Transportation Research Part C 170 (2025) 104894

Contents lists available at ScienceDirect

Transportation Research Part C

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

A User-Driven Prioritisation Process implementation and optimisation for ATFM hotspot resolution

Andrea Gasparin [a], Lorenzo Castelli [a,*], Tatjana Bolić [b], Gérald Gurtner [b], Nadine Pilon [c]

[a] Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Trieste, Italy
[b] University of Westminster, Centre for ATM Research, School of Architecture and Cities, England, United Kingdom
[c] EUROCONTROL Innovation Hub, Brétigny sur Orge, France

ARTICLE INFO

Keywords:
UDPP
ATFM regulations
Hotspot
Optimisation

ABSTRACT

The current and forecast air traffic levels lead to demand-capacity imbalances, which are dealt with by delaying flights through the allocation of air traffic flow management (ATFM) slots. To mitigate the delay impact on airspace users (AUs) and passengers, *User Driven Prioritisation Process (UDPP)* solutions are under development, with the goal to enhance flexibility for airlines to prioritise their own flights in the ATFM regulations. UDPP solutions are developed in collaboration with AUs, achieving high maturity level and even operational use at some airports.

While UDPP solutions in reality are still based on manual or semi-automated procedures, in this paper we show that when an airline has an accurate delay cost model at disposal, the prioritisation process can be fully automated via an integer programming model that provides the prioritisation that optimises the AUs' UDPP exploitation. We use this automated process and the implementation of the UDPP mechanism to provide an estimation of the benefits of UDPP in terms of cost with respect to the current ATFM regulation process.

1. Introduction

The latest EUROCONTROL's forecast for Europe predicts the pre-pandemic number of flights (11.1 million in 2019) to be reached in 2025 (EUROCONTROL, 2023). After 2025, flight growth is expected to average 1.5% per year in the base forecast. Therefore, alongside the traffic recovery we can foresee in the European air transport system the return of demand-capacity imbalances at certain times and locations, which we also refer to as *hotspots*. These imbalances are caused by the mismatch between supply (capacity) and demand (traffic). Sometimes the demand is higher than the capacity, and sometimes expected and unexpected events, like severe weather, low visibility, runway closures, incidents and strikes can temporarily reduce the capacity of a resource (e.g. sectors, airports).

When a hotspot occurs, the air navigation service providers (ANSPs) in collaboration with the Network Manager (NM) can activate an air traffic flow management (ATFM) regulation. The goal of an ATFM regulation is to limit for an appropriate period of time the traffic over the impacted resource to its regulated (i.e. reduced) capacity, by assigning an ATFM slot (or simply *slot* in the following) to all flights scheduled to cross it.[1] An ATFM slot usually implies delays for flights. These delays result in costs

* Corresponding author.
E-mail addresses: ANDREA.GASPARIN@phd.units.it (A. Gasparin), castelli@units.it (L. Castelli), t.bolic@westminster.ac.uk (T. Bolić), g.gurtner@westminster.ac.uk (G. Gurtner), nadine.pilon@eurocontrol.int (N. Pilon).

[1] From here on we use *(ATFM) regulation* and *hotspot* interchangeably.

https://doi.org/10.1016/j.trc.2024.104894
Received 19 January 2024; Received in revised form 11 June 2024; Accepted 18 October 2024
Available online 6 November 2024
0968-090X/

for the European air traffic system that have been estimated to exceed 1B€ per year (Delgado et al., 2021). For this reason, new operational concepts that can make air traffic more sustainable from an economic and environmental point of view have been under consideration for years (EUROCONTROL, 2022).

The slot allocation is currently performed by the NM, using the Computer Assisted Slot Allocation (CASA) algorithm (EUROCONTROL, 2007). CASA is based on the *First Planned First Served* (FPFS) policy that preserves the original planned time ordering of the flights. This policy is well accepted by the airlines, or airspace users (AUs), as it is considered equitable (all flights are treated in the same way) and minimises the total delay (Castelli et al., 2011). However, the same delay might cause different impacts and costs to AUs' operations, depending on specific characteristics of each flight such as the schedule of the day, aircraft rotation, type of aircraft, crew scheduling, turnaround and number of connecting passengers on-board, just to mention some. This implies that a different policy for the slot allocation could result in a lower delay costs for both airlines and passengers. For this reason, AUs are asking for the design of new frameworks to involve them directly in the allocation decision process, in order to better embrace their business needs (EUROCONTROL, 2006).

In response to this request, within the Single European Sky ATM Research (SESAR) programme a User Driven Prioritisation Process (UDPP) Solution has been under development for several years (SESAR, 2007). The aim of this SESAR Solution is to develop different mechanisms and procedures by which airlines can be part of the decision-making process to mitigate the costs caused to them by a hotspot. UDPP mechanisms have been tested in live trials and have shown to enable significant benefits for airlines in terms of better punctuality, avoidance of cancellations, and reduction of number of passengers missing connections (SESAR, 2019). Setting up and running a live trial is a very time and money consuming activity. For this reason, only 3 one-week simulations with 6 airlines and 3 airports participants have been carried out so far (Pilon et al., 2019). These have covered a limited number of flights associated with a single hotspot, which is not always representative of the operational reality of every airport. In fact, so far it was not possible to run fast-time simulations to estimate the impact of UDPP in cases with a larger number of flights because, to the best of our knowledge, a formulation of UDPP did not exist.

And it is precisely in this context that this paper fits, as it presents a mathematical formulation of the core mechanisms planned in UDPP, which in the following is simply referred to as 'UDPP mechanism'. This formulation allows the use of large-scale fast-time simulations, fed by traffic and cost data, to produce a better estimation of the expected cost savings. In fact, this paper provides airlines with a tool to take full advantage of the UDPP and thus strengthen their decision-making role. In the UDPP, airlines can intervene themselves in the final allocation of slots since these are allocated according to priorities that each airline assigns to its flights. At present, the airlines establish these priorities in a manual way, which relies, on a special 'what-if' functionality built into the UDPP prototype. Here, we define an integer programming (IP) model, named *Optimised Prioritisation* (UDPP-OPT) model, which optimises the airline choices through identification of *the priority values that an airline must assign to all its flights in a hotspot in order to maximise the reduction of the cost of delay compared to the FPFS*. The UDPP-OPT model could have an operational value for AUs, as it provides airlines with an *automated tool* to optimise their allocations under UDPP.

Due to the formulation of the UDPP mechanism and the consequent fully automated priorities setting process, we are able to present an extensive quantitative approach (1000 hotspots for which we perform a detailed numeric analysis) instead of the qualitative approach of the previous works, as we run numerous simulations based on traffic data created from historical records. It is thus possible to estimate the overall saving in terms of cost of the delay that is produced by the UDPP compared to the (current) FPFS policy. In the same way, it is possible to compare the results of the UDPP with those obtained from two theoretical reference mechanisms: one determines the slot allocation of the minimum cost while the other always minimises the allocation cost with, however, the constraint that for no airline this cost is greater than the one obtained with the FPFS. Both of these mechanisms are detailed in the 3. We also analyse savings by the size of the airlines and hotspots involved.

This paper is organised as follows. Literature on ATFM regulation resolution and cost models in particular is reviewed in Section 2. Section 4 presents a high-level overview of the UDPP mechanism defined in SESAR (2019), focusing on its main features, followed by the detailed description of the UDPP algorithm. The UDPP-OPT model is defined in Section 5, the simulation experiments are described and presented in Section 6, and results are discussed in Section 7. Finally, Section 8 presents conclusions and future research considerations.

2. Literature review

When the demand exceeds the capacity of an airport or a portion of airspace, ANSPs and the NM activate ATFM regulations. In an ATFM regulation, a delay is assigned to flights that are scheduled to cross that particular resource, thus reducing the demand for the resource in the time period of the regulation activation. For the regulations linked to portions of airspace, AUs have rather flexible choices (Niarchakou and Sfyroeras, 2022). For example, flights can be re-routed around the hotspot, or a better timing can be requested. However, when the hotspot concerns the airport of arrival or departure, the room for flexibility reduces. In this more constrained context, the UDPP is expected to play a role as it includes new mechanisms to favour the active role of AUs when hotspots arise at airports, enabling them to rearrange flights to suit better their business priorities (Pilon et al., 2016, 2021). The priorities could be linked to various reasons including on-time performance or the cost minimisation.

Since regulations can cause delays, and delays imply costs to AUs, an accurate definition of cost-delay relationships (a.k.a. *cost models*) is essential to analyse and develop any slot allocation mechanisms. Cost models are needed to calculate costs associated with the particular amount of delay and flight characteristics because it is a known problem that the delay-cost relationship is not linear (see e.g. Cook and Tanner, 2015; Hansen et al., 2001). Each flight has a very specific cost structure which depends on different factors, like the position in aircraft rotation (e.g. first or last leg of the day), number of passengers carried, number of connecting passengers, duty of care, aircraft type, airline business profile, crew costs, maintenance and airport curfew time. As such, two flights with the same delay can have very different costs of delay. For example, thirty minute delay that prevents a group of connecting passengers from reaching their next flight will likely have much larger operational costs than a delay on a flight with no connecting passengers. The costs of missing connection can include re-booking on another flight, care, accommodation, and compensation (see Regulation 261, European Commission, 2004).

In other circumstances, the cost of delay depends on the current and future legs operated by an aircraft, as for example a later leg may be involved in another hotspot or a subject to another source of delay. This implies that cost models are affected by uncertainty, which still represents an open problem in the research community (Evler et al., 2022).

The complexity of the cost structure and related models implies that in certain real-time instances some AUs might struggle to apply them and assess in detail the cost of their delayed flights. Gurtner and Bolić (2023) use cost models and simulations to collect the true cost of delay and show that the impact of cost approximations have been severely underestimated. However, even when AUs do not know detailed costs, they are aware of *when* they can expect a dramatic increase in costs,[2] which is the reason for introduction of the *Margin* feature to UDPP. The basic idea is that flights should not be assigned slots after the indicated Margin times. Even though the detailed discussion of the Margins algorithm is out of the scope of this paper (it is still under development), it is important to notice that it was developed in order to help AUs find the most convenient way of flight reallocation within their own slots, without sharing the cost data. This is still a manual process where airlines need to input the Margin time for their flights, without necessarily turning to specific cost values, mainly drawing from their experience. By using UDPP-OPT, the whole process could be automatised, which could make Margins irrelevant.

When the cost of the delay is assumed to be known, the slot allocation problem can be tackled via integer (IP) or mixed integer (MIP) programming models to efficiently determine optimal solutions. In this respect, the initial study of Odoni (1987) paved the way of a prolific research activity for the development of more refined formulations as, for instance, the one of Terrab and Odoni (1993) who proposed to interpret the problem as a minimum cost network flow, or the OPTIFLOW mechanism (Vossen and Ball, 2006a). The latter is of particular interest as it introduces two important features: (a) the interpretation of the slot assignment as an inter-airline slot swap mechanism, and (b) the possibility to identify, for each flight, a *goal slot* that an integer programming minimax algorithm considers as the optimisation target of each flight. Relying on these concepts, it is therefore possible to handle the flight reallocation when a hotspot occurs. In addition, the idea of a goal slot leads to a direct involvement of the AUs in the slot assignment process. This is a key aspect in modern ATFM (EUROCONTROL, 2006) striving towards the enhancement of the collaborative ATM framework, inspired by the collaborative decision-making process implemented at airports. A further improvement of OPTIFLOW has been proposed by Vossen and Ball (2006b), with the formulation of the *at least-at most* (AMAL) mechanism. The underlying mechanism is a first attempt to formalise the inter-airlines trade dynamics, assuming that airlines might be interested in accepting a delay increase on some of their less important flights if they can obtain in return a reduction for their most strategical ones. A more refined version of the AMAL mechanism has been proposed by Sherali et al. (2011), which exploits the graph representation of the slot assignment trading mechanism.

An adaptation of the OPTIFLOW mechanism to the European context was proposed by Castelli et al. (2011). Starting from the FPFS assignment, these authors introduce a slot trading mechanism that eventually minimises the overall cost of the slot assignment (MINCOST mechanism). In other words, MINCOST identifies the maximum total cost reduction achievable when starting from the FPFS assignment. It may happen that for some AUs the overall cost of delaying their flights is higher with MINCOST than with FPFS. In that case, it would not be convenient for these AUs to change their FPFS assignment. Thus, it is necessary to handle the delay assignment without favouring or penalising any particular AU, and find a slot allocation process that distributes delays in an "equitable" manner. This concept is named *equity* (in the literature it is sometimes termed *fairness*) and it represents one of the main problems in the ATFM whenever airlines compete for resources, like slots.

