

Capacity-Driven Automatic Design of Dynamic Aircraft Arrival Routes

Joen Dahlberg, Tobias A. Granberg, Tatiana Polishchuk, Christiane Schmidt, Leonid Sedov

Communications and Transport Systems, Linköping University
firstname.lastname@liu.se

Abstract—We present a Mixed-Integer Programming framework for the design of aircraft arrival routes in a Terminal Maneuvering Area (TMA) that guarantee temporal separation of aircraft. The output routes constitute operationally feasible merge trees, and guarantee that the overall traffic pattern in the TMA can be monitored by air traffic controllers; in particular, we ensure that all aircraft on the arrival routes are separated in time and all merge points are spatially separated. We present a proof of concept of our approach, and demonstrate its feasibility by experiments for arrival routes during one hour at Stockholm TMA.

Index Terms—Aircraft Arrival Routes, Automatic Separation, Integer Programming

I. INTRODUCTION

An air traffic controller (ATCO) is responsible for guaranteeing safe separation of aircraft at all times. Traffic is particularly dense in the airspace surrounding one or several aerodromes, a so called Terminal Maneuvering Area (TMA). At most airports redesigned standard routes for departure and arrival are established. Currently, these Standard Instrument Departures (SIDs) and Standard Terminal Arrival Routes (STARs) are designed manually. This design is based on the airspace layout and incorporates constraints like avoidance of no-fly zones, manageability by ATCOs, and others. However, depending on the current traffic load, these routes may result in a high ATCO workload to obtain the required separation, especially at merge points of routes.

As air transportation experienced significant growth over the last decades, and the International Air Transport Association (IATA) projected that the number of passengers will double to reach 7 billion/year by 2034 [2], the workload connected to the safe aircraft separation in the TMA will only increase. Thus, designing arrival procedures that already guarantee safe separation at all points for normal, that is, planned, operation, can be a significant contributor to keep the workload at an acceptable—a safe—level.

Thus, we suggest an optimization framework for daily arrival routes in a TMA: we take the arrival times at entry points of all aircraft during a day into account, and compute arrival routes that ensure safe separation from the entry point to the runway along the entire routes. In particular, we apply Integer

Programming to generate operationally feasible, separation-guaranteeing, dynamic merge trees of curvature-constrained demand-weighted routes with minimum total length of the tree. The output routes guarantee that the overall traffic pattern in the TMA can be monitored by air traffic controllers; in particular, we keep arriving aircraft well separated at merge points.

Our previous work [3], in which we introduced a grid-based MIP approach for finding aircraft arrival routes with limited turning angle, builds the basis for this work. The novel component of this paper is the dynamic design that adjusts the route w.r.t. the demand and the temporal distribution of demand, i.e., aircraft arrival times.

Apart from the aircraft arrival times to the TMA, we are given the locations of the TMA entry points, and the location and direction of the airport runway. As in [3], we discretize the search space by laying out a square grid in the TMA, and snapping the locations of the entry points and the runway onto the grid. The side of the grid pixel is equal to our lower bound on the distance between route vertices—this ensures that the merge point separation is satisfied by any path in the grid. Every grid node is connected to its 8 neighbors. In the output we seek an arrival tree that merges traffic from the entries to the runway, i.e., a tree that has the entries as leaves and the runway as the root, optimized w.r.t. the traffic demand during the given period. Our objective is to minimize the total length of the routes from the entry points to the runway. We consider a weighted version, which takes the number of aircraft using each route into account, that is, we consider the demand-weighted total length of the routes. For this first result, we simplify and assume a unit speed along all edges.

Our model is a mixed-integer program (MIP). We use the AMPL modeling language and the GUROBI solver to model and solve the MIP. We feed the model with real flight data samples (from EUROCONTROL's Demand Data Repository (DDR)). We extract the data for Stockholm Arlanda arrival flights and compare the resulting arrival routes for different sets of arrivals during a day with high traffic load. The resulting arrival tree illustrates a set of optimal demand-weighted aircraft trajectories which yield improved predictability of traffic path planning.

The model under development was discussed with operational experts in LFV (Luftfartsverket, the Swedish ANSP) to provide realistic operational constraints on the route design.

This research is funded by grant 2014-03476 (ODESTA: Optimal Design of Terminal Airspace) from Sweden's innovation agency VINNOVA and in-kind participation of LFV.

The designed framework is flexible and can be applied to other airports and variable arrival traffic demands.

A. Problem Description and Results

In this paper, we present a mathematical programming framework for finding optimal dynamic aircraft arrival trees. Our approach is dynamic in comparison to the standard, static STARs, because our arrival route trees are recomputed for different time periods. Thus, they change during the day to reflect the actual incoming traffic demand.

As part of the input to the problem, we are given locations of the entry points to the Terminal Maneuvering Area (TMA), the location and direction of the airport runway, and the arrival times of aircraft to the entry points within a given time interval T . In the output we seek an arrival tree that merges traffic from the entries to the runway, i.e., a tree that has the entries as leaves and the runway as the root (contrary to the common convention, we assume that the edges of this arborescence are directed from leaves to root), such that all aircraft are separated at all points of the arrival routes.

