A Dantzig-Wolfe Reformulation for Automated Aircraft Arrival Scheduling in TMAs

Roghayeh Hajizadeh[1] [0000−0002−7402−6292], Tatiana Polishchuk[2] [0000−0002−9869−5992], Elina Rönnberg[1] [0000−0002−2081−2888], and Christiane Schmidt[2] [0000−0003−2548−5756]

[1] Department of Mathematics (MAI), Linköping University, Sweden
[2] Department of Science and Technology (ITN), Linköping University, Sweden
firstname.lastname@liu.se

Keywords: Dantzig-Wolfe Decomposition, Aircraft Arrival Routes, Automated Aircraft Separation

1 Introduction

Air traffic volumes have increased for decades, and though there was a significant drop because of the Covid-19 pandemic, IATA [4] projects that the demand for air travel will double by 2040, with an average annual growth rate of 4.3%. These high air traffic volumes result in an elevated environmental impact and significantly increased complexity for air traffic controllers (ATCOs). Both play a particular large role in Terminal Maneuvering Areas (TMAs)—the airspace around one or several aerodromes—where all air traffic merges and which is, hence, particularly impacted by both congestion and noise. To be able to handle the ever increasing volumes, it is crucial to alleviate the environmental impact and the ATCO workload by providing improved arrival and departure procedures, which still enable a high runway throughput.

An approach to lower the environmental impact are so-called continuous descent operations (CDOs), optimal engine-idle descents, which can reduce fuel burn, gaseous emissions, noise and fuel costs [3]. CDOs are optimal for the specific aircraft capabilities. Thus, different aircraft have different optimal trajectories. These do not fit together with the strategical standard terminal arrival routes (STARs) and they do decrease vertical and temporal predictability—a situation to which ATCOs answer with increasing separation buffers, which negatively effects throughput, or with issueing instructions that alter the optimal trajectories, which negatively impacts the environmental benefits. To be able to apply CDOs, tools that supply automated separation to ATCOs are needed.

For example, Choi et al. [2] presented a genetic-algorithm approach to compute aircraft arrival routes and the arrival sequence: they first developed distinct route topologies and then evaluated those with the heuristic-based scheduler.

In a series of papers, a group of authors [1,8,6,5,7] presented a MIP model to design optimal aircraft arrival routes with fully automated scheduling of CDOs with guaranteed aircraft separation and an operational concept that allows the usage of these routes. All aircraft fly according to their optimal neutral CDO speed profiles, where an aircraft's arrival to the TMA entry point can be adapted within a time window. In the model, the correct speed profile is picked by the length of the arrival route from entry point to runway. Moreover, the progress of aircraft along routes within the given grid graph are tracked. While this framework showed the feasiblity of the approach, it suffers from long runtimes: generating the arrival trees for a one-hour scenario took between 1.58h for low-traffic cases to 40.9h for high-traffic cases (with light aircraft added to the flow). This is of course not feasible for the real-world application, where new arrival routes should be computed regularly (ca. every 30 minutes, within the time frame that an aircraft spends in the TMA), and the framework certainly could not handle adding on more features, in particular, the influence of wind (the speed profile does not only depend on the length of the descent, but also on the wind direction in relation to the aircraft's trajectory). Hence, in this paper, we provide a Dantzig-Wolfe refomulation of a simplified model of MIP model in [7] (with fixed entry times and one separation time independent of wake categories) and show that this can yield significantly decreased runtimes—a promising approach for the full model and to handle even more practical aspects like wind in the future.

For the given location of TMA entry points and the runway for an airport, and a set of aircraft planned with fixed arrival time to their entry point, we aim to compute dynamic arrival trees for which

1. No more than two routes merge at a point (merge points require ATCO attention, we aim for the lowest possible complexity).
2. Merge points are separated by a minimum distance (otherwise many merge points could be located within an arbitrarily small area).
3. Routes do not make sharp turns (infeasible by aircraft dynamics).
4. Obstacles, like no-fly zones, are avoided.
5. All aircraft are temporally separated along the arrival route.
6. All aircraft follow CDO speed profiles (dependent on the arrival-route length).

2 Model

We discretize the TMA by creating a square grid with an edge length equal to the lower bound on separation and snapping the locations of both entry points and the runway to the grid. This leads to a bi-directed graph $G = (V, E)$ with nodes N and edges E where each grid node is connected to its 8 neighbors and for any two neighboring nodes i and j, both edges (i, j) and (j, i) exist in E. Let l_{ij} denote the length of edge $(i, j) \in E$, $\mathcal{P}$ the set of entry points, r the runway, $\mathcal{A}_b$ the set of all aircraft arriving at entry point $b \in \mathcal{P}$, $\mathcal{A} = \bigcup_{b\in\mathcal{P}} \mathcal{A}_b$ the set of all aircraft, and $|\mathcal{A}_b|$ the number of aircraft entering entry point b. In addition, $\bar{t}_a$ denotes the planned arrival time of aircraft a to its entry point and $T = \{0, \ldots, \overline{T}\}$ is the considered time interval.

As part of making a Dantzig-Wolfe refomulation, all possible paths from each entry point to the runway are generated beforehand. This is done with an upper bound on the length of feasible paths and with respect to forbidden sharp turns and obstacles avoidance,. Let Π_b denote the set of paths from entry point b to the runway and $\mathbf{\Pi} = \bigcup_{b\in\mathcal{P}} \Pi_b$ be the set of all paths. For any $\pi \in \mathbf{\Pi}$, we define θ_π as the set of edges that path π passes through.

