

An inter-airline ATFM slot trading mechanism for delay cost reduction

Andrea Gasparin, Federico Julian Camerota Verdù, Giulio Alberti, Lorenzo Castelli *

Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Via Alfonso Valerio 6/1, Trieste, 34127, Italy

ARTICLE INFO

Keywords:

Air Traffic Flow Management (ATFM)
ATFM slots
Hotspot
Neural networks
Binary optimisation

ABSTRACT

When unexpected temporary reductions in airspace capacity occur, flights may need to be rescheduled, often resulting in delays and associated costs. Reallocating limited airspace resources in a way that equitably mitigates delay costs across airlines, while directly involving them in the decision-making process, remains a significant operational challenge. Although several approaches have been proposed in the literature, the complexity of tactical air traffic flow management and the diverse requirements of stakeholders have so far limited their practical applicability. In this work, we introduce a novel inter-airline slot trading framework that provides a fair mechanism aligned with airlines' operational needs, while remaining sufficiently simple for real-world deployment. The framework is based on an integer programming formulation, and we show that the resulting trade selection problem can be expressed as a Maximum Weight Independent Set (MWIS) and solved efficiently. To cope with the growing computational burden associated with large-scale instances, we develop a three-step machine-learning-based pre-processing heuristic. This heuristic uses neural networks to discard flight combinations unlikely to generate beneficial trades and to estimate the value of the remaining ones, thereby restricting the search space to the most promising candidates. Computational experiments demonstrate that the proposed framework effectively reduces total delay costs while ensuring an equitable distribution of benefits among airlines. Moreover, the neural-network-based pre-processing component enables the efficient solution of instances that would otherwise be computationally intractable, substantially enhancing the practical applicability of the approach.

1. Introduction

With continuous growth in airspace demand [1], managing the air traffic network is becoming increasingly challenging, potentially leading to a significant rise in flight delays. This represents a major economic loss for both airlines and passengers.

In this paper, we focus on the important problem of the economic impact of delays caused by the temporary disruption of a resource, such as a flight sector or an airport. A resource is defined by its capacity, i.e., the number of flights it can handle per unit of time. Unexpected events (such as adverse weather, operational restrictions, or low visibility) can temporarily reduce a resource's efficiency, necessitating a capacity reduction to maintain safety. This issue falls within the scope of Air Traffic Flow Management (ATFM), a critical component of Air Traffic Management (ATM), which aims to mitigate imbalances between airspace/airport capacity and demand. According to EUROCONTROL, flight volumes returned to pre-pandemic levels in 2025 and continue growing at an average annual rate of +2.2% thereafter [1]. This trend suggests that mismatches between demand and available capacity will become increasingly significant in future

ATFM systems, attracting a considerable research effort to develop solutions that mitigate the effects of this phenomenon [2].

If a disruption occurs during a highly congested period, the resulting demand-capacity imbalance creates what is known as a *hotspot*. In such a situation, a resource may become unavailable to some of the flights that had originally planned to use it. As a result, some flights cannot be accommodated on time and have to be delayed. In Europe, this is managed by the Network Manager (NM) through the allocation of air traffic flow management slots (ATFM slots) [see, e.g., 3,4].

Fig. 1(a) shows a possible hotspot scenario in which the original capacity is reduced by the 50%. We can notice that in this case most of the flights within the hotspot are affected by delay as they are re-assigned to a later time slot with respect to their original one. In Europe, the current policy to perform the flights' slot reallocation follows the *First Planned First Served (FPFS)* rule, meaning that the initial flights' order is preserved, as in the case of Fig. 1(a). Although this assignment seems the natural way to handle such a situation, the fact that different flights generally have different economic values suggests that other

* Corresponding author.

E-mail addresses: andrea.gasparin@dia.units.it (A. Gasparin), federico.camerota@phd.units.it (F.J. Camerota Verdù), giulio.alberti@studenti.units.it (G. Alberti), castelli@units.it (L. Castelli).

<https://doi.org/10.1016/j.orp.2026.100389>

Received 19 December 2025; Received in revised form 21 March 2026; Accepted 10 April 2026

Available online 19 April 2026

2214-7160/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

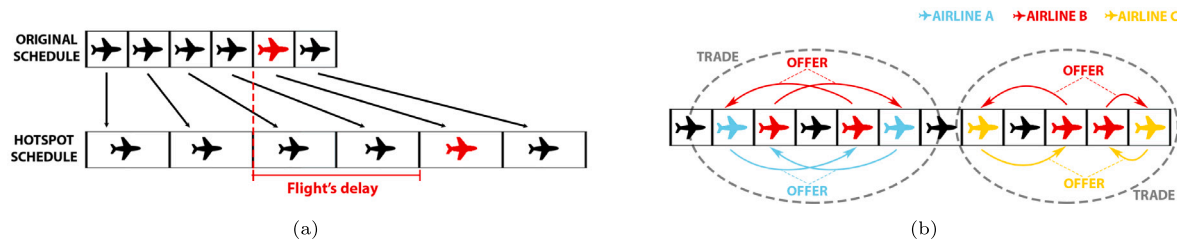


Fig. 1. Examples of hotspot resolution: in (a) FPFS allocation is used, thus delayed flights maintain their original order. Instead, in (b) a trade-based mechanism recombines the original schedule to achieve an improved solution.

reallocation mechanisms may provide more efficient solutions from an airline cost perspective.

However, developing such a solution requires careful consideration of several aspects which go beyond mere compliance with operational constraints. Feedback from various stakeholders [5–7] has indeed highlighted a set of principles that should be taken into account when designing new rescheduling mechanisms. Airlines emphasise that no carrier should be penalised as a result of the reallocation, and that any delay cost savings achieved should be distributed equitably among those involved, a concept often also referred to as *equity* or *fairness*. Stakeholders also express a strong preference for Collaborative Decision Making (CDM) frameworks, wishing to remain directly involved in decisions that affect their operations. From a practical standpoint, the mechanism must strike a delicate balance: it should be simple enough for airlines to understand and use, yet sufficiently automated to be operationally viable. Finally, given the tactical nature of hotspot interventions, any proposed solution must be computationally efficient enough to deliver results within the required time frame. A detailed discussion of the above mentioned requirements will be carried out in Section 3. To the best of our knowledge, the User-Driven Prioritisation Process [UDPP, 8] is the only mechanism that fulfils these requirements. In UDPP, airlines actively participate through a prioritisation mechanism enabling intra-airline flight rearrangements. The framework has gained strong consensus among airlines [9], particularly for its equity interpretation: flexibility is acceptable only if it does not negatively impact other airlines' flights, and non-participating flights experience no delay increase [8]. However, UDPP does not permit inter-airline slot trading, limiting cost reduction potential to airlines with high operational flexibility [10].

Extending the framework to enable inter-airline slot trading is thus crucial for improving delay cost mitigation and broadening benefit distribution, thereby extending the UDPP fairness concept to a broader, inter-airline context. However, this introduces implementation challenges: a fully integrated optimisation model which takes into account all the aforementioned requirement is exceptionally complex, while a fully manual, human-in-the-loop trading system would be prohibitively inefficient for operational use.

A trade off solution has been proposed by Vossen and Ball [11] with a mechanism in which each airline is allowed to propose a set of slot swap offers and the central coordinator (the Federal Aviation Administration (FAA) in the US context or the *Network Manager* (NM) in Europe) plays the role of the mediator, deciding which offers to match. In this context, we refer to an *offer* as a tuple consisting of a set of flights and their desired new position in the schedule, and by *trade* we indicate the matching of offers from different airlines (Fig. 1(b)). Despite showing significant experimental results, this mechanism still requires a huge effort from the airlines to evaluate all possible offer combinations, which grow exponentially with the size of their fleet. In addition, under this framework, all airlines make their offers simply relying on the information regarding their flights and their current schedule, without the possibility to account for the other airlines' situation and reach a better solution for both. With the *Inter-airline Slot Swap Offer Provider* (ISTOP) model, we aim at inverting this process

to overcome these two issues while ensuring full compliance with stakeholders' preferences. The ISTOP introduces a centralised approach to avoid any potential airlines' overhead by letting the NM play the role of the airlines' *broker*, which provides a set of ready-made offers that each airline can evaluate and decide either to accept or refuse, maintaining in such a manner their direct involvement in the allocation process.

Based on a set of information describing the flights' delay cost structure, the NM computes the trades that maximise an objective function which aims at reducing the airlines' costs, trying to equitably spread the benefits of the new allocation among the different stakeholders, and guaranteeing no cost increase for any of the airlines involved.

Within this framework, a slot trade is represented as the matching of several offers of different airlines. If all the offers defining a trade are accepted, the corresponding slot swap eventually takes place; otherwise, all flights involved in the trade retain their original position in the schedule. Still, the optimisation problem underlying ISTOP, formulated as Maximum Weighted Independent Set problem (MWIS), becomes increasingly challenging as the number of flights grows, making the ISTOP in practice not applicable in real large instances. This difficulty arises from the exponential growth in the number of possible trades, which constitute the decision variables of the problem. To address this issue, we developed a tailored Branch&Bound algorithm that exploits the structural properties of the problem to enable efficient solution. In addition, we propose a machine-learning-based heuristic that predicts, for each airline, the flight combinations most likely to appear in an optimal solution, thereby substantially reducing the number of trades considered in the optimisation.

Our selection algorithm is divided into three steps: first, by use of a neural network, we discard the flight combinations of each airline that are unlikely to generate a convenient offer. The remaining ones are then evaluated by a second neural network which predicts the value of the best trade in which they are involved. Finally, the trades are computed by considering only a limited set of combinations per airline, selected according to their predicted value obtained in the second step of the algorithm. The resulting heuristic is capable of providing near-optimal solutions also for instances with a large number of flights in a sufficiently short amount of time.

The contribution of our work is twofold: firstly, we propose an effective ATFM slot allocation mechanism that meets the stakeholders' requirements and offers significant cost savings for airlines. Secondly, our machine learning heuristic ensures that the mechanism remains efficient even when dealing with large instances, ensuring its applicability in all real-world scenarios.

The article is structured as follows. Section 2 provides a literature review of the slot allocation problem, describing the problem statement in depth. In Section 3 we discuss the operational requirements that any acceptable slot reallocation mechanism should satisfy, drawing on stakeholder feedback and existing literature. In Section 4 we explain the details of the ISTOP mechanism and present its mathematical formulation. The analysis of its complexity is carried out in Section 5, where we also introduce the proposed heuristics aimed at tackling large instances. Section 6 describes the assessment methodology and

experimental setup used to evaluate our approach. In Section 7 we present the computational results, first assessing the performance of the heuristic and then measuring the impact of the mechanism in terms of cost reduction and equity. Finally, the conclusions of our work are presented in Section 8.

2. Literature review

Air traffic demand-capacity imbalance is a well-studied issue in the literature, and various solutions have been proposed to mitigate its effects. Initial work by Odoni [12] and Terrab and Odoni [13] set the groundwork to formalise the problem with optimisation models. Subsequent research has diversified this approach, generating several distinct models and refinements across different research directions. In particular, notable efforts have been made to develop frameworks that tackle the issue by supporting a collaborative approach in air traffic management [14], in line with the preference of airlines to be involved in the decision-making process. In this regard, the OPTIFLOW mechanism [15] introduced the concept of modeling slot assignments as inter-airline swaps, allowing each flight to target a specific “goal slot” as part of the optimisation process. Building on OPTIFLOW, Vossen and Ball [11] proposed the *at least-at most* (AMAL) mechanism to formalise inter-airline slot trading. This framework assumes airlines may tactically accept greater delays on less critical flights to secure reductions on more important ones. Further refinements by Sherali et al. [16] enhanced the trading dynamics using graph-based representations. In adapting OPTIFLOW to the European context, Castelli et al. [3] introduced a slot trading mechanism within the FPFS assignment structure to minimise overall slot assignment costs through the MINCOST mechanism. However, this approach can in some cases lead to higher delay costs for certain airlines compared to the FPFS rule, an outcome understandably opposed by the affected airlines. This raises the issue equity, often synonymous with fairness, which is a crucial concept in the ATFM context, where airlines compete for limited resources [17,18]. Attempts to mathematically define equity include Jacquillat and Vaze [19], who proposed distributing delays among airlines based on their number of flights, and Hamdan et al. [20], who introduced penalties to limit flight disruptions in new schedules. Despite these efforts, defining a universally accepted concept of equity remains challenging among stakeholders. Nonetheless, Ruiz et al. [9] highlighted that any equitable delay handling mechanism should ensure that no airline experiences greater costs than with the initial FPFS assignment.

The BEACON project [21] further highlighted that prioritisation mechanisms tend to benefit larger airlines with more flexibility, potentially leaving smaller carriers at a disadvantage [5]. This underscores the need for mechanisms that not only improve efficiency but also ensure a more balanced distribution of benefits across airspace users.

In this respect, the UDPP [22] represents an important step forward, designing a new reschedule mechanism that promotes the active role of airlines when hotspots arise at airports, enabling them to rearrange flights to better suit their business priorities. Periodically validated through human-in-the-loop simulations and received positively by airlines [8], UDPP has garnered significant research attention, prompting numerous efforts to extend and refine its application.

Zhang and Meilong Xu [23] proposed an enhanced UDPP model incorporating selective flight protection, which successfully reduced delay costs. Most recently, Gasparin et al. [10] formalised the mechanism with a dedicated mathematical and optimisation model. A recent study by Zhao et al. [24] further validates the effectiveness of prioritisation processes, demonstrating that, if applied network-wide, they can potentially reduce ATFM delay costs by up to 44.1%.