Our arrival route tree must fulfill a set of operational constraints:

- 1) *Temporal separation of all aircraft along the routes:* Any pair of aircraft moving along the computed arrival tree, starting at their given arrival times at the entry point(s), is separated by at least a temporal distance of S . That is, if aircraft arrive to the TMA within at planned time, all aircraft are safely separated along the arrival routes.
- 2) *No more than two routes merge at a point:* Merge points of routes require an increased attention level from controllers, hence, traffic complexity around the merges should be kept at a minimum [18]. This translates to the requirement that every vertex of the tree must have in-degree less than or equal to 2.
- 3) *Merge point separation:* The constraint of only two routes merging at a point could be circumnavigated for all practical purposes by having merge points very close to one another. This will again create a small zone with several routes merging, which is undesirable for control. Hence, in addition to the operational Constraint 2, we require that the separation between any two merge points is larger than a given distance threshold L [18].
- 4) *No sharp turns:* Aircraft dynamics impose a limit on the angle at which the routes can turn (bank angle) [11, p. 61]. Thus, the turn from a segment of a route to the consecutive segment is required to never undercut a given angle threshold γ . If arbitrarily short edges could be used, a sharp turn could still be simulated by a sequence of many short edges. Thus, the combination of the parameter γ and the limit, L , on the minimum length for any edge enforces the limited turning angle [13]. We assume that the runway is the last segment of every route: this way, the turn onto the runway must also be larger than γ —the aircraft must align with the runway before the touchdown.

- 5) *Obstacle avoidance:* The routes should not pass over a specified set of regions (we do not digress into the specific nature of the obstacles—they may be no-fly zones, noise-sensitive areas, etc.).

Objective functions: It is natural to seek arrival routes that provide short flight routes for aircraft. Thus, one objective in our optimization problem is the *total length of the routes* from the entry points to the runway. Moreover, the arrival route tree should "occupy little space"—both from the ATCO perspective (to minimize attention attraction area), and to avoid spreading the noise and other environmental impact over a larger region. This can be modeled by requiring that the produced tree has small *total length of the edges*. We consider both objectives, and call them *paths length* and *tree weight*, respectively, in particular, we consider convex combinations of these two objectives.

B. Roadmap

In the remainder of this section we review related work. We describe our MIP formulation in Section II: we start with a review of our grid-based MIP for static routes in Subsection II-A, show how we integrate the temporal component in Subsection II-B (we present the complete MIP for our problem in the Appendix A). In Section III we give an experimental study of our approach, both with a theoretical proof of concept in Subsection III-A and experiments on the arrival routes in Stockholm TMA in Subsection III-B. We conclude in Section IV.

C. Related Work

Automatic design of STARs and arrival routes has been studied earlier, but to the best of our knowledge, finding optimal trees, taking into account the temporal component and the turn constraints, has not been considered before. In prior work, the routes have been constructed iteratively (one-by-one) and/or did not adhere to the full set of our constraints and/or did not route the traffic all the way to the runway and/or did not plan arrival routes at which aircraft are fully (temporally) separated at all times.

First, we review the related work on arrival route/STAR design. Pfeil [17] focuses on weather forecast and the redesign or design of TMAs pertaining to different weather scenarios. The author develops an IP model to optimally choose fix locations and corresponding routes in fixed sectors, and to renegotiate sector boundaries. For the TMA design from scratch a two-step solution is presented: first optimal outer fixes are selected with an IP, then 3D routes between fix-runway pairs are chosen with a modified version of the A* algorithm. This defines some first chosen routes as obstacles for later routes, that is, the construction is sequential. All algorithms are presented for the US model of a TMA: two circles of different radii around the runway, where all merges and maneuvers are assumed to be performed within the inner circle and are not considered. The same TMA model is used by Prete et al. [18].

Krozel et al. [13] considered turn-constrained route planning for a *single* path; trees and the merging of paths are not considered. Zhou et al. [21], [22] also construct single, individual routes (not arrival merge trees) through weather-impacted TMA. Similarly, Visser and Wijnen [19] construct single routes, the objective in their work is to minimize noise impact. None of these authors considered a temporal component.

The actual scheduling on routes has been considered in a few papers. Choi et al. [5] claim that using scheduling algorithms that are more efficient than the traditional FCFS approach can increase throughput in congested terminal airspace. But as the route typologies play an important role in the TMA, they aim to consider both routes and scheduling together. They present results on STAR merging, testing the impact of different merge typologies on scheduling of aircraft along the routes; however, the actual location of merge points is not of interest and turn constraints are not taken into account.

A huge body of work has been devoted to aircraft sequencing close to the runway in order to avoid wake vortex effects. The latest work is related to the re-categorization projects on both sides of the Atlantic (see, e.g., [14]) that strive to define separation standards on per-aircraft-type basis, in contrast to the current system where each aircraft is classified into one of the few categories, and the separation is defined based on the categories of the leading and trailing aircraft. A recent paper [16] includes into the equation also the surface movement of the aircraft. On the other end, extended Arrival Management (AMAN) [10], [16] starts taking the separation into account already while the plane is enroute. Our work "fits in the middle" between the Surface Management (SMAN) and Cross-border Arrival Management (XMAN): we show how to ensure the separation within the TMA by taking into account the time component when designing the arrival routes. Differently from [6]–[9] who studied *speed variation* for airborne delay to adjust Calculated Time of Arrival (CTA) of *one* flight, we use *paths adjustment* to coordinate the merge of *multiple* aircraft trajectories. Our framework can be extended to allow for variable speed; combining it with the work [6]–[9] to account for aircraft dynamics is the subject of future research.

Moreover, the idea of assigning not only routes, but also times is known from a non-aviation context, in particular, from dynamic vehicle routing, cp. Bertsimas and van Ryzin [4], and—more general—from dynamic scheduling, see Ouelhadj and Petrovic [15] for a survey on this topic.

