A slot scheduling mechanism at congested airports which incorporates efficiency, fairness and airline preferences

Jamie Fairbrother[1], Konstantinos G. Zografos[1], and Kevin Glazebrook[1]

[1]Centre for Transport and Logistics (CENTRAL), Department of Management Science, Lancaster University Management School, Lancaster, UK

May 30, 2019

Abstract

Congestion is a problem at airports where capacity does not meet demand. At many such airports, airlines must request time slots for the purpose of landing or take off. Given the imbalance between demand and capacity, slot requests cannot always be scheduled as requested. The difference between the requested and allocated time slot is called displacement. Minimization of the total displacement is a key slot scheduling objective and expresses the efficiency of the slot scheduling process. Additionally, fairness has been proposed as a slot scheduling criterion. Fairness relates to the allocation of the total schedule displacement among the various airlines. Single and multi-objective models have been proposed for slot scheduling. However, currently the literature lacks models that incorporate the preferences of airlines regarding the allocation of displacement to their flights. This paper proposes a two-stage mechanism for the scheduling of slots at congested airports. The proposed mechanism considers efficiency and fairness objectives and incorporates the preferences of airlines in allocating the total displacement associated with the flights of each airline. The first stage of the mechanism constructs a reference schedule which is fair to the participating airlines. In the second stage the airlines specify how the displacement allocated to them in the reference schedule should be distributed among their requests. The mechanism then adjusts the fair reference schedule to meet as many of these preferences as possible. The development and implementation of the proposed slot scheduling mechanism is demonstrated using real data from a coordinated airport, and simulated displacement preference data. The proposed slot scheduling mechanism provides useful information to decision makers regarding the equity efficiency trade-off and enhances the transparency and acceptability of the slot scheduling outcome.

1 Introduction

Due to insufficient capacity to meet demand, many airports around the world suffer congestion-related delays and scheduling infeasibilities. Physical and political constraints mean that the expansion of infrastructure is not possible in the short to medium term and so congestion must be mitigated through the management of demand.

Various market-based mechanisms such as congestion-pricing (Vaze & Barnhart, 2012; Morrison & Winston, 2007) and auctions (Rassenti *et al.* , 1982; Ball *et al.* , 2005) have been proposed in the academic literature. However, the dominant mechanism for managing demand at congested airports is an administrative scheme called the IATA Worldwide Slot Guidelines (WSG) (International Air Transport Association, 2017). This is used throughout the world outside of the US, and even forms the basis of EU law on slot allocation (Commission, 1993a,b). Under the IATA WSG, congested airports are designated as *coordinated* and to use airport infrastructure at a coordinated airport, airlines must request and be allocated time slots for landing or take-off. While the WSG specifies the general procedures, priorities and constraints which must be followed, it leaves some discretion to slot coordinators on how exactly slots are allocated.

Various models for slot allocation have been proposed (Zografos *et al.* , 2012; Zografos & Jiang, 2019; Ribeiro *et al.* , 2018). Although some of these models incorporate fairness, the fairness schemes treat all requests equally which indirectly penalizes requests made at off-peak times. Moreover, none of these models explicitly take into account the preferences of airlines regarding how their requests should be displaced. These issues are important because a schedule which is fair and takes into account an airline's preferences is much more likely to be acceptable to that airline. The main contribution of this paper is a mechanism which implements the main features of the IATA WSG scheme and also incorporates fairness and the preferences of the airlines. The mechanism works by first constructing a fair reference schedule and then adjusting this according to airline preferences. The proposed fairness scheme improves on previous schemes by taking into account demand for slots at the requested times. In order to provide a detailed description of this paper's contribution we must first overview the IATA WSG and the current literature.

The paper is organized as follows. In the remainder of this section, we review the IATA WSG, the current literature, and describe in detail the contributions of this paper. In Section 2 we provide a summary of the proposed mechanism, and the main processes involved. In Section 3 we present a basic slot allocation model, originally proposed in Zografos *et al.* (2012), which is extended in later sections to incorporate fairness and airline preferences. In Section 4 we describe a new fairness scheme. In Section 5 we describe our mechanism for incorporating airline displacement preferences. In Section 6 we explain how the proposed mechanism can be used to take into account the main IATA WSG priority classes. In Section 7 we test our methodology using request data for a coordinated airport and simulated preference data. Finally, in Section 8 we make some concluding remarks and discuss future work.

1.1 IATA Slot Allocation Process and Priority Classes

In this section we describe the main features of the IATA slot allocation process. For a more comprehensive description of the IATA WSG we refer the reader directly to (International Air Transport Association, 2017).

Slots at a coordinated airport are allocated for a six month season and the process begins around six months prior to the start of the season. Slot requests are often made in the form of series which consist of multiple requests for the same time, and usually the same day of the week, over a period of at least five weeks. Airlines submit their slot requests using a standardized format which indicates for each request or request series, amongst other data, the requested times for arrival and departure at the airport, the dates for which the request applies, and any priority codes that may apply for the request.