Yet, the UDPP does not include any inter-airline slot swap feature, limiting its beneficial effects on those users that have enough flexibility (number of flights in a hotspot), since they can choose between several options for their fleet rescheduling [10]. Instead, airlines with few flights in the hotspot have limited options for internal rescheduling and

therefore cannot substantially reduce their costs. Hence the necessity for the development of new slot allocation mechanisms, which, in addition to the intra-airline flight rearrangement, include the possibility of inter-airline slot trading.

Several alternative approaches have been explored in the literature to reduce delay costs through inter-airline coordination mechanisms. These include auction-based frameworks [3,25], credit mechanisms [26], and other market-based approaches [4]. While these mechanisms can, under suitable assumptions, achieve economically efficient allocations, they rely on fundamentally different modeling premises compared to our proposal, particularly regarding airline participation, information disclosure, and operational constraints. In auction-based and multi-agent settings performance is highly dependent on the bidding strategies adopted by airlines. Although such frameworks are incentive-compatible in theory, their practical effectiveness may be affected by bounded rationality, incomplete information, and strategic deviations, leading to outcomes that are sensitive to how airlines choose to bid (e.g., cooperatively, competitively, or strategically). In contrast, the mechanism proposed in this paper removes the need for explicit strategic bidding by providing airlines with precomputed, ready-to-evaluate offers, thereby reducing behavioural uncertainty and simplifying participation. Similarly, credit-based mechanisms introduce virtual currencies to enhance flexibility, particularly for low-volume airspace users. However, their benefits materialise over longer time horizons, as airlines must accumulate and manage credits across multiple hotspot events. As a result, their effectiveness depends on inter-temporal strategies and credit management policies, whereas our approach focuses on immediate, instance-based cost reductions, closer to operational decision-making contexts.

As highlighted in the BEACON project [27], evaluating and comparing these mechanisms requires careful consideration of stakeholder priorities, operational constraints, and implementation feasibility. No single approach simultaneously satisfies all stakeholder expectations, and identifying practically viable solutions remains an open challenge.

From a methodological perspective, the mechanism proposed in this paper introduces significant combinatorial optimisation challenges, which we address through a hybrid approach combining mathematical optimisation and machine learning (ML). Data-driven methods have recently gained increasing attention in air traffic management, complementing more traditional physics-based modeling approaches grounded in airspace dynamics and traffic flow theory. Recent surveys [28] highlight successful applications of ML in traffic forecasting [29], trajectory prediction and management [30], and airline and passenger-centric operations [31,32]. Within this broader context, our work aligns with the emerging stream of hybrid optimisation methods, where ML is used to enhance or guide classical solution approaches. Such integration has proven particularly effective for complex combinatorial problems, as documented in recent surveys [33,34]. In line with these advances, we propose a hybrid mechanism that includes ML as a heuristic component for slot trade selection within an optimisation framework. Our computational results further support the effectiveness of this paradigm, demonstrating that hybrid approaches can achieve high-quality solutions while maintaining computational tractability in complex ATM environments.

3. Operational requirements

Before presenting the details of our proposed mechanism, it is essential to establish the principles that guide its design. Any slot reallocation framework intended for operational deployment must not only comply with existing constraints but also address the needs and expectations of the stakeholders involved, primarily airlines and the NM. Over the years, feedback collected from various stakeholder consultation initiatives [5–7] has consistently pointed to a set of key requirements that such a mechanism should satisfy. These range from ensuring that no airline is disadvantaged by the reallocation, to promoting equity in

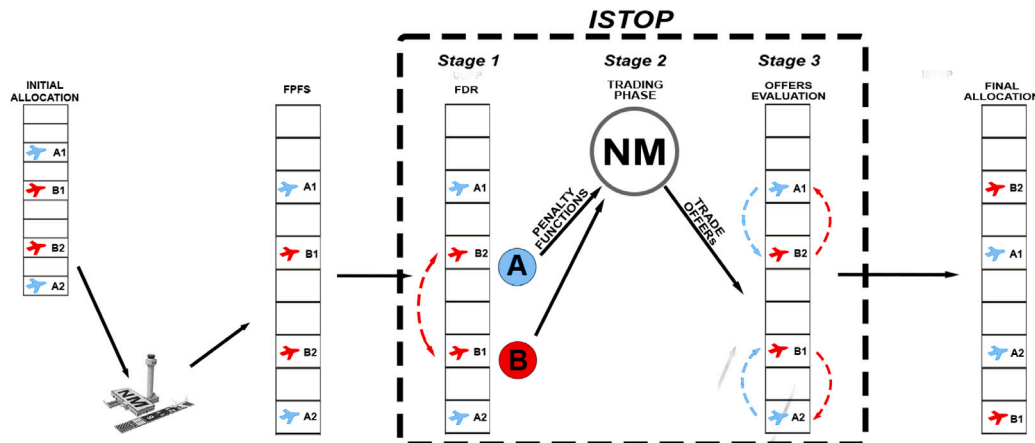


Fig. 2. Flow of the ISTOP mechanism. Starting from the FPFS allocation, the FDR phase (Stage 1) allows airlines to reshuffle the order of their flights within their own slots. Subsequently, in the trading phase (Stage 2), the Network Manager computes the set of trades that minimise the overall penalty and sends the corresponding offers to the involved airlines. Finally, in the third stage, airlines evaluate the received offers, and all trades for which every involved airline accepts its offer are implemented in the final allocation.

the distribution of benefits, to guaranteeing practical usability under tactical time constraints. In this section, we examine each of these requirements in detail, discussing their rationale and implications for the design of ISTOP.

R1 No airline has to be penalised from the new reallocation.

In the context of UDPP development [35], this principle emerged as a widely accepted common ground among airlines following several consultation meetings with carriers. From these discussions, it has been established that: “allowing flexibility to all AUs is considered acceptable only if this has no negative impact on other AUs’ flights. Equity [fairness] means that for each individual flight not participating in UDPP, there is no increase of delay” [9]. In other words, while airlines may benefit from participating in the mechanism, those who choose not to participate, or are unable to, should not see their situation worsen as a consequence of others’ rearrangements.

R2 Any resulting delay cost reduction should be equitably distributed among the airlines involved.

Although several attempts have been made to mathematically define equity in the ATFM context [17–20], the concept remains inherently fuzzy and subject to different interpretations.

The SESAR-funded BEACON project [21] investigated mechanisms for inter-airline slot exchange, with particular attention to equity issues. The project, via computational experiments, real-time and human-in-the-loop simulations, and comparative analyses, investigated how different airlines are affected by different mechanisms, exploring the impact on low-volume airspace users specifically.

Within this context, stakeholders feedback has shed light on what equity means in practice: concerns have been raised that “big airlines get a large advantage because of the number of slots they can trade, and small airlines do not. In this case, airspace users may not be happy with the implementation of a mechanism which will give an advantage to their competitors” [5]. To avoid such a scenario, any acceptable mechanism should ensure that the benefits it generates involve as many actors as possible and are distributed in a balanced manner, preventing the emergence of systematic advantages for larger carriers.

R3 Airlines must be involved in the decision process.

Airlines are more likely to favour Collaborative Decision Making (CDM) frameworks, as they generally wish to be directly involved in decisions that affect their operations. This concept is well established in the air traffic management (as for example 11,36–38) and has become a reference principle in the development of new operational procedures [39]. Direct involvement ensures that airlines retain control over their resources and can align any reallocation with their business priorities, rather than having solutions imposed upon them by a central authority.

R4 The framework must balance simplicity with a high level of automation.

The complexity of operational constraints, combined with the need for collaborative decision-making, imposes a dual requirement on any practical mechanism. On the one hand, it must be simple enough for airlines to understand and use without extensive training or additional workload. On the other hand, it must provide a high level of automation to be applicable in real-world scenarios, where the number of flights and potential interactions can be overwhelming for manual processing. Stakeholder consultations have consistently highlighted this need: users expect a tool that is both accessible and capable of handling complexity autonomously [5,7].

R5 Solutions must be sufficiently quick to compute.

This requirement is straightforward but crucial: since interventions on hotspots occur at the tactical level (typically with limited time available before decisions must be implemented) any proposed solution must be computationally efficient enough to deliver results within the required timeframe. A mechanism that produces high-quality allocations but takes hours to compute would be impractical for operational use, where decisions often need to be made in a matter of minutes.

The objective of the ISTOP mechanism presented in the following section is to address all of the above requirements simultaneously, providing a framework that is both practical and aligned with stakeholder needs.

4. The ISTOP mechanism

This section describes the proposed ISTOP mechanism. ISTOP begins from the baseline allocation provided by the FPFS and aims to reduce airlines’ delay costs while addressing the stakeholder requirements outlined in Section 1. As shown in Fig. 2, the mechanism consists

of three stages. Below, we first provide an overview of its workflow, then analyse each stage in detail.

Starting from the FPFS allocation, the FDR stage allows airlines to reshuffle the order of their flights within their own slots. Subsequently, in the trading phase, the Network Manager computes the set of trades that minimise the overall penalty and sends the corresponding offers to the involved airlines. Finally, in the third stage, airlines evaluate the received offers, and all trades for which every involved airline accepts its offer are implemented in the final allocation.

1. **FDR.** In the first stage, each airline may reorder its own flights among its available slots. This preliminary step enables an airline to minimise its individual costs through internal reassignment without affecting other carriers, thereby improving its position before any inter-airline trading begins. This procedure, introduced in Pilon et al. [22], is referred to as *Flights' Delay Reordering* (FDR). In Fig. 2, for example, airline *B* swaps flights *B1* and *B2*.
2. **Trading phase.** In this stage, each airline submits to the Network Manager (NM) an encrypted surrogate of its flight-delay cost estimation functions (hereafter called *penalties*), which the airlines are assumed to know. Rescaling the penalties preserves cost confidentiality while supplying the NM with the information required to identify beneficial slot trades. Given the penalties, the NM chooses among all feasible trades those that minimise the aggregate penalty, i.e., the sum of the penalties resulting from the new allocation. Each selected trade stems from matching offers submitted by multiple airlines. For instance, in Fig. 2 the trade involving flights *A1*, *A2*, *B1*, *B2* corresponds to matching airline *A*'s offer (*A1* → *B1*, *A2* → *B2*) with airline *B*'s offer (*B1* → *A1*, *B2* → *A2*). The NM then returns the selected offers to the airlines for their consideration.
3. **Offers evaluation by airlines.** For each proposed trade, if all parties accept their respective offers the slot exchanges are executed and recorded in the final allocation (as in the example from Fig. 2). If any party rejects an offer associated with a proposed trade, then all flights involved in that trade retain the slots assigned after the FDR.

4.1. FDR

As noted above, the initial FDR stage provides an immediate reduction in delay costs for each airline while positioning them as favourably as possible before any inter-airline slot trading takes place. The reordering is performed independently by each airline, with the objective of identifying the flight sequence that minimises its own delay costs. This task can be formulated as an integer programming problem through a straightforward adaptation of the MINCOST model [3], in which each airline considers only its own flights and slots to compute the optimal allocation.

4.2. Trading phase

The trading phase computes an optimal set of slot exchanges through a centralised process managed by the NM. To this end, airlines must first provide the NM with the information required to assess the feasibility of trades without revealing confidential cost data. We first describe the mechanism that enables this exchange of secure, surrogate information. We then present the integer programming formulation underlying trade selection and show how the problem can be expressed as a MWIS.

4.2.1. Cost estimation and penalty functions

Estimating flight delay costs is itself a complex task, and extensive research has been devoted to the topic. The relationship between the delay of a flight *i* and the corresponding additional cost—its delay cost function $C_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ —is generally non-linear [40] and depends on factors such as aircraft type, passenger loads, passenger connections, crew and maintenance requirements, and other relevant variables [41]. Fortunately, Cook and Tanner [42] and Cook and Tanner [43] provide a detailed methodology for constructing a mathematical function that, based on these parameters, assigns each delay *d* of flight *i* a cost value $C_i(d)$.

The cost function plays a central role in this work. According to R1, an airline *A* will accept a slot swap only if it leads to a net reduction in delay costs. The cost function therefore allows each airline to compare delay costs before and after a proposed swap. For convenience, let flights be indexed by the index of their initial slot. An offer *o* for airline *A* is then defined as a list of pairs $((i_1, j_1), (i_2, j_2), \dots)$, where each pair (i_k, j_k) indicates that flight i_k (of *A*) is proposed to move to slot j_k . Let d_{ij} denote the delay incurred by flight *i* if assigned to slot *j*.

Airline *A* evaluates whether offer *o* is beneficial by verifying:

$$\sum_{(i,j) \in o} C_i(d_{ij}) < \sum_{i \in o_f} C_i(d_{ii}), \quad (1)$$

where o_f denotes the set of flights involved in offer *o*. This inequality states that the total delay cost under the proposed reassignment (left-hand side) must be lower than the total delay cost of the current assignment (right-hand side), where d_{ii} is the delay of flight *i* in its current slot.

To enable a centralised mechanism in which the NM acts as a broker, the NM must be able to verify condition (1) itself. However, if the NM were to use the true cost functions directly for trade selection, with the objective of maximising absolute cost reduction, this would disproportionately favour airlines with higher baseline costs, as trades producing large absolute savings would dominate the objective.

Moreover, maximising absolute cost reduction could incentivise unfair behaviour: airlines would have an advantage in inflating their reported costs, making the mechanism dysfunctional. To address this issue and promote a more equitable distribution of potential benefits (in line with requirement R2), ISTOP normalises costs by airline. These normalised cost functions are referred to as penalty functions and are formalised as follows:

Definition 1. The **penalty function** $p_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ of a flight *i* operated by airline *A* and involved in a hotspot composed of the set *S* of ATFM slots is defined as

$$p_i(d_{ij}) := \frac{C_i(d_{ij})}{c_{\max}^A}, \quad (2)$$

where

$$c_{\max}^A := \max_{i \in A, j \in S} C_i(d_{ij}) \quad (3)$$

is the maximum value taken by any cost function among the flights of airline *A* within the hotspot.