II. GRID-BASED MIP FORMULATION

We start with a review of our prior MIP-formulation for optimal STAR merge trees [3] in Subsection II-A, we then describe how we integrate the temporal separation into this framework in Subsection II-B.

A. Review of our Grid-based MIP for Static Routes

The problem formulation for the optimal STAR merge trees is the same as the problem description in Subsection I-A

without the temporal component, that is, the arrival times of aircraft are not part of the input, and we do not need to adhere to the operational Constraint 1.

We discretize the search space by laying out a square grid in the TMA, and snapping the locations of the entry points and the runway onto the grid; let \mathcal{P} denote the set of (snapped) entry points, and r the runway. A grid pixel side has a length of L , the lower bound on the distance between route vertices—this ensures that the merge point separation (Constraint 3) is satisfied by any path in the grid. Every grid node is connected to its 8 neighbors (where $N(i)$ = denotes set of neighbors of i , including i), resulting in a bidirectional graph $G = (V, E)$. That is, for any two neighbors i and j , both edges (i, j) and (j, i) are included in E ; the only exceptions are the entry points (they do not have incoming edges) and r (it does not have outgoing edges). The length of an edge $(i, j) \in E$ is denoted by ℓ_{ij} . If we include operational Constraint 5, obstacle avoidance, we delete the edges in that region from our edge set E , and, hence, do not allow routes to include edges in the region.

Our STAR MIP formulation is based on the flow MIP formulation for Steiner trees [12], [20]. We use decision variables x_e that indicate whether the edge e participates in the STAR. In addition, we have flow variables: f_e gives the flow on edge $e = (i, j)$ (i.e., the flow from i to j). The constraints are given in Equations (1)–(4):

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

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

$$f_e \geq 0 \quad \forall e \in E \quad (3)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (4)$$

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

Equation (1) ensures that a flow of $|\mathcal{P}|$ reaches the runway r , a flow of 1 leaves every entry point, and in all other vertices of the graph the flow is conserved. Equation (2) enforces edges with a positive flow to participate in the STAR. The flow variables are non-negative (Equation (3)), the edge variables are binary (Equation (4)).

We might also choose to not only minimize the length of paths from entry points to the runway, but consider a weighted version that minimizes the sum of trajectory lengths flown by all arriving aircraft. That is, each path is counted as many times as it is used by aircraft. Hence, we minimize the demand-weighted distance. We can easily integrate this by changing the right-hand side of Equation (1) (and increase Q accordingly). Let w_b be the number of aircraft entering the TMA via entry point $b \in \mathcal{P}$:

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

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

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

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

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

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

1) *Degree constraints*: Equations (1)-(4) describe a standard MinCostFlow Steiner tree MIP. We add further equations to enforce the constraints defined in Section I. For operational Constraint 2 we require that the outdegree of every node is at most 1 and that the maximum indegree is 2:

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

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

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

$$\sum_{j:(r,j) \in E} x_{rj} \leq 0 \quad (12)$$

$$\sum_{k:(k,i) \in E} x_{ki} \leq 0 \quad \forall i \in \mathcal{P} \quad (13)$$

$$\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \forall i \in \mathcal{P} \quad (14)$$

Equations (11) and (12) ensure that the runway r has one ingoing and no outgoing edges, respectively; Equations (14) and (13) make sure that each entry point has one outgoing and no ingoing edge, respectively; the maximum indegree of 2 for all other vertices is given by Equation (9), the maximum outdegree of 1 by Equation (10).

2) *Turn angle constraints*: Next we take care of operational Constraint 4: If an edge $e = (i, j)$ is used, all outgoing edges at j must form an angle of at least γ with e . Let Γ_e be the set of all outgoing edges from j that form an angle $\leq \gamma$ with e , i.e., $\Gamma_e = \{(j, k) : \angle ijk \leq \gamma, (j, k) \in E\}$, and let $c_e = |\Gamma_e|$. Equation (15) enforces that we may either use edge x_e (which sets the right-hand side to c_e , the upper bound), or we may use any subset of the edges in Γ_e .

$$c_e x_e + \sum_{f \in \Gamma_e} x_f \leq c_e \quad \forall e \in E \quad (15)$$

3) *Auxilliary Constraints to Prevent Crossings*: While route crossings at vertices are prevented by the degree constraints in Subsubsection II-A1, we may still encounter routes crossing within a grid square, and we add auxiliary constraints to prevent this behavior. (Note that in the trees that minimize the length without temporal constraint such a crossing would never occur, because uncrossing the routes would shorten them.) We define V' as the set of all grid nodes without those which belong to the last column and last row, that is, $V' = V \setminus \{\text{last row}\} \setminus \{\text{last column}\}$.

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\} \quad (16)$$

However, if one of the grid points in the grid square is an entry point, one of the four edges crossing the square does not exist (remember, that entry points have no incoming edges). Hence, we need to add Equations (17)-(20). Figure 1 illustrates the four cases depending on the location of the entry point.