The coordinator then proposes an initial allocation of slots to the airlines based on these requests. An airline may accept or reject an offer of a slot. After this initial allocation, the airlines and coordinators meet at the bi-annual IATA slot conference whose purpose is to provide airlines with an opportunity to discuss schedule adjustments with coordinators and to facilitate bilateral trading of slots between airlines. Following the slot conference, a period of continuous slot allocation occurs until the start of the scheduling season when airlines may make new requests or modify or delete existing requests.

The mechanism proposed in this paper aims to produce an initial allocation of slots following the rules which govern this. We now briefly describe these rules. For a more detailed summary of the IATA WSG for the initial allocation of slots see (Zografos *et al.* , 2012; Ribeiro *et al.* , 2018).

Each coordinated airport has a declared capacity which constrains how many slots can be allocated due to availability of resources (e.g. runways, apron stands etc.). This declared capacity is usually expressed in the form of rolling capacity constraints. These limit the number of arrival, departure or total number of slots that can be allocated over a given time interval (e.g. 15 minutes or 1 hour). The term *slot pool* is often used to refer to the availability of slots. It is important to bear in mind that in using this term, we do not refer to a single well defined set of arrival and departure slots, but rather the possible sets of slots that can be allocated subject to the capacity constraints[1]. The schedule must also satisfy *turnaround constraints* which prescribe that, in order to prepare an aircraft for the next flight, a departure must be scheduled a sufficient amount of time after its arrival.

The capacity constraints mean that not all requests can be allocated their preferred slots. While the aim of the slot allocation is to construct a schedule which matches as far as possible the requested times, slots must also be allocated primarily according to the following main priority groups:

1. Historic requests: An airline which had a series of slots in the preceding season (and did not misuse them) is entitled to the same series of slots in the next season. Airlines with historic rights are also entitled to request a change to the time of a slot.

2. New entrant requests: An airline is given the status of *new entrant* if it does not already have significant presence at the airport. Of the remaining slots in the slot pool after historical requests

[1] Note that our use of the term will differ slightly from that in the IATA WSG (International Air Transport Association, 2017). There the term slot pool is used for the slots remaining after all historic slots have been allocated whereas we use it for any stage of the allocation.

have been allocated, 50% should be allocated to new entrants, unless slot requests by new entrants make up less than 50% of the remaining slots.

3. Other requests: All remaining requests have the lowest priority.

Note that the mechanism in this paper assumes that there are enough slots in each day to allocate to the requests. Hence, the 50% rule described above for new entrant requests can be disregarded for the purposes of this paper.

1.2 Literature review

1.2.1 Slot allocation models

Due to the complexity and size of the slot allocation problem, a number of optimization models have been developed to allocate slots. A common concept used in these models is that of the *displacement* of a request, which is the absolute difference in time between the requested and allocated slots of a request. In order to satisfy slot requests as well as possible, these models typically aim to minimize some objective based on displacement. The most common objective is the *schedule displacement* which refers to the total displacement across all slots requests.

The allocation of slots which incorporate the main features of the IATA WSG (scheduling for a season and request priorities) is first explicitly modelled as an optimization problem in the paper by Zografos *et al.* (2012) who formulate it as an integer linear program (ILP). The model allocates requests for series of slots over a scheduling season, includes rolling capacity and turnaround constraints, and minimizes the total schedule displacement. The different priority classes of the requests are taken into account by solving the model *hierarchically*. That is, slots are allocated to the highest priority class by solving the slot allocation model for only those requests, the rolling capacities are updated, the model is solved for the next highest priority group, and so on. The model is solved via a row generation algorithm using request data from three coordinated airports, demonstrating improvements over the actual allocations.

The model of Zografos & Jiang (2016, 2019) extends that of Zografos *et al.* (2012) to incorporate fairness. A bi-objective formulation is proposed to study the trade-off between schedule displacement and fairness, and is solved using the ϵ-constraint method. The model is solved both hierarchically and non-hierarchically for a congested airport. In the non-hierarchical case, priority classes are disregarded, and all requests are allocated slots simultaneously. The analysis reveals that in both cases the fairness of an allocation could be substantially improved at the cost of only a small increase in schedule displacement.

Zografos *et al.* (2017) present two bi-objective formulations. The first formulation minimizes the schedule displacement and the largest single displacement allocated to a request. The second formulation introduces acceptable time windows in which requests should be allocated slots, and seeks to minimize the total schedule displacement and the number of requests allocated a slot outside of their acceptable time window. Airlines tend to be averse to large displacements and so these models aim to construct a schedule which is more acceptable. These models were again solved using the ϵ-constraint method.

Ribeiro *et al.* (2018) present a slot allocation model which uses a different formulation, and which models the IATA WSG rules for the initial allocation to a greater level of detail than has previously been done. This model has four objectives which are minimized lexicographically: the number of rejected requests, the maximum schedule displacement, the total schedule displacement and the number of displaced requests. It also treats separately the IATA priority groups of "historical" and "changes to historical" requests. Finally, priority classes are taken into account by Ribeiro *et al.* (2018) by solving the model lexicographically rather than hierarchically.

