Urban Air Mobility: From Complex Tactical Conflict Resolution to Network Design and Fairness Insights

Mercedes Pelegrín, Claudia d'Ambrosio, Rémi Delmas, Youssef Hamadi

Urban Air Mobility: From Complex Tactical Conflict Resolution to Network Design and Fairness Insights

Mercedes Pelegrín[1,2], Claudia D'Ambrosio[1,2], Rémi Delmas[2], Youssef Hamadi[2]

Abstract

Urban Air Mobility (UAM) has the potential to revolutionize transportation. It will exploit the third dimension to help smooth ground traffic in densely populated areas. This new paradigm in mobility will require methods to ensure safety and maximize efficiency. We propose to use mathematical optimization to address tactical deconfliction in UAM. Our approach is envisioned as a way of modeling and solving tactical conflicts, but also as a means for assessing future infrastructures with potential utility in the design phase. We leverage envisioned UAM corridors to provide a mathematical definition of vehicle separation. A mathematical formulation is then proposed, which minimizes the total deviation from flight schedules needed to avoid loss of pairwise separation. The deconfliction is based on both airborne adjustments (through speed changes) and ground delays (holds relative to the scheduled take-off). Our experimental setup includes three use cases standing for different sources of conflicts and three synthetic UAM network topologies, which represent heterogeneous realistic scenarios. Vehicle and network capabilities are represented through model parameters, which allows us to analyze their impact on the quality of the deconfliction. Finally, insightful comparisons between our approach and both a local deconfliction and a fairness-oriented version are provided.

Keywords: Urban Air Mobility; Tactical Deconfliction; Optimization; Fairness.

1. Introduction

Urban Air Mobility (UAM) designates air transport systems that will move people and goods by air within and around dense city areas. Its purpose is to help smooth urban ground traffic despite the increasing population density. It has been estimated that around 45% of commuters could benefit from this new paradigm, when the roads are highly congested (BOS+21). UAM will exploit new electric vertical take-off and landing (eVTOLs) aircraft to carry passengers and freight. Operating eVTOLs over large and densely populated areas will require an organized approach able to balance efficiency and safety (Ham20; BDH21).

To meet this challenge, industry actors and regulators are gradually refining Concept of Operations (ConOps) for UAM (Fed20). At this stage, this ConOps generalizes existing helicopter corridors into dedicated air lanes for passenger and cargo eVTOLs without relying on traditional Air Traffic Control (ATC) services, see Figure 1. There is a consensus to

Email addresses: pelegringarcia@lix.polytechnique.fr (Mercedes Pelegrín), dambrosio@lix.polytechnique.fr (Claudia D'Ambrosio), remi.delmas.3000@gmail.com (Rémi Delmas), youssefh@lix.polytechnique.fr (Youssef Hamadi)

[1]LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France

[2]Integrated Urban Mobility (IUM) Chair, École Polytechnique

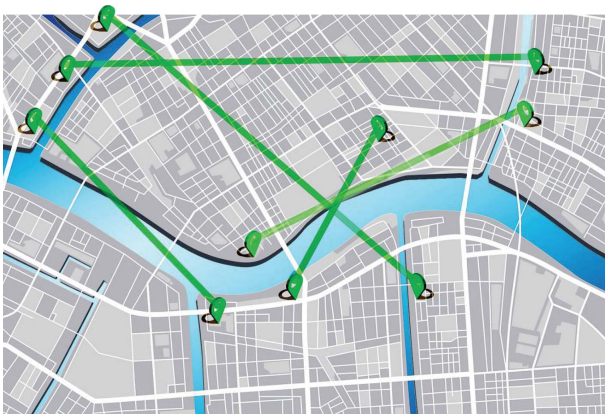


Figure 1: Multiple UAM corridors, (Fed20)

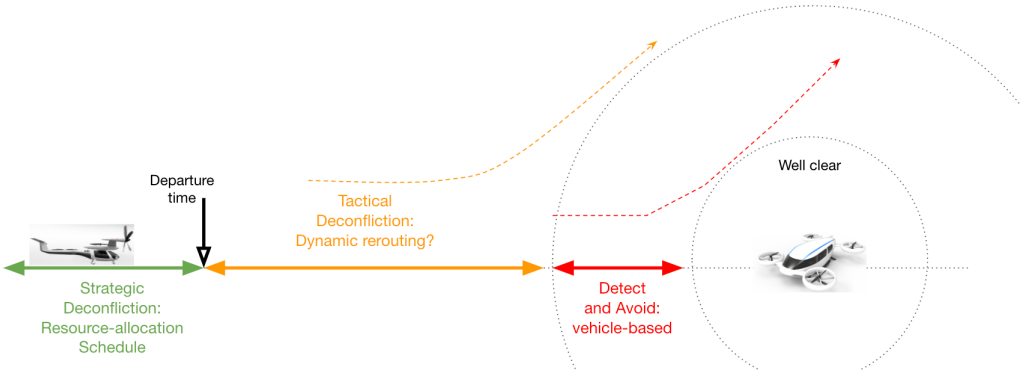


Figure 2: UAM deconfliction layers

consider that scaled UAM operations will require automated traffic management systems backed by human-in-the-loop supervision. In this work, we build on this common framework to provide optimization models and algorithms for general UAM tactical deconfliction. Tactical deconfliction is one of the three layers into which Air Traffic Management (ATM) has been typically divided. In the first layer, called strategic deconfliction, conflicts are detected and solved through the adequate allocation of resources prior to departure. It includes airspace organization and management, demand and capacity balancing, and traffic synchronization. Then, the goal of tactical deconfliction is to maintain separation provision of ongoing traffic. As a last resort, the collision avoidance layer (or detect-and-avoid) uses onboard technologies to evade an imminent collision. In between, the tactical layer is used to adjust operations to perturbation events. Figure 2 presents these general ATM layers in our UAM context.

We consider different perturbations events potentially causing conflicts within strategically scheduled traffic. Other than different sources of traffic disruption, the considered use cases stand for different levels of flight cooperation. Namely, we adjust ongoing traffic to avoid loss of separation provision due to: (i) Random variations: random disruptions within strategically deconflicted scheduled traffic; (ii) Priority traffic: accommodation of new priority operation (Rio18); (iii) Non-cooperative intruder: incursion of unplanned traf-

fic, e.g., an intruder crossing a UAM corridor. We use Mathematical Programming (MP) to approach UAM safety provision, proposing the first Mixed Integer Linear Programming (MILP) formulation that addresses the problem of tactical deconfliction within UAM networks. This model is tested under three different urban topologies, namely: (i) Grid: a city with a grid corridor layout; (ii) Airport: a single hub connected to different parts of the city; (Rio18); (iii) Metroplex: two well-connected urban centers interconnected by a single corridor.

1.1. Contributions and main findings

We present a complete study on optimization for UAM tactical deconfliction that goes from problem modeling based on current ConOps to the simulation of realistic cases. The main contributions of this work can be summarized as follows:

1. New aircraft separation constraints tailored to the UAM context are proposed. Vehicle separation has been typically modeled in classic ATM though nonlinear constraints conceived for free-flight trajectories rather than for corridors.

2. A MILP formulation for UAM tactical deconfliction is proposed. It ensures separation provision at minimum cost, which is measured as the deviation from the strategic flight schedules.

3. Three use cases, corresponding to different sources of disruptions, and three heterogeneous network topologies are studied. Different UAM system performance parameters are studied, including vehicle speed performance and intruder detection capabilities. This provides insightful results for the design phase of UAM infrastructures.

The main insights of this work can be summarized as follows:

1. Conflicts are harder to solve as traffic becomes less cooperative. Namely, speed changes and ground delays adjustments allow solving conflicts due to random variations (drifts and delays) in the flight schedules in less than 2 seconds. For the accommodation of a new priority operation, in 45% of the tested cases, the priority flight does not need to be delayed, while in 65% of the tests it needs to be delayed by 1 minute at most. The conflicts produced by the entrance of a non-cooperative intruder in the UAM network could only be solved in roughly 29% of the tests.

2. Doubling vehicle speed maximum increase rates (from a 5% to a 10% increment with respect to nominal speed) can (i) halve the total deviation from the schedule and avoid non-solved conflicts when random variations occur; (ii) avoid the delay of a new priority operation in the Grid and Airport topologies; (iii) help in solving the conflicts and reducing the deviation due to a non-cooperative intruder, although this is not sufficient to solve the conflicts in over 60% of the cases.

3. The Airport topology is the most affected by the random variations and priority traffic disruptions, requiring more deviation, delays, and speed adjustments to solve the conflicts. On the other hand, the Metroplex topology is the most impacted by a non-cooperative intruder entrance.

4. Regarding the delays affecting priority traffic, when it exists, these are smaller for the Airport topology. These delays were found difficult to avoid when the priority traffic travels within an area featuring small distances and when it has to traverse a single bottleneck corridor that connects two parts of a city.

5. We explore the implications of reducing the solution space. This is typically done by local approaches, which only focus on a reduced set of flights, which are those directly involved in some conflict. A comparative analysis between our approach and its "local" version is presented, from which we conclude that the saving on CPU time of the local version does not compensate for the loss of solution quality and the number of non-solved conflicts. This supports the idea that the "sky is big" assumption on which ATM local approaches were built does not hold in the scaled scenarios of UAM.

6. The cost of having fairer solutions, that is, a more equitable spread of the delays over the flights, is investigated. Our results indicate relatively small improvement of fairness leads to a relatively large degradation of overall delays.

1.2. Outline

We present related work in Section 2. Section 3 fully describes the problem of UAM tactical deconfliction. Section 4 is devoted to the mathematical modeling of flight separation within the UAM network. Potential conflicts are classified into different categories, for which distinct separation constraints are proposed. The mathematical programming formulation used to address tactical deconfliction is presented in Section 5. The experimental results are reported and analyzed in Section 6. Finally, Section 7 closes the paper with some conclusions and future perspectives.

2. Related work

The discussion on UAM traffic management is still at its very beginning. For an overview on open challenges in this emerging topic, we refer the reader to (SRS+20), while some studies on the integration of UAM operations in the airspace can be found in (VMM+20; VCBP20). Nonetheless, the number of works addressing UAM deconfliction at its different layers is growing (Xue20; CGE+21; BL18; RPJ+20; YW20; KMW18).

Instead of focusing on the design of a specific deconfliction algorithm, (Xue20) compares centralized and decentralized architectures on the three deconfliction layers. The work investigates the robustness of these two kinds of approaches to communication and state information errors in terms of safety and efficiency. Randomly generated free-flight trajectories are considered for the experiments. Conflicts are resolved locally for both decentralized and centralized approaches (the latter prioritizes the conflicts first, and resolves them sequentially).

At the strategic layer, (CGE+21) proposes an MP formulation for strategic flight planning that focuses on fairness and restricts deconfliction to a given sector capacity. On the other hand, the authors of (BL18) propose an adaptation of Auto Resolver, a greedy method, to address both climbing scheduling and trajectory management to maintain separation provision in UAM operations. The resulting service has both strategic and tactical capabilities, and is tested by the authors under different conditions. In their experiments, they consider a UAM network that consists of straight-line trajectories between pairs of vertiports.

At the collision avoidance layer, the authors of (RPJ+20) formulate aircraft deconfliction as a MILP, leaving multi-conflicts as future work. Based on their MILP, the authors propose a decentralized algorithm that is not guaranteed to always find conflict-free trajectories, though. Trajectories are simplified as position sequences, which have to be separated in terms of the infinity norm.