We introduce binary variables ρ_π for each path $\pi \in \mathbf{\Pi}$, indicating whether path π is used in the arrival tree and x_{ij} indicating whether the edge (i, j) participates in the arrival tree, our Dantzig-Wolfe-reformulation-based model is:

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

$$\sum_{j:(j,i)\in E} x_{ji} \leq 2 \qquad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{1}$$

$$\sum_{j:(i,j)\in E} x_{ij} \leq 1 \qquad \forall i \in V \setminus \{\mathcal{P} \cup r\} \tag{2}$$

$$\sum_{\pi\in\Pi_b} \rho_\pi = 1 \qquad \forall b \in \mathcal{P} \tag{3}$$

$$\sum_{b\in\mathcal{P}}\sum_{a\in\mathcal{A}_b}\sum_{\pi\in\Upsilon_{ait}} \rho_\pi \leq 1 \qquad \forall i \in V, \forall t \in \{0,\dots,\overline{T}-\sigma\} \tag{4}$$

$$\sum_{\pi\in\mathbf{\Pi}:\ (i,j)\in\theta_\pi} \rho_\pi \leq Q x_{ij} \qquad \forall (i,j) \in E \tag{5}$$

$$\rho_\pi \leq x_{ij} \qquad \forall \pi \in \mathbf{\Pi}, \forall (i,j) \in \theta_\pi \tag{6}$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1$$
$$\forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\} \tag{7}$$

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

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \qquad \forall i : i+1 \in \mathcal{P} \tag{9}$$

$$x_{i,i+1+n,} + x_{i+n+1,i} + x_{i+n,i+1} \leq 1 \qquad \forall i : i+n \in \mathcal{P} \tag{10}$$

$$x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \qquad \forall i : i+n+1 \in \mathcal{P} \tag{11}$$

The objective function is a convex combination of the length of the paths and the tree weight. Constraints (1) and (2) ensure that all the nodes except the entry points and the runway have an indegree of maximum 2 and an outdegree of maximum 1. Constraint (3) states that exactly one path from each entry point should be used. A minimum separation of σ time units between all aircraft at all nodes is given in Constraint (4) where Υ_{ait} is the set of all paths (starting from the corresponding entry point and passing node i) on which aircraft a with entry time $\bar{t}_a$ occupies node i between time t and $t+\sigma-1$. Constraint (5) and/or Constraint (6) connect the variables where Q is a large number (the number of entry points in our case). Because we do not solely aim for shortest paths, we need to prohibit crossing. We could either take the crossings into account when generating the routes or we can prohibit them by Constraints (7)-(11) where $V' = V \setminus \{\text{last grid row}\} \setminus \{\text{last grid column}\}$.

3 Results and Conclusions

We used data for Stockholm-Arlanda-airport TMA with a 15×11 grid (which ensures a separation of 6NM) and solved our model using the Gurobi optimization solver with gurobipy as interface to Python on a MacBook Pro, M1 2020. We have made preliminary experiments where the running time of our reformulated model for a medium-traffic case (20 aircraft in one hour) with fixed arrival time for all aircraft including the generation of the paths was less than one minute.

Comparing our preliminary results to those given in [1,8,6], confirms that our model significantly outperformed in terms of computational efficiency. This gives the possibility for improved runtimes even for the case with flexible entry times and separation based on wake-turbulence categories ([5,7]). Moreover, this indicates that our framework may be capable of handling real-world operations and generating new arrival routes regularly on a personal computer.

Acknowledgements Supported by grant 2022-03178 (OPT@ATM: Optimization Methods for Large Air Traffic Management Problems) from the Swedish Research Council (Vetenskapsrådet). The computational experiments were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

1. Andersson, T., Polishchuk, T., Polishchuk, V., Schmidt, C.: Automatic Design of Aircraft Arrival Routes with Limited Turning Angle. In: Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS) (2016)
2. Choi, S., Robinson, J.E., Mulfinger, D.G., Capozzi, B.J.: Design of an optimal route structure using heuristics-based stochastic schedulers. In: IEEE/AIAA 29th Digital Avionics Systems Conference (DASC) (Oct 2010)
3. Eurocontrol: Continuous climb and descent operations. https://www.eurocontrol.int/articles/continuous-climb-and-descent-operations (2019), accessed December 15, 2017
4. IATA: Global outlook for air transport: Highly resilient, less robust (2023)
5. Polishchuk, T., Polishchuk, V., Schmidt, C., Sáez, R., Prats, X., Hardell, H., Smetanová, L.: 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) (2020)
6. Sáez, R., Prats, X., Polishchuk, T., Polishchuk, V., Schmidt, C.: Automation for Separation with CDOs: Dynamic Aircraft Arrival Routes. AIAA J. Air Transp. **28**(4), 144–154 (2020). https://doi.org/10.2514/1.D0176
7. Sáez, R., Polishchuk, T., Schmidt, C., Hardell, H., Smetanová, L., Polishchuk, V., Prats, X.: Automated sequencing and merging with dynamic aircraft arrival routes and speed management for continuous descent operations. Transportation Research Part C: Emerging Technologies **132**, 103402 (2021)
8. Sáez, R., Prats, X., Polishchuk, T., Polishchuk, V., Schmidt, C.: Automation for separation with CDO:s dynamic aircraft arrival routes. In: Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019) (2019)