Other models have been proposed for slot scheduling which do not incorporate the main aspects of the IATA WSG or which have not been solved for a scheduling season, but which incorporate other problem features. Jacquillat & Odoni (2015) propose a model which incorporates tactical operations such as runway configuration changes in order to calculate the expected queue lengths which arise from a given schedule. This allows capacity levels to be adjusted in such a way that queue lengths do not exceed a given level. This model is extended in Jacquillat & Vaze (2018) to ensure that displacement is apportioned fairly among the competing airlines. All the slot allocation models discussed thus far concern the allocation of slots at a single airport. However, another important issue in the allocation of slots is ensuring that flights have compatible departure and arrival slots at their origin and destination airports. This can be achieved by solving a network-wide slot allocation model, that is, a model which simultaneously allocates slots across a network of airports (Corolli *et al.* , 2014; Castelli *et al.* , 2012; Pellegrini *et al.* , 2017; Benlic, 2018). The sizes and computational complexities of both of these types of models mean that they cannot be solved exactly for a whole scheduling season.

1.2.2 Fairness

Fairness has been widely studied in the context of systems where limited resources or benefits must be shared between a group of participants. Although general axiomatic approaches to fairness exist (Nash, 1950; Kalai & Smorodinsky, 1975), there is no universally agreed definition, and it is often formulated based on problem context. Nevertheless, there is generally a trade-off between fairness and total system efficiency[2].

In the context of air traffic flow management (ATFM), schemes have been proposed for incorporating fairness into traffic management initiatives (TMIs) which reschedule flights in response to reductions in capacity of airports and airspace. A consensus has emerged that the most equitable way of allocating new arrival slots is using the ration-by-schedule (RBS) rule whereby flights are rescheduled on a first-scheduled-first-served basis (Wambsganss, 1996). As this approach can lead to excessive delays, and in the case of multiple reductions of capacity, infeasible schedules (Barnhart *et al.* , 2012), optimization models have been proposed which trade-off overall delay against deviations from the RBS principle. Vossen *et al.* (2003) propose an optimization model to reschedule flights during a ground-delay program[3](GDP) which is found to reduce delays significantly while reducing variation in the average delays across the airlines. The paper by Barnhart *et al.* (2012) generalises the RBS principle to account for the case of multiple reductions of capacity across a network, and proposes an optimization model for rescheduling flights which improves fairness according to this new definition. Bertsimas & Gupta (2015) present a model which reschedules flights across a whole airspace network. As well as minimizing overall delay, this model incorporates fairness by also minimizing reversals in the ordering of pairs of flights in RBS schedules.

Airport slot allocation differs from the ATFM interventions described above in that there is no obvious fair reference schedule. Instead, fairness has been defined by how schedule displacement is distributed across airlines requesting slots. Zografos & Jiang (2017) construct indices which measure how fairly each airline is treated in a given schedule. This fairness index is predicated on the principle that the proportion of total schedule displacement assigned to an airline should be similar to the proportion of requests made by it. Fairness is then incorporated into the slot allocation model by adding constraints involving these fairness indices. The approach of Jacquillat & Vaze (2018), whose model is formulated and solved for a single day, is to incorporate fairness by lexicographically minimizing the schedule displacement normalized by the number of requests for each airline. Note that this is the axiomatic approach to fairness of Kalai & Smorodinsky (1975). A limitation of both of these approaches is that they treat all requests equally, regardless of when they are made. In particular, requests made for off-peak periods may cause airlines to be allocated more schedule displacement, despite the fact that these requests do not contribute to scheduling infeasibility.

1.2.3 Prioritization

The current IATA slot allocation process and existing models do not consider explicitly the airline preferences regarding which requests should be displaced and by how much. By taking these into account one could construct a schedule which is more acceptable to the participating airlines.

In ATFM much work has been done on the involvement of airlines in the decision-making process to improve overall outcomes. One of the most famous examples of this, and which has been in operation for many years, is the schedule compression algorithm used in the GDP in the US. This algorithm encourages airlines to submit accurate information regarding delays by allowing airlines preferential access to new slots when they declare flight delays or cancellations. The system has been demonstrated to reduce delays substantially compared to previous practice (Wambsganss, 1996). Models have since been proposed which allow slots to be exchanged between airlines in more flexible ways and in more general settings (Vossen & Ball, 2006; Bertsimas & Gupta, 2015; Pilon *et al.* , 2016). The works cited here use conditional trades, whereby an airline offers to give up one or more slots in exchange for gaining one or more other slots. There is trade-off with regards to the type of offers airlines are allowed to propose. The more general the type of offer, the more offers an airline can make, but the more computational difficult it becomes to optimize the slot exchange. The less flexible the type of allowed offer, the fewer requests an airline can make, and the less effective the exchange.