A formulation of equity in mathematical terms has been attempted several times, as for example in Bertsimas et al. (2011) or in Bertsimas and Gupta (2016). Other approaches to address equity have been attempted by Jacquillat and Vaze (2018) proposing a mechanism which distributes delays among airlines based on the number of flights of each airline, or more recently by Hamdan et al. (2022) who introduce a penalty mechanism to manage and limit the overtaking between flights when defining the new schedule sequence. Despite all the attempts, it is still hard to find a unique and widely accepted definition of equity by different stakeholders. However, as shown in Ruiz et al. (2019), it has been recognised by the AUs that a basic requirement for a delay handling mechanism in order to be considered equitable is that it should not cause any negative impact on any airline involved, i.e., the delay cost of an AU should not be greater than the initial FPFS one. We refer to this principle as the *No negative impact principle* (NNIP). In the next Section 3 we provide details on the MINCOST formulation and introduce a new formulation that modifies MINCOST with an additional constraint to include the NNIP. We refer to this mechanism as *No Negative Bound* (NNB), and use it to evaluate the performance of the UDPP mechanism with respect to the NNB mechanism that takes equity into account.

3. A methodological approach for mechanism performance analysis

The performance of a slot allocation mechanism can be misrepresented when assessed solely on the basis of the level of cost reduction achieved. Indeed, the characteristics of the hotspot under consideration and the flight cost model may also play an important role in this assessment.

[2] Examples of such events would be the time after which the first connecting passengers miss their outbound flights, the crew exceeds their duty time, the aircraft turnaround delays the next scheduled flight, or a night curfew is reached.

As an illustration, consider a hotspot that is caused by a 50% capacity reduction and involves 15 flights from three different airlines (A, B, and C). It is assumed that all flight cost models exhibit a quadratic relation with delay. This is expressed as a function of the form $C(d) = c_f \cdot d^2$, where d is the delay in minutes and c_f represents the cost per minute, which is a specific parameter defined for each flight. Finally, suppose that the cost of delay is reduced by 25.8% compared to FPFS using a particular mechanism $\mathbb{M}$ for slot allocation.

The mechanism $\mathbb{M}$ may have achieved a significant cost reduction, but it is difficult to determine the extent of this improvement. It is unclear whether this enhancement is solely due to the quality of the mechanism or a result of a poor initial *FPFS* allocation. To address this issue and improve performance indicators, it is important to determine the maximum achievable reduction for a given problem instance and compare the performance of any mechanism in terms of its distance from this upper bound. The upper bound can be determined using the MINCOST mechanism, which minimises overall costs through an assignment problem (Castelli et al., 2011).

Assuming the true cost model C exists and is known for all flights within the hotspot, to formulate the MINCOST we introduce:

- N, the number of slots within the hotspot
- F, the set of flights involved
- S the set of all slots, $S := \{1, \dots, N\}$
- $ETA(i)$, the function returning the slot corresponding to the ETA of flight i
- d_{ij} the delay of flight i if assigned to slot j
- $C_i(d_{ij})$, the cost model of flight i, with $C_i : \mathbb{R}^+ \to \mathbb{R}^+$

and we define as decision variables:

- $x_{ij} \in \{0, 1\}$, flight originally assigned to slot i is assigned to slot j.

The following constraints hold: all flights have to be assigned (Eq. (1)) and a slot can host at most one flight (Eq. (2)):

$$\sum_{j \in S} x_{ij} = 1 \quad \forall i \in F. \tag{1}$$

$$\sum_{i \in F} x_{ij} \leq 1 \quad \forall j \in S. \tag{2}$$

Objective function. The target is to minimise the overall costs:

$$OBJ := \min \sum_{i \in F, j \in S} x_{ij} \cdot C_i(d_{ij})$$

If we now apply MINCOST to the example we have just illustrated, we can see that there is a large gap in terms of total cost reduction between the performance of MINCOST and that of $\mathbb{M}$ (see Fig. 1). MINCOST represents the upper bound for cost reduction by design, thus serving as a reference point for evaluating mechanism $\mathbb{M}$. By looking at the numbers, one could conclude that the performance gap indicates poor quality of $\mathbb{M}$. However, it is possible that $\mathbb{M}$ was designed to promote a more equitable redistribution of the reduction among the AUs involved, which could intrinsically limit its performance in terms of total cost reduction. In such a case, MINCOST may be an inappropriate benchmark. MINCOST's objective function prioritises flights with higher costs, regardless of the airline. In Fig. 1 airlines A and B share the benefits, while airline C experiences a substantial negative impact. This solution may not be considered equitable by airline C. In fact, Airline C's flights have lower costs than those of the other airlines, which explains why they were assigned to the latest positions in the schedule.

The Non-Negative Impact Principle (NNIP) is introduced as a means of enforcing equity constraints on the MINCOST allocation.

Assumption 1. No airline would accept to be penalised by a schedule change in relation to the FPFS.

To include this assumption in the MINCOST, we can just add the following constraint (3): for each airline a, the resulting total delay cost caused by the new slot allocation must be less than or equal to that produced by $FPFS$:

$$\sum_{i \in F_a,\, j \in S} x_{ij} \cdot C_i(d_{ij}) \leq \sum_{i \in F_a} C_i(d_{ii}) \tag{3}$$

where F_a defines the set of flights of airline a. We will refer to the MINCOST mechanism with the addition of constraint (3) as the *No negative bound* mechanism (NNB). Fig. 2 displays the solution achieved by implementing NNB.

The total cost reduction for NNB is not as significant as for MINCOST. However, as anticipated, airline C is not negatively affected this time. Nevertheless, C does not receive any benefit either. This is because, despite the additional constraint, NNB still prioritises high-cost flights. However, the NNB now provides a new upper bound for the total cost reduction that can be used for comparison. For instance, if mechanism M implements equity in a manner that aligns with the NNIP, it can be concluded that its performance is highly satisfactory, as its cost reduction is nearly identical to that of the NNB.

As previously stated, the performance of any mechanism depends on the instance and cost models. In Fig. 3, all flights have the same cost per minute as in the previous examples. However, the initial allocation is different, resulting in a significant decrease in the maximum achievable reduction. Therefore, any mechanism applied to this example cannot provide a reduction greater than 19.8%, which is the upper bound computed with the MINCOST.

FPFS

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
C1	1	1
B2	6	2
A3	5	3
B4	10	4
A5	7	5
B6	4	6
C7	2	7
B8	10	8
C9	3	9
A10	21	10
A11	9	11
B12	11	12
C13	2	13
A14	15	14

MINCOST

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B2	6	0
B4	10	1
A3	5	2
A5	7	3
B8	10	2
A10	21	2
A11	9	3
B12	11	4
A14	15	4
B6	4	14
C9	3	13
C7	2	17
C13	2	13
C1	1	27

$\mathbb{M}$

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
C1	1	1
B2	6	2
B4	10	2
A5	7	3
C9	3	1
A10	21	2
B8	10	6
B12	11	4
C7	2	11
A14	15	16
A11	9	11
A3	5	21
C13	2	13
B6	4	22

AIRLINE	FPFS	MINCOST	$\mathbb{M}$
A	6349	513	3981
B	2552	1000	2536
C	680	2152	584
TOTAL	9581	3665	7101

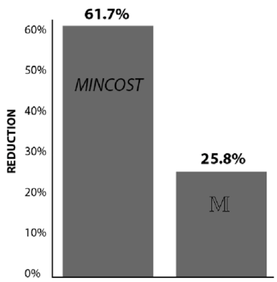
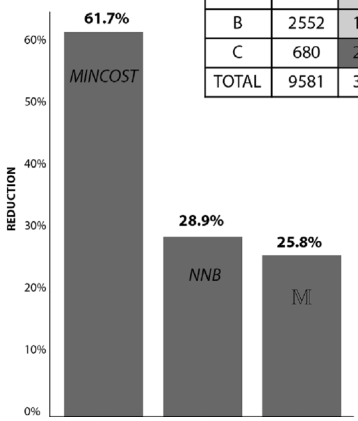


Fig. 1. Comparison between the MINCOST and mechanism $\mathbb{M}$.

NNB

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B2	6	0
B4	10	0
C1	1	5
A5	7	3
B8	10	2
A10	21	2
C9	3	5
B12	11	4
C7	2	11
A14	15	6
A11	9	11
A3	5	21
C13	2	13
B6	4	22



AIRLINE	FPFS	MINCOST	NNB	$\mathbb{M}$
A	6349	513	3981	3981
B	2552	1000	2152	2536
C	680	2152	680	584
TOTAL	9581	3665	6813	7101

Fig. 2. Solution obtained by the *NNB*.

Finally, returning to the example in Fig. 1 and assuming identical initial allocations and cost per minute for each flight, but utilising linear cost models, i.e., $C(d) = c_f \cdot d$, the MINCOST and NNB bounds differ significantly, as illustrated in Fig. 4.

In conclusion, the MINCOST and NNB bounds can improve the evaluation of new mechanisms designed to reduce the impact of delay. These bounds provide two comparative terms that help to understand the performance of the model, regardless of the original schedule or cost model assumptions. However, MINCOST and NNB are not suitable for direct implementation in the operational context as they are two fully centralised (i.e., central optimisation that could be performed by NM, for instance) cost minimisation mechanisms and do not actively involve AUs in the final allocation. In addition, their working assumption is that the central optimisation has access to delay costs of all flights, which AUs are generally reluctant to reveal as they are considered confidential business information. As we shall see in the next Sections 4 and 5, UDPP overcomes both difficulties. First, building on the FPFS solution, UDPP allows AUs to request the reallocation of their own flights, without impacting other AUs' flights, which is an overarching equity principle requested by AUs. Second, cost confidentiality is always preserved. We start in the next Section 4 by describing the UDPP mechanism currently under validation, thus enabling in Section 5 the definition of an IP model designed to find the optimal strategy for any AU to exploit the UDPP mechanism.

4. Overview of the current UDPP mechanism under validation in SESAR

The UDPP SESAR Solution is a mechanism designed to facilitate the active role of the AUs in the process of slot allocation in case of hotspots at the arrival airport. This is achieved by enhancing the AUs' flexibility while reducing their cost of delay, thereby

FPFS

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B1	6	1
A2	5	2
A3	7	3
B2	10	4
A4	21	5
B3	4	6
C1	10	7
B4	11	8
B5	15	9
A5	9	10
A6	3	11
C2	2	12
C3	2	13
C4	1	14

MINCOST

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B1	6	1
A3	7	1
A4	21	1
B2	10	4
C1	10	3
B5	15	3
B4	11	6
A2	5	14
A5	9	8
B3	8	14
A6	3	11
C2	2	12
C3	2	13
C4	1	14

NNB

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B1	6	1
A3	7	1
A4	21	1
B2	10	4
A2	5	8
B5	15	3
C1	10	7
B4	11	8
A5	9	8
B3	4	14
A6	3	11
C2	2	12
C3	2	13
C4	1	14

AIRLINE	FPFS	MINCOST	NNB
A	1871	1947	1287
B	2229	1481	1789
C	1312	912	1312
TOTAL	5412	4340	4388

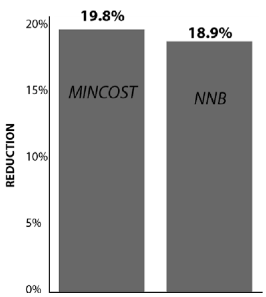


Fig. 3. MINCOST and the NNB for a schedule different from Fig. 1.

FPFS

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
C1	1	1
B2	6	2
A3	5	3
B4	10	4
A5	7	5
B6	4	6
C7	2	7
B8	10	8
C9	3	9
A10	21	10
A11	9	11
B12	11	12
C13	2	13
A14	15	14

MINCOST

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B2	6	0
B4	10	0
A5	7	1
B8	10	0
A10	21	0
B12	11	0
A14	15	0
A11	9	5
A3	5	15
B6	4	14
C9	3	13
C13	2	11
C7	2	19
C1	1	27

NNB

FLIGHT	COST PER MINUTE	DELAY
A0	8	0
B2	6	0
B4	10	0
A5	7	1
B8	10	0
A10	21	0
C9	3	3
A14	15	0
B12	11	4
C7	2	11
C13	2	7
A11	9	11
C1	1	23
A3	5	23
B6	4	22

AIRLINE	FPFS	MINCOST	NNB
A	569	127	221
B	288	56	132
C	68	126	68
TOTAL	925	309	421

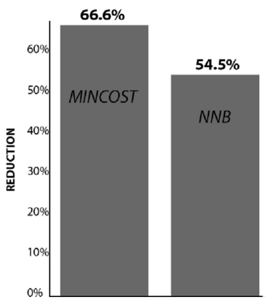


Fig. 4. MINCOST and the NNB for the same initial schedule of Fig. 1 in which linear delay cost model has been adopted.

allowing for the tactical rescheduling of the flights involved in the hotspot. The UDPP mechanism requires that an airline sets a single priority value for each of its flights. The NM then computes a new slot allocation based on these values and two algorithms (UDPPlocal and UDPPmerge). Participation in the UDPP is on a voluntary basis; no airline is obliged to take part. Furthermore, airlines not participating in the UDPP will not experience delays to their flights that are longer than those resulting from the FPFS allocation. Only in very unfortunate circumstances may this not be the case (see Section 4.3.2). These are the reasons why UDPP achieved consensus among AUs, and the development is at a high level of maturity after several human-in-the-loop validation exercises (SESAR, 2019).

