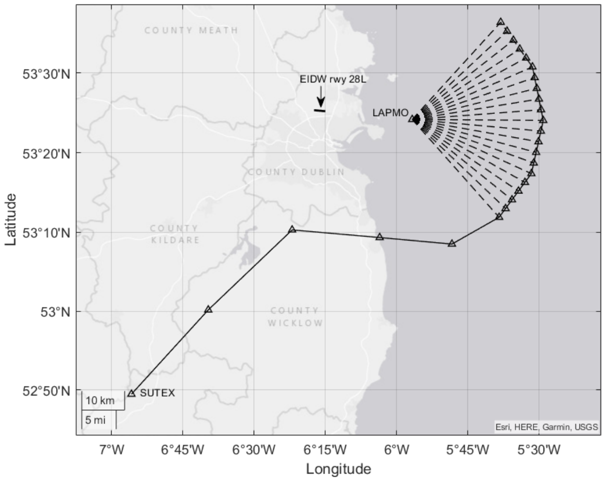


Fig. 2: Example of the 21 possible arrival routes for an aircraft entering the TMA via SUTEX. Common route is in solid line and the variable route parts are depicted as dashed lines.

B. Vertical Profiles

For each arrival route, we create a vertical profile that adhere to the restrictions valid for the published point merge procedures at Dublin airport. More specifically, the sequencing legs in the point merge system are to be flown level at the airspeed (IAS) of max 230 kt, at FL70 or FL80, depending on the flight direction.

We create vertical CDO profiles from the entry to the 50 NM boundary circle until it reaches the start of the point merge system. We match the initial IAS of the real flight, as well as consider the max IAS= 250 kt speed restriction below FL100. Next, we use a constant altitude at the sequencing legs, followed by a CDO to the merge point, to cross it at 3000 ft or above, at max IAS= 180 kt. We model the performance of the descent profiles using Base of Aircraft Data (BADA) v4.2 [14]. For the CDO parts of the profiles, we consider the idle-thrust descent without using speedbrakes, utilizing the BADA idle rating model. We calculate the engine idle thrust and drag at every timestamp and feed it into the Total Energy Model (Equation 1). We set the speed profile designed according to the speed schedule formulas provided in BADA, which we convert to the true airspeed (TAS) and feed into the TEM. From the TEM, we obtain the vertical speed (dh/dt) at every timestamp, starting from the merge point and calculating backwards. By calculating the vertical speed along the trajectory, we obtain the full vertical profile of the CDO parts.

For the level flight parts, where additional thrust is required, we use TEM to find the required thrust. Unlike the CDO case, we know that the vertical speed is zero when the aircraft is not descending. We choose 90% of the maximum landing weight for each aircraft type, specified in BADA.

$$(Th - D) \cdot V_{TAS} = m \cdot g_0 \cdot \frac{dh}{dt} + m \cdot V_{TAS} \cdot \frac{dV_{TAS}}{dt} \qquad (1)$$

Here, Th is the thrust force, D is the drag force, V_{TAS} is the true airspeed, m is the aircraft mass and g_0 is the gravitational acceleration.

We use ECMWF [15] ERA5 reanalysis dataset, provided via the C3S Data Store, to obtain historical data on temperature and wind at different altitudes and positions for imitating the prevailing atmospheric conditions and for conversion between ground speed (GS) to TAS.

C. Optimization Problem

We model the aircraft scheduling problem as MIP. We use the formulation of the landing problem proposed in [10] for trombones as a base, adjusting it to the point merge systems and extending with a set of additional constraints for simultaneous scheduling of arrivals and departures.

1) Arrivals: We input the entry times to the TMA according to the historical data and allow for a flexible arrival times within the fixed-size window. The aircraft profiles are pre-computed (as explained in Section III-B) and are used as the input to the problem. The size of the flexible arrival time window defines the resulting number of profiles per aircraft, as it alters the required times of arrival (RTAs) at the waypoints along the route. Hence, for every additional minute we add to the time window (both before and after the original time), two times the number of original profiles are added to the set of profiles, for each aircraft. In practice, the actual time of arrival is controlled by the speed adjustment in the en-route phase. In [16], the authors investigated the use of en-route speed reduction as a measure for addressing capacity constraints at the airports, or for absorbing air traffic flow management delays, and found that this can be done without increasing the fuel consumption.

Let $\mathcal{A}$ be the set of all aircraft scheduled to land at the airport during a certain period of time. Then, let $\mathcal{P}_a$ be the set containing all possible profiles (a finite number of profiles per available route) that can be flown by a given aircraft a. Finally, let $\mathcal{I}$ be the set of the aircraft-profile pairs (e.g. $((a_i, p_k), (a_j, p_r)))$ representing the incompatibilities between profiles, meaning that there will be a loss of time separation between aircraft a_i and a_j if aircraft a_i flies profile p_k and aircraft a_j flies profile p_r.