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in \mathcal{P} \quad (17)$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \quad \forall i : i+1 \in \mathcal{P} \quad (18)$$

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

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

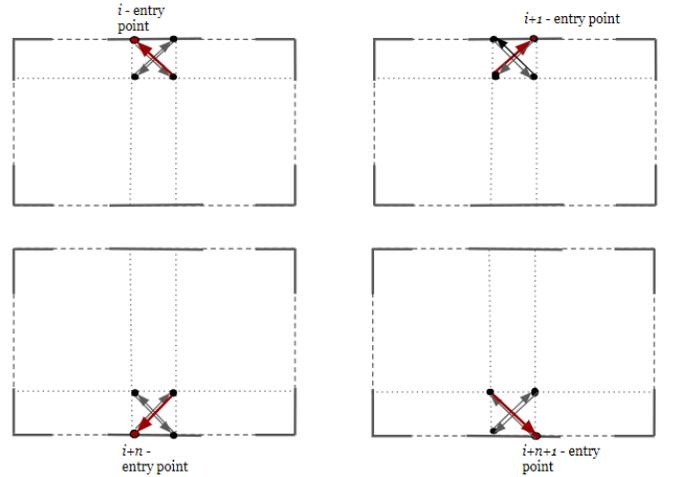


Fig. 1. Four cases with entry point (marked in red) in a grid square: missing edges are shown in red. The figures in order refer to Equations (17), (18), (19) and (20), respectively.

B. Integration of Temporal Separation

We introduce new, binary variables $y_{a,j,t}$ that indicate whether aircraft a occupies vertex j at time t . Moreover, instead of having only an indicator for the edge e participating in the routes, x_e , we have additional indicators $x_{e,b}$ for the edge e participating in the route from entry point b to the runway (for all entry points $b \in \mathcal{P}$). We set the variables $x_{e,b}$ using Equations (21)-(24):

$$x_{e,b} \leq x_e \quad \forall b \in \mathcal{P}, \forall e \in E \quad (21)$$

$$\sum_{j:(b,j) \in E} x_{(b,j),b} = 1 \quad \forall b \in \mathcal{P} \quad (22)$$

$$\sum_{j:(j,r) \in E} x_{(j,r),b} = 1 \quad \forall b \in \mathcal{P} \quad (23)$$

$$\sum_{i:(i,j) \in E} x_{(i,j),b} - \sum_{k:(j,k) \in E} x_{(j,k),b} = 0 \quad \forall j \in V \setminus \{\mathcal{P} \cup r\}, \quad \forall b \in \mathcal{P} \quad (24)$$

Now we can turn our attention to the $y_{a,j,t}$'s: we will set the values according to the arrival time of aircraft a at entry point b , t_a^b , we will make sure that all times are set accordingly for the vertices along the route to the runway, and we will make sure that sufficient temporal separation is kept at all vertices. 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 interval.

$$y_{a,b,t_a^b} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (25)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A} \setminus \mathcal{A}_b, \forall t \in T \quad (26)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall t \in T \setminus \{t_a^b\} \quad (27)$$

$$y_{a,j,t} \leq \sum_{\substack{k \in V: \\ (k,j) \in E}} x_{(k,j)} \quad \forall a \in \mathcal{A}, \forall j \in V \setminus \mathcal{P}, \forall t \in T \quad (28)$$

Equation (25) ensures that aircraft a occupies entry point b at its arrival time t_a^b at this vertex. Equation (26) yields that no aircraft that does not arrive at b occupies this vertex at any time, Equation (27) yields that an aircraft arriving at b occupies this vertex at no time apart from t_a^b . Finally, Equation (28) ensures that 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.

We have set the variable $y_{a,j,t}$ for entry points, but we still need to forward this time information for an aircraft a through its route from the entry point b . When can we set $y_{a,k,t+u} = 1$, that is, when is aircraft a at vertex k at time $t+u$? Aircraft a needs to reach k by traversing an edge from some vertex j to k , this traversal takes, by our assumption, u time units. So, if the aircraft a was at t at some vertex j from which k can be reached on the path from b , it can reach k at $t+u$. If, however, no edge to k exists, or if a was not at some vertex j at time t , it can not reach k at $t+u$, and we set $y_{a,k,t+u} = 0$.

We could achieve this by formulating

$$\sum_{j:(j,k) \in E} x_{(j,k),b} \times y_{a,j,t} = y_{a,k,t+u} \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (29)$$

Unfortunately, Equation (29) contains a multiplication of two binary variables, which we may not include in our MIP. To circumnavigate this, we define a new binary variable $z_{a,j,k,b,t}$ as the product of $x_{(j,k),b}$ and $y_{a,j,t}$ using Equations (30)-(33):

$$z_{a,j,k,b,t} \leq x_{(j,k),b} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (30)$$

$$z_{a,j,k,b,t} \leq y_{a,j,t} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (31)$$

$$z_{a,j,k,b,t} \geq x_{(j,k),b} - (1 - y_{a,j,t}) \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \quad \forall t \in \{0, \dots, \bar{T} - u\} \quad (32)$$

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

With this we can reformulate Equation (29) as:

$$\sum_{j:(j,k) \in E} z_{a,j,k,b,t} - y_{a,k,t+u} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (34)$$

Finally, we can require a minimum separation of S time units between all aircraft at all vertices:

$$\sum_{i=t}^{t+S-1} \sum_{a \in \mathcal{A}} y_{a,j,i} \leq 1 \quad \forall j \in V, \forall t \in \{0, \dots, \bar{T} - S\} \quad (35)$$

C. The Complete MIP

To enhance readability, we present the complete MIP in the Appendix A.

III. EXPERIMENTAL STUDY

The experimental study for our framework is twofold: we present a proof of concept using an artificial instance that highlights the concept, and is of limited size. In addition, we consider arrival routes for aircraft in Stockholm TMA, that is, we apply our framework to real-world instances, and show its feasibility.