4.1. Setting the priorities

The initial step of the UDPP mechanism requires each participating airline to assign a priority value to each of its flights. There are two different ways in which this assignment can be made: *Flight Delay Reordering* and *Selective Flight Protection*. In both cases, the priority values are such that a flight cannot be allocated a slot earlier than its estimated time of arrival (ETA).

4.1.1. Flight delay reordering (FDR)

An airline can freely reorder its flights within its slots as long as for each flight the resulting new time allocation is not earlier than its ETA. This operation does not affect the slots or flights of other AUs. The order of flights is changed by giving each flight a priority number, with the most important flight having priority 1. A flight with a priority number is called an *Nflight*.

4.1.2. Selective flight protection (SFP)

FDR may not always be flexible enough to reduce delays for some important flights. This happens, for example, when an airline owns slots far apart in the hotspot, where the ETA of the flights may not allow a slot swap. For this reason, an airline can use the SFP to *protect* a flight, i.e., it can request for it a slot that it does not own at an earlier time (also referred to as *time not after* - tnA). This flight is labelled P and named *Pflight* (protected flight). In return, the airline must give up an earlier slot it already owns to get this protection. If a flight of airline A is assigned to a slot that is not owned by A, to avoid delaying other flights, the closest slot of A earlier than the new slot must be released to allow all flights between the two slots to be up-shifted. Consequently, if the airline A wants to protect a flight, it must own at least one slot before the flight's *tnA*. If it wants to protect more flights, each flight must be associated with a unique slot before its *tnA*, which has not been previously matched with any other A's Pflight. Once a slot is released by A, it can be assigned to another flight in the hotspot.

The SFP is comparable to the *compression* mechanism (Vossen and Ball, 2006a) and is beneficial for airlines that lack the flexibility to utilise the FDR. This is the case, for instance, when an important flight is scheduled in one of the latest slots in the hotspot and the closest earlier flight of the same AU is too early in the schedule to allow for a swap. Naturally, the ETA must always be respected.

In the end, a unique priority value is defined for each flight: a number if FDR applies (NFlight) or the label P and the requested tnA if SFP applies (Pflight). The current approach to setting these priorities is largely manual, with airlines relying on their own expertise to determine the most optimal course of action. In this paper, we demonstrate that this process can be automated through the use of an optimisation model (UDPP-OPT) that we propose in Section 5.

4.2. The UDPP mechanism

Once the priorities of all AUs have been received, the UDPP mechanism comprises two algorithms (UDPPlocal and UDPPmerge) executed by the NM. The UDPPlocal algorithm is run for each AU separately, resulting in a preliminary, local, slot allocation for each AU (Section 4.2.1). The UDPPmerge algorithm then combines the disparate local allocations into the final slot allocation for all participating AUs (Section 4.2.2).

4.2.1. UDPPlocal

This algorithm is run independently for each AU. For example, UDPPlocal takes the list of flights from airline A, the priorities A assigns to its own flights, and the slots allocated to A ($slots_A$) by the FPFS. The output is called *localSolution*, which is the new slot requests for A's flights. If both SFP and FDR are applied, flights are partitioned into two lists, $Pflights$ and $Nfligths$, respectively. The algorithm initially applies the protections (*ManagePflights*) and then sorts the Nflights according to their priority number (*ManageNflights*). If an airline does not take part in UDPP, it gets as *localSolution* the FPFS allocation.

ManagePflights. This sub-algorithm implements SFP. It:

- sorts the Pflights according to their tnA,
- for each Pflight, sequentially:
  - defines the Pflight's *localSolution* as the slot corresponding to its tnA,
  - identifies the slot s which is the closest one in $slots_A$ earlier than the current Pflight's tnA,
  - removes s from $slots_A$.

After this step, the $slots_A$ list is changed because the removed slots are no longer available for A when exploiting the FDR. This is important because if no flights can request the removed slots, there is a "gap" in the schedule. During the *Merge* phase (Section 4.2.2), flights from other AUs are moved up to fill this gap.

ManagePflights. This sub-algorithm implements FDR, using priority numbers to compute the *localSolution* for A's Nflights. It:

- sorts the Nflights by priority number,
- iteratively processes the Nflights list, and for each Nflight f_n:
  - defines as the *localSolution* of f_n the earliest slot s in $slots_A$ compatible with the f_n's ETA,
  - removes assigned s from $slots_A$.

The same procedure is done for all the airlines in the UDPP. The result is called *localSolution*. This is a set of new slot requests. These consist of re-ordering flights within assigned slots and Pflight requests for new slots in exchange for releasing the same number of own slots. The *localSolution* for airlines not in the UDPP is defined by the FPFS allocation. The UDPPlocal could be managed by the AUs without the NM. However, if airlines were permitted to inform the NM of the slots they desire, there is a possibility that they might misrepresent their *localSolutions* to obtain a more favourable final allocation and gain an advantage. To guarantee that all *localSolutions* comply with the UDPP rules, the UDPPlocal is therefore calculated at the centralised level.

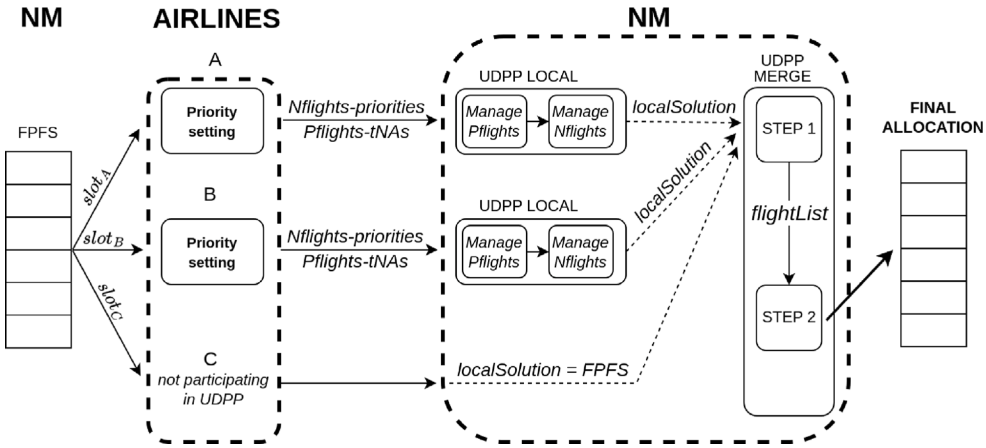


Fig. 5. Schematic representation of the UDPP mechanism flow.

4.2.2. UDPPmerge

The UDPPmerge algorithm tries to respect the slot requests in the AUs' *localSolutions* when producing the final slot allocation. It consists of the following steps:

Step 1. A $flightList$ is defined by sorting all flights according to the time of their *localSolution* slots; in case of multiple flights with the same time request (called *conflict*) the order is defined by the ETAs of the flights.

Step 2. Following the $flightList$ order, each flight is assigned to the first free and compatible slot in the list of all slots within the hotspot. This is the final allocation of slots for all airlines.

Fig. 5 depicts the various steps of the UDPP mechanism.

4.3. Differences between the localSolution and the final allocation

If all *localSolutions* are obtained by FDR operations, the UDPPmerge algorithm merely combines them, as all flights are just rearranged within their airlines' respective slots. When the SFP is applied, the final (merged) solution of some airlines may differ from their *localSolution*. In such instances, some flights may be allocated to an earlier (i.e. positive impact, Section 4.3.1) or later slot with respect to their local assignment. The UDPPmerge algorithm is designed to minimise the likelihood of this latter scenario occurring, although in some rare instances it may still happen (i.e. negative impact, Section 4.3.2).

4.3.1. UDPPmerge positive impact

If a flight of an AU is allocated by the UDPPmerge algorithm to a slot earlier than the one expected by its *localSolution*, that AU is said to be affected by a *positive impact*. Fig. 6 shows an example. Flights $A1$, $A2$ and $A3$ belong to airline A, while flights $F1$, $F2$, $F3$, $F4$, $F5$, $F6$ and $F7$ belong to other airlines that do not participate in the UDPP (hence their *localSolutions* are identical to FPFS (Fig. 6(a), left column)). A decides to protect only $A2$ and handle the rest as *NFlights* through the FDR, with priority numbers $A3:1$ and $A1:2$ (the lower the number, the higher the priority).

The *tnA* of $A2$ is 10:08, which is the time of the earliest slot compatible with ETA of $A2$ (10:07). In return, according to the protection rules (Section 4.1.2), A must release the slot currently occupied by $A1$, since it is the closest A's slot earlier than the requested $A2$ slot. The reordering of the non *Pflights* of A is performed by the UDPPlocal within the remaining slots owned by A (10:14, 10:18). The UDPPmerge will initially order all flights by their *localSolution* time. $A2$ and $F4$ generate a conflict as their *localSolution* slot (10:08) is identical (Fig. 6(b), left column). The conflict is solved by sorting the two flights by their ETAs (Fig. 6(b), middle column). The *flightList* is completed by allocating any remaining flights to the closest available slot (Fig. 6(b), right column).

To summarise, some flights are allocated to slots earlier than expected from the *localSolution* because the $A1$ slot is released to protect $A2$. This "hole" is then filled by the UDPPmerge, which shifts $F2, F3, F4$ to earlier slots.

4.3.2. UDPPmerge negative impact

If a flight is allocated to a slot later than expected by its *localSolution*, this is a *negative impact*. Consider an initial FPFS allocation (Fig. 7), where $A1$ and $A5$ are flights of airline A, $B2$ and $B7$ are flights of airline B, and $F3, F4, F6, F8$ belong to other airlines not participating in the UDPP (their *localSolutions* are identical to the FPFS allocation). Airlines A and B want to protect their flights $A5$ and $B7$ and request the allocation to their respective ETAs (grey arrows in Fig. 7). Consequently, airline A must release the slot occupied by $A1$. Since A has only two flights, in A's *localSolution* $A1$ is moved to the slot previously occupied by $A5$ (upper black arrow in Fig. 7). Similarly, in B's *localSolution* $B7$ is allocated to its ETA slot and $B2$ to the FPFS slot of $B7$.