First, note that the penalty function is a simple linear transformation of the cost function. This allows the NM to still evaluate whether a trade is beneficial for each airline *A*, as condition (1) remains equivalent under this transformation:

$$\begin{aligned} \sum_{(i,j) \in o} C_i(d_{ij}) &\leq \sum_{i \in o_f} C_i(d_{ii}) \iff \sum_{(i,j) \in o} p_i(d_{ij}) = \frac{\sum_{(i,j) \in o} C_i(d_{ij})}{c_{\max}^A} \\ &\leq \frac{\sum_{i \in o_f} C_i(d_{ii})}{c_{\max}^A} = \sum_{i \in o_f} p_i(d_{ii}). \end{aligned} \quad (4)$$

By construction, each penalty satisfies $p_i(d_{ij}) \in [0, 1]$, and for each airline *A* there exists at least one flight-slot pair for which $p_i(d_{ij}) = 1$.

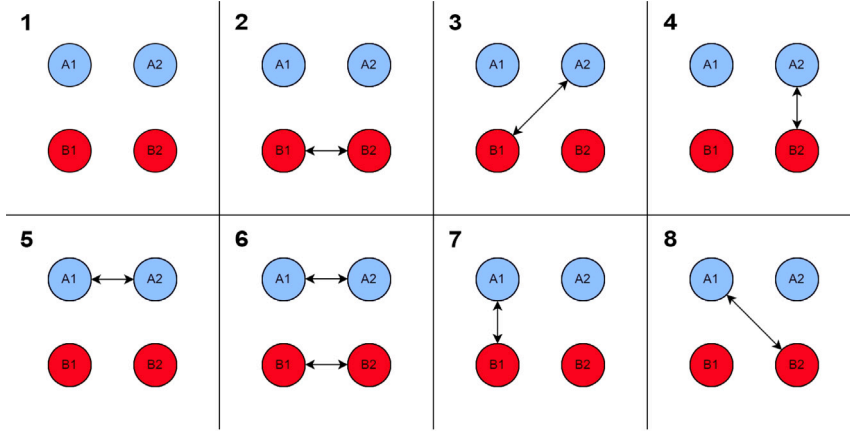


Fig. 3. Representation of the invalid permutations (slot swaps). 1 is the identity, which does not alter any penalty. 2, 5, 6 represent intra-airline swaps. 3, 4, 7 and 8 correspond to swaps between a single flight of A and a single flight of B. Since these inter-airline permutations only switch the order of two flights, they yield a delay reduction for one airline at the expense of an equivalent delay increase for the other.

Thus, penalty functions cap the penalty value to one, regardless of the absolute magnitude of the maximum cost.

This provides clear equity advantages: the cost scale is the same for all airlines, and cost reduction depends only on the relative difference between the costs of two flights operated by the same airline, rather than on absolute cost values. Furthermore, there is no incentive for airlines to inflate their costs, as any inflation would be cancelled out by the normalisation.

It is worth noting that the penalty functions submitted by airlines provide only a transformed version of their true cost structures. While absolute cost values are masked, some level of reverse engineering could still potentially expose sensitive information. Therefore, more robust privacy-preserving mechanisms, such as those recently explored in the literature [44,45], should be considered in future developments of the framework, although they fall beyond the scope of this paper.

4.2.2. Trade computation and selection

Since hotspots are typically predicted only a few hours in advance, decisions must be taken at the tactical level. In this context, a decentralised framework in which airlines negotiate offers directly with one another would lead to negotiation times that are impractically long. To address this issue, the ISTOP slot trading mechanism assigns the NM the role of a central “broker”. Based on the penalty functions provided by the airlines, the NM computes the set of trades that maximises the overall reduction in penalties, defined as the difference between the sum of flight penalties before and after executing the slot swaps associated with the selected trades. To ensure that ISTOP remains both practical and effective, we impose two restrictions on the trade selection process. First, in line with the automation and simplicity requirement R4, we consider only offers involving pairs of flights, so that they are straightforward for airlines to evaluate. Second, we restrict trades to involve at most two or three airlines. When many airlines participate in a single trade, the rejection of any individual offer would invalidate the entire transaction, causing all potential benefits for the remaining participants to be lost. Under these constraints, the identification of the set of trades that minimises the overall penalty is carried out in two phases: *trade computation* and *trade selection*.

Before detailing each phase, we introduce the following notation.

- $\mathbb{A}$, the set of airlines;
- $\mathcal{P}_a$, the set of pairs of flights of airline a ;
- $\mathbb{P}$, the set of pairs of airlines;
- $\mathbb{A}_T$, the set of triples of airlines;
- $N := \{(\rho_{a_1}, \rho_{a_2}) \mid (\rho_{a_1}, \rho_{a_2}) \in \mathcal{P}_{a_1} \times \mathcal{P}_{a_2}, (a_1, a_2) \in \mathbb{P}\}$ the set of all combinations of pairs of flights of two different airlines;

- $M := \{(\rho_{a_1}, \rho_{a_2}, \rho_{a_3}) \mid (\rho_{a_1}, \rho_{a_2}, \rho_{a_3}) \in \mathcal{P}_{a_1} \times \mathcal{P}_{a_2} \times \mathcal{P}_{a_3}, (a_1, a_2, a_3) \in \mathbb{A}_T\}$ the set of all combinations of pairs of flights of three different airlines;
- $\mathbb{T} := N \cup M$, the set of all possible trades
- $\mathcal{T} \subseteq \mathbb{T}$ the set of all feasible trades;
- $ETA(i)$ the function that returns the index of the slot corresponding to the ETA of flight i .

Trade computation. In the trade computation phase, we construct the set $\mathcal{T}$ of all *feasible* trades, defined as those composed exclusively of offers that are convenient for all airlines involved. Let us focus on the elements of N (an analogous argument applies to those of M). For a given $n \in N$, determining whether a trade exists (i.e., a reassignment of slots among the flights in n) that is beneficial for all parties requires examining all permutations of flights that correspond to inter-airline slot exchanges. Only those permutations that exclude intra-airline swaps and yield a reduction in penalty for every participating airline are considered feasible (Fig. 3 illustrates the 8 out of 24 permutations that can be excluded *a priori*).

Let Σ_4 denote the set of permutations of four flights that yield a valid inter-airline slot swap involving two airlines. Valid permutations are all those except: (i) the identity; (ii) permutations involving only a two-flight swap; and (iii) permutations corresponding to intra-airline swaps only (see Fig. 3 for examples of invalid cases). The elements being permuted are the indices of the flights, which indicate the slots to which they are currently assigned. For example, if we were given the list of flight slot indices $s = [11, 17, 34, 41]$ and the permutation $\sigma = (2, 4, 1, 3)$, the new slot assignment would be obtained applying σ to s , yielding $\sigma(11) = 17$, $\sigma(17) = 41$, $\sigma(34) = 11$, and $\sigma(41) = 34$.

Finally, we define $\mathcal{R}$ as the set of penalty reductions associated with each feasible trade. The procedure used to compute $\mathcal{T}$ and $\mathcal{R}$ is described in Algorithm 1, referred to as *TwoAirTradeComputation*. The algorithm first computes, for each $n \in N$, the initial total penalty (line 5), defined as the sum of the penalties incurred by all flights under their current delays, as well as the total penalty for each airline whose flights are included in n (lines 6 and 7). It then iterates over Σ_4 , evaluating all valid slot swaps by checking whether the reassigned slots are compatible with the flights’ ETAs (line 10) and whether they yield a penalty reduction for all involved airlines (line 13). If these conditions are satisfied for at least one permutation σ , the trade n is added to $\mathcal{T}$, and the maximum total penalty reduction achievable for that trade is recorded in $\mathcal{R}$ (lines 15 and 16).

To obtain the final sets $\mathcal{T}$ and $\mathcal{R}$, an analogous procedure—referred to as *ThreeAirTradeComputation*—is applied to all elements of M , this time considering trades involving three airlines rather than two. The

Algorithm 1 TwoAirTradeComputation

```

1: for  $n \in N$  do
2:    $a_1, a_2$  = airlines which pair of flights are in  $n$ 
3:    $f_i, f_j$  = flights of the pair  $\rho_{a_1}$  of  $n$ 
4:    $f_k, f_l$  = flights of the pair  $\rho_{a_2}$  of  $n$ 
5:    $P_{init} = p_i(d_{ii}) + p_j(d_{jj}) + p_k(d_{kk}) + p_l(d_{ll})$ , initial total penalty of
     flights in  $n$ 
6:    $P_{a_1} = p_i(d_{ii}) + p_j(d_{jj})$ , initial penalty of airline  $a_1$ 
7:    $P_{a_2} = p_k(d_{kk}) + p_l(d_{ll})$ , initial penalty of airline  $a_2$ 
8:    $r_n = 0$ , best penalty reduction obtained swapping flights in  $n$ 
9:   for  $\sigma \in \Sigma_4$  do
10:    if  $\sigma(i) \leq ETA(i)$  and  $\sigma(j) \leq ETA(j)$  and  $\sigma(k) \leq ETA(k)$  and
       $\sigma(l) \leq ETA(l)$  then
11:       $P_{a_1}^* = p_i(d_{i\sigma(i)}) + p_j(d_{j\sigma(j)})$ , penalty of airline  $a_1$  after the
      swap
12:       $P_{a_2}^* = p_k(d_{k\sigma(k)}) + p_l(d_{l\sigma(l)})$ , penalty of airline  $a_2$  after the
      swap
13:      if  $P_{a_1} - P_{a_1}^* > 0$  and  $P_{a_2} - P_{a_2}^* > 0$  then
14:         $r_n^* = P_{init} - P_{a_1}^* - P_{a_2}^*$ 
15:        if  $r_n^* > r_n$  then
16:           $r_n = r_n^*$ 
17:    if  $r_n > 0$  then
18:      Append  $n$  to  $\mathcal{T}$  and append  $r_n$  to  $\mathcal{R}$ 
19: return  $\mathcal{T}, \mathcal{R}$ 

```

main difference lies in replacing Σ_4 with Σ_6 : the set of permutations of six elements that yield a valid inter-airline slot swap involving three airlines. Following the same reasoning adopted for the two-airline case, a permutation is considered invalid—and thus excluded from Σ_6 —if it corresponds to any of the following cases: (i) the identity; (ii) permutations that result in a swap involving only two flights (i.e., only one pair of flights exchanges slots, while the others remain in place); (iii) permutations that correspond to swaps occurring entirely within a single airline (i.e., no inter-airline exchange takes place); (iv) permutations that involve only two airlines, even if all their flights are swapped (i.e., the third airline's flights are unaffected).

The remaining 584 valid permutations constitute Σ_6 , representing all possible ways to reallocate slots across three airlines such that at least one inter-airline swap occurs and all three airlines are involved in the exchange. Algorithm 2 summarises the overall flow of the trade computation phase.

Algorithm 2 Trade computation

```

1:  $\mathcal{T}_2, \mathcal{R}_2 = \text{TwoAirTradeComputation}()$ 
2:  $\mathcal{T}_3, \mathcal{R}_3 = \text{ThreeAirTradeComputation}()$ 
3:  $\mathcal{T} = \mathcal{T}_2 \cup \mathcal{T}_3$ 
4:  $\mathcal{R} = \mathcal{R}_2 \cup \mathcal{R}_3$ 
5: return  $\mathcal{T}, \mathcal{R}$ 

```

Trade selection. Once the trade computation phase is completed, the trade selection phase is performed, which consists of identifying the subset of $\mathcal{T}$ that maximises the overall penalty reduction. This problem can be formulated as a *Maximum Weighted Independent Set* (MWIS) problem [46], since the selected trades must not involve any common flights, as described in 4.2.2. Otherwise, at least one flight would be assigned to multiple slots, resulting in an infeasible schedule.

To cast the trade selection problem as a MWIS, each trade t is represented as a vertex x_t in a graph $G = (V, E)$, where an edge between two vertices indicates that the corresponding trades share at least one flight. The weight r_t of vertex x_t is defined as the maximum penalty reduction achievable by trade t . The trade selection problem can then

be formulated as:

$$\begin{aligned}
 \max \quad & \sum_{t \in \mathcal{T}} r_t x_t \\
 \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall (i, j) \in E \\
 & x_t \in \{0, 1\}, \quad j = 1, \dots, |\mathcal{T}|,
 \end{aligned} \tag{5}$$

This formulation coincides with the standard definition of the MWIS problem. The objective function represents the overall penalty reduction, while the constraints ensure that if two trades share at least one flight, at most one of them can be selected. The MWIS problem has been extensively studied, and numerous exact solution approaches are available [47]. However, the present application exhibits two specific characteristics that require a tailored solution strategy.

The first is a favourable structural property: in a hotspot with n flights, the number of trades in any feasible solution cannot exceed $\lfloor n/4 \rfloor$, since each trade involves at least four flights and each flight can participate in at most one trade. This bound enables the development of an efficient, problem-specific Branch&Bound algorithm, which exploits trade ordering and a compatibility matrix to effectively prune the search space (the complete algorithm is reported in Appendix). Computational experiments show that this approach outperforms a general-purpose solver (Gurobi) on our instances, both in terms of computational time and memory usage.

The second, and more critical, characteristic is that the number of potential offers grows exponentially with the number of flights, making direct optimisation intractable for large hotspots. This intrinsic scalability issue provides the main motivation for the heuristic framework introduced in Section 5.