We solve our MIP using Gurobi server installed on a LUNARC Aurora server [1], utilizing the nodes with two 10-core Intel E5-2650v3 2.3 GHz CPUs (Haswell), 64 RAM and 1.7 TB temporary disk space.

A. Proof of Concept

In this subsection, we start from presenting an artificial example that highlights how the optimal dynamic aircraft arrival route that guarantee temporal separation differ from the optimal static routes without this condition, and how different aircraft arrival times influence the route structure. Then we also show how we integrate the capacities of the flows from different entry points.

Time Separation: We are given locations for four entry points and a runway for an artificial TMA with a 10x10 grid laid out on top of it. The runway is located at the center of the TMA, at the point with coordinates (4,4), and the four entry points are placed at the following grid locations: (0,6), (3,9), (6,9), and (9,6). We choose the parameter S of Equation (35) as one time unit, that is, $S = 1$.

Figure 2 shows the configurations of arrival trees that change depending on the aircraft arrival time sequences to provide automatic time separation. The aircraft arriving from the entry point at (0,6) are scheduled at the times 1, 3, 6 and 9, and the aircraft from the entry point located at (3,9) are arriving at 2, 5, 8 and 11. Tracking the times of all aircraft arrivals at the merge points in the resulting arrival trees (merge points for the routes corresponding to the given two entry points are at (3,6) and (4,5)), we confirm that there are no conflicts and all aircraft are separated by at least 1 time unit.

In Figure 2(b) we give an example that shows the resulting optimal arrival tree changes to avoid conflicts. We change the arrival time sequences for the same two entry points as follows: four aircraft arrive from the entry point located (0,6) at times 1, 3, 5 and 7, and four aircraft arrive from the entry point located at (3,9) at times 2, 4, 6 and 8.

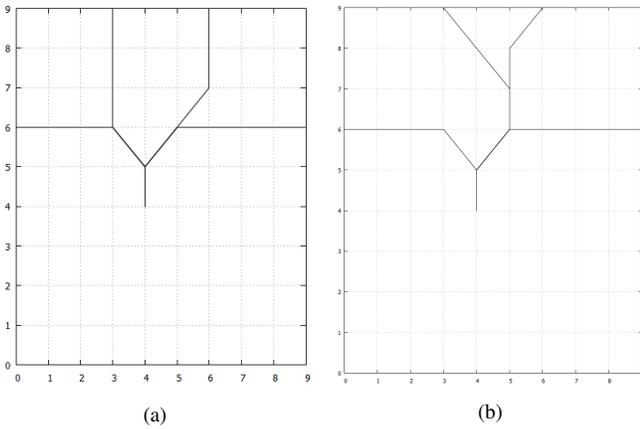


Fig. 2. Arrival route trees calculated for the following arrival time sequences: (a) four aircraft arrive from (0,6) at times 1, 3, 6 and 9, and four aircraft arrive from (3,9) at times 2, 5, 8 and 11; (b) four aircraft from (0,6) at times 1, 3, 5, 7 and four aircraft from (3,9) at times 2, 4, 6 and 8.

Capacity-Driven Arrival Routes: The following example shows how the arrival route trees adapt to the amount of traffic arriving from different entry directions.

We position four entry points in the same 10x10 grid at (0,9), (4,9), (9,9), and (4,0). Figure 3 demonstrates how the configurations of the arrival trees change depending on the amount of scheduled traffic from different entry points.

First, we schedule 10 aircraft arriving from the entry point located at (0,9) and one aircraft arriving from each of the other three entry points. Figure 3(a) shows that in the resulting arrival tree the direction from where the majority of the aircraft arrive is prioritized by providing the shortest path from the entry point (0,9). Figure 3(b) supports this concept by

demonstrating how the other direction with the largest amount of arriving traffic is prioritized, when we schedule 10 aircraft arriving from (9,9) and one from each of the remaining entry points.

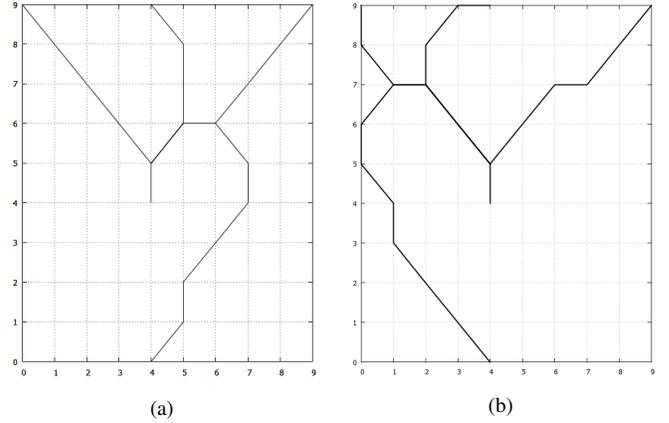


Fig. 3. Arrival route trees, prioritizing the routes with larger capacities. In (a) 10 aircraft arrive from the entry point located at (0,9) and one aircraft from each of the other three entry points; in (b) 10 aircraft are arriving from the entry point located at (9,9) and one aircraft from each the other three entry points.

B. Arlanda Airport

In this subsection, we consider arrival routes for Stockholm TMA. That is, we use the aircraft arrival times at TMA entry points during one hour of airport operation as input, and compute the dynamic arrival routes with guaranteed temporal separation for this time interval.