Mechanisms have also been proposed which allow airlines to have a say on parameters of TMIs such the GDP (e.g. scope and duration), by submitting their preferences on system-wide performance objectives (e.g. predictability and capacity). See (Swaroop & Ball, 2013; Evans *et al.* , 2016) for instance. All these schemes relate to TMIs which are implemented in the case of capacity reduction. A prioritization

[2]Usually defined as the total benefit to all participants.

[3]This is an initiative used in the US to hold aircraft on the ground to avoid airborne delays.

scheme has also been developed for the routing of flights across an airspace network (Dal Sasso *et al.* , 2019). Here, an optimization model and heuristic algorithm are proposed which attempt to minimize various objectives based on the deviation of a flight from the airline's ideal route, and according to a ranking by each airline of all of its flights.

In the context of slot allocation, the incorporation of airline preferences is comparatively less well studied. A slot exchange mechanism was proposed by Pellegrini *et al.* (2012) to facilitate the secondary trading of slots after the initial allocation according to the IATA WSG (see Section 1.1 above). However, the model was only formulated for a single day, and relied on monetary transactions which is inappropriate for the initial allocation of slots. The idea of encoding preferences by weighting schedule displacement in slot allocation models has also been explored.

In this vein, Jacquillat & Vaze (2018) suggest that airlines could use weights to rank the importance of requests, or have an allocation of credit to apportion across all of their flights. The use of weights in the calculation of schedule displacement gives rise to two issues. Firstly, in order to assign appropriate weights to requests, an airline should have some knowledge of which requests are likely to be displaced due to high demand. It would be wasteful for an airline to heavily weight requests which are not liable to be significantly displaced. Secondly, weighting requests does not give any guarantees to an airline regarding how much a request is displaced.

1.3 Contributions

The work in this paper builds upon work previously presented in the extended abstract (Fairbrother & Zografos, 2018). The main contribution of this paper is a two-stage mechanism for slot allocation which incorporates efficiency, fairness and airline preferences, and which implements the main features of the IATA WSG (scheduling for a season and request priorities). In the first stage of the mechanism, a reference schedule is constructed which is fair with regards to how much displacement is allocated to each airline. In the second stage, airlines suggest adjustments to this reference schedule and the schedule is adjusted to satisfy these requested changes as far as possible. The development of both of these stages constitute contributions in themselves.

With regards to the first stage, a new fairness metric is proposed which takes into account each airline's contribution to scheduling infeasibility. We call this *demand-based fairness*. In this way, the new fairness approach overcomes the problem with respect to the previous ones discussed above by not penalizing requests made for off-peak periods. We also identify the issue of *airline Pareto optimality* for (demand and non-demand based) fairness metrics of this type, and how the issue can be overcome. The definition of peak and off-peak periods which is proposed for the slot allocation problem also constitutes a contribution.

The second stage of the mechanism is essentially a slot exchange. It is novel in that it is, to our knowledge, the first slot exchange mechanism proposed for an entire scheduling season. It is also novel in how adjustments to the schedule are proposed by airlines. Rather than propose concrete conditional exchanges, airlines instead use a "displacement budget" in the form of a "credit" allocated to each airline to be used to describe how much displacement it would prefer each flight to have. The mechanism then adjusts the schedule to meet requested adjustments as far as possible subject to constraints which ensure airlines do not lose out by participating in the mechanism. Such an approach allows airlines to flexibly specify their preferences in a simple way.

2 Summary of Overall Mechanism

The mechanism proposed in this paper consists of two main stages. In the first stage the mechanism constructs a fair *reference schedule* based on demand for slots in peak periods. During this stage, calculations must first be performed to determine for each period in the scheduling season whether demand is peak or off-peak. In the second stage of the mechanism the airlines specify adjustments to the reference schedule, and the mechanism then adjusts this to meet as many of these preferences as possible. Our scheme is similar to that of Bertsimas & Gupta (2015) who propose a two-stage mechanism for assigning routes to flights in an ATFM problem. Their mechanism also constructs a reference schedule before adjusting it to better satisfy airline priorities.

The main aspects of our scheme are illustrated in Figure 1. Each process within the scheme is represented by a box, and this is annotated with the number of the section in which this is detailed.

For the purpose of constructing a fair reference schedule, we propose a new approach to fairness. The approach taken in this paper is an enhancement of the proportionality principle of Zografos & Jiang