4.3. Offer evaluation by airlines

The offer evaluation stage is essential to the practical applicability of the mechanism, as optimal trades are computed using penalty functions derived from deterministic estimates of the true delay cost functions. These estimates do not account for the fact that delay costs are often subject to significant uncertainty [48]. Providing a rigorous mathematical representation of this uncertainty remains a challenging research problem [49]. As a result, airlines typically rely on heuristic assessments that consider the overall network situation and draw on the experience of flight dispatchers to support decision making.

Fig. 4 illustrates this issue. Airline A is offered a trade that reduces the delay of one of its flights (shifting from the 9:20 to the 9:05 slot) at the expense of an increased delay for another flight (shifting from the 9:30 to the 9:50 slot). Eq. (4) ensures that the NM selects this offer because, according to the penalty functions provided by airline A, the resulting schedule yields a net benefit. However, examining the cost function of the 9:30 flight (thick blue line in Fig. 4) reveals a sharp increase corresponding to a delay threshold beyond which some passengers miss their connections, leading to additional compensation costs for the airline. The proposed new slot for 9:30 flight lies before this threshold, and the associated delay cost is therefore limited.

However, if passenger connection times are uncertain—due, for example, to gate congestion or disruptions in baggage transfer—the 9:50 slot might still not guarantee passengers to catch their connection. Consequently, the estimated cost function, and hence the penalty function, may be inaccurate. This uncertainty highlights the importance of allowing airlines to evaluate offers before committing to schedule changes. In the example of Fig. 4, if airline A anticipates that passenger connection time could potentially increase by up to 10 min around the nominal value, effectively shifting the cost function threshold backward by 10 min (dashed lines), the decision to accept or reject the offer ultimately rests on the dispatcher's experience and risk assessment. This example illustrates how the possibility to refuse an offer, even when initial estimates suggest a potential benefit, is essential for the effectiveness of the mechanism. By leaving the final decision to the airlines, the framework fully respects the Collaborative Decision Making principle R3, ensuring that carriers remain in control of their operations and can adapt to uncertain or evolving circumstances.

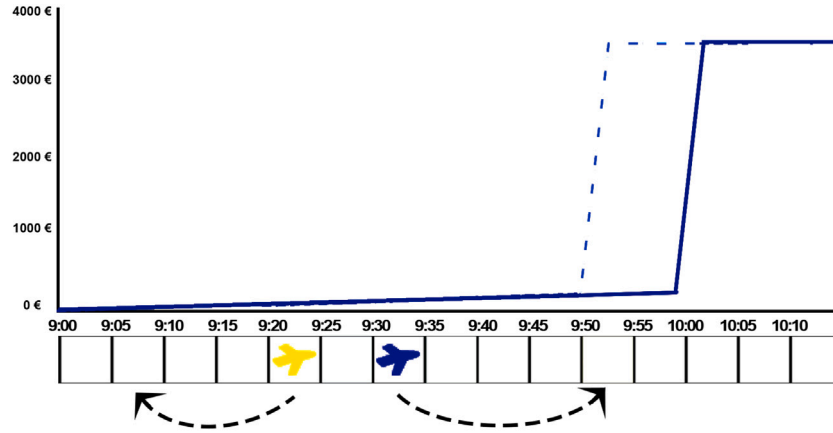


Fig. 4. Impact of cost uncertainty on the evaluation of a trading offer. A trade is proposed that prioritises the 9:20 flight (yellow) at the expense of the 9:50 flight (blue). The blue flight's actual cost function (thick line) shows a sharp increase beyond a certain delay, which may correspond to missed passenger connections. The dashed line illustrates the uncertainty in this cost estimate (e.g., due to variable connection times). Such uncertainty must be taken into account by the airline before committing to the offer and accept the swap.

5. Heuristic

As explained in Section 4.2.2, selecting the optimal set of trades reduces to solving a MWIS problem, for which several efficient algorithms have been proposed in the literature. However, as the number of variables grows, exact methods increasingly struggle to deliver solutions within acceptable computation times. This limitation is particularly critical at the tactical level, where decisions must be taken promptly, making the development of efficient heuristic approaches essential.

An analysis of European traffic in 2019 shows that hotspots typically involve, on average, 85 flights, with a standard deviation of approximately 80. During the preliminary phase of our experimental study, we observed that instances with up to 120 flights (corresponding to about 75% of the hotspots in our real-world dataset) can generally be solved to optimality within a few seconds. Nevertheless, in some instances with flights ranging between 80 and 120, and in most cases exceeding the 120-flight threshold, solving the problem to optimality becomes computationally intractable. This difficulty stems from the exponential growth in the number of possible trades $|\mathbb{T}|$. Denoting by γ_a the number of flights operated by airline $a \in A$, we obtain:

$$|N| = \sum_{i,j \in \mathbb{A}_P} |\mathcal{P}_i| \cdot |\mathcal{P}_j| = \sum_{i,j \in \mathbb{A}_P} \binom{\gamma_i}{2} \cdot \binom{\gamma_j}{2}, \quad (6)$$

and similarly:

$$|M| = \sum_{i,j,k \in \mathbb{A}_T} |\mathcal{P}_i| \cdot |\mathcal{P}_j| \cdot |\mathcal{P}_k| = \sum_{i,j,k \in \mathbb{A}_T} \binom{\gamma_i}{2} \cdot \binom{\gamma_j}{2} \cdot \binom{\gamma_k}{2}, \quad (7)$$

thus, resulting in a total number of possible combinations which amounts to

$$|\mathbb{T}| = |N| + |M|. \quad (8)$$

To provide some perspective, consider a hotspot involving six airlines, each operating 25 flights (150 flights in total). In this case, the number of possible trades amounts to 162,540,000. Such a large trade set makes Algorithm 2 impractical, as it would require excessive computation time.

However, our analysis of generated data (see Section 6.1 for details) shows that nearly 40% of the trades in these large instances are typically discarded because they do not yield benefits for all parties involved. Moreover, among the remaining feasible trades, a substantial fraction provides only negligible penalty reductions. Fig. 5(a) illustrates this phenomenon by showing the distribution of trades according to their penalty reduction. We observe that approximately 44.7% of the

feasible trades (corresponding to 27.2% of $\mathbb{T}$) yield a penalty reduction below 0.1.

To reduce the computational burden of Algorithm 2, rather than considering the entire sets $\mathcal{P}_a$ to compute the feasible trades $\mathcal{T}$, we restrict the search to a subset $\hat{\mathcal{P}}_a \subset \mathcal{P}_a$ for each airline, containing only flight pairs that are most likely to participate in trades yielding a significant penalty reduction.

The procedure, described in Algorithm 3 and illustrated in Fig. 6, can be summarised as follows: we first define a maximum number of trades B (or budget) to be considered in the optimisation. For each airline a and each pair of flights $\rho^a \in \mathcal{P}_a$, we compute an estimate $\hat{r}_{\rho^a}$ of the maximum reduction r_{ρ^a} achievable by any trade including ρ^a .

Next, for each airline a , we construct the set $\hat{\mathcal{P}}_a := \{\rho^a \in \mathcal{P}_a \mid \hat{r}_{\rho^a} > 0\}$, which contains only flight pairs estimated to be part of feasible trades. The elements of each $\hat{\mathcal{P}}_a$ are then sorted in descending order of their estimated reduction values $\hat{r}_{\rho^a}$.

Finally, to effectively reduce the search space, we iteratively populate the subsets $\hat{\mathcal{P}}_a$ by adding elements from $\hat{\mathcal{P}}_a$ one at a time until the total number of offers, calculated using Eq. (8), reaches the predefined budget B .

Algorithm 3 Pair selection

```

1:  $i = 1$ 
2: initialise  $\bar{\mathcal{P}}_a = \{\}$  for all  $a \in \mathbb{A}$ 
3:  $stop = false$ 
4: while not  $stop$  do
5:    $A = \{a \in \mathbb{A} : |\hat{\mathcal{P}}_a| \leq i\}$ 
6:   sort  $A$  in descending order according to the elements  $\hat{\rho}_i^a$  of each set  $\hat{\mathcal{P}}_a$ 
7:   for  $a \in \mathbb{A}$  and if not  $stop$  do
8:     add  $\rho_i^a \in \hat{\mathcal{P}}_a$  to  $\bar{\mathcal{P}}_a$ 
9:     if  $\sum_{i,j \in \mathbb{A}_P} |\bar{\mathcal{P}}_i| \cdot |\bar{\mathcal{P}}_j| + \sum_{i,j,k \in \mathbb{A}_T} |\bar{\mathcal{P}}_i| \cdot |\bar{\mathcal{P}}_j| \cdot |\bar{\mathcal{P}}_k| > B$  then
10:       remove  $\rho_i^a$  from  $\hat{\mathcal{P}}_a$ 
11:        $stop = true$ 
12:     else
13:        $i += 1$ 

```

In more detail, at each iteration i we first identify the list of airlines A for which $\hat{\mathcal{P}}_a$ contains at least i pairs. We then sort A in descending order according to the estimated reduction of the i th pair ρ_i^a of each $\hat{\mathcal{P}}_a$ (recall that each $\hat{\mathcal{P}}_a$ has already been sorted by estimated reduction).

For each airline $a \in A$, we add ρ_i^a to $\bar{\mathcal{P}}_a$, checking at each step that the addition does not exceed the budget B , i.e.:

$$\sum_{i,j \in \mathbb{A}_P} |\bar{\mathcal{P}}_i| \cdot |\bar{\mathcal{P}}_j| + \sum_{i,j,k \in \mathbb{A}_T} |\bar{\mathcal{P}}_i| \cdot |\bar{\mathcal{P}}_j| \cdot |\bar{\mathcal{P}}_k| \leq B. \quad (9)$$

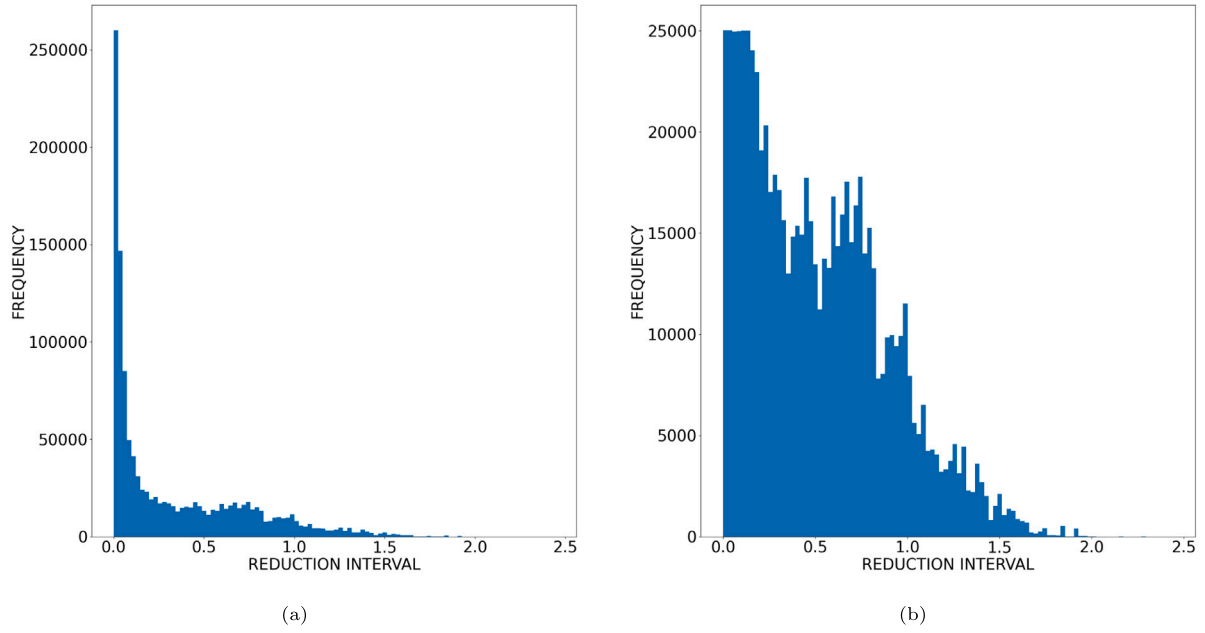


Fig. 5. (a) Histogram of the feasible trades frequency, grouped by penalty reduction. (b) Histogram of the feasible trades frequency, grouped by penalty reduction after the balancing pre-process.