Most of the existing literature on tactical deconfliction falls in the context of ATM, where the focus is on giving support to human air traffic controllers. Tactical deconfliction typically receives a set of rectilinear trajectories crossing an observed air sector, and the task is to provide new trajectories that are conflict-free. ATM tactical deconfliction has been modeled as an optimization problem, where the objective function can be chosen among a range of possibilities such as the number of plan deviations, maneuvers implemented, fuel costs, or airlines fairness, among others. Existing MP formulations are varied, and a classifying taxonomy can be consulted in (PD21). Existing mathematical formulations of ATM tactical deconfliction represent aircraft separation provision in different ways. The common starting point is to consider the Euclidean distance between every pair of aircraft along their trajectories and enforcing it to be greater than a minimum threshold. However, different approaches are used to approximate these quadratic equations, which also define a non-convex space. This includes geometric (Bil00; PFB02), analytic (CD14; CD18), and time-separation (Ome15; RRFF16) approaches (see (PD21) for a detailed analysis). Among the three approaches, the latter suits UAM settings the most. Since UAM traffic is envisioned to travel on a virtual network, it has to be orchestrated around disputed resources (such as network junctions or vertiports) by imposing time lapses between the different passage times. In this work, we define these time lapses based on the UAM network layout, namely, the crossing angles featured at the network junctions.

Works addressing UAM deconfliction at the tactical layer are limited. We can cite (YW20), which proposes a distributed algorithm that runs on each eVTOL to provide tactical guidance to avoid loss of separation. The problem is formulated as a Multiagent Markov Decision Process and solved with a Monte Carlo Tree Search algorithm. Experiments are performed on free-flight two-dimensional trajectories using a simulator built by the authors. On the other hand, (KMW18) presents a MILP that just focuses on vehicle arrivals sequencing in real-time, which considers the battery status of the eVTOLs.

As opposed to existing distributed algorithms, we present a centralized global approach for UAM tactical deconfliction. Unlike decentralized algorithms, centralized deconfliction approaches are consistent with the UAM development strategic plan, which includes human-in-the-loop supervision (con20; NL20). Our approach is designed for network-structured problems instead of free-flights, which are a typical assumption of ATM approaches. We propose a fully declarative model for UAM tactical deconfliction, which is extensible in contrast to existing ad-hoc approaches. The parameters of the proposed model allow us to investigate different configurations and to draw insightful conclusions.

3. Problem definition

As envisioned by regulations (Fed20), we will assume that aircraft are routed through a well-defined *track network*. A track network brings internal structure to UAM corridors (Fed20). It adds directional transit lanes with specific performance requirements that help to support an increased operational tempo and refined control. Figure 3 illustrates UAM tracks. In this work, we model track networks as directed graphs. More specifically, the nodes of the graph consist of vertiports and cruise level nodes, where the cruise altitude is fixed. Cruise level nodes include one node per take-off and one per landing pad (which corresponds to the associated "entering"/ "exiting" point into/from the track network), and nodes corresponding to the intersection of the different tracks. Edges of the graph connect either pairs of nodes on the same track, or vertiports to the entering/exiting points at cruise level. A given corridor structure is only valid for some amount of time (weeks or months),

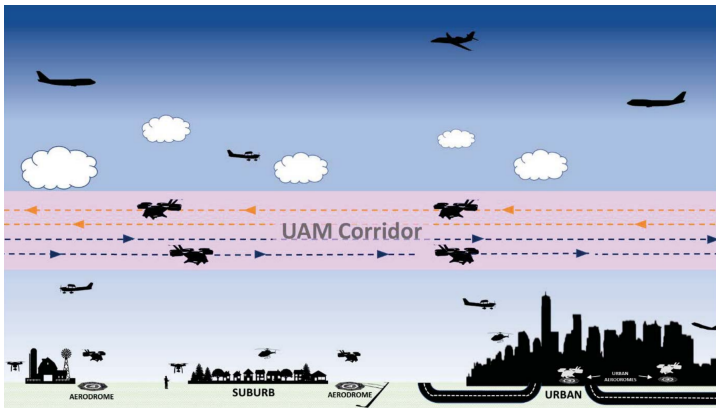


Figure 3: UAM Corridor with tracks, (Fed20)

and the track configuration can be switched over time within a given corridor structure. Since we focus on tactical deconfliction and short time horizon we assume the graph to stay the same during the deconfliction process.

We assume the following setting: strategic deconfliction produces a *nominal schedule* for a number of flights (or trips) over a given time window, in which each flight gets assigned several waypoints and associated traversal time intervals (which are delimited by an earliest and latest traversal times). In addition, the nominal schedule is conflict-free. That is, every pair of flights is separated at least by the minimum established safety distance throughout their paths on the track network. However, at the tactical level, the execution of the nominal schedule will most likely suffer small deviations due to disruptions, and may also include new flights such as emergency operations or network intruders. These perturbations can cause a loss of separation for some of the flights. The Tactical Deconfliction (TD) optimization problem can be stated as follows: Given a track network, a nominal schedule, and a set of conflict-causing perturbations, find a new conflict-free schedule with minimal deviation from the nominal schedule.

Our work focuses on this critical tactical deconfliction layer, where we will handle three different use cases. Each use case stands for a source of traffic disruption, as detailed below:

- Random variations: Some trips missed their departure due time and/or an in-flight trip suffered a drift. All the flights are willing to cooperate to ensure safety, regardless of who caused the conflict.

- Priority traffic: A new operation must be accommodated with high priority. The flights in the nominal schedule are willing to cooperate to give way to the new incoming traffic. This traffic is willing to implement modifications if this is the only solution available to avoid conflicts.

- Non-cooperative intruder: A non-cooperative intruder is expected to cross the UAM network. The flights in the nominal schedule are willing to cooperate to avoid conflicts due to the intruder's entrance, who is not willing to modify its trajectory.

The first use case is the most common and arises due to natural contingencies including passenger delays, weather conditions, or aircraft performance alterations. On the other hand, the need to integrate regular traffic with emergency or priority operations has been

identified in existing literature, (Rio18; VMM+20; RPJ+20). The last use case covers the event of unplanned traffic flying outside the UAM network that crosses some of its tracks at some point. In this case, existing traffic on the network must be adjusted to avoid conflicts with the intruder, which will not modify its trajectory. The three proposed use cases present a gradual level of cooperation, which goes from full-cooperative in the first use case to some non-cooperative trips in the latter, including different trip priorities in the second one.

The single MILP formulation we introduce in this work is general enough to address the three cases. It considers both airborne adjustments (through speed changes) and ground delays (holds relative to the scheduled take-off) as possible corrective actions to avoid conflicts. Before presenting it, we describe how the UAM separation constraints used in the formulation are derived.

4. Modeling of UAM aircraft separation

The main assumptions of all ATM aircraft separation approaches are rectilinear trajectories, i.e., heading angles are the same before and after the crossing points of trajectories. This is not the case in the UAM setting, where crossing points can also correspond with a heading change in the UAM track. In this section, we present specific separation constraints tailored for UAM track networks.

Flight separation for future UAM networks will be ensured by a specific lapse between passage times of different flights through the same network junction. For tactical separation, the considered common junction is always a cruise-level node. Although we assume that nodes traversal times are given by intervals, we analyse separation on single time instants for the sake of simplicity. The resulting separation inequalities will be then applied on the appropriate extremes of the arrival intervals so that safety is ensured. Let us denote by f_i,f_j a pair of flights and by $\mathcal{P}_i, \mathcal{P}_j$ their paths, which are sequences of nodes in the graph of the track network. For each common node $x \in \mathcal{P}_i \cap \mathcal{P}_j$, the passage times of the flights are denoted by t_{ix} and t_{jx}. UAM separation provision can be modeled by the following general-form separation constraint

$$|t_{jx} - t_{ix}| \geq \delta_{ijx}. \tag{1}$$

The required lapse δ_{ijx} will depend on the track network topology and, in theory, on vehicle performance profiles as well. However, according to current developments, a vehicle will have authorization to enter a track if an only if it has enough performance. In a general case we could say that each track segment has a performance profile constraint. Here, we propose an approach to define this lapse so that a minimum safety distance, D, will be always maintained between any pair of flights, bridging UAM separation with that of classical ATM. We will use v_{ix} to denote the speed at which f_i traverses x. Since speed changes are only allowed at the nodes of the graph, v_{ix} is assumed to be constant throughout the edge ending at x.

We distinguish two main families of conflicts, namely *trailing* and *non-trailing*. The former occurs when two flights are travelling along the same edge of the network, the latter when they are travelling on two different edges sharing a common source or destination node. Without loss of generality, we assume that $t_{ix} \leq t_{jx}$ in the rest of our analysis.

4.1. Trailing conflicts

Two flights f_i, f_j travel through the same edge (x, x') at the same time, if and only if $t_{ix'} > t_{jx}$. Figure 4 illustrates such conflict scenario. Since overtaking is forbidden

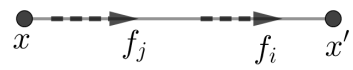
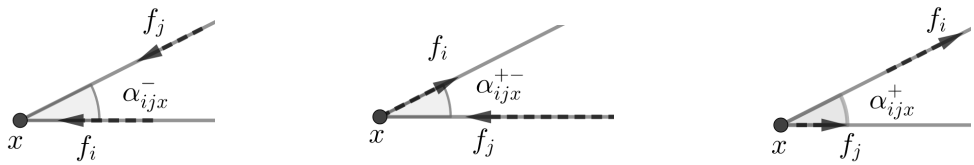


Figure 4: A trailing conflict



(a) Merging (b) Divergent (c) Splitting

Figure 5: Classification of non-trailing conflicts

within tracks and speed is assumed to be constant throughout each edge, the flights will be separated if passing times through x and x' are arbitrated as in (1). Classical ATM separation would be to ensure that vehicle positions at t_{jx} and $t_{ix'}$ are at a distance of at least D units:

$$v_{ix'}(t_{jx} - t_{ix}) \geq D \tag{2a}$$

$$v_{jx'}(t_{jx'} - t_{ix'}) \geq D, \tag{2b}$$

that is, $\delta_{ijx} = D/v_{ix'}$ and $\delta_{ijx'} = D/v_{jx'}$.

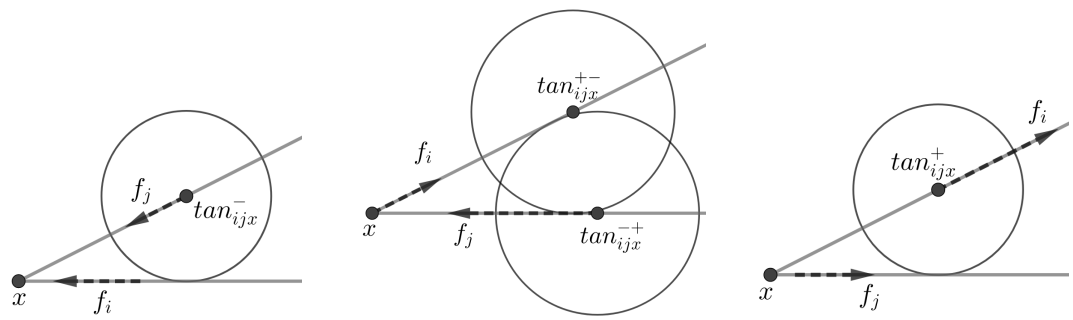
4.2. Non-trailing conflicts

When two flights traverse edges of the network that have only one of source or destination node in common, we speak of a non-trailing conflict. They can be classified into three different groups, which correspond to the possible relative positions of the vehicles, illustrated in Figure 5:

- *Merging*: both f_i and f_j fly on different edges towards x.
- *Diverging*: f_i is flying from x while f_j is moving towards x.
- *Splitting*: both f_i and f_j fly on different edges from x.