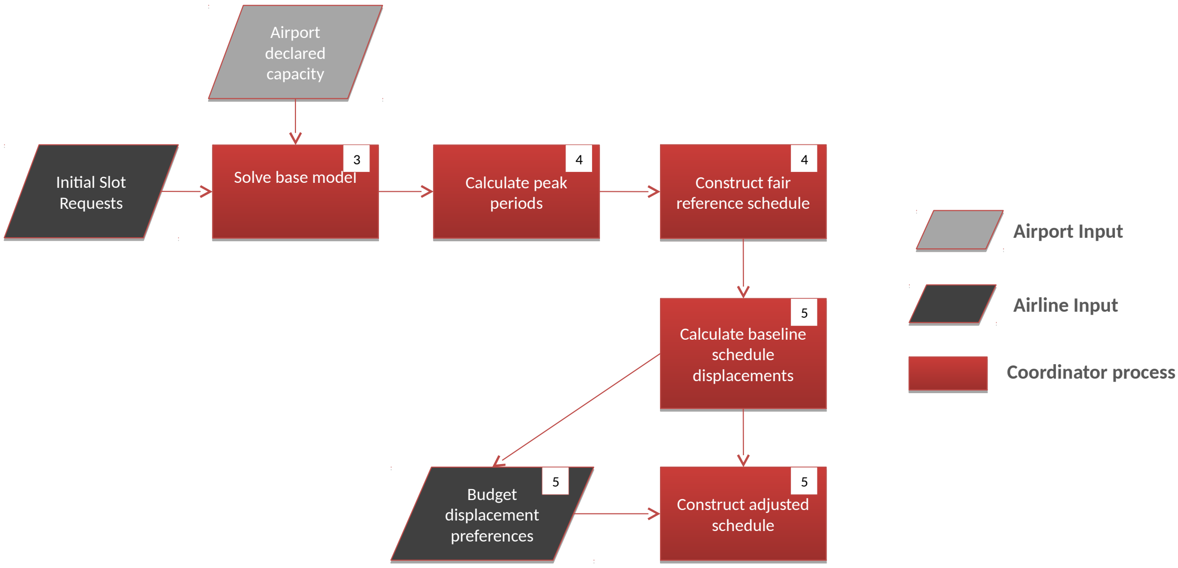


Figure 1: Overall slot allocation scheme

(2016). In our approach every period in the scheduling season is marked as either peak or off-peak, depending on whether or not there is any slack capacity, and our new fairness principle requires that the proportion of total displacement allocated to an airline be approximately equal to the proportion of requests it makes during peak periods.

The prioritization stage of our slot allocation mechanism adjusts the reference schedule according to the airlines' preferences. The reference schedule is used to calculate for each request a *baseline displacement* and the total baseline displacement for each airline is referred to as its *displacement budget*. The airlines must specify for each request a *preferred displacement* such that the sum of the preferred displacements of their requests is at least equal to its displacement budget. In this way, airlines can specify a smaller preferred displacement than the baseline for requests which should be prioritized in return for specifying larger preferred displacements for requests which are not a priority. The mechanism then adjusts the reference schedule so that the requests are allocated a displacement as close to the airlines' preferred values as possible. This is done in a way which guarantees that the displacement of a request will not be greater than the airline's preferred displacement, or the baseline displacement if this is larger.

Unlike the method of weighting the displacement, as suggested as a way of prioritizing requests by Jacquillat & Vaze (2018), our prioritization mechanism is done with respect to a reference schedule, and therefore the airlines can specify their preferences in the knowledge of which of their requests are at risk of a large displacement. Our prioritization mechanism also does not require the airlines to reveal any of their operating costs to the slot coordinator. Finally, our mechanism allows airlines to specify their preferences in a much more flexible way than other schedule adjustments approaches such as 2-swap (e.g. see Bertsimas & Gupta (2015)).

3 Base slot allocation model

In this section we define key terminology and notation related to our approach, and present an integer linear program which forms the core of our mechanism and which is extended in later sections.

3.1 Basic concepts and terminology

The slot allocation problem takes place over a *scheduling season*. The scheduling season consists of a set of days $d \in \mathcal{D}$ with each day divided into a set of equally-sized *(coordination) time intervals* $t \in \mathcal{T} = [0, T-1]$. For the airport which we use for numerical testing in Section 7, the coordination time interval has a length of 15 minutes, but for many other airports it may be 5 or 10 minutes. We reserve the term *time period* to mean a pair $(d, t) \in \mathcal{D} \times \mathcal{T}$ which corresponds to a time interval for a particular day in the scheduling season. An arrival (departure) slot is a permission to use airport infrastructure for the purposes of landing (take-off) during a given time period (d, t).

For the purposes of this paper, a *request series*, $m \in \mathcal{M}$, is a set of requests made by an airline for a series of arrival (or departure) slots for the time interval $t_m \in \mathcal{T}$, on days $\mathcal{D}_m \subseteq \mathcal{D}$, and by *request*, we refer to an individual request for an arrival (or departure) slot which is made as part of a series. All requests in a single request series must be allocated slots for the same time interval. The purpose of request series is thus to allow airlines to create consistent schedules. Note that some request series may only consist of a single request. An individual request is referred to by a pair of indices (m, d) where $m \in \mathcal{M}$ and $d \in \mathcal{D}_m$. Note that our definition of request series is more general than that given in the IATA WSG, where a request series comprises of requests of a period over at least five weeks. Our definition is only necessary for the purposes of exposition, and we process requests according to the definition in the IATA WSG.