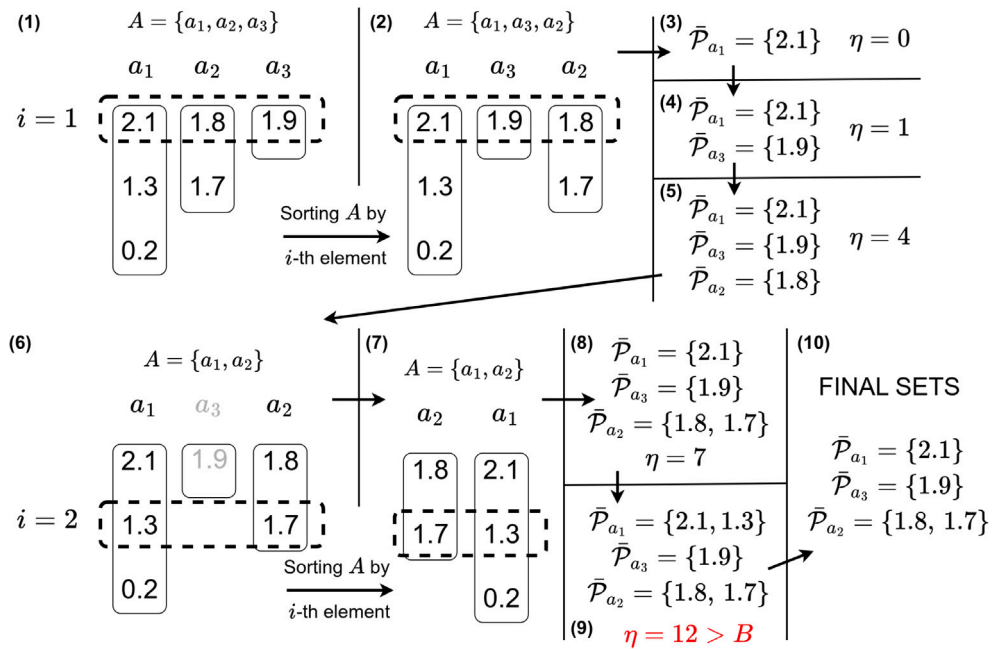


Fig. 6. Example of application of Algorithm 3 in a case with three airlines, a_1, a_2, a_3 , and $B = 10$. Here, the pairs ρ_i^a are represented by their estimated value $\hat{\rho}_i^a$, and the number of possible offers $\sum_{t,i,j \in \mathcal{A}_B} |\bar{\mathcal{P}}_t| \cdot |\bar{\mathcal{P}}_j| + \sum_{t,j,k \in \mathcal{A}_B} |\bar{\mathcal{P}}_t| \cdot |\bar{\mathcal{P}}_j| \cdot |\bar{\mathcal{P}}_k|$ is denoted by η . (1) The pairs with the highest estimated penalties for each airline are considered and (2) sorted in descending order. (3) The first pair is added to the corresponding airline pair set, in this case a_1 . The number of possible combinations is initially $\eta = 0$, so the algorithm continues. (4) Next, the second pair is added to its respective set—in this case, that of a_3 ; one combination becomes possible, and the algorithm proceeds with the third pair (5). Since the budget has not yet been exceeded ($\eta = 4$), the algorithm continues. Once the best pairs of each airline have all been added, the algorithm moves on to consider the second-best pairs of each airline (6). (7) As before, the pairs are reordered by penalty value, and the highest-valued pair is added to its respective set—in this case, that of a_2 . Since the number of combinations remains $\eta = 7 < 10 = B$, the algorithm proceeds. (8) However, adding the next pair would increase the number of combinations to 12, exceeding the budget. Therefore, the addition of this last pair is not finalised, and the algorithm stops, providing the final sets for each airline (10).

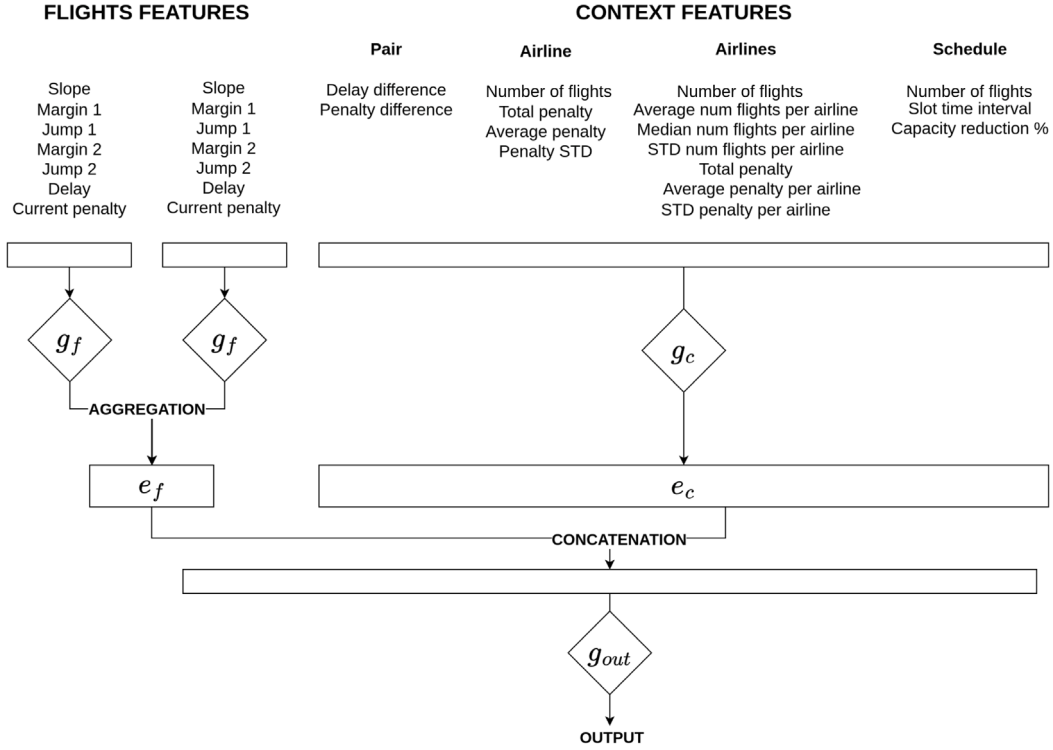


Fig. 7. Sketch of both g^1 and g^2 neural networks architecture.

If adding ρ_i^a would violate this condition, the pair is removed from $\bar{P}_a$ and the algorithm terminates; otherwise, the procedure continues with the next airline in A .

5.1. Estimation of penalty reduction

The method described above for restricting the search space can significantly reduce computational costs while maintaining strong performance. However, manually defining a rule to estimate r_{pa} proves extremely challenging. To address this, we exploit the large dataset available and adopt a two-stage neural network framework capable of efficiently performing this task.

The two-stage design is motivated by a key characteristic of the dataset: most trade pairs are infeasible. This imbalance makes it difficult for a single model to both filter out infeasible pairs and accurately predict reduction values for feasible ones. To address this, we train two neural networks, g^1 and g^2 . The first network, g^1 , performs an initial selection of flight pairs likely to be included in a feasible trade (classification task). The second network, g^2 , then performs a regression task, predicting the reduction value r_{pa} for each pair selected by g^1 .

Both networks share the same architecture, illustrated in Fig. 7, and are trained within a supervised learning framework. As input, they receive features describing the flights, providing the network with relevant information about each flight pair. In particular, each flight's penalty function is encoded using five parameters that define its step-wise linear interpolation (Fig. 8(b)). This representation is characterised by a constant slope and two jump discontinuities, each described by the delay at which it occurs (Margin 1 and Margin 2) and its magnitude (Jump 1 and Jump 2). Such a step-wise parametrisation of the cost function has been widely used in the literature [26,41], and the analysis of the common structure of the penalty functions observed in our dataset empirically confirmed its appropriateness (Fig. 8(a)).

Within the flight's features, the *Delay* and *Current penalty* features are included to provide the network with information on the current state of each flight in the pair. In addition, context features are provided to summarise characteristics of the hotspot. Among these, *Pair features*

capture the current delay difference and penalty difference between the two flights. Instead, the *Airline* features describe the situation of the involved airline while the *Airlines* features summarise the situation of all airlines. Lastly, the *Schedule* features include a set of potentially relevant statistics describing the current hotspot, such as the number of flights, the length of slots and capacity reduction.

The network architecture is designed to process the two flights and the context features separately. This design avoids the network being influenced by the order in which the flights are presented in the input vector. Each flight's feature vector is processed independently by the same feed-forward network g_f , and the resulting outputs are combined via a simple vector sum. This produces a single aggregated embedding e_f , which is then concatenated with the context embedding e_c obtained from a separate feed-forward network g_c . The concatenated embedding is subsequently processed by another feed-forward network g_{out} to generate the final output.

For the classification task, standard machine learning practice is followed: the output of g^1 is mapped to the interval (0,1) using a sigmoid function, and the network is trained with the cross-entropy loss.

For the regression task, the network g^2 is trained using a custom loss function designed to facilitate learning:

$$(1 + r_{pa})^2 \cdot (r_{pa} - \hat{r}_{pa})^2, \quad (10)$$

where r_p and $\hat{r}_p$ are the observed and predicted reduction values, respectively. In this loss function, the second term represents the standard squared error, while the first term serves as a weighting factor that penalises errors more heavily for pairs associated with higher penalty reductions.

6. Methodology

In this section, we present the methodology used to assess our work. The assessment focuses on two main aspects: computational and operational.

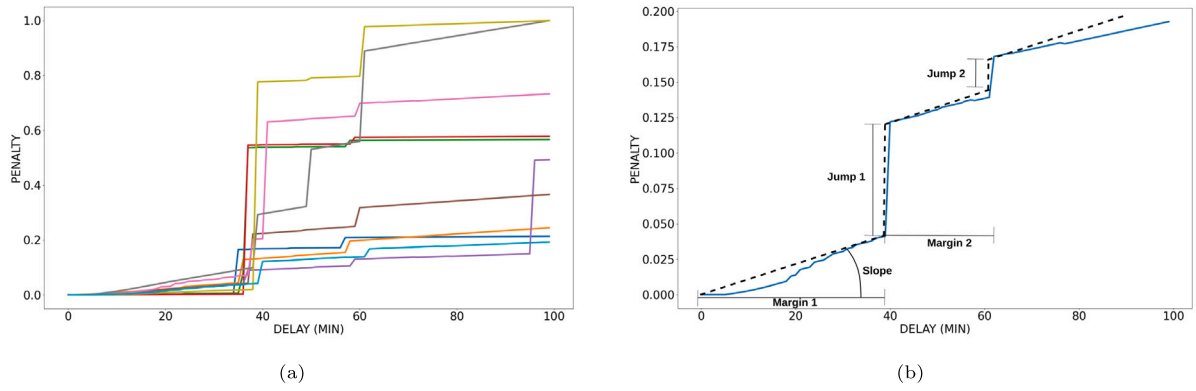


Fig. 8. (a) Examples of flight cost functions sampled from different hotspot scenarios. (b) Interpolation of a cost function using the five selected parameters: slope, two margins, and two jumps.

As discussed, the ISTOP mechanism entails a computationally challenging problem. For the mechanism to be feasible in a real-world scenario, sufficient optimisation efficiency must first be guaranteed, i.e., it must be able to identify good trading opportunities within the time constraints of tactical operations. Once this computational feasibility is established, the next step is to assess the operational side: namely, the actual effectiveness of ISTOP in mitigating delay costs and its impact on fairness across airlines. More specifically, for the computational aspect, our experiments have been designed to address the following research questions:

- Q1 Are the two neural networks effective in identifying the most promising flight pairs?
- Q2 How close to the optimum are the solutions obtained using only the pairs selected by the heuristic?
- Q3 Does our selection strategy outperform a random approach in identifying flight pairs?

Q1 aimed to verify whether the machine learning component can successfully rank and select pairs with high potential for delay cost reduction, thus reducing the solution space without discarding valuable opportunities. Q2 aimed to assess the quality of the heuristic solutions by comparing them against the exact optimum obtained when considering all possible flight pairs, thereby assessing the trade-off between computational efficiency and solution accuracy. Finally, the goal of Q3 was to ensure that the neural network-based selection is genuinely effective and not merely equivalent to a naive random sampling of flight pairs.

To assess the actual operational effectiveness of ISTOP mechanism in mitigating delay costs, our experiments addressed the following questions:

- Q4 What level of cost reduction is achieved by ISTOP?
- Q5 How does ISTOP perform with respect to equity considerations?

Q4 aimed to quantify the delay cost savings generated by the proposed inter-airline slot trading mechanism under various traffic scenarios. Q5 instead, aims to verify whether the benefits of slot trading are distributed across airlines and, more specifically, to investigate if ISTOP is capable of providing benefits also to those carriers that do not have enough flexibility to exploit FDR.

6.1. Data preparation and experimental setup

To test our model, we use hotspot data covering 56 days, from 20 June 2019 to 14 August 2019, obtained from EUROCONTROL's DDR2 database. In addition, passenger-related information is incorporated from itineraries collected on 11, 12, and 13 September 2014 [50].

Rather than relying on a limited set of real samples, we adopt a generative approach: from these datasets we compute statistical distributions of key operational characteristics—including the number of flights per hotspot, airline frequencies, capacity reductions, and passenger and connecting passenger volumes per flight. These distributions are then used to generate synthetic yet realistic hotspot scenarios, following the methodology established in Gasparin et al. [10] for UDPP implementation and optimisation. Flight cost functions are generated according to Cook and Tanner [51] and Cook and Tanner [43]. This approach enables more robust statistical analyses while preserving real-world operational characteristics, offering a significant advantage over using a small number of actual samples.

With the above procedure, we generated 2000 hotspots, each containing between 70 and 120 flights, resulting in a dataset of 2,008,662 flight pairs. To train the neural network for the classification task, labels were assigned by applying Algorithm 2 to each hotspot: a pair was labelled 0 if no feasible trade included it, and 1 otherwise. Subsequently, we split the data randomly, and 50% of the data was used as the training set, with the remainder reserved for testing. For classification, training was conducted using the Adam optimiser and learning rate scheduler from Smith [52] with parameters reported in Table 2. Instead, for the regression task, only pairs involved in feasible trades were considered (1,219,358 out of 2,008,662). Each pair p was associated with a target given by r_p , the maximum penalty reduction achievable in any trade containing that pair. Because a large fraction of pairs exhibit low penalty reductions (44.7% below 0.1, see Fig. 5(a)), the data was balanced by dividing the pairs into 100 bins according to penalty reduction and randomly selecting up to 25,000 pairs per bin, producing the distribution shown in Fig. 5(b). This resampling reduces the dominance of low-reduction pairs and provides a more informative training distribution for learning to predict meaningful reductions, making the neural network performance more robust across the range of different reductions. Also in this setting, the data was split randomly, and 50% of the selected pairs were used for training, with the remainder reserved for testing. Training was performed using the same optimiser and learning rate scheduler as for the classification task, but with the parameters in Table 1. Both neural networks were trained once with the above settings and then used throughout all the experimental analysis. All experiments were conducted on a machine with an Intel Core™ i7-10700 (2.90 GHz), 16 cores, 96 GB of memory, and a GeForce RTX 2070 SUPER GPU. The neural networks were implemented in PyTorch, while the optimisation problem was addressed using a custom Branch&Bound algorithm developed in C++, as anticipated in Section 4.2.2. Further implementation details are provided in Appendix.