To obtain the incompatible profiles, we start by identifying the violations in the required time separation t_{sep0} at the waypoints shared by the profile pairs. If the separation is violated, we mark the pair as incompatible and include it into the set $\mathcal{I}$. Algorithm I illustrates the pseudocode for the deconfliction at the pre-processing step.

Algorithm 1 Deconfliction pre-processing step

```
for each pair of a/c do
    for $i = 1$ to #profiles of a/c 1 do
        for $j = 1$ to #profiles of a/c 2 do
            for $k = 1$ to #waypoints in profile i do
                for $l = 1$ to #waypoints in profile j do
                    if $i$ and $j$ share a waypoint $w$ then
                        if difference in time at $w < t_{sep0}$ then
                            mark the profile pair as conflicting
                        end if
                    end if
                end for
            end for
        end for
    end for
end for=0
```

The estimated time of arrival ETA$_a$ for each aircraft is the time the aircraft arrives at the merge point if it flies along the shortest route from the entry to the 50 NM circle to the runway. The required time of arrival RTA$_{a,p}$ is the arrival time at the metering fix. The vertical part to the direct route, that, in combination with the speed profile, determines the required travel time, is calculated as a CDO, using part of the methodology explained in Section III-B.

Let $x_{a,p}$ be the binary decision variables, which equal 1 when aircraft a flies profile p, and 0 otherwise. The constraints corresponding to the aircraft arrivals are shown in Equations (2) and (3). Equation (2) ensures that if there are incompatibilities between two aircraft flying two profiles, at most one aircraft will be allowed to fly its profile. Otherwise, there would be a loss of separation. Equation (3) ensures that each aicraft fly only one profile.

$$x_{a_i,p_k} + x_{a_j,p_r} \leq 1,\ \forall a_i, a_j \in \mathcal{A},\ \forall p_k \in \mathcal{P}_{a_i},\ \forall p_r \in \mathcal{P}_{a_j} \mid ((a_i, p_k), (a_j, p_r)) \in \mathcal{I} \quad (2)$$

$$\sum_{p \in \mathcal{P}_a} x_{a,p} = 1, \quad \forall a \in \mathcal{A} \quad (3)$$

The objective is to minimize the difference between $\text{RTA}_{a,p}$ and ETA_a for each particular flight, assuming that this way the cost of the operation is minimized:

$$\min \quad J := \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}_a} x_{a,p} \cdot |\text{RTA}_{a,p} - \text{ETA}_a| \quad (4)$$

Late and early arrivals are penalized symmetrically.

2) Departures: Since Dublin uses a single runway for both arriving and departing aircraft, the runway is not available to accept arriving aircraft at a constant rate. Based on the analysis of the Opensky data, we assume a minimum of $t_{sep1} = 1.5$ minutes separation between the times when the departing aircraft initiates its takeoff roll, until the succeeding arriving aircraft touches down on the runway. Furthermore, we assume that a departing aircraft may not initiate its takeoff earlier than t_{sep1} after a preceding, arriving aircraft has landed. The two separation assumptions set a minimum separation of $t_{sep2} = 3$ minutes between two arriving aircraft, in order to provide for an intermediate departure. Note that this separation requirement differs from the minimum separation required between two consecutive arrivals in the air t_{sep0}.

Since our optimized arrival trajectories end at the merge point, LAPMO, we need to make an assumption of the time it takes for the aircraft to reach the runway threshold. According to the Opensky data, the average time is about $t_{land} = 4$ minutes, which is the time we need to add to the arrival time at LAPMO to find the time the aircraft reaches the runway.

Let $\mathcal{D}$ be the set of aircraft scheduled to depart within the given period of time, corresponding to the arrival schedule. In order to add some flexibility to the departure times, we use X minutes time window, which allow aircraft to takeoff a maximum of X minutes after its original take off time of the aircraft i, $t_{dep_{os_i}}$, obtained from the Opensky data network. This time window is controlled by the following constraints:

$$t_{dep_{os_i}} \leq t_{dep_i} \leq t_{dep_{os_i}} + X;\ i \in \mathcal{D} \quad (5)$$

where t_{dep_i} is the variable departure time for aircraft i. We consider the departure time delay only, assuming that the real aircraft lined up on the runway immediately, without any prior waiting at the holding point, hence, taking off earlier would not be possible.