Note that more than one of the previous situations can take place for the same pair of trips and common node. Indeed, a pair of flights with crossing paths can pass through the three situations.

To arbitrate the traffic traveling through x, we need to consider the angle made by the edges starting/ending at this node. If this angle is very narrow, the paths are close and the time elapsed between t_{ix} and t_{jx} must be longer. To denote the angle made by the paths of f_i and f_j around x, we use $\alpha^-_{ijx}, \alpha^{+-}_{ijx}, \alpha^+_{ijx}$, respectively for the three groups of non-trailing conflicts (see Figure 5). Non-trailing conflicts take place in a surrounding of the common node x that we call the *conflict zone*. We consider *safety disks* of radius D around both f_i and f_j. The conflict zone begins at the instant at which one path is tangent to the safety disk of the other flight. We propose to provide time separation of passage through the conflict zones. This relates nicely to ATM strategies, since it ensures a minimum distance of D between the flights all their way throughout their paths. Namely, we propose the following arbitrating rule (we recall that we are assuming $t_{ix} < t_{jx}$):



(a) Merging (b) Divergent (c) Splitting

Figure 6: Conflict zones and separation on different conflict scenarios

$\mathcal{R}$: "f_j does not enter its conflict zone until f_i does not leave its own"

Figure 6 depicts the safety disks for the three identified groups of conflicts, whose centers are denoted by tan^-_{ijx}, tan^{-+}_{ijx}, tan^{+-}_{ijx}, and tan^+_{ijx} (note that we implicitly assume waypoints to be separated by at least D). The figure illustrates rule $\mathcal{R}$, which applies to the identified groups as follows:

- Merging: f_j does not traverse tan^-_{ijx} until f_i has passed x. The conflict zone of f_j starts when it gets to tan^-_{ijx}, which delimits the fragment of the edge traveled by f_j that is at a distance smaller than D from that traveled by f_i . On the other hand, the conflict zone of f_i in this scenario ends when it leaves x. We can shortly write $\mathcal{R}$ as

$$t_{ix} \leq time(j, tan^-_{ijx}), \tag{3}$$

  where $time(i, \bar{x})$ represents the arrival time of f_i at the point of the network $\bar{x}$.

- Diverging: f_j does not traverse tan^{-+}_{ijx} until f_i has passed tan^{+-}_{ijx}. In this case, both safety disks are needed in the rule statement, which reads

$$time(i, tan^{+-}_{ijx}) \leq time(j, tan^{-+}_{ijx}). \tag{4}$$

- Splitting: f_j does not traverse x until f_i has passed tan^+_{ijx}. In this scenario, the conflict zone of f_j starts when it is leaving x. Once f_i passes trough tan^+_{ijx}, it is always separated from f_j by at least D. In this case, $\mathcal{R}$ can be stated as

$$time(i, tan^+_{ijx}) \leq t_{jx}. \tag{5}$$

The arrival time of a flight f_i at $\bar{x}$, $time(i, \bar{x})$, can be obtained by considering the edge (x, x') in the path $\mathcal{P}_i$ that contains $\bar{x}$:

$$time(i, \bar{x}) = t_{ix} + \frac{d(x, \bar{x})(t_{ix'} - t_{ix})}{d(x, x')}, \tag{6}$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance between two points of the network. We note that the separation constraints defined by (3)-(5) are conservative, i.e., they are sufficient (but not necessary) to ensure a minimum safety distance D between the aircraft. However,

as we show next, these constraints are linear, which makes our model simpler. To rewrite (3)-(5) using (6), we just need to have an analytical expression for the distance between the crossing node x and the centers tan^{-}_{ijx}, tan^{-+}_{ijx}, tan^{+-}_{ijx}, and tan^{+}_{ijx}.

If the crossing angle of the paths in one of the three scenarios (namely, α^{-}_{ijx}, α^{+-}_{ijx}, or α^{+}_{ijx}) is greater than or equal to 90 degrees, then the corresponding safety disk(s) has (have) its (their) center at a distance D from x. Otherwise, to calculate the desired distance, let us consider the equations of two rays that have the origin as common point:

$$R_1 \equiv \lambda(1,0), \qquad \lambda \geq 0,$$

$$R_2 \equiv \gamma(\cos\alpha, \sin\alpha), \qquad \gamma \geq 0.$$

Note that these lines can model two edges ending/starting at node x, thus representing any of our three scenarios where α stands for the crossing angle of the paths in each case. The circle with center at R_2 and radius D has the following equation in coordinates (X, Y):

$$(X - \gamma\cos\alpha)^2 + (Y - \gamma\sin\alpha)^2 = D^2.$$

If we substitute (X, Y) by $(\lambda, 0)$, we get the intersection of this circle with R_1. The solution λ is unique (i.e., the circle is tangent to R_1) if and only if $\gamma = \frac{D}{\sin\alpha}$. Then, the circle centred at R_2 tangent to R_1 has its center at $\frac{D}{\sin\alpha}(\cos\alpha, \sin\alpha)$, being $\frac{D}{\sin\alpha}$ the distance we are looking for.

We can now present our UAM separation constraints, which are easily yielded from the previous development together with equations (3)-(6):

$$v_{jx}(t_{jx} - t_{ix}) \geq D \qquad \text{if } \alpha^{-}_{ijx} \geq \pi/2 \tag{7a}$$

$$v_{jx}(t_{jx} - t_{ix}) \geq \frac{D}{\sin\alpha^{-}_{ijx}} \qquad \text{if } \alpha^{-}_{ijx} < \pi/2 \tag{7b}$$

$$t_{jx} - t_{ix} \geq D \cdot \left(\frac{1}{v_{i,suc_i(x)}} + \frac{1}{v_{jx}}\right) \qquad \text{if } \alpha^{+-}_{ijx} \geq \pi/2 \tag{7c}$$

$$t_{jx} - t_{ix} \geq \frac{D}{\sin\alpha^{+-}_{ijx}} \cdot \left(\frac{1}{v_{i,suc_i(x)}} + \frac{1}{v_{jx}}\right) \qquad \text{if } \alpha^{+-}_{ijx} < \pi/2 \tag{7d}$$

$$v_{i,suc_i(x)}(t_{jx} - t_{ix}) \geq D \qquad \text{if } \alpha^{+}_{ijx} \geq \pi/2 \tag{7e}$$

$$v_{i,suc_i(x)}(t_{jx} - t_{ix}) \geq \frac{D}{\sin\alpha^{+}_{ijx}} \qquad \text{if } \alpha^{+}_{ijx} < \pi/2, \tag{7f}$$

where $v_{i,suc_i(x)}$ denotes the speed of f_i along the edge of $\mathcal{P}_i$ that leaves x.

5. Mathematical programming formulation

We propose a mathematical formulation to solve the TD problem within a given track network. To perform the deconfliction, the model decides of speed change maneuvers, which occur at cruise-level nodes, and of ground holding delays, which occur at vertiport nodes. As opposed to local approaches, these adjustments are decided on the global schedule. That is, they may not only focus on the conflicting aircraft or the instants at which the conflicts take place.

The mathematical model of this section considers two types of traffic, namely high priority traffic and the rest. If a flight has high priority, the model will try to preserve

its departure time. That is, this flight will be delayed only if no other adjustment of non-priority traffic can solve the conflicts. An example of such a situation is when a priority flight demands to leave a vertiport right after another flight has initiated its take-off from the same vertiport. The objective of the model is to minimize the total deviation, which is the overall difference between the new and nominal scheduled times of non-priority flights throughout their paths. We optimize the delay of priority traffic and the total deviation of non-priority traffic by weighting the priority traffic delay in the objective function with a sufficiently large constant.

Before presenting our mathematical programming formulation, we introduce some notations. We use $pre_i(\cdot)$ and $suc_i(\cdot)$ to denote the predecessor and successor of a node in $\mathcal{P}_i$, respectively. The following parameters and sets appear in the model:

- $\mathcal{F}$ the set of flights, $\mathcal{F} := \mathcal{F}^1 \cup \mathcal{F}^2 \cup \mathcal{F}^3$, where $\mathcal{F}^1, \mathcal{F}^2, \mathcal{F}^3$ represent, respectively, the regular, priority, and intruder flights.
- $\mathcal{P}_i = (x_1^i, \ldots, x_{k_i}^i)$, the path of flight f_i, where k_i is the length of the path and (x_r^i, x_{r+1}^i) are eges of the graph, $r = 1, \ldots, k_i - 1$.
- t^0, the beginning of the time horizon, i.e. the time at which the TD model is called.
- $\hat{t}_{ix}^{ear}$, $\hat{t}_{ix}^{lat}$ the scheduled earliest and latest traversing times of each waypoint $x \in \mathcal{P}_i$ for each flight $f_i \in \mathcal{F}$.
- $\underline{v}_{ix}, \bar{v}_{ix}$, minimum and maximum allowed speed of the flight f_i through $(pre_i(x), x) \in \mathcal{P}_i$. These bounds are determined by the vehicle performance profile and rules on the road.
- v_i^{climb}, the climbing speed of f_i.
- T, L, J, the set of vertiport nodes corresponding to take-off and landing pads, and the set of cruise-level nodes, respectively.
- $Conf := \{(i, j, x) : f_i, f_j \in \mathcal{F},\ x \in \mathcal{P}_i \cap \mathcal{P}_j,\ i < j\}$, set of potential conflicts.
- $Trail := \{(i, j, x, x') : f_i, f_j \in \mathcal{F},\ (x, x') \in \mathcal{P}_i \cap \mathcal{P}_j,\ x, x' \in J,\ i < j\}$, set of potential trailing conflicts between in-flight traffic.
- $Climb := \{(i, j, x) \in Conf :\ x \in T \cup L\}$, set of potential climbing conflicts.
- $Merge := \{(i, j, x) \in Conf :\ pre_i(x) \neq pre_j(x)\}$, set of potential merging conflicts.
- $Diver := \{(i, j, x) \in Conf :\ x \in J\}$, set of potential divergent conflicts.
- $Split := \{(i, j, x) \in Conf :\ suc_i(x) \neq suc_j(x)\}$, set of potential splitting conflicts.
- $S(\alpha) := \frac{1}{\sin \alpha}$ if $\alpha < \pi/2$, 1 otherwise.

The model's variables are:

- $t_{ix}^{ear}, t_{ix}^{lat}$ new scheduled times defining a safe time interval $[t_{ix}^{ear}, t_{ix}^{lat}]$ for every $f_i \in \mathcal{F}$ to traverse each $x \in \mathcal{P}_i$. Instant t_{ix}^{ear} (resp. t_{ix}^{lat}) is the earliest (resp. latest) time at which f_i can safely traverse x.

- $\underline{t}_{ix}, \bar{t}_{ix}$ bounding times to ensure that $f_i \in \mathcal{F}$ traverses $x \in \mathcal{P}_i$ without violating the speed limits, i.e. the following constraint

$$\underline{t}_{ix} \leq t_{ix}^{ear} \leq \bar{t}_{ix},$$

  must hold for every trip f_i and waypoint $x \in \mathcal{P}_i$.