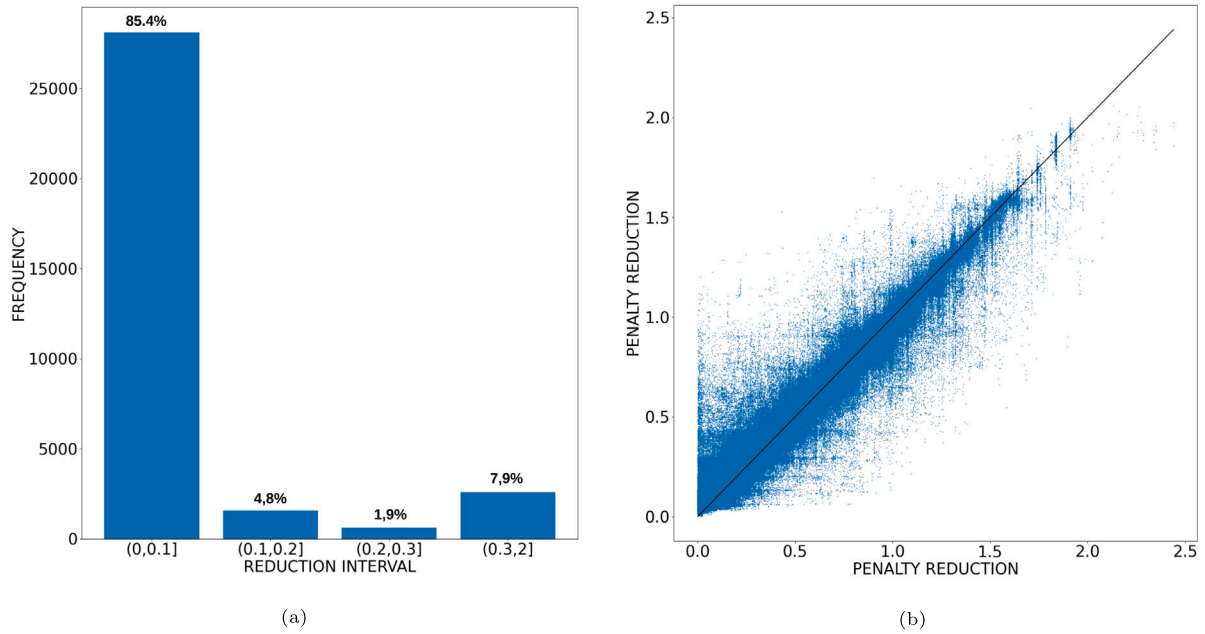


Fig. 9. (a) Distribution of the false negatives pairs misclassified by the neural network g^1 , aggregated by reduction interval. (b) Regression assessment: the black line indicates the ground truth and the blue dots the values predicted by the neural network g^2 .

Table 1
 g^1 and training parameters detail.

Parameter	Value ($RI=Relu$, $Dr = Dropout$)
g^1_f	128, RI , $Dr(0.4) \rightarrow 256$, $RI \rightarrow 64 - RI$
g^1_c	128, RI , $Dr(0.4) \rightarrow 256$, $RI \rightarrow 64 - RI$
g^1_{out}	128, RI , $Dr(0.4) \rightarrow 256$, $RI \rightarrow 128$, $RI \rightarrow 64$
batch size	256
epochs	2000
learning rate scheduler	cyclical, triangular2 [52]
scheduler cycle step	100,000
learning rate lowest value	10^{-8}
learning rate highest value	10^{-3}
up cycle	1000
down cycle	5000
weight decay	10^{-2}

7. Results

7.1. Q1 Are the two neural networks effective in identifying the most promising flight pairs?

Training the classification neural network was successful, yielding an accuracy of 93.8% on the test set. Misclassifications were split into 47.3% false positives and 52.7% false negatives. Importantly, as shown in Fig. 9(a), 85.4% of the false negatives—pairs incorrectly discarded by the network—correspond to pairs with very low penalty reduction (between 0 and 0.1).

For the regression task, training the neural network achieved a mean squared error of 0.058 and a mean error of 0.306 on the test set.

The quality of the regression fit is evident in Fig. 9(b), where the blue points, representing predicted penalty reductions for each pair, closely cluster around the red line, which denotes the ground truth values. This alignment demonstrates that the model's predictions closely match the actual data.

7.2. Q2 How close to the optimum are the solutions obtained using only the pairs selected by the heuristic?

The objective of this test was to compare solutions for the trading phase obtained using all possible flight pairs of all airlines (exact

approach) with those obtained by considering only the pairs selected through Algorithm 3, which relies on the predictions of the two neural networks (heuristic approach).

We generated 100 hotspots, each containing between 70 and 120 flights. This range was chosen because it is sufficiently large to challenge the heuristic, yet still manageable for the exact approach. For the offers budget B in the heuristic, we set a value of 300,000, which preliminary tests indicated strikes a good balance between solution quality and computational efficiency. Since ISTOP is intended for tactical-level implementation, a time limit of 10 min was imposed for optimising the B offers selected by the heuristic. For the exact approach, the solver was allowed up to one hour of computation. While this comparison is unfavourable in terms of runtime for the heuristic, the focus here is on solution quality—specifically, the gap between the heuristic and the optimal solution.

Table 3 shows that the average optimality gap of the heuristic is 30.195%, with a standard deviation of 33.039%. While this result is acceptable, it is not exceptional. However, Fig. 10(a) shows that poorer heuristic performance is concentrated in instances where the exact objective value is small, that is, hotspots in which ISTOP is not expected to deliver a significant penalty reduction. Considering only the 77 instances where the penalty reduction exceeds 0.1, the average gap drops to 19.532% (standard deviation 22.332%), demonstrating improved performance in cases where achieving a good solution has a meaningful impact.

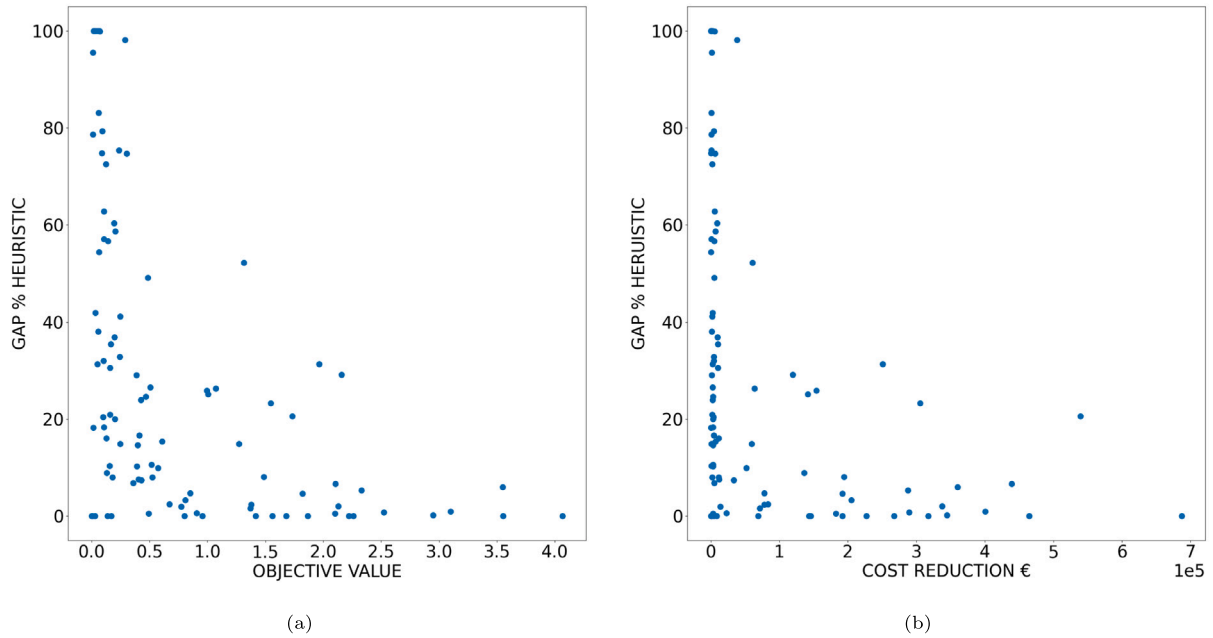
A critical consideration is whether the penalty function accurately reflects absolute cost behaviour. If it does not, the heuristic could fail to identify offers that provide small penalty reductions but yield significant cost savings. Fig. 10(b) indicates that this is not a prevalent issue, as poorer heuristic performance occurs mainly in instances where the absolute cost reduction (from the exact solver) is itself small. Focusing on the 76 instances with cost reductions above €2000, the average gap is 22.132% (standard deviation 25.636%), which further decreases to 11.877% (standard deviation 18.178%) for the 44 instances where reductions exceed €10,000. Notably, in 12 instances, the heuristic found the exact optimal solution; all of these cases involved significant penalty reductions (greater than 0.1) and cost reductions exceeding €2000.

Table 2
 g^2 and training parameters detail.

Parameter	Value (Rl=Relu, Dr = Dropout)
g_f^2	64, Rl, Dr(0.1) $\rightarrow$ 128, Rl, Dr(0.1) $\rightarrow$ 128, Rl $\rightarrow$ 36, Rl
g_c^2	64, Rl, Dr(0.1) $\rightarrow$ 128, Rl, Dr(0.1) $\rightarrow$ 128, Rl $\rightarrow$ 36, Rl
g_{out}^2	64, Rl, Dr(0.1) $\rightarrow$ 128, Rl, Dr(0.1) $\rightarrow$ 160, Rl, Dr(0.1) $\rightarrow$ 182, Rl, Dr(0.1) $\rightarrow$ 160, Rl $\rightarrow$ 160, Rl, Dr(0.1) $\rightarrow$ 128, Rl $\rightarrow$ 64, Rl $\rightarrow$ 36, Rl $\rightarrow$ 64, Rl $\rightarrow$ 64, Rl
batch size	1024
epochs	3000
learning rate scheduler	cyclical, triangular2 [52]
scheduler cycle step	100.000
learning rate lowest value	10^{-8}
learning rate highest value	10^{-3}
up cycle	1000
down cycle	5000
weight decay	10^{-2}

Table 3
Performance statistics of the three algorithms considered for the tests.

Model	Trades		Time		Gap %		Time limit
	mean	std	mean	std	mean	std	
Heuristic	2.393,390	4.020,187	20.969	77.005	30,195	33,039	6
Random	2000	5392	0033	0016	91,348	21,518	0
Exact	11.512,950	25.890,147	45.118	143.603	1000	0000	21

**Fig. 10.** Scatter plots of heuristic accuracy against (a) the overall penalty (objective value) and (b) the cost reduction. The accuracy tends to improve with the value of the penalty and similarly with the value of the cost.

In summary, the heuristic is effective when potential benefits are substantial. Its performance is strongest in scenarios where penalties (and thus underlying delay costs) are significant, confirming its practical utility for high-impact cases.

Regarding computational time, Table 3 shows that the heuristic is generally fast, averaging roughly twice the speed of the exact approach and reaching the time limit in only six instances, compared to 21 for the exact solver. This speed advantage holds despite the overhead of the pair selection phase and is largely due to the reduced pre-processing time and the smaller number of offers, which simplifies the optimisation. For fairness, average execution times were computed excluding instances in which either approach reached its time limit.

7.3. Q3 Does our selection strategy outperform a random approach in identifying flight pairs?

To justify the additional complexity introduced by the machine learning-based selection, we conducted an ablation study to assess its necessity beyond the performance analysis presented in List Q2. A natural validation step is to compare the proposed method against a naive baseline. Specifically, we replace the network-based selection in Algorithm 3 with a purely random selection process. This is implemented by considering the full set of flight pairs P_a for each airline and replacing both sorting operations in the algorithm (the initial ordering

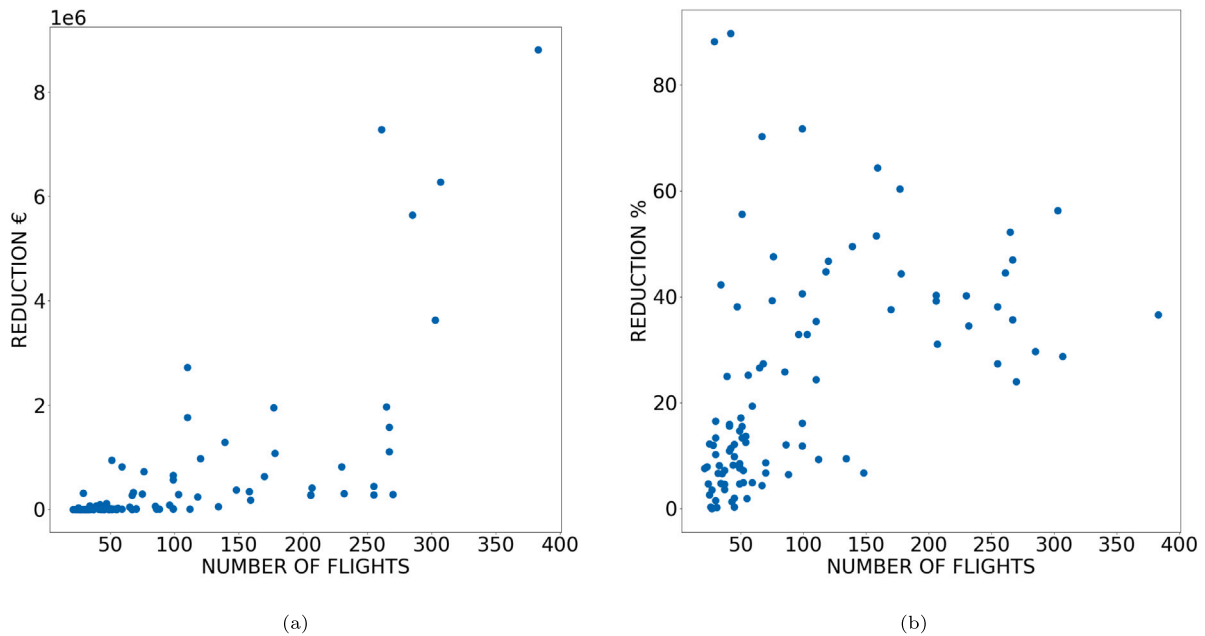


Fig. 11. (a) Number of flights in the hotspot vs ISTOP reduction in terms of absolute costs. (b) Number of flights in the hotspot vs ISTOP percentage reduction.

of the pair sets and the ordering of the airline list within the *while* loop) with random shuffling.