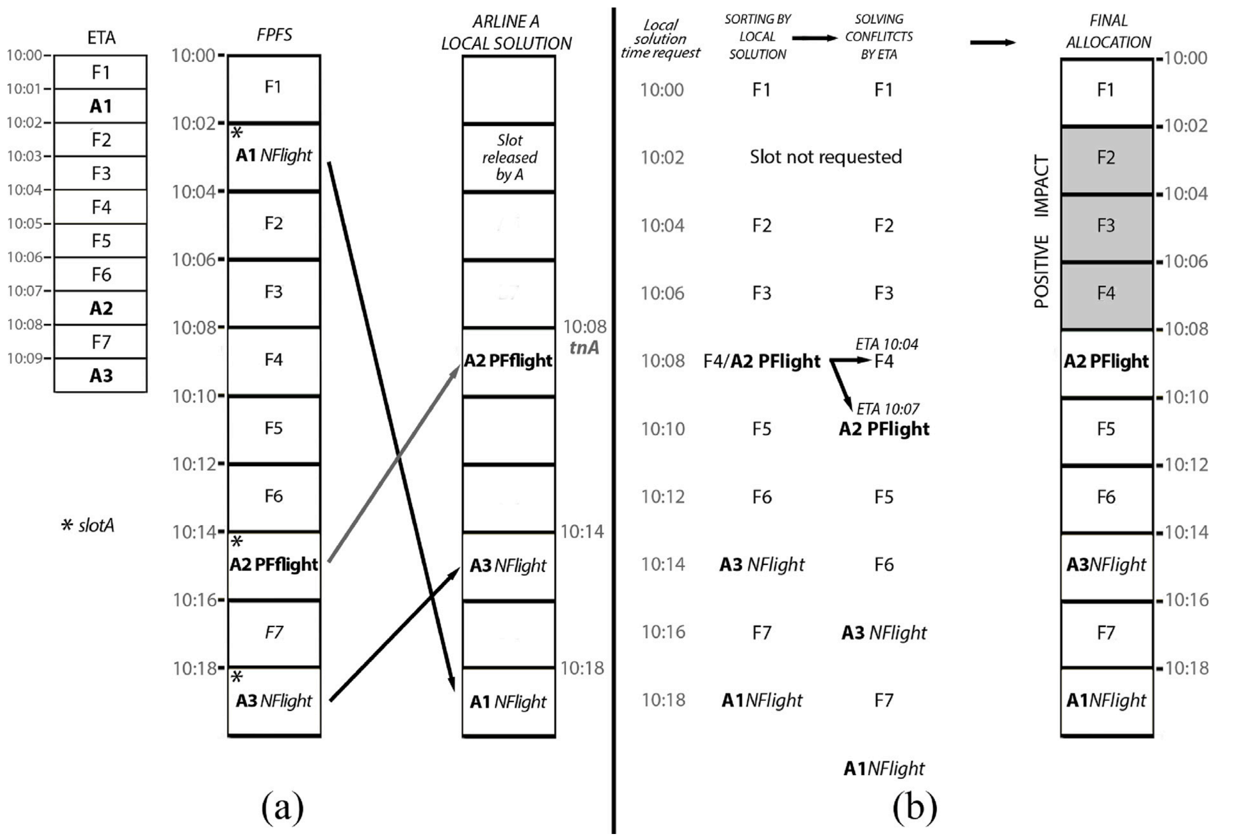


Fig. 6. Positive impact with UDPPmerge: (a) FPFS allocation and airline A *localSolution*. (b) UDPPmerge steps.

The UDPPmerge generates the $flightList$ shown in Fig. 7(a). Initially flights are ordered by their *localSolution* time; $F3$ and $A5$ are in conflict as well as $F4$ and $B7$. Both conflicts are solved considering the flights' ETA, so $F3$ will take the first place in the list, $A5$ the second, $F4$ the third, and $B7$ the fourth. When the allocation process begins (Fig. 7(b)), the algorithm is unable to assign the 10:00 slot to $F3$ as it is earlier than its ETA. Instead, it is allocated to the next available slot, at 10:02. $A5$ is unable to take the 10:00 and 10:02 slots, as its ETA is 10:04, so $A5$ has to be allocated to the third slot (10:04). The same issue arises when allocating the following flights (Fig. 7(c)). Hence, the first slot is not allocated. As a result, flight $F8$ is allocated to time 10:16, resulting in further delays with respect to FPFS (Fig. 7(d)).

This example is rare because it occurs when the protection of some flights causes the UDPPmerge to create a "hole" in the slot list that cannot be filled due to flight ETA restrictions. However, in the live trials carried out so far UDPP (Pilon et al., 2019) allows a flight to arrive earlier than its ETA (typically 5 min) to guarantee flexibility and reduce the possibility of negative impacts. In practice, a slot is compatible with a flight if:

$$slot\ time \leq flight's\ ETA - HFES, \tag{4}$$

where *HFES* (Hotspot Flight Earlier Schedule) represents the tolerance.

4.4. The need for optimisation

An AU's willingness to participate in the UDPP would likely increase with an efficient strategy for determining convenient priorities. Accurately identifying flights to protect (Pflights) and assigning priority numbers to Nflights is in the AU's best interest to maximise savings. As shown in Sections 4.3.1 and 4.3.2, the final slot allocation may differ from individual *localSolutions*, and an AU cannot predict other AUs' priorities. Thus, finding the best priorities for the final allocation is impossible. However, an AU can adjust its priorities to improve its *localSolution*. So far, priority assignments in UDPP validations have been based on the dispatchers' experience and educated guesses. However, this manual approach becomes problematic when there are dozens of flights in a hotspot to prioritise, which can happen in practice (see Section 6). It is therefore necessary to implement a decision support tool that identifies optimal priorities.

This paper shows that it is possible to optimise AUs' choices by using an integer linear programming formulation of the flights' prioritisation process, named Optimised Prioritisation Model (UDPP-OPT). The model is presented in the next Section 5. It identifies

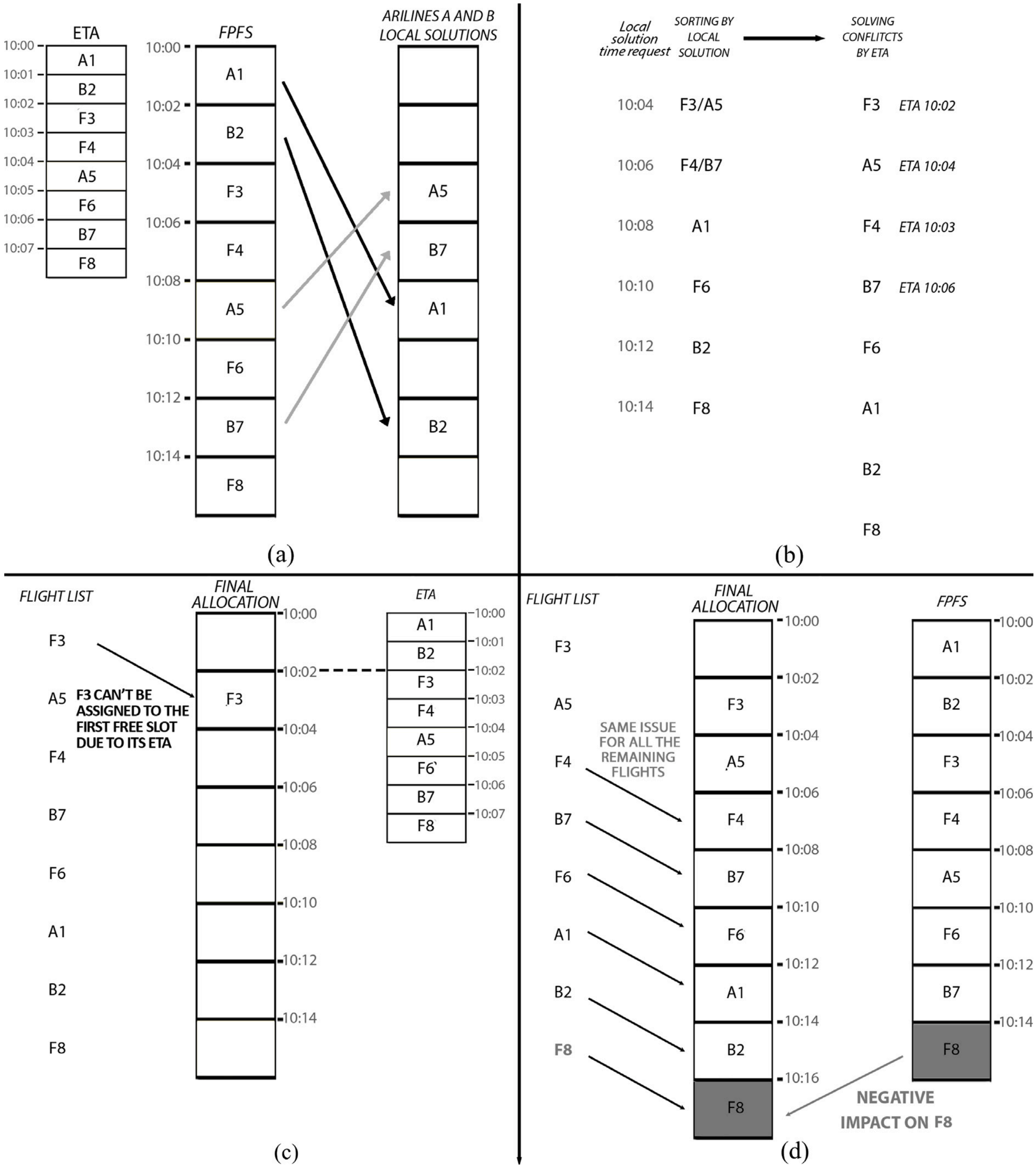


Fig. 7. Negative impact caused by UDPP merge: (a) FPFS allocation and airlines A and B *localSolutions*. (b) UDPPmerge first step: sorting flights and conflict resolution. (c) Allocation of the first flight in the *flightList*. (d) Final schedule.

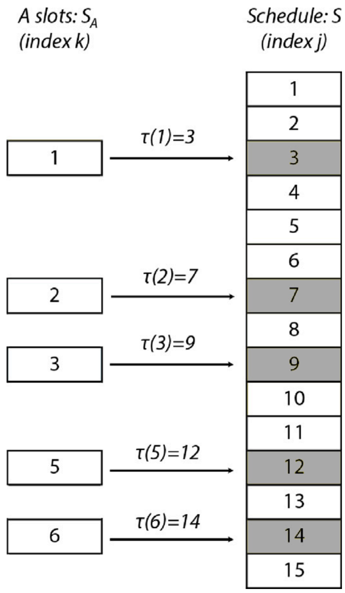


Fig. 8. Correspondence between S_A and S indexes.

the optimal *localSolution* for each AU, which flights to protect and the order of the remaining flights to minimise the AU cost of delay under the assumption that it has at its disposal accurate cost models for all its flights.

5. Optimised prioritisation model for the $localSolution$

This section introduces the UDPP-OPT model, along with additional notation with respect to that introduced in Section 3. It also describes the decision variables, objective function, constraints, and the retrieval of priorities for AUs from the optimised *localSolution*.

5.1. Sets

The following sets are identified:

- A, the AU under study.
- F_A, the set of the flights belonging to A, $F_A := \{1, \dots, K\}$.
- S, the set of all slots within the hotspot, $S := \{1, \dots, N\}$.
- S_A, the set of slots assigned to A. $S_A := \{1, \dots, K\}$. Since each flight $i \in F_A$ needs a slot and is initially allocated to the ith slot of A, index i either indicates the ith flight or the ith slot in S_A. Due to this index correspondence, when convenient, the two sets will be used as equivalent.

As a standard practice, all sets' indexes are sorted in accordance with the FPFS time, with index one representing the earliest flight or slot and indexes K and N representing the last flight or slot, respectively.

5.2. Functions

The following functions are used:

- $\xi(i)$, with $\xi : F_A \to S$. This function returns the index in S of the slot corresponding to flight i's ETA.
- $\xi_A(i)$, with $\xi_A : F_A \to S_A$. This function returns the index in S_A of the A's slot closest to the flight i's ETA (before ETA). This function will be used to determine the A's slot to release in case of protection of flight i.
- $\tau(i)$, with $\tau : S_A \to S$, mapping S_A's indexes with their corresponding indexes in S (see Fig. 8).

5.3. Decision variables

Two sets of variables are used.

$$x_{ik} = \begin{cases} 1 & \text{if flight originally allocated to slot } \tau(i) \text{ is allocated to slot } \tau(k) \\ 0 & \text{otherwise.} \end{cases} \qquad \forall i \in F_A, k \in S_A$$

Variables x_{ik} describe the FDR allocation. We remark that the meaning of the variables is the same as for the model of Section 3, the only differences are the sets of the indices i and k. If for a flight i, and some $k \in S_A$, the equation $x_{ik} = 1$ holds, i is an Nflight.

$$y_{ij} = \begin{cases} 1 & \text{if flight originally allocated to slot } \tau(i) \text{ is allocated to slot } j \\ 0 & \text{otherwise.} \end{cases} \qquad \forall i \in F_A, j \in S$$

Variables y_{ij} describe the SFP allocation. If for a flight i, and some $j \in S$, it is $y_{ij} = 1$, i is a Pflight, and slot j's time represents the Pflight's tnA.

5.4. Objective function

The optimisation objective is to minimise A's costs.

$$OBJ := \min \left[\sum_{i \in F_A, k \in S_A} C_i(d_{ik}) x_{ik} + \sum_{i \in F_A, j \in S} C_i(d_{ij}) y_{ij} \right] \tag{5}$$

5.5. Constraints

The following constraints are identified.