- τ_i delay of the priority flight $f_i \in \mathcal{F}^2$.

Our mathematical formulation of TD reads as:

$$\min \sum_{f_i \in \mathcal{F}^1} \sum_{\substack{x \in \mathcal{P}_i: \\ x \neq x_2^i, x_{k_i}^i}} |\hat{t}_{ix}^{ear} - t_{ix}^{ear}| + M \sum_{f_i \in \mathcal{F}^2} \tau_i \tag{8a}$$

$$\text{s.t.} \quad t_{ix_1^i}^{ear} \geq \hat{t}_{ix_1^i}^{ear} \qquad \forall f_i \in \mathcal{F} : \hat{t}_{ix_1^i}^{ear} > t^0 \tag{8b}$$

$$t_{ix}^{lat} = t_{ix}^{ear} + \hat{t}_{ix}^{lat} - \hat{t}_{ix}^{ear} \qquad \forall f_i \in \mathcal{F}, x \in \mathcal{P}_i \tag{8c}$$

$$t_{ix}^{ear} = \hat{t}_{ix}^{ear} + \tau_i \qquad \forall f_i \in \mathcal{F}^2, x \in \mathcal{P}_i \tag{8d}$$

$$\underline{t}_{ix} \leq t_{ix}^{ear} \leq \bar{t}_{ix} \qquad \forall f_i \in \mathcal{F}, x \in \mathcal{P}_i \tag{8e}$$

$$\underline{t}_{ix} = t_{i,pre_i(x)}^{ear} + \frac{d(pre_i(x), x)}{\bar{v}_{ix}} \qquad \forall f_i \in \mathcal{F}, x \in \mathcal{P}_i : pre_i(x), x \in J \tag{8f}$$

$$\bar{t}_{ix} = t_{i,pre_i(x)}^{ear} + \frac{d(pre_i(x), x)}{\underline{v}_{ix}} \qquad \forall f_i \in \mathcal{F}, x \in \mathcal{P}_i : pre_i(x), x \in J \tag{8g}$$

$$\underline{t}_{ix} = \bar{t}_{ix} = t_{i,pre_i(x)}^{ear} + \frac{d(pre_i(x), x)}{v_i^{climb}} \qquad \forall f_i \in \mathcal{F}, x \in \mathcal{P}_i : pre_i(x) \in T \text{ or } x \in L \tag{8h}$$

$$\underline{v}_{ix'}(t_{jx} - t_{ix}) \geq D \text{ or } \underline{v}_{jx'}(t_{ix} - t_{jx}) \geq D \qquad \forall (i, j, x, x') \in Trail \tag{8i}$$

$$\underline{v}_{jx'}(t_{jx'} - t_{ix'}) \geq D \text{ or } \underline{v}_{ix'}(t_{ix'} - t_{jx'}) \geq D \qquad \forall (i, j, x, x') \in Trail \tag{8j}$$

$$t_{jx}^{ear} - t_{ix}^{lat} \geq S(\alpha_{ijx}^{-}) \frac{D}{\underline{v}_{jx}} \text{ or } t_{ix}^{ear} - t_{jx}^{lat} \geq S(\alpha_{ijx}^{-}) \frac{D}{\underline{v}_{ix}} \qquad \forall (i, j, x) \in Merge \tag{8k}$$

$$\begin{aligned} & t_{jx}^{ear} - t_{ix}^{lat} \geq S(\alpha_{ijx}^{+-}) \left(\frac{D}{\underline{v}_{jx}} + \frac{D}{\underline{v}_{i,suc_i(x)}} \right) \text{ or} \\ & t_{ix}^{ear} - t_{jx}^{lat} \geq S(\alpha_{ijx}^{+-}) \left(\frac{D}{\underline{v}_{ix}} + \frac{D}{\underline{v}_{j,suc_j(x)}} \right) \end{aligned} \qquad \forall (i, j, x) \in Diver \tag{8l}$$

$$\begin{aligned} & t_{jx}^{ear} - t_{ix}^{lat} \geq S(\alpha_{ijm}^{+}) \frac{D}{\underline{v}_{i,suc_i(x)}} \text{ or} \\ & t_{ix}^{ear} - t_{jx}^{lat} \geq S(\alpha_{ijm}^{+}) \frac{D}{\underline{v}_{j,suc_j(x)}} \end{aligned} \qquad \forall (i, j, x) \in Split \tag{8m}$$

$$t_{ix_1^i}^{ear} \in \mathbb{Z} \qquad \forall f_i \in \mathcal{F} \tag{8n}$$

$$\tau_i \in \mathbb{Z} \qquad \forall f_i \in \mathcal{F}^2 \tag{8o}$$

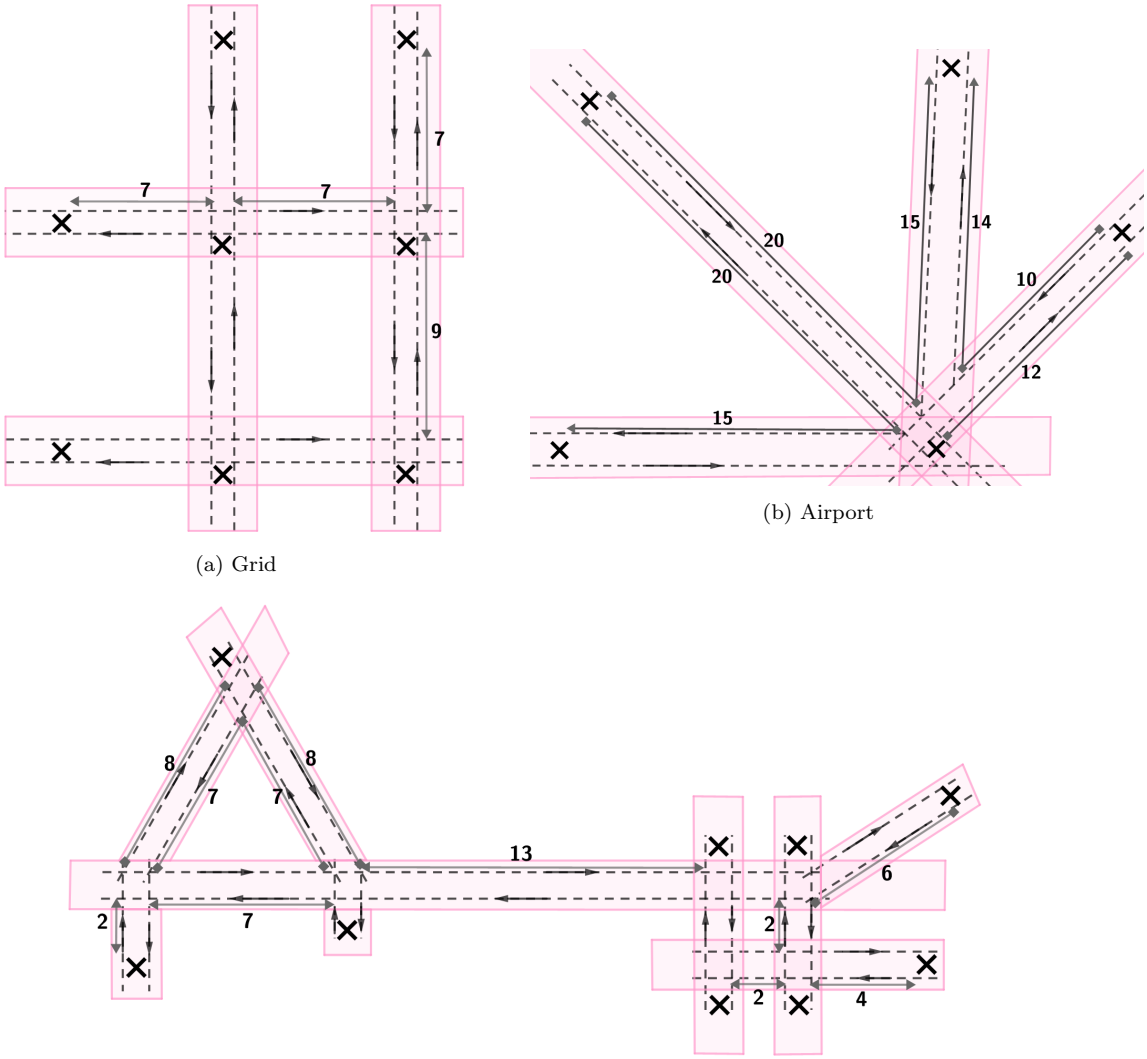
The objective function (8a) consists of two terms, namely, the total deviation from the nominal schedule and the delay in the departure of the priority traffic. The total deviation is the sum of the absolute differences between the nominal and newly scheduled times of passage at all cruise-level nodes. The first node at cruise level ($x_2^i \in \mathcal{P}_i$) and the destination node ($x_{k_i}^i \in \mathcal{P}_i$) for each flight f_i are not considered since the climbing speed after take-off is fixed. On the other hand, only earliest passage times are considered in the summation, since latest passage times are a dependent variable of the first, see constraints (8c). The second term in (8a) is zero if there is no priority traffic. Otherwise, a large coefficient M ensures that only the strictly necessary delays are applied to priority traffic. The first term in the objective (8a) favours nominal flight speed recovery as a byproduct. In effect, flights will come back to their nominal plans in any optimal solution, as far as this does not involve conflicts. The absolute value in (8a) can be easily linearized by introducing additional continuous non-negative variables and linear constraints. However, we chose this non-linear representation of the objective function for the sake of readability.

The constraints impose relations between time instants, (8b)-(8d); ensure that speed bounds are respected, (8e)- (8h); and enforce separation (8i)-(8m). More precisely, (8b) restrict delays to ground traffic only. In fact, all variables t^{ear} and t^{lat} corresponding to already passed waypoints (before t^0) will be fixed to the corresponding nominal schedules given by $\hat{t}^{ear}$ and $\hat{t}^{lat}$. Then, (8c) states the value of the latest arrival time in terms of the earliest one, by adding up the same margin as that of the nominal schedule. Equations (8e)-(8g) establish the lower and upper bounds on passage times at cruise level, which respond to operational constraints on the speed. Supposing that speed changes are not allowed for some flight f_i at some waypoint x, one just sets maximum and minimum speed to a nominal value $\hat{v}$, $\underline{v}_{ix} = \bar{v}_{ix} = \hat{v}$. On the other hand, constraints (8h) fix the required time for taking off and landing, for which speed adjustments are not permitted. Finally, the rest of the constraints ensure flight separation as detailed in Section 4. All of them are disjunctive constraints, where disjunctions stand for the passage order of the trips though the common node. In mathematical programming, disjunctive constraint are usually modeled with binary variables, in our case:

$$z_{ijx} = 1 \text{ if i passes through } x \text{ before } j, \text{ 0 otherwise.}$$

We use these variables when one of the trips belongs to $\mathcal{F}^2$ or $\mathcal{F}^3$ (use cases of priority traffic and non-cooperativee intruder), or when it suffered a delay disruption (random variations use case). Otherwise, we keep the passage order given by the nominal schedule, i.e., we only impose one of the constraints for each disjunction in (8i)-(8m). Finally, (8n) and (8o) enforce departures to happen at integral time values (e.g. at minute 1, 2, ...).