A *request pair*, $(m_1, m_2) \in \mathcal{P}$, consists of an arrival request series, m_1 and a subsequent departure request series, m_2, which are to be operated by the same aircraft, and which must be allocated slots compatible with an aircraft's turnaround time. Specifically, for a request pair $(m_1, m_2) \in \mathcal{P}$, the absolute difference between the time intervals of the slots allocated to the arrival and departure must be greater than the *minimum turnaround time*, $\underline{l}_{m_1 m_2}$, and less than the *maximum turnaround time*, $\bar{l}_{m_1 m_2}$. We assume throughout that all request series are made as part of a request pair.

The number of slots which can be allocated over a sequence of consecutive time periods is limited by the *rolling capacity constraints*. For a duration c, and time period (d, t), the number of arrival slots which can be allocated for the time intervals $\{t, \ldots, t + c - 1\}$ on day d is at most α_c^{dt}. Similarly, the number of departure slots, and the total number of arrival and departure slots allocated in these periods are at most β_c^{dt} and γ_c^{dt} respectively.

A *schedule* is an allocation of slots to the set of request series $\mathcal{M}$, and is *feasible* if it satisfies airport capacity and turnaround constraints. By *displacement* we mean the absolute difference measured between the requested time intervals of a request series, and the slots allocated to that request series. That is, if a request series m is allocated slots at time interval t, then the displacement, denoted f_m^t, is $|\mathcal{D}_m||t - t_m|$. The *total displacement* of a schedule, or *schedule displacement*, is the sum of displacements over all requests. The aim of the base slot allocation model is to construct a feasible schedule which minimizes total displacement.

3.2 Optimization model

We present a modified version of the model by Zografos *et al.* (2012). The sets, parameters, and variables used in this model and its extensions are given in Table 1. The full optimization model is given in (1)-(9). We call this the *base model*.

The main decision variables in this model, x_m^t, are binary and indicate whether or not a request series $m \in \mathcal{M}$ is assigned slots at $t \in \mathcal{T}$. The objective in (1) minimizes the total displacement of the constructed schedule. The constraints (2) ensure that each request series is allocated a set of slots.

Sets	
$\mathcal{A}$	set of airlines
$\mathcal{M}$	set of request series
$\mathcal{M}^{A(D)} \subset \mathcal{M}$	set of all arrival (departure) requests series
$\mathcal{M}_a \subset \mathcal{M}$	set of requests series made by airline $a \in \mathcal{A}$
$\mathcal{M}_a^{A(D)} \subset \mathcal{M}^{A(D)}$	set of arrival (departure) request series made by airline a
$\mathcal{D}$	set of days in scheduling season
$\mathcal{D}_m \subseteq \mathcal{D}$	set of days for which request series m must be scheduled
$\mathcal{P} \subset \mathcal{M} \times \mathcal{M}$	set of request series pairs (m_a, m_d) where m_d corresponds to the departure of an aircraft following the arrival m_a
$\mathcal{C}$	set of durations of rolling capacity constraints
$\mathcal{T} = \{1, \ldots, T\}$	set of coordination time intervals
Parameters	
$t_m \in \mathcal{T}$	requested time for request series m
r_a	proportion of peak requests made by airline a
f_m^t	displacement from assigning slot t to request series m
$\underline{l}_p$ $(\bar{l}_p)$	minimum (maximum) turnaround time for request pair $p \in \mathcal{P}$
a_m^d	binary value indicating whether request series $m \in \mathcal{M}$ includes a request for day $d \in \mathcal{D}$
α_c^{ds}	rolling capacity on arrivals for period $[s, s+c-1]$ on day $d \in \mathcal{D}$
β_c^{ds}	rolling capacity on departures for period $[s, s+c-1]$ on day $d \in \mathcal{D}$
γ_c^{ds}	rolling capacity on total movements for period $[s, s+c-1]$ on day $d \in \mathcal{D}$
Decision variables	
x_m^t	indicates whether request series m is assigned slots at t
y_{dt}^A	aggregate number of arrival slots allocated for period (d,t)
y_{dt}^D	aggregate number of departure slots allocated for period (d,t)

Table 1: Notation used for single airport slot allocation model

$$\text{minimize} \quad \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} f_m^t x_m^t \tag{1}$$

$$\text{subject to} \quad \sum_{t \in \mathcal{T}} x_m^t = 1, \; m \in \mathcal{M} \tag{2}$$

$$y_{dt}^A = \sum_{m \in \mathcal{M}^A} a_m^d x_m^t, \; t \in \mathcal{T}, \; d \in \mathcal{D} \tag{3}$$

$$y_{dt}^D = \sum_{m \in \mathcal{M}^D} a_m^d x_m^t, \; t \in \mathcal{T}, \; d \in \mathcal{D} \tag{4}$$

$$\sum_{t \in [s, s+c-1]} y_{dt}^A \leq \alpha_c^{ds}, \; c \in \mathcal{C}, \; d \in \mathcal{D}, \; s \in [0, T-c] \tag{5}$$