The flexible departure times are modeled using two pairs of constraints for each departing aircraft, where the asymmetric constraints in Equations (6) ensure the separation between the departing aircraft and arrivals before and after, while a pair of symmetric constraints expressed in Equations (7) provide a required separation of $t_{sep} = 1.5$ minutes between the two consecutive departures. For an aircraft landing prior to a departure, we obtain the latest allowed RTA at LAPMO by deducting both the fixed time it takes from the merge point to the runway (t_{land}) and the required separation (t_{sep1}), from the departure time. For an aircraft landing after a departure, we calculate the earliest allowed RTA at LAPMO by deducting the fixed time it takes from LAPMO to the runway (t_{land}), from the departure time, adding the required separation (t_{sep1}).

$$\begin{aligned} t_{land} + t_{sep1} + x_{a,p} \cdot RTA_{a,p} - t_{dep_i} &\leq M \cdot y_{i,a,p}; \\ & i \in \mathcal{D}, a \in \mathcal{A}, p \in \mathcal{P}_a \\ t_{dep_i} - x_{a,p} \cdot RTA_{a,p} + t_{land} - t_{sep1} &\leq M \cdot (1 - y_{i,a,p}); \\ & i \in \mathcal{D}, a \in \mathcal{A}, p \in \mathcal{P}_a \end{aligned} \quad (6)$$

where t_{dep_i} is a variable for the departure time of each aircraft and $y_{i,a,p}$ is a binary variable that activates one of the mutually exclusive constraints. M is a very large number.

$$\begin{aligned} t_{dep_i} - t_{dep_j} + t_{sep2} &\leq M \cdot y_i;\ i \in \mathcal{D}, j \in \mathcal{D} \setminus \{i\} \\ t_{dep_j} - t_{dep_i} + t_{sep2} &\leq M \cdot (1 - y_i);\ i \in \mathcal{D}, j \in \mathcal{D} \setminus \{i\} \end{aligned} \quad (7)$$

The described MIP is NP-hard in general [10], but selected instances corresponding to the real operational scenarios are of relatively small size and are solved in reasonable time using Gurobi solver, as presented in Section IV.

D. Performance Metrics

We choose the following set of performance indicators to compare the arrival performance of the optimized flights scheduled with the proposed optimization framework, and the actual flights using the historical trajectories obtained from the Opensky network. For more details on the performance metrics, refer to [2], [17].

1) Entry Conditions: For evaluating the entry conditions, we calculate the minimum time to final, spacing deviation, sequence pressure, throughput and metering effort indicators defined as follows.

Minimum time to final. We plot all the flown trajectories of the given dataset and overlay a rectangular grid with the cell side of ≈ 1 NM, over the TMA and calculate the minimum time needed from any point within the cell of the grid to the final (merge point) along any of the aircraft trajectories passing through the cell. We assign infinite (or a very large value) to the cells through which no trajectories pass during the considered time period. For visualisation of the resulting assignment, we plot a *heatmap* of the minimum time to final on a grid.

Spacing deviation. 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) at time

t is calculated for a pair of aircraft tagged as the leader and the trailer. The leader is the aircraft that arrives at the final point first, and the trailer is the aircraft that arrives second. The spacing deviation is calculated using the following equation:

$$sd(t) = min\_time(\text{trailer}(t)) - min\_time(\text{leader}(t - s_{rwy})),$$

where s_{rwy} is the temporal separation at the runway, and *min_time* is the minimum time to final. The spacing deviation reflects information about the control error, which is the accuracy of spacing around the airport.

Sequence pressure for an aircraft at time t is the number of aircraft with the same time to final within a given time window ω; it reflects the aircraft density at different time *t*. It is calculated for each aircraft at any time of its presence within the TMA with the discrete time steps. In this work, we set $\omega = 120s$.

Throughput at a given time horizon t is calculated by counting the number of aircraft with the minimum time to final within a given time window. In this work we calculate the throughput crossing iso-minimum time lines from 900 to 30s to final, sampled at a 30s rate over 5-minute periods.

Metering effort is defined as the difference between the throughput at the given time horizon and the one close to the final (30s in this work). It quantifies the controllers effort for metering, and may be used as a proxy to controllers workload.

2) Horizontal Flight Efficiency: is assessed by calculating the **Horizontal distance** inside the 50 NM boundary circle from the entry to the merge point.

In addition, we consider the **Horizontal spread** to estimate the surface of the terminal area occupied by the flights and to quantify the dispersion of the arrival flows. It is calculated as the ratio of the number of cells through which at least one trajectory passes to the total number of grid cell covering the TMA. A smaller Horizontal spread indicates that the aircraft mainly follow similar arrival paths.

3) Time in TMA: We calculate the arrival **Time in TMA** as the difference between the time at the merge point and the time when aircraft enter the 50 NM boundary circle around the airport.