Constraint 1. *Protecting a flight (i.e., applying SFP feature) means allocating it to an earlier slot not owned by A. Otherwise, the flight would be subject to an FDR operation:*

$$y_{ij} = 0 \quad \forall i \in F_A \ \forall j \in S \mid j > i$$
$$y_{ij} = 0 \quad \forall i \in F_A \ \forall j \in \mathcal{IS}(S_A)$$

Constraint 2. *The earliest flight, i.e., the flight with index 1, cannot be protected as no earlier slot can be released in exchange. Therefore it must be allocated to a slot owned by A:*

$$\sum_{k \in S_A} x_{1k} = 1 \quad \textit{and} \quad y_{1j} = 0 \quad \forall j \in S \setminus S_A.$$

Constraint 3. *All other flights have to be either protected or allocated to a slot owned by A:*

$$\sum_{k \in S_A} x_{ik} + \sum_{j \in S \setminus S_A} y_{ij} = 1 \quad \forall i \in F_A \setminus \{1\}$$

Constraint 4. *All slots can be allocated to at most one flight:*

$$\sum_{i \in F_A} x_{ik} \le 1 \quad \forall k \in S_A$$
$$\sum_{i \in F_A} y_{ij} \le 1 \quad \forall j \in S \setminus S_A$$

Constraint 5. *None of the flights can be allocated to a slot earlier than their respective ETA:*

$$x_{ik} = 0 \quad \forall i \in F_A, \forall k < \xi_A(i), \ k \in S_A$$
$$y_{ik} = 0 \quad \forall i \in F_A, \forall j < \xi(i), \ j \in S$$

Constraint 6. *If a protection is applied, the closest earlier slot to the Pflight's tnA has to be released.*

This means that if $y_{ij} = 1$ for some $j \in S$, and if $k \in S_A$ is the greatest index such that $\tau(k) < j$ then we have $x_{ik} = 0 \ \forall i \in F$ (i.e., flights cannot be allocated to the kth slot owned by A, which is the $\tau(k)$-th in the schedule). To achieve this we have to introduce a new set of decision variables:

- $z_k \in \mathbb{N}$, the number of Pflights allocated between $\tau(k)$ and $\tau(k+1)$, which are formally defined as (see Fig. 9),

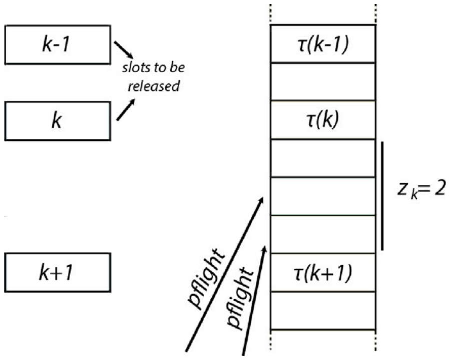


Fig. 9. Meaning of z_k variables.

$$z_k = \sum_{i \in F,\ \tau(k)<j<\tau(k+1)} y_{ij} \quad \forall k \in S_A - \{K\}$$

Constraint 7. *To release the slots of A consistently with the protection feature,*

we introduce the following decision variables:

- $r_k \in \{0, 1\}$, indicating whether the slot k has been released, which are formally defined as

$$r_k = (1 - \sum_{i \in F} x_{ik}) \quad \forall k \in S_A - \{K\}$$

Constraint 8. *Slots have to be adequately freed when protections are applied.*

More precisely, the following inequality ensures that if a certain number of flights z_k are prioritised between slots $\tau(k)$ and $\tau(k+1)$, then z_k slots will be released before (and including) $\tau(k)$:

$$r_k \cdot \mathcal{M} \geq z_k + \sum_{h>k, h \in S_A} (z_h - r_h) \quad \forall k \in S_A - \{K\}$$

where $\mathcal{M} \gg 1$.

In this expression, $z_h - r_h$ represents the 'surplus' of slots prioritised between $\tau(h)$ and $\tau(h+1)$, i.e. the number of slots prioritised minus the number of slots released. Indeed, if more than one flight is prioritised in this interval, not only $\tau(h)$ has to be released, but also another, earlier slot. In the next interval, between $\tau(h-1)$ and $\tau(h)$, the previous surplus has to be added to the surplus of the current interval. Surplus thus need to be carried from one interval to the other, and $\sum_{h>k, h \in S_A} (z_h - r_h)$ is the surplus carried so far from the last slot, until slot k. Hence, if this carried surplus is non-null *or* if z_k is non-null, then the right side is non-null and r_k has to be equal to 1, i.e. slot k has to be released. This has to be the case for every slot belonging to the airline, i.e. $S_A - \{K\}$.

5.6. Retrieving the priorities from the optimal localSolution

Once the optimal *localSolution* is obtained, the corresponding priorities can be retrieved from the indexes of the x_{ik} and the y_{ij} non-zero variables. The *priority number* of Nflights is determined by the non zero x_{ik} variables, in the following manner:

- define X as the list of non zero x_{ik} variables sorted by index k
- sequentially, for each variable x_{ik} in X, assign to the ith flight the *priority number* corresponding to the position of the variable in the list

If for some i, j the variable $y_{ij} = 1$, it means that the ith flight is a Pflight and its desired tnA is represented by the time of the jth slot in the schedule (note that due to Constraint 1, the jth slot belongs to a different AU). An example of prioritisation retrieval is provided in Fig. 10.

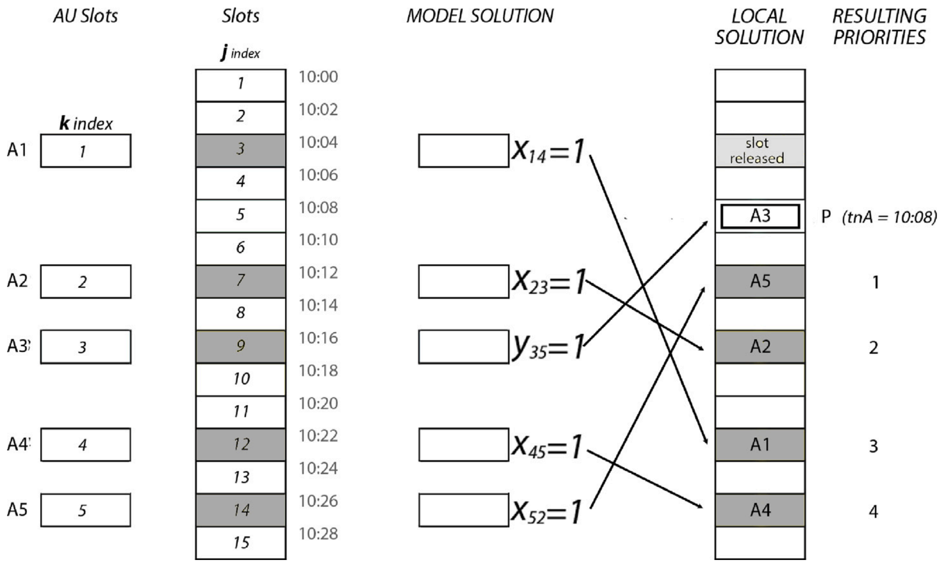


Fig. 10. Retrieving priorities from the UDPP-OPT solution.

6. Simulation design

The goal of UDPP is to improve the allocation of flights to ATFM slots, using the input from AUs, which are based on their priorities. The proposed UDPP-OPT model minimises the cost of delay for each *localSolution* of an airline participating in UDPP. In order to analyse the performance. i.e., the cost decrease the UDPP can provide (when optimised via UDPP-OPT), we turn to simulations. The assumptions under which the simulations are run are: the AUs can provide flight cost models, all AUs participate in UDPP and make use of UDPP-OPT to define their priorities.

To better evaluate the optimised UDPP performance we use two comparison mechanisms, MINCOST and NNB, which are described in Section 3. These mechanisms provide two different (theoretical) upper bounds to the delay cost reductions achievable in each hotspot: the MINCOST determines the allocation that minimises the overall costs, whilst the NNB minimises the overall costs on the condition that the new allocation, with respect to the FPFS, does not produce additional costs for any participating AU.

The simulations consist of generating 1000 instances of hotspots using synthetic data that are described below. The generated hotspots are solved, or in other words, final slot allocations are derived from the mechanisms. The difference in flight delay costs between the FPFS allocation and the final slot allocation obtained are calculated across all instances, for the three mechanisms. Results are analysed either in aggregate manner and through more detailed analyses like taking into account the number of flights in the hotspot and the impact on single AU (see Section 7).

In order to keep our experiments as realistic as possible, all test cases were delineated through data driven-procedures. To produce a hotspot instance, two main classes of information are required:

1. Hotspot configurations, i.e., the initial capacities, the capacity reductions, the number of flights involved, and the hotspot start and end times.
2. Models for the cost of flight delays. Appendix A provides an illustration of the main features on which these models are based.

Most of the information is retrieved from the EUROCONTROL's DDR2 database. The database is divided in Aeronautical information regulation and control (AIRAC) periods, each composed of 28 days. In this study, we use AIRAC 1907 and 1908 covering 56 days, from the 20th June 2019 to the 14th August included. The flight information in AIRACs contains the data required to define hotspot configurations and flight cost models, except for passenger data. The passenger-related data (load factors and potentially missed connections due to delay) are taken from 11-12-13 September 2014 passenger itinerary data (source Paxis[3]), describing the flights taken by groups of passengers (direct and connecting). To maintain consistency between the two data sets, the number of passengers per flight is inflated by 2.9%, due to the increase of the mean load factor of European carriers from 84.7% in September 2014 (IATA, 2014) to 86.6% in September 2019 (IATA, 2019).

From the dataset, the majority of the airport hotspots in Europe (1880 out of 3285) happened at the 25 major European airports (the ones with more than 10M passengers per year, Table 1). Also, as Fig. 11 shows, the most critical hotspots (higher capacity reduction and greater number of flights involved) usually occur at these 25 airports. For these reasons, we restricted our study to these airports.

[3] A data provider, now IATA Passenger and Traffic data.

Table 1
Airports used in the analysis.

EGLL	LFPG	EHAM	EDDF	LEMD
LEBL	LTFM	EDDM	EGKK	LIRF
EIDW	LFPO	LOWW	LSZH	LPPT
EKCH	LEPA	EGCC	LIMC	ENGM
EGSS	EBBR	ESSA	LGAV	EDDL

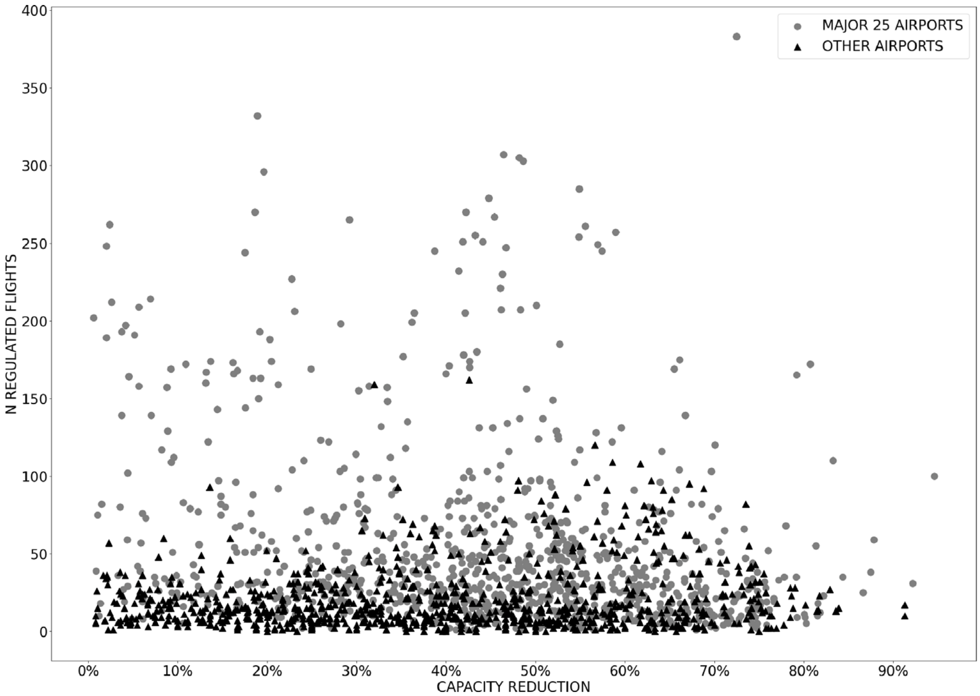


Fig. 11. Hotspot distribution in terms of number of flights and capacity reduction.

We generate/simulate life-like hotspots that we use to run the MINCOST, the NNB and the UDPP-OPT mechanisms. The procedure applied to create a hotspot instance is divided into two steps: we first sample the hotspot configuration parameters (airport, capacity, etc.) and then we generate the flights and their delay costs.