Remark 1. *Formulation (8a)-(8n) can be used to address TD in any of the use cases described in Section 3. The less intuitive case is that of the non-collaborative intruder. However, to tackle this case, it suffices to consider an "extended graph" that includes the trajectory of the intruder. New nodes will be added for each intersection of its trajectory with existing tracks, which will be linked by new edges. The resulting graph will be our input graph, the nominal plan will be updated with the intruder's schedule, and we will fix* $t_{ix}^{ear} = \hat{t}_{ix}^{ear}, t_{ix}^{lat} = \hat{t}_{ix}^{lat}$, *for all* $f_i \in \mathcal{F}^3$ *and* $x \in \mathcal{P}_i$ *(since* f_i *is non-cooperative). In all the use cases, we distinguish between in-flight and ground traffic, using the temporal reference* t^0. *Ground traffic can be delayed and/or receive speed adjustments, while in-flight traffic only admits the latter.*



(a) Grid

(b) Airport

(c) Metroplex

Figure 7: Track network topologies

6. Computational experiments

This section presents the computational experiments resulting from solving model (8) for the three presented use cases under three UAM network topologies. The model is solved with a commercial MP solver. We first present the experimental design, namely, the testing instances and experimental setup, including the software used and the model's parameters choice. Then, the results from our tests are discussed. This includes a thorough study of each use case, and a comparative analysis between formulation (8) and its local and fair versions, which are also introduced within the section.

6.1. Testing instances

We consider three UAM network topologies, illustrated on Figure 7, which shows traveling times in minutes and vertiports represented by the symbol $\times$. These represent different types of layouts that might be encountered in future UAM corridors:

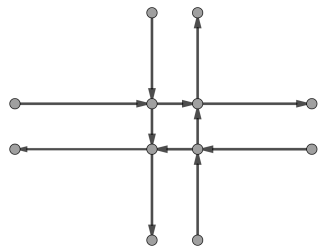


Figure 8: Modeling of track crossing through a directed graph

	req.	trips	trip duration			trip waypoints [3]			in-conflict flights		
			mean	min	max	mean	min	max	rand.	prior.	intru.
Grid	25	93	16.9	10	25	7.5	5	10	1.2	2.4	4.2
	27	103	16.9	10	25	7.3	5	10	3.5	1.6	3.6
Airport	25	45	21.3	16	24	6.3	6	8	1.2	1.6	2
	30	43	21.1	16	24	6.4	6	8	1.3	1.8	3.6
Metroplex	27	115	14.0	7	24	8.3	5	12	2.3	2.2	6.4
	28	117	14.0	7	24	8.1	5	12	2.4	1.6	6

Table 1: Properties of used SD schedules

1. Grid: a regular city with 8 vertiports, where the track network has a grid layout.
2. Airport: an urban area with 4 vertiports connected to an airport through a fifth vertiport.
3. Metroplex: two densely-connected city centers with 3 and 6 vertiports, respectively, connected through a single corridor.

These topologies are modeled with directed graphs, where the crossing of two tracks is designed as depicted in Figure 8. Since speed changes are supposed to happen at the nodes of the graph, we add "breakpoints" nodes along the edges (every 1 min. of travelling time) to allow for higher flexibility.

For each topology, two nominal schedules generated by the Strategic Deconfliction (SD) layer are considered. The SD schedules are generated by considering 5 batches of 10 minutes each. A few minutes before the beginning of each batch, the SD service receives a set of requests for trips. The requests are defined by origin and destination pairs, and trip duration in minutes is supposed to be within the interval $[7, 25]$. We run the SD service with a fixed number of requests per batch, which we increase until maximum capacity is reached. That is, the number of requests is increased until the number of accommodated trips converges to its maximum. Table 1 summarizes some information about the resulting SD schedules. The columns show: the trip requests per batch, the total number of accommodated trips, the mean/min/max trip duration in minutes, the mean/min/max number of waypoints per trip, and the average number of in-conflict flights per use case.

Once a SD schedule is generated, we consider it in its entirety framed into a single time horizon, which is approximately 50 minutes (10 minutes per batch, plus some more minutes depending on the latest arrival time of the trips to their destination). The generated SD

[3] waypoints do not include the "breakpoints" nodes

schedules are disrupted to simulate perturbation events at tactical level, according to the three typical use cases described in Section 3. The instant time at which each disruption takes place is randomly chosen within the time horizon. In the following, we detail how these disruptions are generated for each use case:

- **Random variations**: disruptions consist in randomly drifting or delaying some trips. Drifts and delays are also considered to happen together. Drifts are deviations (ahead or delays) of a flight from its earliest scheduled arrival at one waypoint, and are randomly generated in the interval $(0, 0.25]$ minutes. On the other hand, a trip is delayed if it is still on ground when its latest departure time is due. If so, the pilot has to wait to receive a new schedule for safely performing its trip.

- **Priority traffic**: one request for a (new) emergency trip is generated at a randomly chosen time instant, t. The origin and destination of this trip is also randomly generated. This trip will be ideally scheduled to depart at time $\lceil t \rceil$. Five random paths of the priority flight are generated at different time instants.

- **Non-cooperative intruder**: one (new) non-cooperative intruder is generated at a random time instant. The trajectory of the intruder is supposed to be rectilinear and to cross the track network. In this case, five random intruder's trajectories are generated at different time instants.

6.2. Experimental setup

All experiments were run on a 16-core Intel i9-9880H CPU at 2.30GHz running under Ubuntu 18.04.4 LTS 64 bits. The mathematical programming model was implemented in the algebraic modeling language AMPL (Version 20200501) and solved with the commercial solver CPLEX (Version 12.10).

Recent UAM research conducted at NASA Ames Research Center assumed that the capacity at each vertiport was one arrival and one departure per minute, see (VMM+20). We also assume the same capacity at vertiports, and allocate flight departures at the beginning of the minute. As mentioned earlier in the text, vertiports and cruise nodes are at two different altitude levels, which are separated by 1.5 NM in our experiments. We use a minimum separation provision of $D = 0.5$NM, a standard also considered in related works (CB15; YW20). We consider vehicle default and climbing speeds to be respectively $v = 150$ mph, and $v^{climb} = 90$ mph. The actual optimal speed of eVTOLs will depend on the development of the electric propulsion technology, but former studies estimate it to range between 150 and 200 mph (HG16). The allowed speed variation is a tuning parameter in our study. We set it either to $\pm 5\%$ or $-5\%, +10\%$, which gives the bounds $\underline{v}_{ix} = 142.5$ mph and $\bar{v}_{ix} = 157.5$ mph or $\bar{v}_{ix} = 165$ mph, for all $f_i \in \mathcal{F}$ and $x \in \mathcal{P}_i$. As already mentioned, the schedule of trips is a set of earliest and latest arrival times at waypoints. The width of these intervals is a parameter, $t_{err} = \hat{t}^{lat}_{ix} - \hat{t}^{ear}_{ix}$ for all flights $f_i \in \mathcal{F}$ and waypoint $x \in \mathcal{P}_i$. We assume $t_{err} = 1$ minute. This implies that at least 2 minutes should be blocked for the departure/arrival of a flight: one minute from the earliest until the latest time for its departure/arrival, and one minute of separation with the departure/arrival of the next flight.

There are some additional parameters characterizing each of the considered use cases. Most of them measure the degree of disruption affecting the schedules. Namely, for the random variations use case, we have the number of trips suffering a drift and the number of delayed trips. The number of trips affected by a drift is at most one. The detection

of a drift and the call to the tactical deconfliction model is supposed to happen at once, then the accumulation of several drifts is unlikely. As for the ground delays, we do tests with 1, 2, and 3 delayed trips for the same instance. This choice is proportional to the dimension of the considered instances, which feature between 5 and 9 vertiports (see Figure 7). Finally, we also consider the case of one ground delay and one flight affected by a drift happening together. For the priority traffic and non-cooperative intruder use cases, due to the on-demand nature of tactical deconfliction, the number of new in-coming operations is assumed to be always one. Since tactical deconfliction service is assumed to be called on demand, we assume that not much unplanned flights might come into place at the same time, which motivates our choice of parameters. Finally, there is a last parameter used for the non-cooperative intruder use case. It is denoted by r, and it represents the length of the time window starting when the intruder is detected and ending at the instant at which it intersects the first network track. We assume that the deconfliction is performed immediately after detection. This parameter is tuned in our experiments, $r \in \{1, 5, 10\}$ minutes. That is, the tactical deconfliction model is called r minutes before the first intersection of the intruder's trajectory with a track of the UAM network. This allows us to analyse our model performance under different foreseeing capabilities of the detection system.

The generation of the test instances together with the different values of the tuning parameters makes a total of 60 tests for each of the random variations and priority traffic use cases, and 180 test for the non-cooperative intruder use case.

6.3. Results and analysis

In this section we discuss computational results for each of the use cases, see Sections 6.3.1-6.3.3. Moreover, in Section 6.3.4 we compare the proposed mathematical formulation with its local version, which only allows adjustments on a reduced set of flights. Finally, in Section 6.3.5 we analyse the cost of fairness in the solution.

6.3.1. Random variations

Table 2 shows average results on the performance and solutions of formulation (8) for the random variations use case. It displays the mean and maximum running times over the tests, and the number of infeasible instances. All instances and configurations were solved in less than two seconds, and only one was found to be infeasible. It corresponds to the Metroplex topology, and contains a single drift of -0.22 minutes (before schedule) and a single delayed flight. Speed could be adjusted in $\pm 5\%$ in the corresponding configuration. With maximum speed variation set to 10%, the same instance is deconflicted in 0.14 seconds.

As for the optimal solutions found, Table 2 shows average results on the deviation from the nominal schedule, delays, and required speed changes. The total deviation accounts for the value of the first term in the objective (8a), that is,

$$\sum_{f_i \in \mathcal{F}^1} \sum_{\substack{x \in \mathcal{P}_i : \\ x \neq x_2^i, x_{k_i}^i}} |\hat{t}_{ix}^{ear} - t_{ix}^{ear}|.$$

The mean deviation is obtained when the total deviation is divided by

$$\sum_{f_i \in \mathcal{F}^1} (|\mathcal{P}_i| - 2),$$

which is the total number of waypoints of the different paths where the deviation is measured. The table also displays some results concerning the delay at the origin and destination

		Grid	Airport	Metroplex	Overall
cpu (s.)	mean	0.18	0.35	0.29	0.27
	max	0.73	1.65	1.06	1.65
	num. infeasible	0	0	1	1
deviation	total	137.24	225.10	151.82	171.72
	mean (per f_i, x)	0.16	0.31	0.19	0.22
delay at O	% trips	5.35	19.88	7.02	10.81
	total	10.20	13.55	13.26	12.32
	mean (per f_i)	0.14	0.33	0.16	0.21
	max (per f_i)	5.50	2.85	5.68	4.66
delay at D	% trips	5.75	27.37	7.89	13.77
	total	7.56	11.67	10.00	9.74
	mean (per f_i)	0.10	0.28	0.13	0.17
	max (per f_i)	5.26	2.14	5.30	4.22
speed changes	% trips	11.34	33.71	12.56	19.32
	% waypoints	9.97	29.53	11.83	17.20
	max (per f_i)	15.10	18.10	14.32	15.86
% obj. improv.	mean	43.71	51.77	41.81	45.90
	max	75.23	75.23	75.23	75.23

Table 2: Use case of random variations, 60 tests (20 tests per topology). Deviations and delays in minutes.

over the trips. Namely, we report the percentage of trips affected by these delays, the total delay in the overall schedule, and the mean and maximum delay over the trips. Finally, the last rows of the table account for the speed changes in the solution. They show the percentage of trips/waypoints that perform speed changes/at which speed changes are performed, and the maximum number of speed changes performed per eVTOL. The percentage of trips is always calculated with respect to the total number of on-flight and ready-to-departure flights (some eVTOLs might have already arrived to their destination when (8) is executed, which are discarded).