4) Vertical Flight Efficiency: The **Time flown level** is calculated using the technique proposed by EUROCONTROL in [18] with small changes. We identify a level segment when the aircraft is flying with a vertical speed below 300 feet per minute for at least 30 seconds, and these 30 seconds are subtracted from each level duration. Only level flight segments after top of descent (ToD) are considered, which means that we ignore any parts where aircraft are still on their cruise altitude in TMA.

5) Fuel Efficiency: is used to assess the *environmental impact* of the arrivals. We calculate the **Fuel consumption** according to the formula provided in the BADA manual (Equation 8).

$$F = \delta \cdot \theta^{\frac{1}{2}} \cdot m \cdot g_0 \cdot a_0 \cdot L_{HV}^{-1} \cdot C_F \tag{8}$$

Here, δ is the pressure ratio, θ is the temperature ratio, m is the reference mass, g_0 is the gravitational acceleration, a_0 is the speed of sound at sea level, L_{HV}^{-1} is the fuel lower heating value and C_F is the fuel coefficient. For the CDO part of the descent profile, we use the idle thrust fuel coefficient, and for the part where additional thrust is required, we obtain the fuel coefficient via the thrust, obtained in Equation (1). We use the weather data described in Section III-B for the fuel calculations.

IV. Experimental Evaluation

In this section, we demonstrate how to apply the proposed methodology on example of a high-traffic one-hour scenario at Dublin airport implementing point merge techniques. Using a set of performance metrics, we evaluate the arrival performance based on the optimized flight trajectories and compare it against the performance of the actual historical flights.

A. Dataset

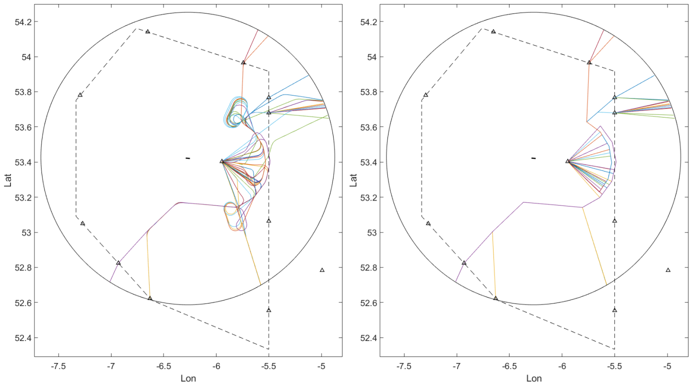
The example scenario is based on the historical data obtained from the Opensky Network [12], [19], which provides an open-source data in a form of aircraft state vectors for every second of the trajectories inside the terminal area.

We analysed the Opensky data for the year 2019 and identified the busiest month of 2019 to be October, and choose one of the busiest hours on October 4, 16:00-17:00 UTC, with 32 arriving aircraft in TMA during this hour. The actual trajectories of the arriving aircraft are shown in Figure 3 (a). All aircraft belong to the Medium wake turbulence category, four of which are turboprop aircraft and the rest are propelled by jet engines. Given this set of aircraft, we use $t_{sep0} = 120$ s according to [20]. In addition, there are five Medium wake turbulence category jet aircraft taking off from the same runway during this hour.

The Opensky data for 2019 does not cover aircraft movements on the ground at Dublin airport, but the first recording is registered just outside the departure end of the runway while the aircraft is climbing. In order to estimate the actual takeoff time, we assume a takeoff roll duration of 30 seconds, followed by a 10 seconds climb, before the aircraft is first detected by an Opensky Network connected ADS-B receiver. We set the flexible time window for the departures to $X = 60$ s.

B. Optimized Arrival Schedule

We solve our MIP using the Gurobi optimization solver installed on a powerful Tetralith server [21], utilizing Intel HNS2600BPB computer nodes with 32 CPU cores, 384 GB, provided by the Swedish National Infrastructure for Computing (SNIC). We set the entry time window to ± 2 minutes, which represents the minimum time window to allow all 32 aircraft in this scenario to be scheduled. This results in 4704 profiles, for which we run a deconfliction program in Matlab, executed on the same machine as for the MIP. The number of profiles contributes significantly to the execution time of the deconfliction calculations, taking about 51 hours for 4704 profiles, while the MIP itself is solved in less than 15 seconds. In general, the pre-processing part is solved in polynomial time, $\mathcal{O}(n^2)$, where n is the number of profiles. The time to check whether an aircraft



(a)

(b)

Fig. 3: Actual (a) and optimized (b) arrival trajectories.

pair is in conflict is constant and does not depend on the input size. The computational complexity of the deconfliction at the pre-processing step can be reduced by considering smaller time periods when aircraft are mutually present in the TMA. In fact, we have improved the complexity of the deconfliction from the previous work [1], where it was modeled as a part of MIP.