We have chosen data samples for October 4, 2017, one of the busiest days of that year with 432 aircraft arrivals. We use data samples of one hour per experiment. The resulting arrival trees are presented in Figure 4. Figure 4 (a)-(f) shows arrival trees for different hours during that day, and Figure 4 (g) depicts a family of all arrival trees during that day. We observe that when the trees need to be recomputed, they do not differ completely. This may be an advantage for actual operation.

We use a 10x10 grid, which automatically guarantees separation of about 9-10 nm, which is more than the common standard of 6 nm. In fact, with our current setup we could not solve the problem for some time periods with higher number of aircraft arrivals. For example, for the current grid there was no feasible solution for the time slot between 5 am and 6 am. The current computational resources did not allow to use a finer grid with smaller cells, but in the future we plan to continue our experiments on more powerful servers.

Figure 5 shows an arrival tree for 10 aircraft entering Stockholm TMA between 6 am and 7 am. Locations of the three merge points are marked by M1, M2 and M3. We track how each aircraft progresses along the tree and consider the simulated arrival times at entry points, merge points and the runway, Table I presents these times for all aircraft. Several aircraft arriving simultaneously at a merge point would be a conflict, which our approach excludes.

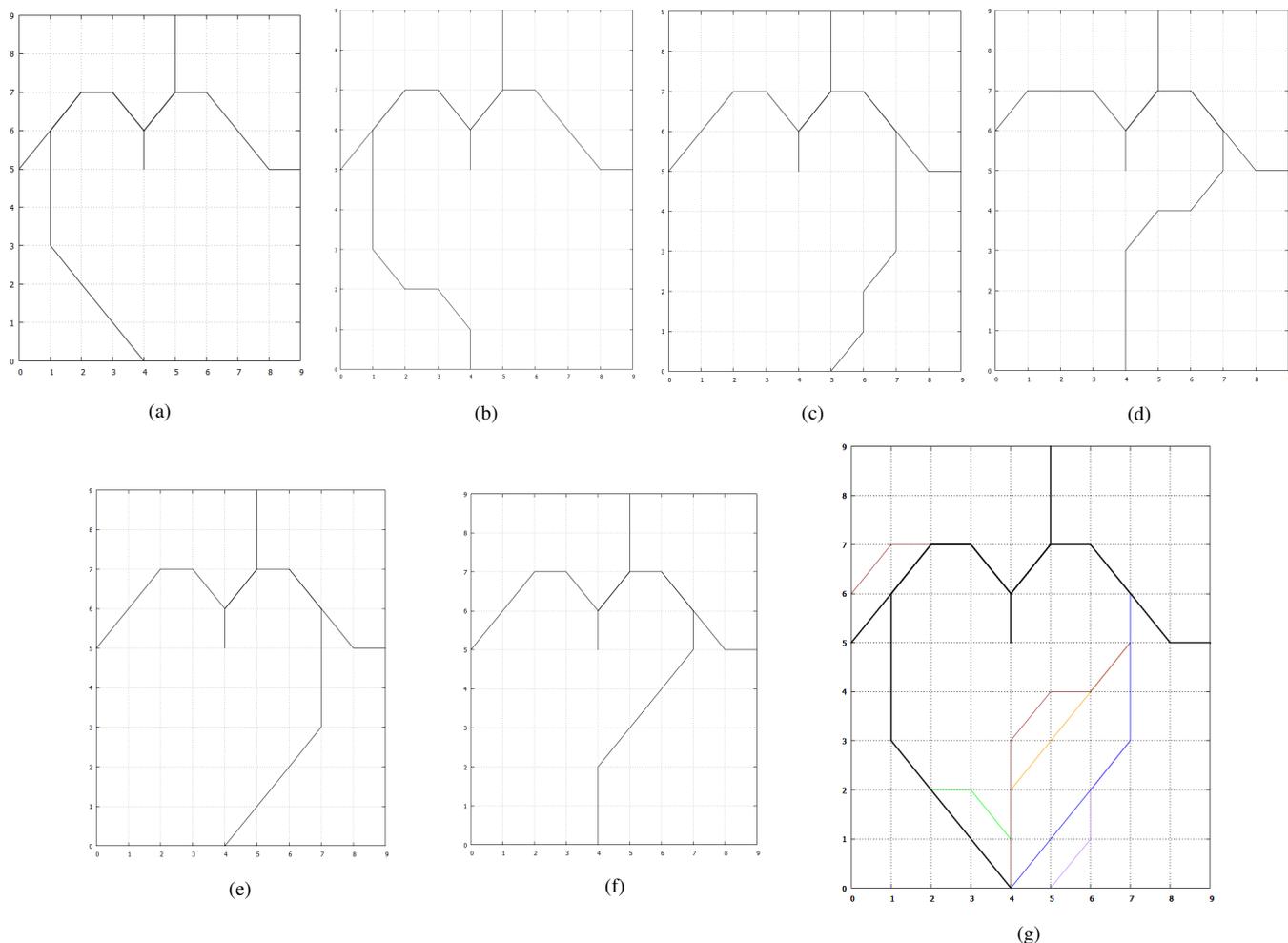


Fig. 4. Arrival route tree calculated for one hour of Stockholm Arlanda operation in October 4th, 2017, optimized for aircraft arrived at the following times: (a) 6 am to 7 am, (b) 7 am to 8 am (c) 8 am to 9 am, (d) 12 am to 1 pm, (e) 3 pm and 4 pm, (f) 4 pm to 5 pm. (g) shows all computed arrival trees in one figure.