6.1. Generate hotspot configuration

The hotspot configuration as intended in the simulations can be observed in Fig. 6(a). The first column (or grid) shows the initial arrival schedule composed of 10 flights. The initial capacity is 60 flights per hour which determines the ETA time (ETA slot) separation of 1 min. The second column represents the hotspot time grid, which in this case is the result of a capacity reduction of 50% (30 flights per hour) imposed on the 10 flights in Fig. 6(a). The slot time separation then becomes 2 min. These two time grids define the hotspot configuration, which in order to be generated in a life-like manner we constructed with the following procedure:

- Pick at random a hotspot from the dataset. The hotspot information defines the destination airport, initial capacity, start time, capacity reduction, and the number n of flights involved.
- The initial capacity determines the ETA slot time separation, which, together with the start time and the number of flights involved defines the initial slot times (ETAs).
- The capacity reduction determines the new, enlarged slot separation, which together with the start time, defines the hotspot slot grid.

6.2. Generate flights and delay costs

The grids obtained at the previous step, so far empty, have to be filled with flights. Having in mind the generation of life-like hotspots, for each slot i in the grid:

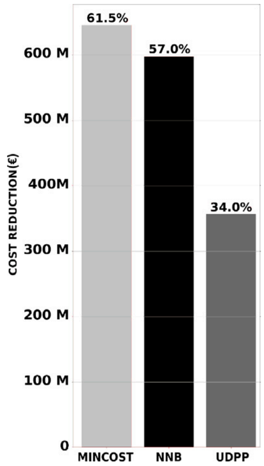


Fig. 12. Total cost reduction and percentage total reduction per mechanism.

- we pick at random a flight from the flights arriving at the selected airport. The flight data contains the AU, aircraft type, and haul (information required to build delay cost model),
- we assign to the flight the ith initial slot, which defines its ETA,
- we assign to the flight the ith hotspot slot, i.e., the FPFS assignment and corresponding delay,
- in order to determine the number of passengers (direct and connecting), we pick from the passenger itinerary database a flight with the same destination, AU and aircraft type. If a flight with such characteristics cannot be found, the number of passengers is determined by applying a load factor of 0.89 to the maximum passenger capacity of the aircraft type, all being direct passengers.
- the hotspot delay might cause the aircraft to hit the curfew in one of the subsequent legs, which might increase the cost of delay. To check for the curfew, we look for a flight (in the dataset) operated by the same AU, and aircraft type, at the selected airport, with arrival time within an hour around the assigned ETA. If such a flight is found, we check the next legs for this aircraft. Considering a minimum turnaround time of 30 min and the next legs, we determine the critical delay threshold that would cause the aircraft to break the curfew in one of its subsequent legs. If such a flight is not found, we assume no curfew issues.

7. Results

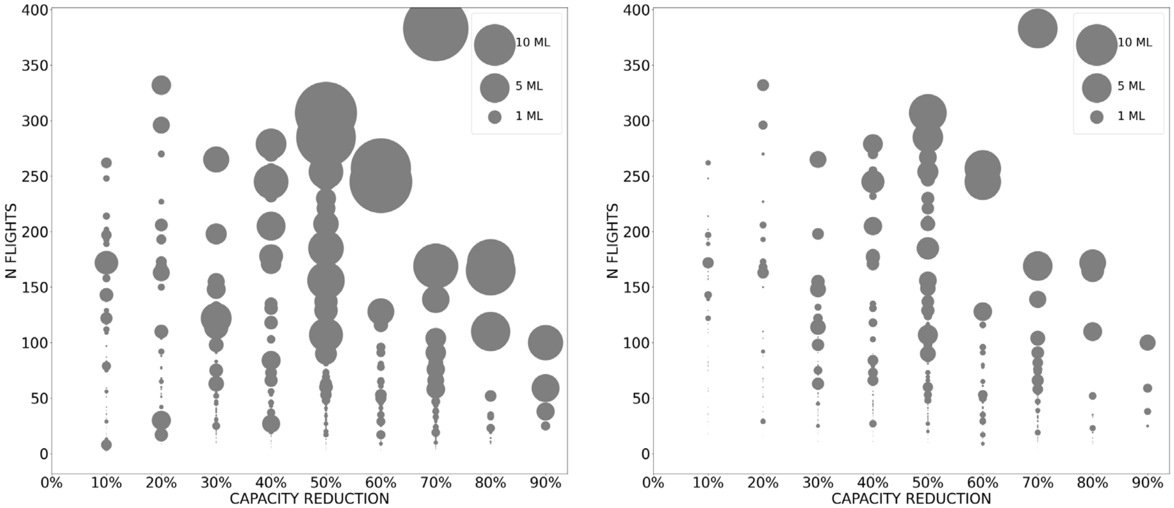
7.1. Aggregated cost reductions

We start by solving the 1000 hotspots, once for each mechanism: MINCOST, NNB and UDPP, defining the AUs' priorities via the UDPP-OPT and setting the parameter $HEFS = 0$. In these simulations the total delay costs (over all fights and airlines) amount to 1.05B€ in the initial FPFS. UDPP provides a reduction of 356M€ (34.0% of reduction), NNB reduction of 597M€ (57.9%), and MINCOST of 645M€ (61.5%), see Fig. 12.

MINCOST, which represents the maximum possible reduction of total cost that one can achieve through slot swapping procedure, reduces significantly the total cost to airlines from the initial FPFS allocation, highlighting the difference between a delay-minimisation and a cost-minimisation procedure. NNB also provides a significant cost reduction, indicating that most of the reductions when applying the MINCOST mechanism is achieved without negative impact on airlines. In other words, including the equity in MINCOST reduces the cost savings by 4.5% only. Finally, the UDPP mechanism reduces costs by 34%. The figure is notably smaller when compared to the two benchmark mechanisms, albeit very significant for airlines.

The first row of Table 2 reports the average cost reduction for the three mechanisms, the corresponding standard deviations and the average percentage reduction per hotspot. It can be seen that, as in the aggregate case, the greatest cost reductions occur for MINCOST, followed by NNB and then UDPP. However, the average percentage reduction per hotspot may significantly differ from the total percentage reduction (56.6% vs. 61.5% for MINCOST, 29.9% vs. 57.0 for NNB, and 16.8% vs 34.0% for UDPP). This, and the high values of the standard deviation suggest that the results of all three mechanisms depend heavily on the hotspots themselves.

Therefore, it is necessary to analyse how different hotspot characteristics affect the results. The reduction of the capacity increases the slot width, implying higher delays, as does the duration of the hotspot (the longer is the hotspot duration, it impacts more flights and higher delays are created). As costs generally increase with the delay, higher delay costs would be expected with higher capacity reduction, and longer hotspot duration. Fig. 13(a) shows that this is the case. A higher capacity reduction increases the total initial



(a) Initial FPFS total costs in euros vs capacity reduction and total number of flights.

(b) Total cost reduction in euros from UDPP vs capacity reduction and total number of flights.

Fig. 13. Initial FPFS costs vs UDPP cost reduction, across capacity reduction and number of flights involved in hotspots.

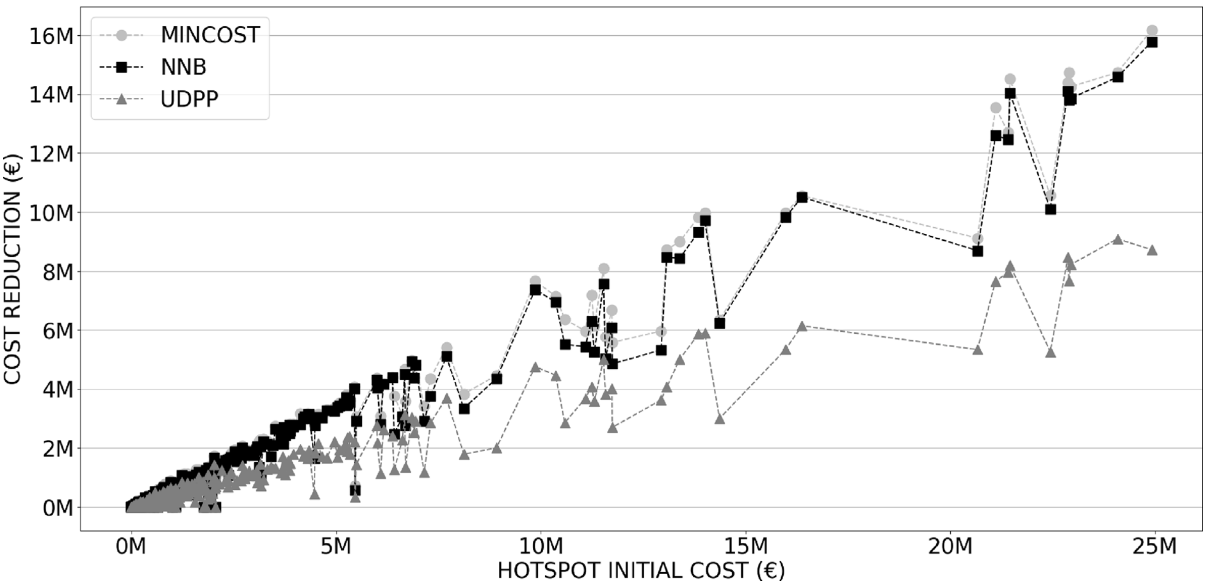


Fig. 14. Initial total costs in euros vs total reduction for the three mechanisms.

Table 2
Cost reduction performance in Euros of three mechanisms.

	MINCOST			NNB			UDPP		
	avg	std	%	avg	std	%	avg	std	%
Per hotspot	645k	1,83M	56,6	598k	1,76M	29,9	356k	1,05M	16,8
Per AU	27,8k	189k	−676,0	25,7k	160k	32,8	15,3k	167k	4,3

(i.e. FPFS) delay costs of the hotspot. Interestingly, the instances of high initial delay costs, are also the ones where airlines generally benefit the most from the possibility to change the originally assigned slot in the hotspot, as shown in Fig. 13(b). The figure shows the cost savings with respect to the FPFS when UDPP is applied.

A statistical test confirms this dependency to initial FPFS costs, as we find that the UDPP cost reduction is highly correlated with the initial costs (Bravais-Pearson correlation index $\approx 0,98$), as also shown in Fig. 14, visualising total cost reductions per mechanism, across cost amounts. Fig. 14 shows that also MINCOST and the NNB cost reductions are similarly correlated with the initial costs, as each of them experiences a Bravais-Pearson correlation index of $\approx 0,98$ with FPFS costs. For this reason, in order to compare the UDPP with the benchmark mechanisms, we can simply look at the cost reduction as a function of the initial costs.

7.2. Cost reduction and airline size

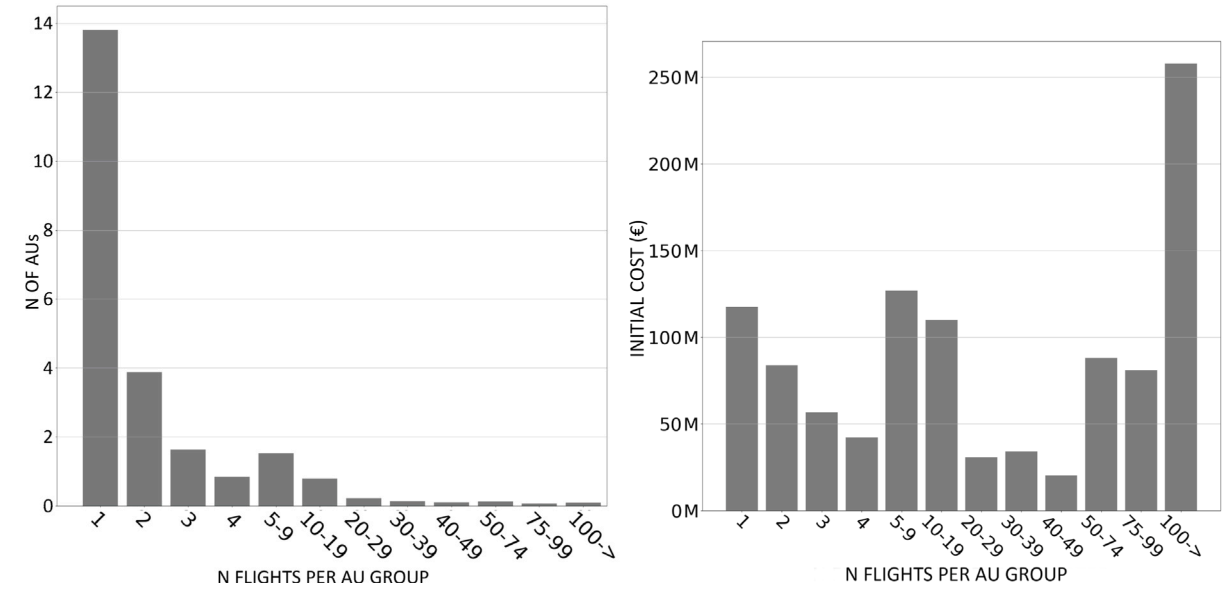
The second row of Table 2 shows the impact on individual AUs per applied mechanism. Again, the greatest mean cost reductions occur for MINCOST, followed by NNB and then UDPP. However, the average percentage cost reduction per AU reveals quite counter intuitive results, particularly for the MINCOST. The reason of this discrepancy lies in the fact that the percentage cost reduction for each AU is relative to their initial costs: if an AU A has initially 100 000€ of delay costs and obtains a reduction of 50 000€, the percentage reduction is 50%; instead, if initial costs of another AU B are 1 000€, and the final allocation costs are 4 000€, B suffers an increase of costs of 3 000€, resulting in a negative reduction, with a percentage reduction of −300%. Although the total reduction (considering the sum of both AUs reductions) is positive and amounts to $50\,000 - 3\,000 = 47\,000$€, the average percentage reduction is negative and amounts to $(50 - 300)/2 = -125\%$. This phenomenon can arise whenever negative reductions occur, which is the case of MINCOST: since optimisation takes place globally, it is possible that the (few) flights of one AU, all with a low delay cost, are further delayed in order to anticipate flights of another airline with a high delay cost (see airline C in Fig. 1). Thus, the AU's costs increase (i.e., there is a negative reduction). This demonstrates the impact of not taking equity into account in the MINCOST, which favours AUs with higher costs. The high value of the standard deviation indicates that it is hard to predict the effect of the mechanism on a particular AU in a given hotspot without further characterisations of the hotspot and the AU.