Table I presents the final schedule for the arriving and departing aircraft, obtained through the optimization. We see that in the resulting schedule a minimum separation of 2 minutes is provided at every waypoint, as well as that there is at least 1.5 minutes between an arriving and a departing aircraft, as well as 1.5 minutes between the two consecutive departures. The deviations from the the initial times of arrival to TMA and from the departure times are also presented in the table (row 2), with the average of 84 seconds for arrivals and 29 seconds for departures. The corresponding optimized arrival routes are shown in Figure 3 (b).

The minimum time separation at the runway, between two arriving aircraft, is 2 minutes, with the maximum (not considering the extra separation to allow for departures) at 2.23 minutes and 2.32 in the average, considering all arriving aircraft. As we are ensuring extra spacing to allow for departures, an average spacing of 2 minutes at the runway is not achievable. If we compare to a separation of 2 minutes between two arriving aircraft, and given the scheduling of departing aircraft presented in Table I, the minimum achievable average separation between two arriving aircraft is 2.18 minutes. Our resulting schedule is quite close to this, indicating high runway utilization.

C. Performance Evaluation

1) Time in TMA: The comparison of the time aircraft spend in TMA for the actual and optimized arrival flights is shown in Figure 5 (a) and in Table II. According to this metric, the arrivals following optimized trajectories spend less time in TMA. Similar trend is also captured by the minimum time to final metric, illustrated with the heatmaps in Figures 6 (a)-(b).

2) Horizontal Flight Efficiency: The statistics for the distance in TMA metric, for the actual and optimized flights, is shown in Figure 5 (b), with average values presented in Table II. We can see that this metric also takes lower median, average and variation values for the optimized trajectories in comparison to the actual ones.

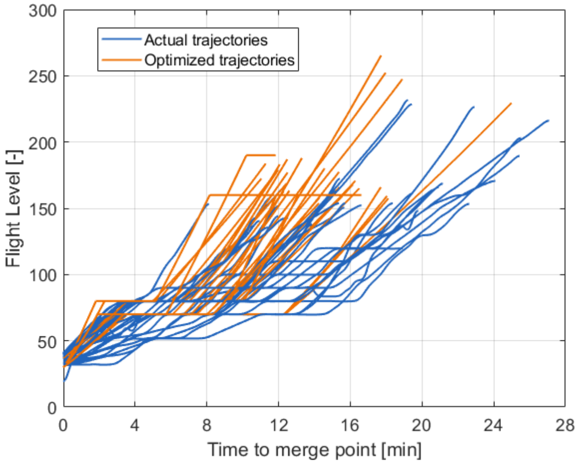


Fig. 4: Vertical profiles of the actual and optimized trajectories.

3) Vertical Flight Efficiency: In Figure 4, we observe that the optimized trajectories, in general, are steeper than the real profiles, for the part of the flight until leveling off at the sequencing legs. As the vertical profiles of the optimized arrivals have to adhere to the level flight segments at the sequencing legs, there is no possibility to keep the CDOs along the whole arrival. However, there are no other level segments but those mentioned, apart from any initial cruise phase in TMA, for aircraft with a low cruise altitude.

The distribution of the time flown level metric (expressed in %), for the actual and optimized scenario, is shown in Figure 5 (c), with average values presented in Table II. We observe that the median and average time flown level is greater for the optimized flights compared to the actual ones, while the variation is smaller. However, for the optimized scenario, the only part of the vertical profile that is level, corresponds to the sequencing legs. The results indicate that the optimized arrivals spend a greater portion of their total time in TMA on the sequencing legs, compared to the actual trajectories. In terms of the absolute values, the average time flown level is 4.42 minutes for the actual trajectories and 4.38 minutes for the optimized trajectories.

4) Fuel Efficiency: The distribution of the fuel consumption, for the actual and optimized scenario, is shown in Figure 5 (d), with average values presented in Table II. Even though the absolute values of the time flown level is similar for the actual and the optimal arrivals, there is a significant fuel saving in the optimized scenario, providing for a reduction of 22%. Hence, most of the fuel savings for the optimized arrivals can be attributed to other factors, such as better idle thrust CDO execution and less average time and distance spent in TMA in total, compared to the actual trajectories.