Speed changes and delays at the origin can be seen as a measurement of the adjustments needed to solve conflicts. On the other hand, deviations and delays at the destination are a consequence of the adjustments made, and give an idea of the impact that they have on the nominal schedule. From Table 2, we can see that tactical deconfliction requires more adjustments for the Airport topology, which produce a greater impact on the schedules. This is a consequence of the strong dependency between the paths in this topology, which all converge to the same hub. The percentage of delayed trips at origin and destination is similar on average, except for the Airport topology. This can be explained by the greater number of speed changes featured by this topology, which increases the number of trips that arrive at their destination with delay. Indeed, this greater number of maneuvers corresponds in a larger extent to actions needed to avoid en-route conflicts (rather than to recover the original plan). The ultimate objective of the deconfliction, given in (8), should not be forgotten. This is to minimize the deviation from the nominal times, which includes recovering the original schedule after adjustments such as delays or speed changes are implemented. Indeed, the effect of recovery can be appreciated by comparing the total, mean, and maximum delays at origin and destination, the latter being smaller.

Finally, at the bottom of the table, a comparison on the two speed changes bounds

		Grid	Airport	Metroplex	Overall
cpu (s.)	mean	0.29	0.31	0.36	0.32
	max	0.46	0.47	1.81	1.81
	num. infeasible	0	0	0	0
deviation	total	157.66	763.94	775.54	565.71
	mean (per f_i, x)	0.11	0.93	0.54	0.53
delay prior.	% instances	60	45	60	55
	total	3.95	0.70	2.70	2.45
	max (among instances)	14	5	8	14
delay at O	% trips	9.79	56.40	18.59	28.26
	total	12.55	38.95	66.00	39.17
	mean (per f_i)	0.13	0.89	0.57	0.53
	max (per f_i)	2.00	2.20	1.55	1.92
delay at D	% trips	11.26	56.46	19.60	29.11
	total	7.83	43.24	56.71	35.93
	mean (per f_i)	0.08	0.98	0.49	0.52
	max (per f_i)	1.50	1.92	1.30	1.57
speed changes	% trips	15.13	61.95	20.99	32.69
	% waypoints	12.92	63.77	19.70	32.13
	max (per f_i)	19.30	21.85	8.70	16.62
% obj. improv.	mean	48.15	18.64	7.29	24.70
	max	99.82	99.43	32.73	99.82

Table 3: Use case of priority traffic, 60 tests (20 tests per topology). Deviations and delays in minutes.

used is given. Namely, we compare the value of (8a) in the optimal solution for both configurations, i.e. valid changes in $[-5, 5]\%$ and $[-5, 10]\%$ with respect to the default speed. To that aim, we calculate the following percentage for each instance k,

$$100 \cdot \frac{Obj_k^{5\%} - Obj_k^{10\%}}{Obj_k^{5\%}},$$

where $Obj_k^{5\%}$ and $Obj_k^{10\%}$ represent the objective value (8a) in the solution for valid changes in $[-5, 5]\%$ and $[-5, 10]\%$, respectively (note that $Obj_k^{10\%} \leq Obj_k^{5\%}$). The rows display the mean and maximum of the above percentages over the set of instances. These results show that the total deviation from the nominal schedule (which in this use case coincides with (8a)) can be halved by doubling the allowed speed increase rates.

6.3.2. Priority traffic

Table 3 summarizes average experimental results for the priority traffic use case. This table collects similar information to that in Table 2. In this case, deviations, delays, and speed changes refer only to non-priority traffic. This table also includes statistics on the delay of the priority traffic (which always consists in a single trip). Namely, we can find the percentage of instances for which the priority flight suffers delay; the average total delay of priority traffic; and the maximum number of minutes that the priority flight has to wait until it can depart, among the different instances tested.

All the instances were solved to optimality in less than two seconds. The deviation from the nominal plan is of less than one minute per flight and waypoint on average. As for the

delay of the priority traffic, which is the main objective, the best performance is obtained in the Airport topology. In this topology, tracks are dedicated to single trajectories, which limits the interference with the scheduled traffic when it comes to accommodate the priority flight. For the Grid and Metroplex topologies, some of the tests yield important waiting times for an emergency operation (14 and 8 minutes, respectively). Regarding the delay of non-priority traffic at origin and destination, this is slightly above 2 minutes at most on average, which seems reasonable. Similarly to the previous use case, the Airport topology is the most affected by the delays, as it is also the one requiring speed changes the most. To sum up, the Airport topology has lower delays for the priority flight, which results in larger delays for the rest of the traffic.

Finally, as for the comparison between the two vehicle performances, we conclude that doubling the valid rates of speed increase allows to avoid delaying the priority traffic in some instances belonging to the Grid and Airport topologies (which accounts for the percentages of objective improvement close to 100). In most of the other cases, the delay of the priority flight is reduced (up to 9 and 5 minutes, respectively). The Metroplex topology is an exception: here the increase in the allowed speed changes barely affects the delay of the priority traffic, which remains the same or decreases in one minute. This might be revealing an underlying complexity linked to the network structure rather than to the vehicles' performance rates. Indeed, when we look at the path of the priority flight, we find that, for those paths within the triangle part of the Metroplex (see left-hand side of Figure (7c)), priority traffic have 0 or 1 minute of delay. On the other hand, paths within the square part (see right-hand side of Figure (7c)), which features smaller traveling distances, have delays ranging from 4 to 8 minutes. Finally, those paths passing through the single corridor connecting the triangle and the square area have the greatest delays, between 7 and 8 minutes.

6.3.3. Non-cooperative intruder

Table 4 gathers average results concerning the non-cooperative intruder use case. The table includes, other than analogous information to that in Tables 2 and 3, some additional details on the infeasible instances, which abound for this use case. Namely, the number of infeasible instances is expressed as a function of r. We recall that, for a given topology, the total number of tests when r is fixed is 20. It can be seen from the table that the number of infeasible instances decreases when r increases. However, nearly half of the instances are still infeasible with a time window of $r = 10$ minutes to implement adjustments before the intruder intersects the network.

As for the consequences of the deconfliction, the Metroplex topology seems to be the most impacted. The differences with respect to the nominal schedule are the greatest if we look at the deviation and delays, and the percentage of trips and waypoints receiving speed adjustments is also higher.

The improvement of the objective as a result of more flexible rates of speed increase is expressed only for those instances that are feasible. In addition, the number of instances that became feasible when the allowed speed changes are in $[-5, 10]\%$ is displayed. The results suggest that vehicle performance can help in the deconfliction against a non cooperative intruder, but is not sufficient. Other adjustments that allow a greater degree of freedom to deconflict the traffic should be considered for this use case, such as re-routing or other emergency maneuvers, e.g., exiting a UAM corridor.

Table 5 provides an overview on the computational results obtained for the three studied use cases. The overall conclusion from our study is related to the degree of freedom of the

		Grid	Airport	Metroplex	Overall
cpu (s.)	mean	0.60	0.05	0.28	0.17
	max	0.20	0.08	0.37	0.37
	num. infeasible	41	46	41	128
	inf. $r = 1$	16	20	19	55
	inf. $r = 5$	15	16	14	45
	inf. $r = 10$	10	10	8	28
deviation	total	303.83	83.11	1162.59	558.19
	mean (per f_i, x)	0.21	0.14	0.79	0.40
delay at O	% trips	12.75	7.80	44.65	23.08
	total	21.89	4.57	101.68	46.38
	mean (per f_i)	0.22	0.12	0.88	0.44
	max (per f_i)	2.74	1.57	3.63	2.75
delay at D	% trips	17.40	18.75	43.06	27.14
	total	16.57	3.95	77.49	35.43
	mean (per f_i)	0.17	0.10	0.67	0.33
	max (per f_i)	1.94	0.60	3.21	2.04
speed changes	% trips	20.83	40.44	56.63	39.19
	% waypoints	18.09	17.61	47.47	28.69
	max (per f_i)	21.63	20.07	21.58	21.19
% obj. improv.	mean	29.55	49.08	63.31	46.38
	max	44.66	71.69	88.93	88.93
	become feas.	3	2	5	10

Table 4: Use case of non-cooperative intruder, 180 tests (60 tests per topology). Deviations and delays in minutes.

		Rand.	Prior.	Intru.
num. tests		60	60	180
cpu (s.)	mean	0.27	0.32	0.17
	max	1.65	1.81	0.37
	num. infeasible	1	0	128
deviation	total	171.72	565.71	558.19
	mean (per f_i, x)	0.22	0.53	0.40
delay at O	% trips	10.81	28.26	23.08
	total	12.32	39.17	46.38
	mean (per f_i)	0.21	0.53	0.44
	max (per f_i)	4.66	1.92	2.75
delay at D	% trips	13.77	29.11	27.14
	total	9.74	35.93	35.43
	mean (per f_i)	0.19	0.52	0.34
	max (per f_i)	4.22	1.57	2.04
speed changes	% trips	19.32	32.69	39.19
	% waypoints	17.20	32.13	28.69
	max (per f_i)	15.86	16.62	21.19
% obj. improv.	mean	45.90	24.70	46.38
	max	75.23	99.82	88.93

Table 5: Summary results for the three use cases.

proposed model, which allows changing eVTOLs cruise speeds and delaying flights ready to depart. These adjustments are found enough to tackle the conflicts originated by delays and/or small drifts of on-flight traffic. Indeed, the random variations use case is the one needing from less delays and speed adjustments according to Table 5, which produce smaller deviations with respect to the nominal schedule as a result. When it comes to give way to a priority flight, the vehicle speed increase rates gain special relevance, avoiding the delay of the priority operation when up to a 10% of increase if allowed. Nevertheless, the high delay in the solution of some hard instances is not reduced by increasing vehicles performance. This suggests the need for complementary actions such as re-routing. As for the possible entrance of a non-cooperative intruder crossing the UAM network, our tactical deconfliction method could not solve more than 70% of the instances. Even with 10 minutes to perform adjustments before the intruder intersects the network, over 50% of the instances could be solved. The low running times of a few tenths of seconds suggests the use of the proposed tactical deconfliction model as a first resource, which should be complemented with additional methods if it does not succeed.

6.3.4. Comparison with a local version

Formulation (8) is a centralized and global approach in the sense that (i) the information of the positions and trajectories of the vehicles is known by the conflict resolution service, which sends trajectory amendments to the pilots; (ii) all the conflicts (those explicit and the ones generated after adjustments are made) are resolved at once (multi-conflict) rather than in several sequences (pair-wise). In this section, we simulate the behaviour of a local tactical deconfliction approach that only makes changes to those trajectories that are directly in conflict, and compare its performance with our global approach. The set of benchmarking

instances used is the same as those in previous sections (60 random variations instances; 60 priority traffic instances; and 180 non-cooperative intruder instances).

To have a local version of formulation (8), we consider the disrupted nominal schedule and check the separation conditions for every pair of aircraft passing through a common node. The in-conflict pairs are noted down, together with the aircraft involved. All variables t_{ix} in (8) not corresponding to in-conflict aircraft are fixed to their nominal values $\hat{t}_{ix}$. The resulting model is a centralized local approach that only produces speed adjustments and ground delays for in-conflict flights. In this section, we compare formulation (8) with this *local model*.