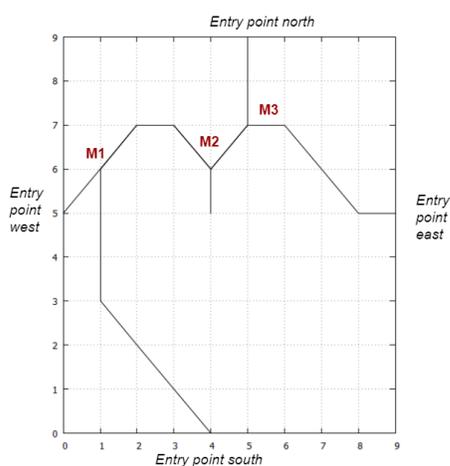


Fig. 5. Arrival route tree calculated for one hour of Stockholm Arlanda operation in October 4th, 2017, for aircraft arriving between 6 am and 7 am. Entry points and merge points are reference points for the time schedule presented in Table I.

TABLE I
EXAMPLE TIME SCHEDULE FOR 10 AIRCRAFT ARRIVED BETWEEN 6 AM AND 7 AM AT STOCKHOLM TMA ON OCTOBER 4, 2017.

Aircraft	Arrivals Entry point	Simulated time				RWY
		Entry	M1	M2	M3	
a1	south	1	7	10	x	11
a2	south	9	15	18	x	19
a3	south	10	16	19	x	20
a4	north	1	x	4	3	5
a5	north	11	x	14	13	15
a6	east	27	x	30	29	31
a7	east	1	x	6	5	7
a8	west	18	19	22	x	23
a9	west	23	24	27	x	28
a10	north	30	x	33	32	34

IV. CONCLUSION AND DISCUSSION

We presented a MIP-based approach to compute arrival routes that guarantee temporal separation of aircraft at all times for normal, that is, planned, operation. This can be

an important contributor to keep the ATCO workload at an acceptable level at all times. We gave a proof of concept and applied our framework to the arriving aircraft in Stockholm TMA.

Currently, we assume a unit speed on all edges, that is, we do not incorporate the slow-down of the aircraft, we plan to integrate this in future work—we plan to combine our work with [6]–[9] to account for aircraft dynamics.

We compute the set of trees for consecutive time periods during the day, hence, the tree suggested by the control system will switch from period to period. Sliding time windows may be applied in the future to handle aircraft movements that span two periods.

Uncertainties due to changing weather conditions or deviating aircraft arrival times are planned to be incorporated into the model in later stages as well. This could include varying aircraft speed for different possible weather scenarios. We plan to use robust optimization to make our model adaptable to deviating aircraft arrival times.

Our current model can incorporate static obstacles, e.g., no-fly zones. In future work we plan to integrate flexible obstacles, which can be recomputed dynamically together with the resulting arrival routes.

Moreover, currently we consider 2D routes with a temporal component, we plan to extend our study to 3D routes with a temporal component: 4D routes.

ACKNOWLEDGMENT

We thank Billy Josefsson (project co-PI, LFV) and the members of the project reference group—Eric Hoffman (Eurocontrol), Håkan Svensson (LFV/NUAC), Anne-Marie Ragnarsson (Transportstyrelsen, the Swedish Traffic Administration), Anette Näs (Swedavia, the Swedish Airports Operator), Patrik Bergviken (LFV), Håkan Fahlgren (LFV) for discussions.

REFERENCES

- [1] Aurora server, LUNARC, Lund University. <http://www.lunarc.lu.se/resources/hardware/aurora/>. Accessed on 29/05/2018.
- [2] IATA air passenger forecast shows dip in long-term demand. <http://www.iata.org/pressroom/pr/Pages/2015-11-26-01.aspx>, November 2015. accessed on June 2, 2016.
- [3] T. Andersson, T. Polishchuk, V. Polishchuk, and C. Schmidt. Automatic Design of Aircraft Arrival Routes with Limited Turning Angle. In *ATMOS*, 2016.
- [4] Dimitris J. Bertsimas and Garrett van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4):601–615, 1991.
- [5] S. Choi, J. E. Robinson, D. G. Mulfinger, and B. J. Capozzi. Design of an optimal route structure using heuristics-based stochastic schedulers. In *Digital Avionics Systems Conference (DASC), 2010 IEEE/AIAA 29th*, pages 2.A.5–1–2.A.5–17, Oct 2010.
- [6] Ramon Dalmau Codina, Ronald Verhoeven, Xavier Prats Menéndez, Frank Bussink, and Bart Heesbeen. Performance comparison of guidance strategies to accomplish RTAs during a CDO. In *Proceedings of the 36th DASC-Digital Avionics Systems Conference*, 2017.
- [7] Ramon Dalmau and Xavier Prats. How much fuel and time can be saved in a perfect flight trajectory? In *International Conference on Research in Air Transportation (ICRAT)*, 2014.
- [8] Ramon Dalmau and Xavier Prats. Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations. *Transportation Research Part D: Transport and Environment*, pages 62–71, 2015.