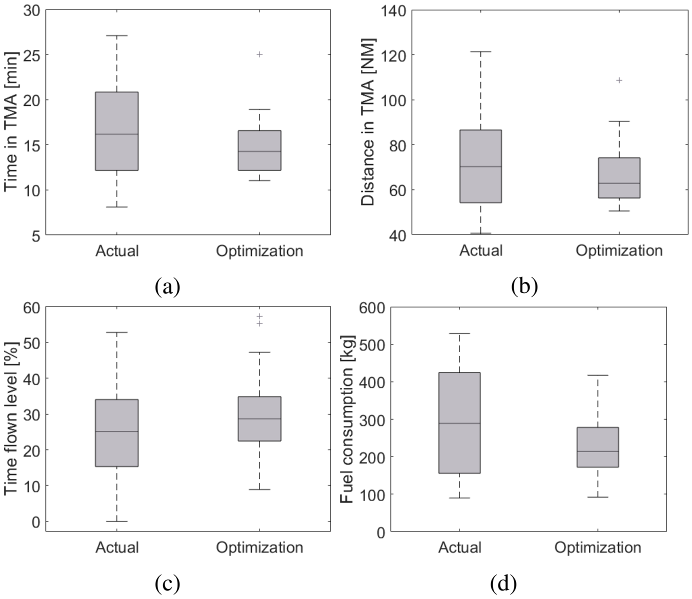
TABLE I. Optimized schedule for the 32 arriving (arr) and 5 departing (dep) aircraft in Dublin TMA between 16:00 to 17:00 on October 4, 2019. Times are expressed in MIN:SEC. Times followed by ^{1} belong to hour 15 and times followed by ^{2} belong to hour 17. Aircraft are ordered by the time they use the runway. Note that the entry point VATRY falls outside the 50 NM boundary circle, hence, VATRY in this table correpsonds to the intersection between the VATRY-SORIN route and the 50 NM circle.

	arr 1	arr 2	arr 3	arr 4	arr 5	arr 6	dep 1	arr 7	arr 8	arr 9	arr 10	dep 2	dep 3	arr 11	arr 12	arr 13	arr 14	arr 15	arr 16	arr 17
Time shift [s]	-120	-60	-120	-120	-120	-120	1	-60	0	-120	-120	43	58	-120	-120	-120	-120	120	60	-120
SUTEX																				20:40
BUNED								55:48^1						05:42						
VATRY	48:33^1		51:57^1						58:09^1	05:29	07:48					17:19				
BAGSO		53:37^1		56:25^1	00:31												21:47	24:00	26:41	
BOYNE						02:57														
NIMAT															17:22					
DIRUM								59:44 ^{1}						09:48						22:49
ADSIS		54:42^1		57:30^1	01:39	04:28											22:54	25:06	27:48	
SORIN	54:40^1		57:29^1					06:21	03:33	11:02	13:24			16:37		22:38				29:04
KERAV		55:46^1		58:34^1	02:43	05:32									21:30		23:57	26:09	28:51	
LAPMO	00:23	02:28	04:32	06:33	08:39	10:42		13:46	15:51	18:05	20:05			24:36	26:36	28:42	30:55	33:07	35:19	37:23
THR 28L	04:23	06:28	08:32	10:33	12:39	14:42	16:16	17:46	19:51	22:05	24:05	25:36	27:06	28:36	30:36	32:42	34:55	37:07	39:19	41:23

	dep 4	arr 18	dep 5	arr 19	arr 20	arr 21	arr 22	arr 23	arr 24	arr 25	arr 26	arr 27	arr 28	arr 29	arr 30	arr 31	arr 32
Time shift [s]	3	60	40	0	120	60	60	-120	0	-60	-60	120	0	-120	0	60	120
SUTEX																	
BUNED									28:58								
VATRY						29:36				41:50	44:04				48:50		
BAGSO		29:17					37:28	43:08					55:28	52:53		57:41	59:46
BOYNE					33:44							48:23					
NIMAT				29:20													
DIRUM									33:47								
ADSIS		30:23			35:15		38:44	44:19				50:03	56:40	54:09		58:56	01:03^2
SORIN						35:23			41:40	47:28	50:20				54:41		
KERAV		31:27		33:28	36:18		39:51	45:24				51:13	57:45	55:17		00:03^2	02:10^2
LAPMO		40:30		43:30	45:35	47:39	49:39	51:52	54:01	56:07	58:19	00:27^2	02:41^2	04:49^2	06:57^2	09:07^2	12:16^2
THR 28L	42:53	44:30	46:00	47:30	49:35	51:39	53:39	55:52	58:01	00:07^2	02:19^2	04:27^2	06:41^2	08:49^2	10:57^2	13:07^2	16:16^2

TABLE II. Average time, distance, fuel burn and time flown level for the actual and optimized arrival trajectories

	Average time [min]	Average distance [NM]	Average time flown level [%]	Average fuel burn [kg]
Actual	17.0	73.0	23.7	294.5
Optimization	14.7	66.9	29.8	229.3