This variant, hereafter referred to as *random selection*, still ensures a substantial reduction in the number of pairs considered in Algorithm 1, and thus limits the number of offers included in the final optimisation step. As shown in Table 3, the random selection leads, as expected, to a significant reduction in computation time, even compared with the proposed heuristic. This behaviour is explained by the very small number of feasible offers typically identified when pairs are selected at random.

However, this gain in speed comes at the cost of solution quality. The random selection yields extremely poor results, with an average optimality gap of 91.320%. Moreover, it never outperforms the proposed heuristic and matches its solution in only nine instances. In particular, all these cases correspond to scenarios in which ISTOP is unable to deliver meaningful benefits: the best among them achieves a penalty reduction of only 0.066 and a maximum cost reduction of €2684.

7.4. Q4 What level of cost reduction is achieved by ISTOP?

To evaluate the performance of ISTOP in terms of cost reduction, we consider both the total reduction achieved and the individual contributions of the FDR and the trading phases. Since the objective is to assess the mechanism's potential, the experiments are conducted under the assumption of deterministic flight delay cost functions. Accordingly, all proposed offers are assumed to be accepted by the involved parties, as airlines have no incentive to reject solutions that guarantee a reduction in costs in the absence of uncertainty.

As in the experiment described in 7.2, we generated a new set of 100 hotspots, this time without imposing bounds on the number of flights, which was directly sampled from the empirical distribution inferred from the data. Each instance was solved using the exact approach whenever the number of possible offers (computed over all airlines' flight pairs) was below 300,000, and using the heuristic approach otherwise. In all cases, a time limit of 10 min was imposed on the solver.

Starting from an initial total cost of €159,059,058, ISTOP achieved an overall cost reduction of €56,928,652 (35.8%). Of this amount, €48,568,380 (30.5%) is attributable to the FDR phase, while €8,360,272 (5.3%) is generated by the trading phase.

As shown in Fig. 11(a), the largest absolute reductions are observed in hotspots involving a high number of flights. However, total cost reduction alone should not be regarded as the sole evaluation criterion, as the results may be disproportionately influenced by a limited number of large hotspots. Fig. 11(b) instead reports the relative reduction, defined as the percentage decrease with respect to each hotspot's initial cost. The substantial variability observed indicates that, in relative terms, ISTOP's effectiveness is not solely driven by the number of flights involved.

By partitioning the analysis into four scenarios based on the number of flights (Table 4), we observe that, in relative terms, ISTOP exhibits fairly consistent behaviour. The total cost reduction ranges between approximately 27% and 37%, with 20%–32% attributable to the FDR phase and between 3% and 7% to the trading phase.

In terms of computational performance, the average running time is 69.4 s, with a standard deviation of 190.1 s. Fig. 12(a) shows that, as expected, the most challenging instances are those involving a larger number of flights. In particular, the red dots correspond to instances solved using the heuristic approach, and all eleven cases in which the ten-minute time limit was reached belong to this group.

7.5. Q5 How does ISTOP perform with respect to equity considerations?

As noted in the introduction, a key motivation behind ISTOP is not only to reduce delay costs, but also to distribute the benefits of a revised allocation across multiple airlines. The FDR phase provides substantial flexibility to airlines with a large presence in a hotspot, as they have numerous opportunities to reshuffle their own flights. This effect is evident in Fig. 13, which reports ISTOP performance statistics for airlines grouped by the number of flights they operate within a hotspot.

As shown in Fig. 13(a), the FDR phase (green bars) is particularly beneficial for airlines operating more than 50 flights, with its contribution steadily decreasing as the number of flights declines. For these large airlines, the degree of internal flexibility is such that the trading phase becomes almost negligible in terms of additional cost reduction. This observation is further supported by Fig. 13(b), which reports the average percentage reduction.

A different picture emerges for the remaining airline groups, which typically constitute the majority within each hotspot, as illustrated

Table 4
Report of the ISTOP absolute reduction and percentage reduction aggregated by hotspot number of flights.

Num flights	Instances	Final reduction		Int-opt reduction		FDR %	Trade %	Final %
		mean	std	mean	std			
<50	43	18.116	52.550	13.842	32.646	20,82	6,43	27,25
50–100	27	181.539	292.582	164.043	257.559	32,02	3,41	35,43
100–200	14	847.864	823.622	677.339	709.226	26,77	6,74	33,51
200>	16	2.461.122	2.910.770	2.128.827	2.535.894	31,85	4,97	36,82

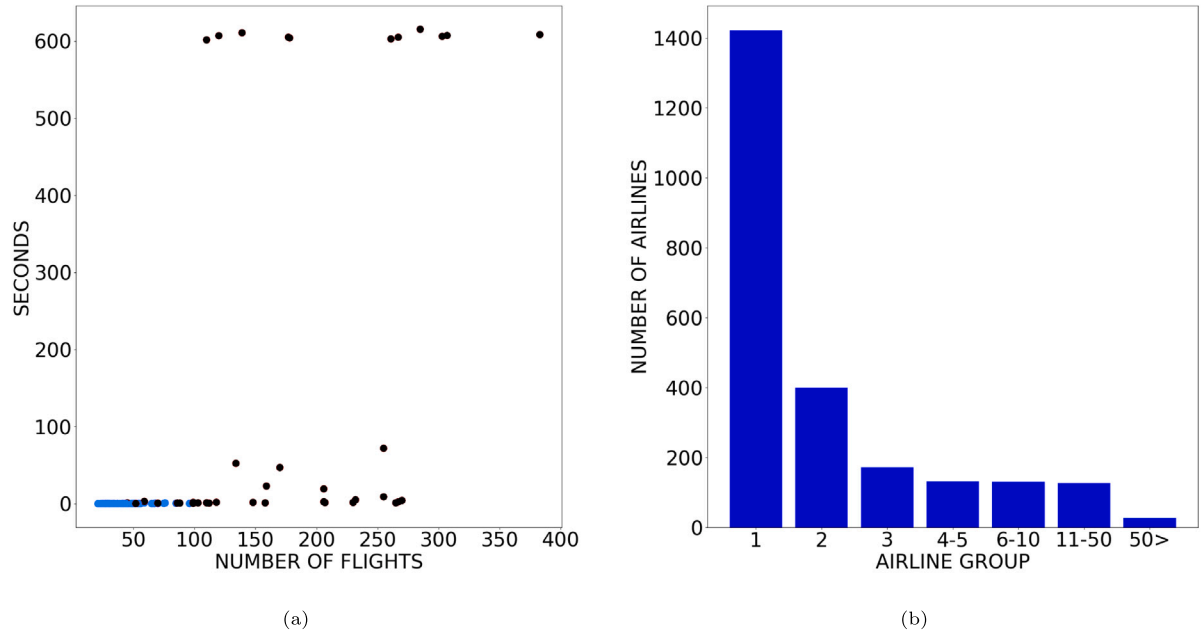


Fig. 12. (a) Number of flights vs Execution time. The black dots represent the instances solved with the heuristic, while the blue dots represent the ones solved without it. (b) Frequency plot representing the occurrences of the various airline groups.

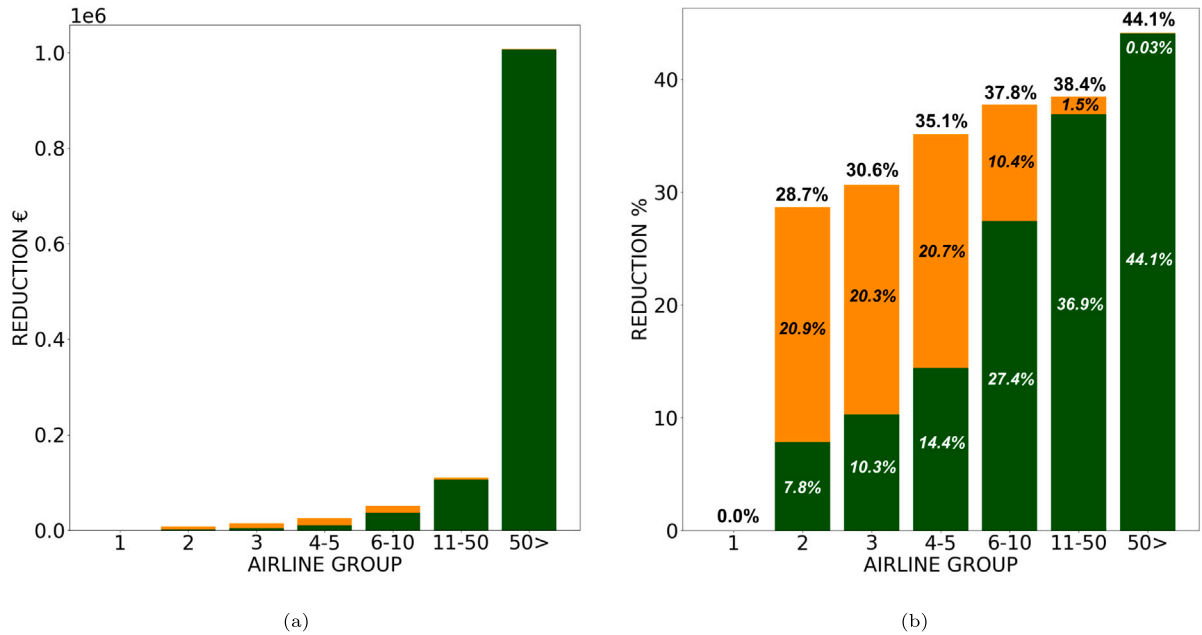


Fig. 13. (a) Total cost reduction per airline group. (b) Percentage cost reduction per airline group. In both figures, the green portion of the bars represents the contribution of the FDR, while the orange the one of the trading phase.

in Fig. 12(b). With the exception of airlines operating a single flight (which cannot benefit from ISTOP) airlines with a small number of flights can only partially exploit the FDR phase due to their limited

internal flexibility. In these cases, the trading phase (orange bars) plays a crucial role, accounting for more than half of the total reduction for airlines operating 2, 3, 4, or 5 flights.

8. Conclusions

In this work, we introduced ISTOP, a novel mechanism for addressing the ATFM slot allocation problem. ISTOP aims to reduce delay costs while ensuring non-penalising and equitable outcomes. Its design deliberately centralises the optimisation complexity within the Network Manager, while offering airlines a simple and transparent framework that directly involves them in the decision-making process.

The computational experiments show that ISTOP consistently achieves its objectives. On average, the mechanism yields a 35.8% reduction in total delay costs and improves upon UDPP solutions. It proves particularly effective for airlines that typically lack flexibility due to a limited number of flights. In this respect, the trading phase represents a crucial complement to the FDR, providing an additional average cost reduction of 5.3%. More importantly, it plays a key role in enhancing equity: the analysis by airline groups indicates that the effectiveness of the FDR is strongly dependent on the number of flights an airline operates within a hotspot, favouring primarily larger operators. The trading phase mitigates this imbalance by enabling smaller airlines, with limited internal flexibility, to achieve meaningful cost savings through cross-airline exchanges. As a result, ISTOP is especially beneficial for the majority of airlines in a hotspot, which typically operate only a small number of flights.

At its core, ISTOP relies on a complex combinatorial optimisation problem. While this problem can be solved in negligible time for most instances, it rapidly becomes intractable as the number of flights increases. To address this issue, we developed a hybrid approach that combines classical optimisation with machine learning techniques. This integration allows large-scale instances to be handled efficiently, achieving a favourable trade-off between solution quality and computational effort.

Despite these results, airlines operating a single flight within a hotspot cannot benefit from the current features of the mechanism. Extending ISTOP to cover these cases therefore remains an open challenge and represents a natural direction for future research.

CRedit authorship contribution statement

Andrea Gasparin: Writing – original draft, Supervision, Software, Methodology, Data curation, Conceptualization. **Federico Julian Camerota Verdù:** Writing – original draft, Formal analysis, Conceptualization. **Giulio Alberti:** Visualization, Software, Methodology. **Lorenzo Castelli:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors have declared no conflict of interest.

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.

Appendix. Branch&bound for the ISTOP MWIs

For solving the MWIS that selects the final trades in the ISTOP trading phase, we exploit a property that is specific to this setting: given a hotspot with n flights, the number of trades in an optimal solution cannot exceed $\lfloor n/4 \rfloor$. This follows from the fact that each flight can be involved in at most one trade, and that each trade requires at least four flights (two pairs).