- [9] Ramon Dalmau and Xavier Prats. Assessment of the feasible CTA windows for efficient spacing with energy-neutral CDO. In *International Conference on Research in Air Transportation (ICRAT)*, 2016.
- [10] EUROCONTROL. <http://www.eurocontrol.int/articles/xman>. Accessed May 22, 2018.
- [11] EUROCONTROL. European airspace concept handbook for PBN implementation, edition 3.0. <http://www.eurocontrol.int/publications/airspace-concept-handbook-implementation-performance-based-navigation-pbn>, 2013.
- [12] Michel X. Goemans and Young-Soo Myung. A catalog of Steiner tree formulations. *Networks*, 23(1):19–28, 1993.
- [13] Jimmy Krozel, Changkil Lee, and Joseph S.B. Mitchell. Turn-constrained route planning for avoiding hazardous weather. *Air Traffic Control Quarterly*, 14(2), 2006.
- [14] NATS and Eurocontrol. Operational service and environment definition (OSD) for time based separation for arrivals (TBS), 2013. Project Number 06.08.01.
- [15] Djamila Ouelhadj and Sanja Petrovic. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4):417, Oct 2008.
- [16] G. Pavese, M. Bruglieri, A. Rolando, and R. Careri. DMAN-SMAN-AMAN Optimisation at Milano Linate Airport. In *Eighth SESAR Innovation Days*, 2017.
- [17] Diana Michalek Pfeil. *Optimization of airport terminal-area air traffic operations under uncertain weather conditions*. PhD thesis, Massachusetts Institute of Technology. Operations Research Center, 2011.
- [18] Joseph Prete, Jimmy Krozel, Joseph Mitchell, Joondong Kim, and Jason Zou. Flexible, Performance-Based Route Planning for Super-Dense Operations. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2008.
- [19] H. G. Visser and R. A. Wijnen. Optimization of Noise Abatement Departure Trajectories. *Journal of Aircraft*, 38(4):620–627, jul 2001.
- [20] Richard T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28(3):271–287, 1984.
- [21] Jun Zhou, Sonia Cafieri, Daniel Delahaye, and Mohammed Sbihi. Optimization of Arrival and Departure Routes in Terminal Maneuvering Area. In *ICRAT 2014, 6th International Conference on Research in Air Transportation*, Istanbul, Turkey, May 2014.
- [22] Jun Zhou, Sonia Cafieri, Daniel Delahaye, and Mohammed Sbihi. Optimizing the design of a route in Terminal Maneuvering Area using Branch and Bound. In *EIWAC 2015, ENRI International Workshop on ATM/CNS*, Tokyo, Japan, November 2015. ENRI.

APPENDIX

A. Complete MIP

$$\min \beta \sum_{e \in E} \ell_e x_e + (1 - \beta) \sum_{e \in E} \ell_e f_e \quad 0 \leq \beta \leq 1 \quad (8)$$

s.t.

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

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

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

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

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

$$\sum_{j:(r,j) \in E} x_{rj} \leq 0 \quad (12)$$

$$\sum_{k:(k,i) \in E} x_{ki} \leq 0 \quad \forall i \in \mathcal{P} \quad (13)$$

$$\sum_{j:(i,j) \in E} x_{ij} = 1 \quad \forall i \in \mathcal{P} \quad (14)$$

$$c_e x_e + \sum_{f \in \Gamma_e} x_f \leq c_e \quad \forall e \in E \quad (15)$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in V' \setminus \{\mathcal{P} \cup r\} : i+1+n, i+n, i+1 \notin \{\mathcal{P} \cup r\} \quad (16)$$

$$x_{i,i+1+n} + x_{i+n,i+1} + x_{i+1,i+n} \leq 1 \quad \forall i \in \mathcal{P} \quad (17)$$

$$x_{i,i+1+n} + x_{i+1+n,i} + x_{i+1,i+n} \leq 1 \quad \forall i : i+1 \in \mathcal{P} \quad (18)$$

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

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

$$x_{e,b} \leq x_e \quad \forall b \in \mathcal{P}, \forall e \in E \quad (21)$$

$$\sum_{j:(b,j) \in E} x_{(b,j),b} = 1 \quad \forall b \in \mathcal{P} \quad (22)$$

$$\sum_{j:(j,r) \in E} x_{(j,r),b} = 1 \quad \forall b \in \mathcal{P} \quad (23)$$

$$\sum_{i:(i,j) \in E} x_{(i,j),b} - \sum_{k:(j,k) \in E} x_{(j,k),b} = 0 \quad \forall j \in V \setminus \{\mathcal{P} \cup r\}, \forall b \in \mathcal{P} \quad (24)$$

$$y_{a,b,t_a} = 1 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b \quad (25)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A} \setminus \mathcal{A}_b, \forall t \in T \quad (26)$$

$$y_{a,b,t} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall t \in T \setminus \{t_a^b\} \quad (27)$$

$$y_{a,b,t} \leq \sum_{k \in V : (k,j) \in E} x_{(k,j)} \quad \forall a \in \mathcal{A}, \forall j \in V \setminus \mathcal{P}, \forall t \in T \quad (28)$$

$$z_{a,j,k,b,t} \leq x_{(j,k),b} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (30)$$

$$z_{a,j,k,b,t} \leq y_{a,j,t} \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (31)$$

$$z_{a,j,k,b,t} \geq x_{(j,k),b} - (1 - y_{a,j,t}) \quad \forall a \in \mathcal{A}, \forall j, k \in V, \forall b \in \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (32)$$

$$\sum_{j:(j,k) \in E} z_{a,j,k,b,t} - y_{a,k,t+u} = 0 \quad \forall b \in \mathcal{P}, \forall a \in \mathcal{A}_b, \forall k \in V \setminus \mathcal{P}, \forall t \in \{0, \dots, \bar{T} - u\} \quad (34)$$

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

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (4)$$

$$f_e \geq 0 \quad \forall e \in E \quad (3)$$