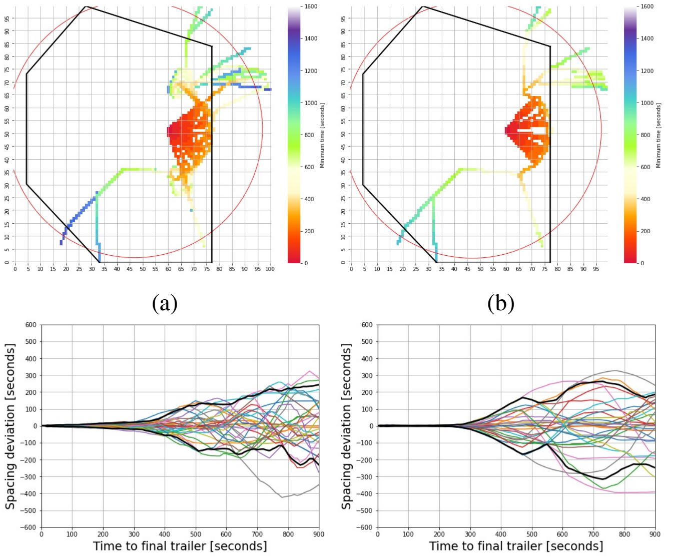


(a) (b)

(c) (d)

Fig. 5: Time in TMA (a), distance in TMA (b), time flown level (c) and fuel consumption (d) for the actual and optimized trajectories.

5) Entry Conditions: Figures 6 (a)-(b) show the minimum time to final heatmaps for the actual and optimized arrival trajectories. We see that the results are in line with what we obtained for the time in TMA (Figure 5 (a)). We observe that the optimized arrivals marginally reduce the horizontal spread of the trajectories (10.3% and 7.2% for the actual vs. optimized trajectories), due to the fact that the arrivals follow only the published point merge procedures and are not put in holdings. Moreover, the maximum and average throughput (7 and 2.54 for the actual trajectories) is lower for the optimized arrivals (5 and 2.34), as illustrated in Figures 7 (c)-(d). The corresponding results for the metering effort (with the maximum at 3 for the actual trajectories and reduced to 2 for the optimized ones) confirm this finding, which roughly indicates that less effort would be required to sequence and merge the arrival traffic flows along the optimized routes from the entry to the final point.

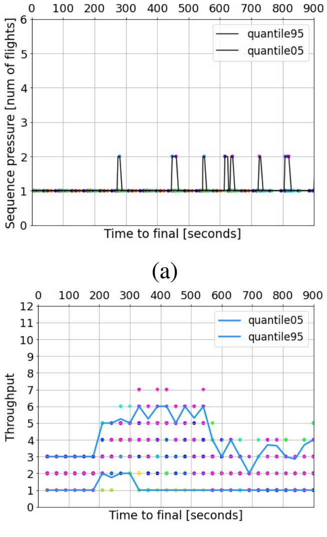


(a) (b)

(c) (d)

Fig. 6: Minimum time to final heatmap and spacing deviation for actual (a,c) and optimized trajectories (b,d).

Despite that sequence pressure indicator does not vary much when we compare the corresponding statistics for the actual versus optimized flights (maximum and average are 2 and 1.03 for the actual trajectories, and 2 and 1 for the optimized ones), Figures 7 (a)-(b) demonstrate that in the optimized scenario there are less occurrences with two aircraft sharing the window of 120 seconds at any point within TMA, which indicates that the traffic is more uniformly distributed in time within TMA. Spacing deviation evolution curves in Figures 6 (c)-(d) support this observation: for the optimized flights they converge to 0 quite quickly around 200 s to final and less outliers are observed outside of the *95-5*-th quantile band. In general, we may conclude that the traffic flows scheduled with the proposed optimization framework are better organized in space and time, requiring less control effort in TMA.



(a) (b) (c) (d)

Fig. 7: Sequence pressure and throughput for the actual (a,c) and optimized arrivals (b,d).

V. Conclusions and Future Work

Point merge was initially designed to facilitate greener arrivals and improve efficiency in the terminal airspace. However, without proper support tools and efficient utilization of the arrival procedures, the full benefits of point merge may not be explored.

We proposed an optimization framework that can be used as a tool for ATCOs to calculate the best combination of flight profiles and up-stream to the cockpits in the pre-sequencing area, facilitating fully-automated fuel-efficient descents even in high-traffic scenarios. The resulting schedule provides synchronization of the arrivals and departures, in a single-runway or mixed-mode operation on the example of Dublin airport. Furthermore, the framework is highly flexible and can be applied to any airport implementing point merge, with all the parameters, including the separation requirements, easily adjustable to the operational constraints at the airport of interest.