$$\sum_{t \in [s, s+c-1]} y_{dt}^D \leq \beta_c^{ds}, \; c \in \mathcal{C}, \; d \in \mathcal{D}, \; s \in [0, T-c] \tag{6}$$

$$\sum_{t \in [s, s+c-1]} y_{dt}^A + y_{dt}^D \leq \gamma_c^{ds}, \; c \in \mathcal{C}, \; d \in \mathcal{D}, \; s \in [0, T-c] \tag{7}$$

$$\underline{l}_{m_1, m_2} \leq \sum_{t \in \mathcal{T}} t x_{m_1}^t - \sum_{t \in \mathcal{T}} t x_{m_2}^t \leq \bar{l}_{(m_1, m_2)}, \; (m_1, m_2) \in \mathcal{P} \tag{8}$$

$$x_m^t \in \{0, 1\}, \; m \in \mathcal{M}, \; t \in \mathcal{T} \tag{9}$$

The constraints (3) and (4) define the auxiliary variables y_{dt}^A and y_{dt}^D to be, respectively, the number of arrival and departure slots allocated for time period (d, t). Although this model could be formulated without these, we include them for notational convenience. These are used in Section 4 for the characterization of peak and off-peak periods.

The constraints (5)–(7) enforce rolling capacity constraints. In particular, constraints (5) limit the rolling *number of arrivals*, constraints (6) limit the rolling *number of departures*, and constraints (7) limit the rolling *total number of movements*. For notational simplicity we have assumed that all three types of capacity constraints use the same set of durations. This leads to no loss of generality, as we can simply take the capacity to be infinity in cases where there is no limit specified for a particular duration and movement type. Finally, constraints (8) bound the turnaround time for each request pair $(m_1, m_2) \in \mathcal{P}$.

4 Construction of a fair reference schedule

In this section we present a new approach to fairness which distinguishes between peak and off-peak time periods, and explain how this is used to construct a reference schedule for our slot allocation mechanism. In Sections 4.1 and 4.2 we define the related concepts of slackness, saturation and peak and off-peak periods. In Section 4.3 we use peaks periods to define a new fairness metric. In Section 4.4 we describe a technical issue related to incorporating fairness into the base model. Finally in Section 4.5 we present an algorithm for constructing a reference schedule.

4.1 Slackness and saturation

Previously, we said that a *peak period* could be thought of as one where there is no slack capacity, that is, if no extra slots for the period can be allocated without breaking a capacity constraint. We call such a period *saturated*. We now give precise mathematical definitions to these concepts. Note that since arrival and departure slots are subject to different capacity constraints, slackness and saturation must be defined differently for each type of runway movement.

Definition 4.1. *Suppose* $\tilde{\mathbf{y}} := (\tilde{y}^M_{dt})^{M\in\{A,D\}}_{d\in\mathcal{D},t\in\mathcal{T}}$ *is the aggregate schedule corresponding to a (possibly infeasible) solution to problem* (1)–(9). *The arrival, departure and total movement slacks at a time period* (d,t) *are defined to be, respectively:*

$$\bar{\alpha}_{dt}(\tilde{\mathbf{y}}) = \min_{c\in\mathcal{C}} \min_{s\in[\max(t-c+1,0),t]} \left(\alpha^{dt}_c - \sum_{\tilde{s}=s}^{s+c-1} \tilde{y}^A_{d\tilde{s}}\right)_+,$$

$$\bar{\beta}_{dt}(\tilde{\mathbf{y}}) = \min_{c\in\mathcal{C}} \min_{s\in[\max(t-c+1,0),t]} \left(\beta^{dt}_c - \sum_{\tilde{s}=s}^{s+c-1} \tilde{y}^D_{d\tilde{s}}\right)_+,$$

$$\bar{\gamma}_{dt}(\tilde{\mathbf{y}}) = \min_{c\in\mathcal{C}} \min_{s\in[\max(t-c+1,0),t]} \left(\gamma^{dt}_c - \sum_{\tilde{s}=s}^{s+c-1} \left(\tilde{y}^A_{d\tilde{s}} + \tilde{y}^D_{d\tilde{s}}\right)\right)_+,$$

where $u_+ = \max(u, 0)$.

The arrival slack gives the number of extra arrival slots for time period (d,t) that can be allocated before an arrival constraint is broken. Departure and total movement slack can be interpreted similarly.

Definition 4.2. *Suppose* $\tilde{\mathbf{y}}$ *is an aggregate schedule. Then, the corresponding* saturation *indices are defined as follows:*

$$A^t_d(\tilde{\mathbf{y}}) = \begin{cases} 1 & \text{if } \bar{\alpha}_{dt}(\tilde{\mathbf{y}}) = 0 \text{ or } \bar{\gamma}_{dt}(\tilde{\mathbf{y}}) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$D^t_d(\tilde{\mathbf{y}}) = \begin{cases} 1 & \text{if } \bar{\beta}_{dt}(\tilde{\mathbf{y}}) = 0 \text{ or } \bar{\gamma}_{dt}(\tilde{\mathbf{y}}) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

A period (d,t) *is said to be* arrival-saturated *with respect to* $\tilde{\mathbf{y}}$ *if* $A^t_d(\tilde{\mathbf{y}}) = 1$. *Similarly* (d,t) *is* departure-saturated *with respect to* $\tilde{\mathbf{y}}$ *if* $D^t_d(\tilde{\mathbf{y}}) = 1$.