Figure 9 illustrates the result of the comparison for random variations. The average total deviation in the solution, depicted by Figure 9a, is significantly increased when only local adjustments are considered. In addition, the dispersion of the objective value (i.e. the deviation from schedule) is much greater for the local model. In contrast, locally solving the conflicts takes less time, as demonstrated by Figure 9b. All the instances are solved in a few tenths of a second with the local model. On the other hand, formulation (8) finds the optimal solution in no more than 0.55 seconds in most of the cases, with a few exceptions. This magnitude of computing time is affordable for a tactical level of deconfliction, where the time to take actions before conflicts occur is typically of seconds or even some minutes. The deconfliction on the Airport topology is the one that needs more time.

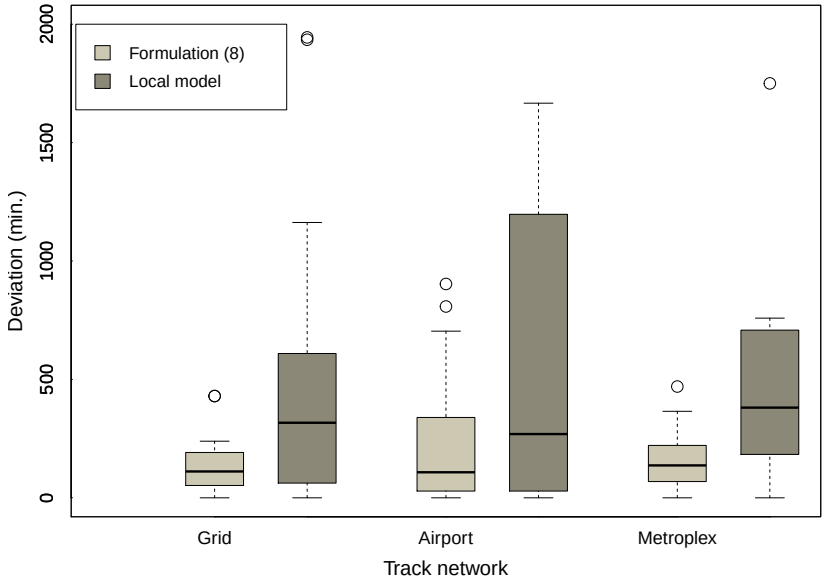
The comparison for the use case of priority traffic is illustrated in Figure 10. At the top, Figure 10a shows the delay in the departure of priority traffic. An extremely different performance of formulation (8) and its local version can be observed. The local approach cannot provide valid solutions for an emergency operation. Indeed, for the Grid and Airport topology local solutions always feature more than 35 min of delay of the priority traffic, while for the Metroplex high delays are also often produced. On the other hand, formulation (8) provides an optimal solution in a very reasonable amount of time (see Figure 10b), which sometimes consists in delaying the priority flight of several minutes. This suggest that, for congested scenarios such as those tested here, additional solutions such as re-routing or emergency landing of in-flight traffic might be needed to give way to priority operations.

Finally, concerning the non-cooperative intruder case, the local approach could only solve 3 of the 180 testing instances. In these cases, which correspond to the Airport topology and time to take actions to give way to the intruder of $r = 10$, $r = 10$, and $r = 5$ minutes, the optimal solution found was the same as that of formulation (8). These results suggest that local approaches might be neglected when it comes to ensure safety against a non-cooperative intruder.

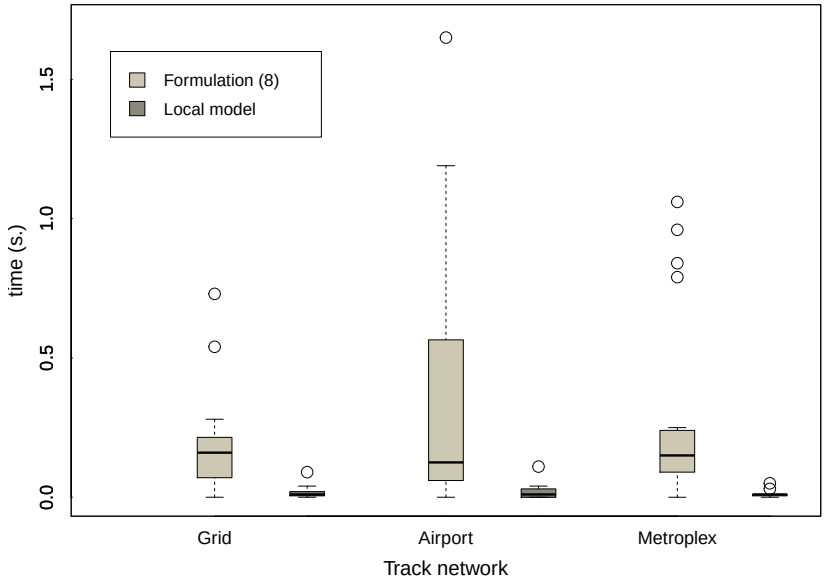
6.3.5. Impact of fairness on the solution

In this section, we explore alternative solutions to the tactical deconfliction problem, with the aim of improving fairness. Fairness is typically understood as an equitable spread of some metric (e.g. delay, or some economic indicator) among the different actors of an optimization problem (here the different trips). This equitable spread can lead to a degradation of other metrics such as the overall schedule delay or number of speed adjustments, which we call the cost of fairness and analyse in this section.

To aim at obtaining fair solutions, we consider the alternative objective of minimizing the sum of the squared differences between the delay at the destination of each trip and the average of these delays. That is, if we define τ_i for $i \in \mathcal{F}^1$, the delay of the non-priority

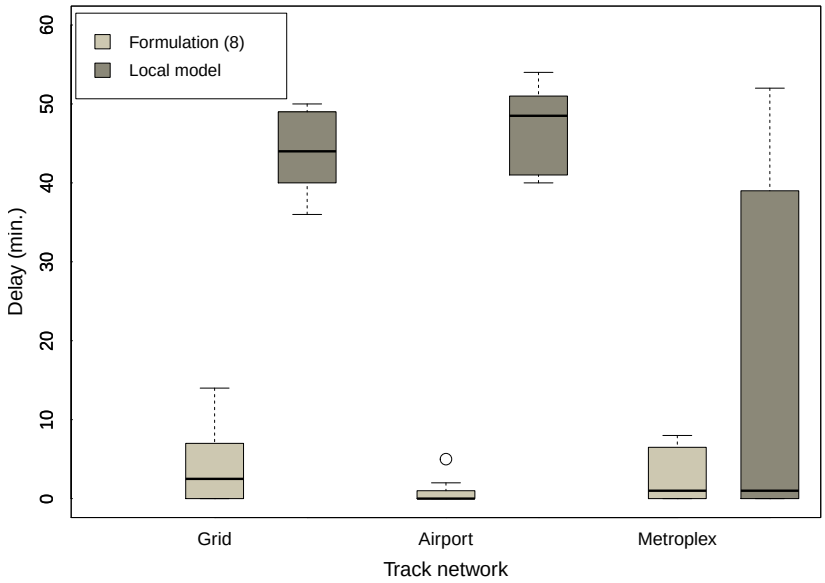


(a) Total deviation of the optimal solution

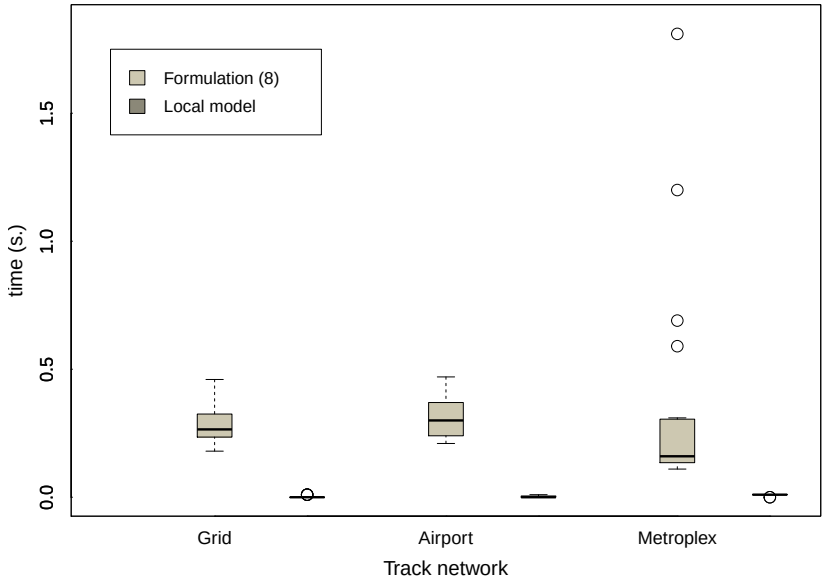


(b) CPU time to obtain the optimal solution

Figure 9: Comparison of performance against the local model for random variations



(a) Total delay of priority traffic in the optimal solution



(b) CPU time to obtain the optimal solution

Figure 10: Comparison of performance against the local model for priority traffic

		Rand.		Prior.		Intru.	
		(8)	fair	(8)	fair	(8)	fair
value of (9)	mean	39.36	23.99	36.72	19.02	38.86	19.99
	max	272.62	255.40	795.50	411.60	210.71	127.70
	% gain	-	35.06	-	39.87	-	45.29
delay at D	mean	9.74	37.62	35.93	65.39	35.43	77.50
	max	49.80	191.76	627.15	710.20	221.63	295.53
	% degrad.	-	478.76	-	430.39	-	397.72
min delay at D	mean	0.00	0.07	0.00	0.02	0.00	0.04
	max	0.00	1.11	0.00	0.42	0.00	0.37
max delay at D	mean	4.22	3.30	1.57	1.58	2.04	2.04
	max	15.52	15.52	8.00	8.00	6.76	6.76

Table 6: Comparison between formulation (8) and the fair model

flight f_i at its destination, the "fair" objective is to minimize:

$$\sum_{f_i \in \mathcal{F}^1} (\tau_i - \bar{\tau})^2 + M \sum_{f_i \in \mathcal{F}^2} \tau_i, \tag{9}$$

where $\bar{\tau} := \frac{1}{|\mathcal{F}^1|} \sum_{f_i \in \mathcal{F}^1} \tau_i$. We will refer to the resulting model where (8a) is replaced by (9) as the "fair model".

Table 6 gives a comparison of formulation (8) and its fair version under the three considered use cases (we consider the same instances as in previous sections). The first rows refer to the fairness of the optimal solution (on average), which is measured using (9). This value is naturally smaller for the fair model, which has (9) as objective function. The improvement of fairness with respect to the original formulation is greater for the priority traffic and non-cooperative intruder use cases when we compare the mean and max values of the first term in (9) (on the two first rows of the table). To have a better idea of the gain on the fairness in the solution when the fair model is considered, we can look at the third row of the table. For each test k, we have calculated the following percentage

$$100 \cdot \frac{F_k^8 - F_k^9}{F_k^8},$$

where F_k^8 and F_k^9 denote, respectively, the value of the first term in (9) in the solution of formulation (8) and in that of the fair model. The average of these percentages, which measures the gain of the fair model with respect to formulation (8), is displayed on the third row. From these data, we can conclude that the non-cooperative intruder use case is the one for which a greater improvement can be obtained on average.