Our solution demonstrates that point merge system can be used in more efficient way with better utilization of its sequencing legs and less holdings, providing better organized arrival flows, significantly reducing time and distance in TMA, and requiring less control effort as a result. The improved performance comes with a fuel saving of 22%, corresponding to an average of 65 kg per flight.

In the future, we plan to evaluate noise impact and non-CO_2 emissions, associated with the improved efficiency provided by our approach. We will also test our optimization framework at a dual-runway airport, where aircraft may arrive and depart from either runway and perform additional case studies with different aircraft fleet mix and weather conditions. Additionally, we consider exploring the trade-offs between the robustness against uncertainties and arrival efficiency.

References

[1] R. Sáez, T. Polishchuk, C. Schmidt, H. Hardell, L. Smetanová, V. Polishchuk, and X. Prats. Automated Sequencing and Merging with Dynamic Aircraft Arrival Routes and Speed Management for Continuous Descent Operations. *TR-C: Emerging Technologies*, 132, 2021.

[2] H. Hardell, A. Lemetti, T. Polishchuk, L. Smetanova, and K. Zeghal. Towards a Comprehensive Characterization of the Arrival Operations in the Terminal Area. In *SIDs*, 2021.

[3] H. Hardell, A. Lemetti, T. Polishchuk, and L. Smetanová. Evaluation of the Sequencing and Merging Procedures at Three European Airports Using Opensky Data. *Engineering Proceedings*, 13(1):13, 2022.

[4] S. Choi, J. E. Robinson, D. G. Mulfinger, and B. J. Capozzi. Design of an optimal route structure using heuristics-based stochastic schedulers. In *IEEE/AIAA 29th DASC*, October 2010.

[5] A. Michelin, M Idan, and J. L. Speyer. Mergin of Air Traffic Flows. *Journal of Guidance, Control and Dynamics*, 34(1), Feb 2011.

[6] M. Samà, A. D'Ariano, K. Palagachev, and M. Gerdts. Integration methods for aircraft scheduling and trajectory optimization at a busy terminal manoeuvring area. *OR Spectrum*, 41:641–681, Aug 2019.

[7] R. Sáez, S. Prats, T. Polishchuk, V. Polishchuk, and C. Schmidt. Automation for Separation with CDOs: Dynamic Aircraft Arrival Routes. *AIAA Journal of Air Transportation*, 28(4):144–154, 2020.

[8] T. Polishchuk, V. Polishchuk, C. Schmidt, R. Saez, X. Prats, H. Hardell, and L. Smetanova. How to Achieve CDOs for All Aircraft: Automated Separation in TMAs (Enabling Flexible Entry Times and Accounting for Wake Turbulence Categories). In *SIDs*, 2020.

[9] J.M. de Wilde. Implementing point merge system based arrival management at amsterdam airport schiphol. 2018.

[10] R. Sáez, X. Prats, T. Polishchuk, and V. Polishchuk. Traffic Synchronization in Terminal Airspace to Enable Continuous Descent Operations in Trombone Sequencing and Merging Procedures: An Implementation Study for Frankfurt Airport. *TR-C: Emerging Technologies*, 121, 2020.

[11] Emad A Alharbi, Layek L Abdel-Malek, R John Milne, and Arwa M Wali. Analytical model for enhancing the adoptability of continuous descent approach at airports. *Applied Sciences*, 12(3):1506, 2022.

[12] Opensky Network. https://opensky-network.org/, last accessed on 11.02.2022.

[13] Irish AIP. http://iaip.iaa.ie/iaip/aip_directory.htm, last accessed on 11.02.2022.

[14] EUROCONTROL. User Manual for the Base of Aricraft Data (BADA) Family 4, 2014.

[15] Copernicus Climate Change Service (C3S) Data Store, European Centre for Medium-Range Weather Forecasts (ECMWF). https://cds.climate.copernicus.eu, last accessed on 11.02.2022.

[16] L. Delgado and X. Prats. Operating cost based cruise speed reduction for ground delay programs: Effect of scope length. *Transportation Research Part C*, 48:437–452, 2014.

[17] H. Hardell, T. Polishchuk, and L. Smetanova. Fine-Grained Evaluation of Arrival Operations. In *SIDs*, 2020.

[18] EUROCONTROL. Analysis of Vertical Flight Efficiency During Climb and Descent. 2017.

[19] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm. Bringing Up OpenSky: A Large-scale ADS-B Sensor Network for Research. In *IPSN'14*, 2014.

[20] SKYbrary. Mitigation of wake turbulence hazard, wake turbulence separation minima. https://skybrary.aero/articles/mitigation-wake-turbulence-hazard, last accessed on 11.02.2022.

[21] Tetralith server, NSC, Linköping University. https://www.nsc.liu.se/systems/tetralith/, last accessed on 11.02.2022.