To this aim, we investigate the structure of the hotspots in terms of number of flights and number of airlines involved. Given a particular hotspot, one can divide all airlines involved in groups, depending on the number of flights they have in the hotspot. Fig. 15(a) illustrates the average number of airlines of each group in a hotspot, showing that hotspots generally involve a high number of AUs with a small number of flights, the so-called low-volume operators in constraint (LVOCs, i.e. ≤ 3 flights). Likewise, Fig. 15(b) shows the average initial (i.e., FPFS) costs across the groups.

An AU may fully exploit UDPP features if it has slots to swap (FDR) and/or slots to release (SFP). As such, the LVOCs do not have enough flexibility to take advantage of UDPP, while they statistically represent the very large majority of AUs (83%) in an hostpot (Fig. 15(a)), and hold a quite significant portion (25%) of the total initial costs (Fig. 15(b)).

The inability to use UDPP features converts into smaller cost reductions for LVOCs, as illustrated in Fig. 16(a). While NNB and MINCOST can reduce costs for LVOCs, sizeable cost reductions with UDPP only happen for high-volume users. Fig. 16(b) illustrates the same point by displaying the average of the cost reduction percentage for each AU group. More specifically, we compute the percentage cost reduction for each airline and each hotspot independently (with respect to their own initial costs) and average the values obtained within the different groups.

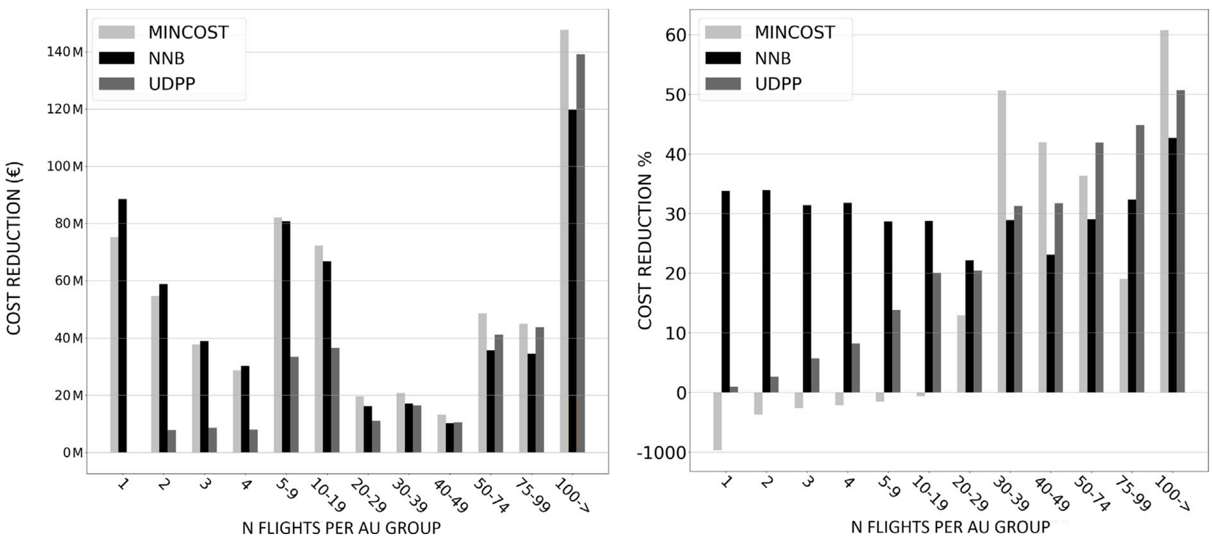
MINCOST provides a negative average percentage cost reduction for LVOCs, even though the total gains for these airlines are positive (see Fig. 16(a)). In the case of UDPP, as expected, we have almost no effect on the average cost reduction for LVOCs but significant gains for high-volume AUs. Finally, NNB provides very good cost reductions across different airline sizes, slightly less performing than UDPP for high volumes AUs, but particularly convenient for LVOCs. Note that this fact is not trivial, since the impact of NNB on low-volume users might have been simply neutral. Interestingly, NNB provides good equity across airlines, while having excellent performance from the total cost reduction point of view. This fact can be explained as follows. In order to reduce the delay of some flights, some others have to be further delayed in exchange. Let us assume for simplicity that high and low delay cost flights are uniformly distributed among airlines. MINCOST does not take into account flights' ownership, so the further delayed flights will be picked simply in order to optimally reduce the objective function, regardless of whom they belong to. Since LVOCs own the majority of the flights, we expect that on average the majority of the flights further delayed belong to the LVOCs. Instead, NNB is constrained to not provide a negative impact to any AU. Consequently, LVOCs' flights are still eligible to obtain a delay reduction, but they are less likely to be chosen for a penalisation, and, in particular, those which represent the only flight of the respective AUs cannot be further delayed. In summary, for NNB the majority of the candidate flights for a delay reduction are within the LVOCs, but the ones that have to be further delayed are more likely to be chosen within the AUs with a higher number of flights.



(a) Average number of AUs per AU group.

(b) Average initial costs in euros per AU group.

Fig. 15. Distribution of AU groups and initial costs.



(a) Total cost reduction in euros per AU group with respect to the FPFS.

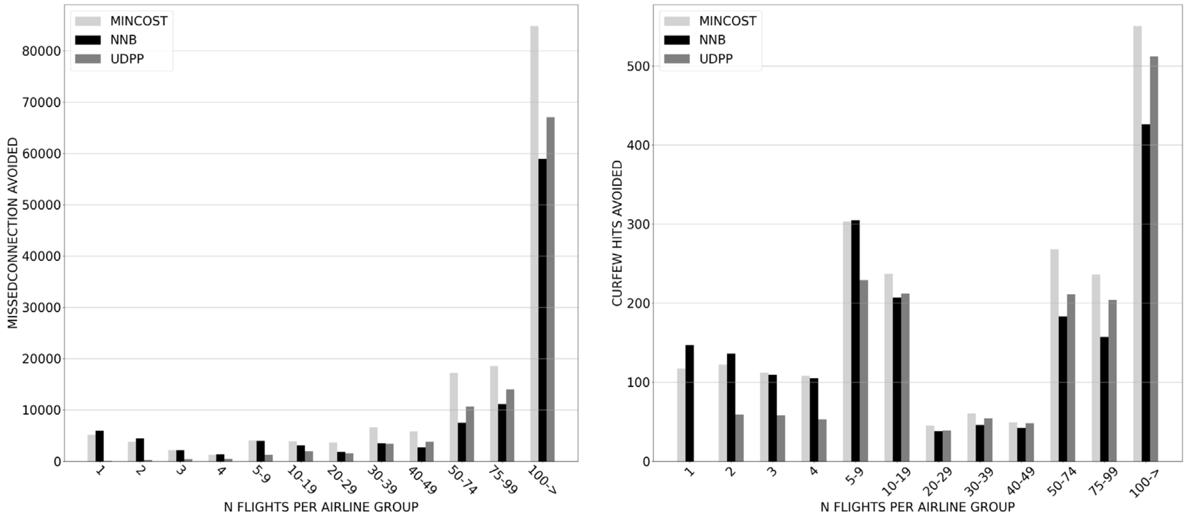
(b) Average percentage cost reduction per AU group with respect to the FPFS.

Fig. 16. Total and average percentage cost reduction with respect to FPFS.

7.3. Negative impact of UDPP

A review of the potential negative impact of UDPP, as described in Section 4.3.2, reveals that they occur in only 49 cases (4.9% of the time), affecting 1 605 flights for a total of 6 448 minutes (4.0 min per flight on average). Conversely, in 458 instances (45.8% of the total), 7 371 flights exhibited a positive impact, amounting to 39 790 minutes in total (5.4 min per flight on average). Although these figures demonstrate that the negative consequences of the UDPPmerge are limited and do not have a significant influence on airlines, we are nevertheless interested in determining whether allowing some flexibility to allocate flights earlier than their arrival time would improve the situation.

To do this, we set the $HFES$ parameter, introduced in Section 4.3.2, to 5 min. We thus perform the same simulations including this parameter.



(a) Number of avoided missed connections per AU group.

(b) Number of avoided curfew hits per AU group.

Fig. 17. The impact of UDPP on passenger missed connections and flight curfews.

Thus, we compare the UDPP-OPT under HFES settings of 0 and 5 min. In terms of total costs, the reduction with HFES of 5 min amounts to 360M€ which is 1% more than the 356M€ obtained with HFES of 0 min. In fact, the only difference in the final allocation are those 49 instances of UDPP negative impact when HFES equals 0. In fact, with HFES= 5 the negative impact is completely avoided.

7.4. Impact on passenger missed connections and flight curfew hits

So far, our analysis has focused on the impact of the UDPP mechanism on airlines, in comparison with the other mechanisms (MINCOST and NNB). However, it is important to note that the adoption of one or the other of these mechanisms also has an influence on other air transport stakeholders, such as passengers and airports.

As far as passengers are concerned, one of their main disruptions is a missed connection. In this context, we focus on the impact on mechanisms in terms of the number of passengers missing their connections (or simply *missed connections*). The initial total number of missed connections under FPFS was 308,694. MINCOST reduces this to 151,729 (−50.8%), NNBOUND to 202,251 (−34.5%) and UDPP to 203,898 (−33.9%). Fig. 17(a) shows the distribution of the avoided missed connections among the different airline groups. The trend is consistent across all mechanisms, with the number of flights an airline has increasing the likelihood of missed connections being avoided. Airlines with a large number of flights tend to benefit more from UDPP than NNB, confirming the potential of this mechanism that combines flexibility with equity. Conversely, the UDPP is less effective for airlines with a small number of flights, as they lack the flexibility to benefit from it.

Regarding the impact due to airport operations, an interesting measure is the number of curfew hits: flights which delay caused directly or indirectly (the delay being passed on to the next aircraft rotations) the failure to obtain permission to land at the arrival airport. The total number of curfew hits under FPFS is 3 957. MINCOST reduces this number to 1 750 (−56%), NNBOUND to 2 056 (−48%) and UDPP to 2 278 (−42%). Fig. 17(b) demonstrates that UDPP outperforms NNB for a diverse range of airlines, including those with at least 10 flights in a hotspot. As anticipated, UDPP encounters greater challenges with airlines with fewer flights, as they have limited flexibility to reallocate their slots. However, in contrast to the missed allocations scenario depicted in Fig. 17(a), UDPP still enhances the FPFS solution even for airlines with minimal flight operations (two or three flights).

7.5. Computational analysis

The mechanisms are implemented in Python using the linear optimisation library provided by *Gurobi* and the simulations are run on a Dell XPS 8940, Intel Core i7-10700 8 Core 800 MHz machine. The computational time for a UDPP-OPT instance turned out to be negligible, with an average time of 0.3 s. The combinatorial nature of the problem implies a theoretical exponential growth of the computational effort with the increase of the number of flights; however, as the number of flights in the actual hotspots is limited (as shown in Fig. 11), this does not appear to be an issue. The maximum computational time required to solve UDPP-OPT for an AU with 198 flights amounts to 17.7 s.

8. Conclusions

In this paper we showed that (1) the UDPP mechanism can be successfully formulated in order to test it in fast-time simulations; (2) the way airlines set their priorities to use UDPP can be optimised, and (3) UDPP allows significant cost savings, but these savings are very dependent on the number of flights an airline has in the hotspot, and the total number of flights in the hotspot.