This observation enables the design of an efficient Branch&Bound strategy, which proceeds as follows:

- The set of trades $\mathcal{T}$ is sorted in descending order according to their penalty reduction.
- A compatibility matrix $\mathcal{M}$ is constructed, where $\mathcal{M}_{ij} = 1$ if trade $i < j$ is compatible with trade j (i.e., they share no flights), and $\mathcal{M}_{ij} = 0$ otherwise. By construction, $\mathcal{M}$ is upper triangular.
- The best solution s_b is initialised with the first element of $\mathcal{T}$, with corresponding value r_b .
- For each trade $i \in \mathcal{T}$, a partial solution $s = \{i\}$ with value $r = r_i$ is built, together with its compatibility vector $o = \mathcal{M}_i$ (the i th row of $\mathcal{M}$). For each trade compatible with i (i.e., each non-zero entry of o), the following recursive procedure is applied:

- If $|s| = \lfloor n/4 \rfloor$, no additional trades can be added to s , and the current branch is pruned.
- Otherwise:

- * Trade j is added to s , and the solution value is updated as $r \leftarrow r + r_j$.
- * If $r > r_b$, the incumbent best solution is updated.
- * The compatibility vector is updated by element-wise multiplication $o \leftarrow o \cdot \mathcal{M}_j$. After this operation, the non-zero entries of o correspond to trades that are compatible with j and with all other trades already in s .
- * Let $m = \lfloor n/4 \rfloor - |s|$ denote the maximum number of additional trades that can still be added to s . Since $\mathcal{T}$ is sorted, if $\tilde{\mathcal{T}}$ denotes the set of the first m remaining trades (or fewer, if not available) that are compatible with all trades in s (i.e., the first m non-zero entries of o), then

$$\bar{r} = r + \sum_{i \in \tilde{\mathcal{T}}} r_i$$

provides an upper bound for the current branch. If $\bar{r} < r_b$, the branch is pruned.

- * Otherwise, the recursion continues by considering each trade compatible with all trades in s (the non-zero entries of o).

Two aspects are central to the effectiveness of this algorithm. First, the upper-triangular structure of the matrix $\mathcal{M}$ ensures that each combination of trades is explored only once, since the compatibility vector o is obtained through successive multiplications of rows of $(\mathcal{M})$. Second, sorting $\mathcal{T}$ in descending order guarantees the validity of the bound $\bar{r}$, as the first m trades compatible with the current solution s are also those with the largest reduction values. Implementing such a finely tuned Branch&Bound algorithm within a commercial solver is non-trivial, as it relies on low-level customisation (such as the non-linear update of the compatibility vector) that is not readily supported. For this reason, a dedicated implementation in C++ was developed. To evaluate its efficiency, we compared its computational performance with that of a Gurobi implementation of problem (5), augmented with the valid inequality

$$\sum_{i \in \mathcal{T}_i} x_i \leq \lfloor n/4 \rfloor \quad \forall i \in \mathcal{T},$$

where $\mathcal{T}_i$ denotes the set of trades compatible with trade i .

The experiments were conducted on 100 instances generated as in Section 7.2, with the number of flights sampled between 40 and 70 and a time limit of 10 min. The average solution time of the proposed Branch&Bound algorithm was 14.9 s, with a standard deviation of 90.1 s, compared with an average of 33.0 s for Gurobi (standard deviation 134.2 s). In two instances, Gurobi ran out of memory and failed to produce a solution, while the Branch&Bound algorithm reached the time limit but still returned feasible (and non-zero) solutions. In the remaining 98 instances, Gurobi reached the time limit in five cases, whereas the Branch&Bound algorithm always found the optimal solution. In four of these five cases, the Gurobi solutions were suboptimal, with an average optimality gap of 0.9%.

Data availability

The authors do not have permission to share data.

References

- [1] EUROCONTROL. EUROCONTROL Forecast Update 2025–2031 Autumn. 2025, URL: <https://www.eurocontrol.int/publication/eurocontrol-forecast-update-2025-2031>. [Last accessed: 20 April 2026].
- [2] Mannino C, Nakkerud A, Sartor G. Air traffic flow management with layered workload constraints. *Comput Oper Res* 2021;127:105159.
- [3] Castelli L, Pesenti R, Ranieri A. The design of a market mechanism to allocate Air Traffic Flow Management slots. *Transp Res Part C: Emerg Technol* 2011;19(5):931–43.
- [4] Brugnara I, Castelli L, Pesenti R. A market mechanism for multiple air traffic resources. *Transp Res Part E: Logist Transp Rev* 2023;178:103255.
- [5] Gurtner G, Mocholí D. BEACON project, D3.1, High-level modelling requirements. 2021, URL: <https://www.beacon-sesar.eu/wp-content/uploads/2021/09/893100-BEACON-D3.1-High-level-modelling-requirements-v01.00.00.pdf>. [Last accessed: 20 April 2026].
- [6] SESAR. SESAR Solution PJ07.02 SPR-INTEROP/OSD for V2 - Part I. 2019.
- [7] Pilon N, Guichard L, Cliff K. Reducing impact of delays using airspace user-driven flight prioritisation. Ninth SESAR Innov Days 2019.
- [8] Pilon N, Guichard L, Bazso Z, Murgese G, Carré M. User-driven prioritisation process (UDPP) from advanced experimental to pre-operational validation environment. *J Air Transp Manag* 2021;10:97:102124.
- [9] Ruiz S, Guichard L, Pilon N, Delcourte K. A new air traffic flow management user-driven prioritisation process for low volume operator in constraint: Simulations and results. *J Adv Transp* 2019;2019(1):1208279.
- [10] Gasparin A, Castelli L, Bolić T, Gurtner G, Pilon N. A user-driven prioritisation process implementation and optimisation for ATFM hotspot resolution. *Transp Res Part C: Emerg Technol* 2025;170:104894.
- [11] Vossen TW, Ball MO. Slot trading opportunities in collaborative ground delay programs. *Transp Sci* 2006;40(1):29–43.
- [12] Odoni AR. The flow management problem in air traffic control. In: *Flow control of congested networks*. Springer; 1987, p. 269–88.
- [13] Terrab M, Odoni AR. Strategic flow management for air traffic control. *Oper Res* 1993;41(1):138–52.
- [14] EUROCONTROL. Airport Collaborative Decision-Making (A-CDM) Implementation Manual. 2017, URL: https://www.eurocontrol.int/archive_download/all/node/10373. [Last accessed: 20 April 2026].
- [15] Vossen T, Ball M. Optimization and mediated bartering models for ground delay programs. *Naval Res Logist* 2006;53(1):75–90.
- [16] Sherali HD, Hill JM, McCrea MV, Trani AA. Integrating slot exchange, safety, capacity, and equity mechanisms within an airspace flow program. *Transp Sci* 2011;45(2):271–84.
- [17] Bertsimas D, Farias VF, Trichakis N. The price of fairness. *Oper Res* 2011;59(1):17–31.
- [18] Bertsimas D, Gupta S. Fairness and collaboration in network air traffic flow management: an optimization approach. *Transp Sci* 2016;50(1):57–76.
- [19] Jacquillat A, Vaze V. Inter-airline equity in airport scheduling interventions. *Transp Sci* 2018;52(4):941–64.
- [20] Hamdan S, Cheaitou A, Jouini O, Andersson Granberg T, Jemai Z, Alsyouf I, Betayeb M, Josefsson B. Central authority-controlled air traffic flow management: An optimization approach. *Transp Sci* 2022;56(2):299–321.
- [21] BEACON Consortium. BEACON - Behavioural economics for ATM concepts. 2022, URL: <https://www.sesarju.eu/projects/beacon>. [Last accessed: 20 April 2026].
- [22] Pilon N, Cook A, Ruiz S, Bujor A, Castelli L. Improved flexibility and equity for airspace users during demand-capacity imbalance—an introduction to the user-driven prioritisation process. Sixth SESAR Innov Days 2016.
- [23] Zhang QL, Meilong Xu Y. Collaborative delay management towards demand-capacity balancing within user driven prioritisation process. *J Air Transp Manag* 2021;91:102017.
- [24] Zhao Z, Li Y, Liu X, Zhu J, Zhao S. Research on ATFM delay optimization method based on dynamic priority ranking. *Aerospace* 2025;12(9):793.
- [25] Feichtenschlager P, Schuetz K, Jaburek S, Schuetz CG, Gringinger E. Privacy-preserving implementation of an auction mechanism for ATFM slot swapping. In: 2023 integrated communication, navigation and surveillance conference. ICNS, 2023, p. 1–12.
- [26] Gurtner G, Bolić T. Impact of cost approximation on the efficiency of collaborative regulation resolution mechanisms. *J Air Transp Manag* 2023;113:102471.
- [27] Gasparin A, Gurtner G, Bolić T. D6.2 final concept assessment report. Project deliverable 893100, Brussels: BEACON Consortium; 2022, URL: <https://www.beacon-sesar.eu/wp-content/uploads/2023/02/893100-BEACON-D6.2-Final-Concept-Assessment-report-V.01.02.00.pdf>. [Last accessed: 20 April 2026].
- [28] Pinto Neto EC, Baum DM, Almeida JRd, Camargo JB, Cugnasca PS. Deep learning in air traffic management (ATM): A survey on applications, opportunities, and open challenges. *Aerospace* 2023;10(4).
- [29] Xu Q, Pang Y, Liu Y. Air traffic density prediction using Bayesian ensemble graph attention network (BEGAN). *Transp Res Part C: Emerg Technol* 2023;153:104225.
- [30] De Giovanni L, Lancia C, Lulli G. Data-driven optimization for air traffic flow management with trajectory preferences. *Transp Sci* 2024;58(2):540–56.
- [31] Hızır AE, Barnhart C, Vaze V. Large-scale airline crew recovery using mixed-integer optimization and supervised machine learning. *Transp Sci* 2025.
- [32] Xu Y, Shen Z, Liu Z, Sun X. Machine learning assisted dynamic copy generation approach for integrated aircraft and passenger recovery problem. *Transp Res Part C: Emerg Technol* 2025;178:105208.
- [33] Azevedo BF, Rocha AMAC, Pereira AI. Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Mach Learn* 2024;113(7):4055–97.
- [34] Boschetti MA, Maniezzo V. Contemporary approaches in matheuristics an updated survey. *Ann Oper Res* 2024;343(2):663–700.
- [35] SESAR Joint Undertaking, EUROCONTROL. Step 1 V3 UDPP final OSD (Operational service and environment definition). Project deliverable D66, SESAR Joint Undertaking; 2015, URL: https://www.sesarju.eu/sites/default/files/documents/solution/Sol57_3_Local_UDPP_OSED.pdf. [Last accessed: 20 April 2026].
- [36] Wambgans MC. Collaborative decision making in air traffic management. In: Bianco L, Dell’Omo P, Odoni AR, editors. *New concepts and methods in air traffic management*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001, p. 1–15.
- [37] Murça MCR. Collaborative air traffic flow management: Incorporating airline preferences in rerouting decisions. *J Air Transp Manag* 2018;71:97–107.
- [38] Erkan H, Erkip NK, Şafak Ö. Collaborative decision making for air traffic management: A generic mathematical program for the rescheduling problem. *Comput Ind Eng* 2019;137:106016.
- [39] EUROCONTROL. Airport collaborative decision-making (a-CDM). 2016, URL: <https://www.eurocontrol.int/concept/airport-collaborative-decision-making>. [Last accessed: 20 April 2026].
- [40] Hansen MM, Gillen D, Djafarian-Tehrani R. Aviation infrastructure performance and airline cost: a statistical cost estimation approach. *Transp Res Part E: Logist Transp Rev* 2001-03;37(1):1–23.
- [41] Evler J, Schultz M, Fricke H, Cook A. Development of stochastic delay cost functions. Tenth SESAR Innov Days 2020.
- [42] Cook AJ, Tanner G. European airline delay cost reference values, Final report (Version 3.2). Technical report, EUROCONTROL Performance Review Unit; 2011.
- [43] Cook AJ, Tanner G. European airline delay cost reference values, Updated and extended values (Version 4.1). Technical report, EUROCONTROL Performance Review Unit; 2015.
- [44] Le HM, Pham DT, Duong VN. A multi-agent decision-making approach for automated inter-airline slot swapping in ground delay program with airline privacy preservation. In: 2025 integrated communications, navigation and surveillance conference. ICNS, 2025, p. 1–7.
- [45] Gruber S, Feichtenschlager P, Schuetz CG. Using genetic algorithms for privacy-preserving optimization of multi-objective assignment problems in time-critical settings: An application in air traffic flow management. In: *Proceedings of the genetic and evolutionary computation conference. GECCO '24*, New York, NY, USA: Association for Computing Machinery; 2024, p. 1246–54.
- [46] Pardalos P, Xue J. The maximum clique problem. *J Global Optim* 1994;4:301–28.

- [47] Lamm S, Schulz C, Strash D, Williger R, Zhang H. Exactly solving the maximum weight independent set problem on large real-world graphs. In: 2019 proceedings of the meeting on algorithm engineering and experiments. ALENEX, 2019, p. 144–58.
- [48] Gurtner G, Cook A. The hidden cost of uncertainty for airspace users. *J Air Transp Manag* 2021;91:102002.
- [49] Evler J, Schultz M, Fricke H, Cook A. Stochastic delay cost functions to estimate delay propagation under uncertainty. *IEEE Access* 2022;10:21424–42.
- [50] Paxis. Paxis passenger and traffic data. 2014, Data provider. Now part of IATA Passenger and Traffic data.
- [51] Cook AJ, Tanner G. The cost of passenger delay to airlines in Europe - consultation document. Univ Westminster 2015.
- [52] Smith LN. Cyclical learning rates for training neural networks. In: 2017 IEEE winter conference on applications of computer vision. WACV, 2017, p. 464–72.