The second group of rows refers to the overall delay at destination of the deconflicted schedules in the solution. In this case, the displayed data gives an idea of the cost of fairness in terms of the overall delay at destination. The percentage of degradation of the delay when the fair model is used is calculated as the average of

$$100 \cdot \frac{|E_k^8 - E_k^9|}{E_k^8},$$

where E_k^8 and E_k^9 denote the value of $\sum_{f_i \in \mathcal{F}^1} \tau_i$ in the solution of formulation (8) and in that of the fair model in the test k. This value gives an idea of the cost of fairness: the greater

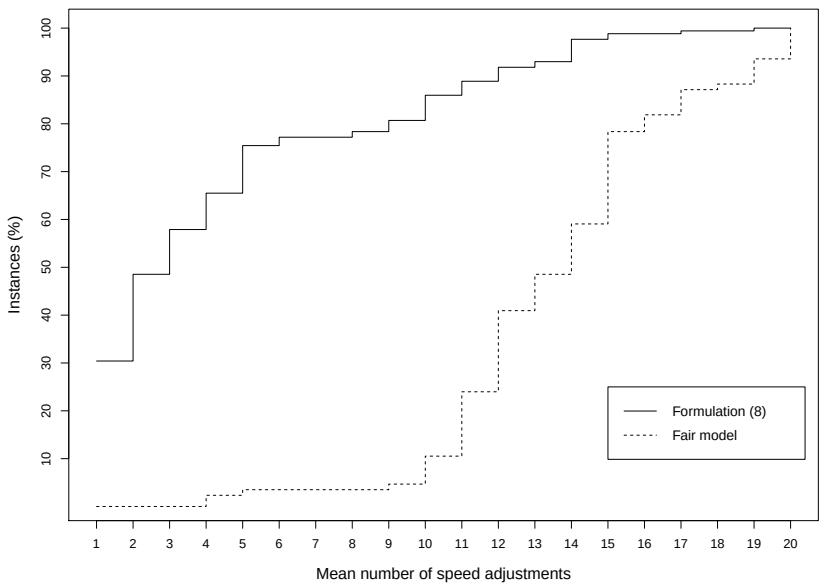


Figure 11: The cost of fairness: Mean number of speed adjustments made per trip

it is the higher cost. When comparing among the different use cases, the random variations one features the greatest cost of fairness. On the other hand, when comparing the gain and cost of fairness, the improvement of solution fairness does not compensate the degradation of the overall delay at destination (if both are considered to be equally important).

Finally, the last two groups of rows in Table 6 refer to the minimum and maximum delay at the destination among the trips of a given schedule. The minimum delay, which is always 0 for formulation (8), increases up to 1 minute and 6 seconds when the fair model is used. We observe that maximum delays are reduced by the fair model in the use case of random variations. However, they are barely modified for the priority traffic use case, and remain unchanged for the non-cooperative intruder one. For these last two use cases, fairer solutions are achieved by increasing the delay of trips in order to minimize the distance to the mean. However, for the random variations, there is usually an actual fairer redistribution of the delays among the trips. Indeed, the number of instances for which the fair model reduces the maximum delays of the solutions found by formulation (8) are 22, 2, and 0 for the random variations, priority traffic, and non-cooperative intruder use cases, respectively. This may also suggest a higher degree in the multiplicity of the existing solutions for this use case. In conclusion, even though the random variations use case is the one obtaining the smaller gain on fairness, the results suggest that it is also the only one allowing for non-trivial fairer solutions.

Figure 11 illustrates the cost of fairness in terms of the number of speed adjustments made. The ordinate axis shows the percentage of the tested instances whose solution feature a mean number of speed changes per trip fewer than or equal to the number depicted on the abscissa axis. Half of the solutions found by formulation (8) required from no more than two speed adjustments per trip (on average). However, in the case of the fair model, half of the solutions needed up to 13 speed adjustments per trip. In general, the vertical distance between the two graphs gives us an idea of the differences between the approaches. This distance, which reaches 70 percent-points, illustrates a significant cost of fairness in terms of manoeuvres. The maximum number of speed adjustments per trip was of 19.3 for the

fair model and 18.2 for formulation (8).

7. Conclusions and perspectives

To the best of our knowledge, this work presents the first Mathematical Programming formulation for UAM Tactical Deconfliction. We propose a set of linear equations for ensuring spatial separation within the UAM setting, which features non-rectilinear trajectories. The new equations depend on vehicle speeds and trajectories crossing angles. The resulting model allows to handle three operational use cases, corresponding to realistic scenarios envisioned for this future service, in foreseen realistic UAM topologies. Our study provides insight into the reach and limitations of tactical deconfliction of UAM traffic under different conditions. It helps to understand and identify some of the key aspects influencing the complexity of TD in this new envisioned environment.

The proposed centralized global model efficiently solved the conflicts originated by random variations of the flights schedules. When unplanned traffic, which was not scheduled during the strategic phase, comes into place, this model might result insufficient to satisfy the timing requirements in the new schedules– when the unplanned traffic has priority—, or to solve the conflicts—when the unplanned traffic is non-cooperative. To sum up, our results reveal a major adversity to solve the conflicts with the decrease of the level of cooperation. On the other hand, using speed increase rates of up to 10% (with respect to the default speed) was proved decisive for some priority traffic instances, and helped to reduce the total deviation from the nominal schedule in general. The particularities in the layouts of the Airport and Metroplex topologies were found to influence the deconfliction, including higher deviations and delays in the optimal solutions. From the comparison with a local version of our model, we conclude that the faster running times do not compensate the poorer quality of the solutions. In addition, for the non-cooperative use case, the local model reveals ineffective. Finally, from our analysis of solution fairness, we conclude that fairness implies a considerable cost in terms of delays and number of speed adjustments. This opens the questions of whether the fairness discussion is suitable at this level of deconfliction.

As for future extensions and considerations, we highlight the need of a well-defined energy consumption model. This would allow to incorporate new features into the models, for instance, a different impact of acceleration and deceleration maneuvers. On the other hand, devising strategies to cluster eVTOL trajectories could help in the deconfliction, the aim being to send the same adjustments for the vehicles in the same cluster. Considering the robustness of the solution against new future disruptions as an optimization goal would be also an interesting direction to explore. New use cases might be considered for future studies, for instance, to include uncertainty in the non-cooperative intruder's trajectory. This might be approached by considering a spherical radius around the intruder.

Acknowledgments

This publication was supported by the Chair "Integrated Urban Mobility", backed by L'X - École Polytechnique and La Fondation de l'École Polytechnique and sponsored by Uber. The Partners of the Chair shall not under any circumstances accept any liability for the content of this publication, for which the author shall be solely liable.

References

[BDH21] M. Bennaceur, R. Delmas, and Y. Hamadi. Passenger-Centric Urban Air Mobility: Fairness Trade-Offs and Operational Efficiency, 2021. arXiv:2103.09839.

[Bil00] K.D. Bilimoria. A geometric optimization approach to aircraft conflict resolution. In *18th Applied aerodynamics conference*, page 4265, 2000. URL: https://doi.org/10.2514/6.2000-4265.

[BL18] C. Bosson and T.A. Lauderdale. Simulation evaluations of an autonomous urban air mobility network management and separation service. In *Aviation Technology, Integration, and Operations Conference*, page 3365, 2018.

[BOS+21] V. Bulusu, E.B. Onat, R. Sengupta, P. Yedavalli, and J. Macfarlane. A traffic demand analysis method for urban air mobility. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–9, 2021.

[CB15] S.P. Cook and D. Brooks. A quantitative metric to enable unmanned aircraft systems to remain well clear. *Air Traffic Control Quarterly*, 23(2-3):137–156, 2015.

[CD14] S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, 58(4):613–629, 2014. URL: https://doi.org/10.1007/s10898-013-0070-1.

[CD18] S. Cafieri and C. D'Ambrosio. Feasibility pump for aircraft deconfliction with speed regulation. *Journal of Global Optimization*, 71(3):501–515, 2018. URL: https://doi.org/10.1007/s10898-017-0560-7.

[CGE+21] C. Chin, K. Gopalakrishnan, M. Egorov, A. Evans, and H. Balakrishnan. Efficiency and fairness in unmanned air traffic flow management. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2021.

[con20] SESAR consortium. Strategic research and innovation agenda. Digital European Sky Technical Report, September 2020. URL: https://www.sesarju.eu/node/3697.

[Fed20] Federal Aviation Administration. Urban Air Mobility Concepts of Operations (v1.0), June 2020.

[Ham20] Y. Hamadi. Optimization for Urban Air Mobility. In *Learning and Intelligent Optimization - 14th International Conference, LION*, volume 12096 of *Lecture Notes in Computer Science*, pages 1–8, 2020. doi:10.1007/978-3-030-53552-0_1.

[HG16] J. Holden and N. Goel. Fast-forwarding to a future of on-demand urban air transportation. Technical report, Uber Elevate, 2016.

[KMW18] I.C. Kleinbekman, M.A. Mitici, and P. Wei. eVTOL arrival sequencing and scheduling for on-demand urban air mobility. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–7, 2018.

[NL20] NASA and Deloitte Consulting LLP. UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4 Version 1.0, December 2020. URL: https://ntrs.nasa.gov/citations/20205011091.

[Ome15] J. Omer. A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. *Computers and Operations Research*, 58:75–86, 2015. URL: https://doi.org/10.1016/j.cor.2014.12.012.

[PD21] M. Pelegrín and C. D'Ambrosio. Aircraft deconfliction via mathematical programming: Review and insights. https://hal.archives-ouvertes.fr/hal-02902566/, 2021.

[PFB02] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE transactions on intelligent transportation systems*, 3(1):3–11, 2002. URL: https://doi.org/10.1109/6979.994791.

[Rio18] J. Rios. UAS Traffic Management (UTM) project strategic deconfliction: System requirements final report, 07 2018. National Aeronautics and Space Administration.

[RPJ+20] A. Rodionova, Y.V. Pant, K. Jang, H. Abbas, and R. Mangharam. Learning-to-fly: Learning-based collision avoidance for scalable urban air mobility. https://arxiv.org/abs/2006.13267, 2020.

[RRFF16] D. Rey, C. Rapine, R. Fondacci, and N.E. El Faouzi. Subliminal speed control in air traffic management: Optimization and simulation. *Transportation Science*, 50(1):240–262, 2016. URL: https://doi.org/10.1287/trsc.2015.0602.

[SRS+20] A. Straubinger, R. Rothfeld, M. Shamiyeh, K.D. Büchter, J. Kaiser, and K.O. Plötner. An overview of current research and developments in urban air mobility—setting the scene for UAM introduction. *Journal of Air Transport Management*, 87:101852, 2020.

[VCBP20] P.D. Vascik, J. Cho, V. Bulusu, and V. Polishchuk. Geometric approach towards airspace assessment for emerging operations. *Journal of Air Transportation*, 28(3):124–133, 2020.

[VMM+20] S.A. Verma, S.C. Monheim, K.A. Moolchandani, P. Pradeep, A.W. Cheng, D.P. Thipphavong, V.L. Dulchinos, H. Arneson, T.A. Lauderdale, C.S. Bosson, E.R. Mueller, and B. Wei. Lessons learned: Using UTM paradigm for Urban Air Mobility operations. *Aviation*, 2020.

[Xue20] M. Xue. Urban air mobility conflict resolution: Centralized or decentralized? In *AIAA AVIATION 2020 FORUM*, page 3192, Virtual, 2020.

[YW20] X. Yang and P. Wei. Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility. *Journal of Guidance, Control, and Dynamics*, 43(8):1473–1486, 2020.