The UDPP mechanism developed by SESAR has been geared towards usability right from its birth. Indeed, airlines could in principle already swap slots, even between them, but only through a cumbersome and time consuming process. UDPP made it simple thanks to a few rules, and the live experiments within SESAR showed the potential of the mechanism (Pilon et al., 2019). However, no large-scale simulations existed until now to estimate the impact of larger UDPP deployment. The formulation of UDPP presented here makes it possible.

Moreover, while the implementation of the mechanism itself is required to estimate its impact, one also needs a model of the decision-making process in airlines, who have to set the priorities for UDPP. In this article, we presented an optimisation model (UDPP-OPT) that finds the best possible priorities given the cost structure of the airline. This can be used to model airlines in fast-time simulations, as we did here.

The UDPP-OPT model could be used directly by airlines in the future, provided that they have sufficient data on their costs. The AUs' requirements of data confidentiality are respected as the optimisation, the only stage in which the cost knowledge is exploited, is performed locally by the AUs without sharing cost information with external parties. Thus, a tool like UDPP-OPT could potentially make the UDPP more widespread. Furthermore, the run times are of the order of a couple of seconds in most cases. Such automation of the internal slot swapping process would enable airlines to include all of their impacted flights (not only those where the highest costs would be incurred), and be a part of the potential dynamic use of UDPP when hotspot characteristics change (might happen often as ANSPs are constantly trying to improve the situation).

Our estimation of the cost reduction yielded by this 'optimised' version of UDPP is significant, with 356k€ per hotspot and 15.3k€ per airline on average. Moreover, our simulations showed that this represents roughly half of the maximum possible reduction obtainable through slot swapping. In the vast majority of cases, no airline loses from the final allocation of UDPP, in accordance with UDPP rules, and even the rare cases of negative impact are completely eliminated when a tolerance for early arrival (i.e. before their ETA) of 5 min is introduced. Note that this tolerance is actually already used in the actual ATFM regulations.

In order to be able to run the UDPP-OPT model, an airline needs to have the information on flight schedules, passengers and their connections, flight operational costs and initial FPFS allocation. All of this information is present in the airline systems, but not necessarily available to the flight dispatchers at the time (and promptness) of need. To facilitate the potential use of UDPP-OPT, we are sharing the libraries used for cost modelling in this work (https://github.com/andygaspar/Hotspot). The libraries are easy to customise to own costs and needs.

The simulation showed that UDPP is able to provide significant benefits to those airlines which have enough flights to exploit the UDPP features. However, we have also seen that generally a great portion of the AUs involved in hotspots is composed of LVOCs, users that might benefit from positive impact generated by other airlines' protections but that struggle to directly take advantage of UDPP. For this reason further development of UDPP solutions are necessary in order to improve the chances of LVOCs to reduce their delay costs. Another limitation of UDPP, is that currently the possibility of a direct inter-airline slot exchange is not included. The extension of UDPP towards this direction might lead to further improvement in terms of delay reduction and flexibility. The ER-4 BEACON SESAR project (https://www.beacon-sesar.eu/) for instance studied the feasibility of credit-based mechanisms, allowing airlines to swap slots across regulations.

Finally, in this work we suggested a procedure for the performance evaluation of mechanisms aimed at reducing the delay costs, allowing to go beyond the simple comparison with the FPFS. In this respect, the MINCOST and the NNB mechanisms represent two useful benchmarks that provide the minimum cost allocations achievable. Given any mechanism, the cost reduction with respect to the FPFS clearly remains an important performance assessment indicator. However, we might want to know how the specific mechanism compares to the theoretical benchmarks like MINCOST and NNB.

CRediT authorship contribution statement

Andrea Gasparin: Writing – original draft, Validation, Software, Methodology, Data curation, Conceptualization. **Lorenzo Castelli:** Supervision, Methodology, Funding acquisition, Conceptualization, Writing – review & editing. **Tatjana Bolić:** Writing – original draft, Conceptualization. **Gérald Gurtner:** Writing – original draft, Methodology, Formal analysis, Funding acquisition. **Nadine Pilon:** Conceptualization.

Acknowledgements

The work presented here is a result of the BEACON project. This project has received funding from the SESAR Joint Undertaking under grant agreement No. 893100 of the European Union's Horizon 2020 research and innovation programme. The opinions expressed herein reflect the author's view only. Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein. More information can be found here: https://www.beacon-sesar.eu/.

Table A.3
Flight cost model dependencies.

Cost item	Dependency
Passenger hard	Non Low cost/Low cost AU, destination, flight haul, number of pax, itineraries
Passenger soft	Non Low cost/Low cost AU, destination, number of pax
Maintenance & crew	Non Low cost/Low cost AU, destination, aircraft type
Curfew	Aircraft type, rotations, turnaround, next flight duration

Appendix A. Cost delay model

The work in this article revolves around the possibility for airlines to reduce their costs. Hence, the UDPP mechanism explained here needs to be tested by using airlines' cost estimation, for different situations.

Airlines' costs in general are complex, and airlines themselves have heterogeneous capabilities when it comes to cost estimates, in particular during day-to-day operations. However, while delay is an acceptable proxy for many applications, cost represents a more important metric for airlines in the long run, simply because ultimately they are profit-driven companies.

Hence, some effort was dedicated more than 15 years ago to come up with estimates of these costs, that was compiled in Cook and Tanner (2011). This study has been updated twice (Cook and Tanner, 2015; Cook et al., 2021), as it became an industry standard since, see for example the *EUROCONTROL Standard Inputs for Economic Analysis* (EUROCONTROL, 2024), which is a guideline for performing economic analyses in ATM. Furthermore, the airlines were consulted during the preparation of the study.

The specific approach can be found in the references, but the main relevant pieces of information for this article are the following:

- The cost is given as a function of the delay of the flight, here the difference between the expected time of arrival and the actual arrival.
- The cost is divided in several components:
  - Cost of extra maintenance:
  - Cost of crew: this component estimates the cost calling back crews, extra shift, etc; due to delay. This cost is estimated as a high-level average.
  - Cost of curfew: this component estimates the average cost of a flight infringing a curfew at the end of the day. It is a fixed number for all flights, output of an estimate made on various situations.
  - Passenger costs:
    - Hard costs, that comprise of:
      - Duty of care: covers mainly meals provided to passengers in case of departure delay.
      - Compensation: estimates the cost of paying for compensation according to the European Regulation 261 (European Commission, 2004).
      - Re-accommodation costs: covers the price of paying for accommodation for passengers would get stuck for the night before of their final destination.
    - Soft costs: cover the potential loss of market (due to degradation of brand image).

These costs also depend on different factors as summarised in Table A.3.[4]

The behaviour of these different components are quite diverse with delay, due mainly to the type of estimation. For instance, the cost of maintenance, crew, and the passenger soft costs are very smooth with the delay. They are modelled with well-behaved functions (see Cook and Tanner (2011) for more details on the type of function used), typically linear or parabolic. Conversely, costs of curfew, compensation, and re-accommodation are modelled as step functions, yielding a very non-linear behaviour for the total cost.

Furthermore, these costs can be specified for three scenarios: low, base and high. The scenarios allow different costs to be used depending on the airline's business model. For example, a low cost scenario can be used for a low cost point-to-point carrier, and a high cost for a legacy network carrier.

Data availability

The authors do not have permission to share data.

[4] Flight haul is computed using the great-circle distance between the origin and the destination of a flight. Flights ≤1500 km are considered short-haul, medium haul 1500–3500 km and long haul >3500 km.

References

Bertsimas, D., Farias, V.F., Trichakis, N., 2011. The Price of Fairness. Oper. Res. 59 (1), 17–31.

Bertsimas, D., Gupta, S., 2016. Fairness and collaboration in network air traffic flow management: an optimization approach. Transp. Sci. 50 (1), 57–76.

Castelli, L., Pesenti, R., Ranieri, A., 2011. The design of a market mechanism to allocate air traffic flow management slots. Transp. Res. C 19 (5), 931–943.

Cook, A.J., Tanner, G., 2011. European Airline Delay Cost Reference Values, Final Report (Version 3.2). Technical Report, EUROCONTROL Performance Review Unit.

Cook, A.J., Tanner, G., 2015. European Airline Delay Cost Reference Values, Updated and Extended Values (Version 4.1). Technical Report, EUROCONTROL Performance Review Unit.

Cook, A.J., Tanner, G., Bolic, T., 2021. D3.2 - Industry Briefing on Updates to the European Cost of Delay. Technical Report, ER4 SESAR Project BEACON.

Delgado, L., Gurtner, G., Bolić, T., Castelli, L., 2021. Estimating economic severity of air traffic flow management regulations. Transp. Res. C 125, 103054.

EUROCONTROL, 2006. Airport CDM Operational Concept Document.

EUROCONTROL, 2007. Air Traffic Flow and Capacity Management Operations Users Manual.

EUROCONTROL, 2022. High level network concept of operations CONOPS 2029.

EUROCONTROL, 2023. EUROCONTROL forecast update 2023–2029.

EUROCONTROL, 2024. EUROCONTROL standard inputs for economic analysis.

European Commission, 2004. Regulation (EC) No 261/2004 of the European Parliament and of the Council of 11 February 2004 Establishing Common Rules on Compensation and Assistance to Passengers in the Event of Denied Boarding and of Cancellation or Long Delay of Flights, and Repealing Regulation (EEC) No 295/91 (Text with EEA relevance) - Commission Statement. Technical Report, European Commission.

Evler, J., Schultz, M., Fricke, H., Cook, A., 2022. Stochastic Delay Cost Functions to Estimate Delay Propagation Under Uncertainty. IEEE Access 10, 21424–21442.

Gurtner, G., Bolić, T., 2023. Impact of cost approximation on the efficiency of collaborative regulation resolution mechanisms. J. Air Transp. Manag. 113, 102471.

Hamdan, S., Cheaitou, A., Jouini, O., Andersson Granberg, T., Jemai, Z., Alsyouf, I., Bettayeb, M., Josefsson, B., 2022. Central authority–controlled air traffic flow management: An optimization approach. Transp. Sci. 56 (2), 299–321.

Hansen, M.M., Gillen, D., Djafarian-Tehrani, R., 2001. Aviation infrastructure performance and airline cost: a statistical cost estimation approach. Transp. Res. E: Logist. Transp. Rev. 37 (1), 1–23.

IATA, 2014. Air Passenger Market Analysis, September. Technical Report, IATA.

IATA, 2019. Air Passenger Market Analysis, September. Technical Report, IATA.

Jacquillat, A., Vaze, V., 2018. Inter-airline equity in airport scheduling interventions. Transp. Sci. 52 (4), 941–964.

Niarchakou, S., Sfyroeras, M., 2022. ATFCM Operations Manual.

Odoni, A.R., 1987. The flow management problem in air traffic control. In: Flow Control of Congested Networks. Springer, pp. 269–288.

Pilon, N., Cook, A., Ruiz, S., Bujor, A., Castelli, L., 2016. Improved flexibility and equity for airspace users during demand-capacity imbalance. In: Sixth SESAR Innovation Days.

Pilon, N., Guichard, L., Bazso, Z., Murgese, G., Carré, M., 2021. User-Driven Prioritisation Process (UDPP) from advanced experimental to pre-operational validation environment. J. Air Transp. Manag. 97, 102124.

Pilon, N., Guichard, L., Cliff, K., 2019. Reducing impact of delays using airspace user-driven flight prioritisation. In: Ninth SESAR Innovation Days.

Ruiz, S., Guichard, L., Pilon, N., Delcourte, K., 2019. A new air traffic flow management user-driven prioritisation process for low volume operator in constraint: Simulations and results. J. Adv. Transp. 2019.

SESAR, 2007. Milestone Deliverable D3 – The ATM Target Concept.

SESAR, 2019. SESAR solution PJ07.02 SPR-INTEROP/OSED for V2 - part I.

Sherali, H.D., Hill, J.M., McCrea, M.V., Trani, A.A., 2011. Integrating slot exchange, safety, capacity, and equity mechanisms within an airspace flow program. Transp. Sci. 45 (2), 271–284.

Terrab, M., Odoni, A.R., 1993. Strategic flow management for air traffic control. Oper. Res. 41 (1), 138–152.

Vossen, T., Ball, M., 2006a. Optimization and mediated bartering models for ground delay programs. Naval Res. Logist. 53 (1), 75–90.

Vossen, T.W., Ball, M.O., 2006b. Slot trading opportunities in collaborative ground delay programs. Transp. Sci. 40 (1), 29–43.