A period is arrival (departure) saturated if no more arrival (departure) slots can be allocated without breaking a capacity constraint.

4.2 Characterisation of Peak Periods

A natural definition for arrival (departure) peak periods are those which are arrival (departure) saturated with respect to the *default schedule*. By default schedule, we mean the schedule where every request series is allocated its preferred slots. Mathematically, the default schedule is characterized by the following solution:

$$\tilde{x}^t_m = 1 \Leftrightarrow \; t = t_m \text{ for all } m \in \mathcal{M}$$

The periods which are saturated with respect to the default schedule are those periods where the number of requests equals or exceeds the available capacity. However, this definition for peak periods is flawed. The default schedule will typically be infeasible (if it were feasible there would be no need to solve the slot allocation problem). When constructing a feasible schedule, some requests will be displaced, which may lead to other periods becoming saturated. These periods cannot accommodate any extra slots without breaking capacity constraints, and so should be considered to be peak. A definition for peak periods which overcomes this issue is the following:

Definition 4.3 (Period sensitivity)**.** *A period (d, t) is said to be arrival (departure) sensitive if an single extra arrival (departure) request for that period increases the optimal total displacement as calculated in problem* (1)–(9)*.*

Given that the problem (1)–(9) is an integer program with a complex structure, the calculation of whether each period is arrival or departure sensitive would largely have to be done by brute force. That is, for each day and time period we would have to adjust the requests and re-solve the problem. We will instead use an approximation of sensitive periods to define peak and off-peak periods, which can be calculated easily after solving the problem (1)–(9) once, and which is conservative, in the sense that any period that is sensitive would be deemed a peak period.

Definition 4.4. *Let $(\mathbf{x}^*, \mathbf{y}^*)$ be an optimal solution to the slot allocation problem* (1)–(9)*. Then, a period (d, t) is an arrival peak period if it is arrival-saturated with respect to $\mathbf{y}^*$. Similarly, (d, t) is a departure peak period if it is departure-saturated with respect to $\mathbf{y}^*$.*

The downside of this definition of peak periods is that it depends on the optimal solution which may not be unique. However, as well as having the advantage of being easy to calculate, the next result shows that for any optimal solution, the arrival (departure) peak periods must cover all arrival (departure) sensitive periods.

Proposition 4.5. *Suppose the time period (d, t) is arrival (departure) sensitive, then $A^t_d(\mathbf{y}^*) = 1$ ($D^t_d(\mathbf{y}^*) = 1$) for any optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ of the base problem* (1)–(9)*.*

The proof of this result can be found in Appendix A. Unfortunately the converse of the result is not true and so we also give a counter-example to this in the same appendix.

One could imagine a peak period indicator which not only indicated the presence of peak demand, but also the severity. A possible way of measuring this would be to look at the actual change in optimal solution value rather than simply whether it increases or not. However, such an approach would likely be very computationally expensive, and we are not aware of any numerically efficient approximation approaches.

4.3 Demand-based fairness

For notational convenience, unless it is necessary, from now on we will suppress the dependency on an aggregate schedule $\mathbf{y}^*$ from our notation for peak periods and simply write them as A^t_d and D^t_d. Using the decision variables and parameters from the base model, the amount of schedule displacement for each airline can be expressed as follows:

$$s_a = \sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f^t_m x^t_m = \sum_{t \in \mathcal{T}} |\mathcal{D}_m||t - t_m| x^t_m \qquad \text{, with}$$
$$S = \sum_{a \in \mathcal{A}} s_a$$

for the aggregate displacement across airlines, in other words, the total displacement.

Given arrival and departure peak indicators, the number of peak requests made by an airline a is given by

$$C_a := \sum_{m\in\mathcal{M}_a^A}\sum_{d\in\mathcal{D}_m} A_d^{t_m} + \sum_{m\in\mathcal{M}_a^D}\sum_{d\in\mathcal{D}_m} D_d^{t_m}$$

and proportion of peak requests made by airline a is:

$$r_a := \frac{C_a}{\sum_{a\in\mathcal{A}} C_a}$$

assuming that $\sum_{a\in\mathcal{A}} C_a \neq 0$.

We propose that the proportion of total displacement allocated to an airline should be similar to its proportion of peak requests. In this way, airlines will not be penalized for requests made during off-peak periods. We call this the *peak request proportionality principle.* The new fairness index for an airline is defined to be the proportion of total displacement allocated to an airline divided by its proportion of peak requests, with special cases to account for the cases where an airline makes no peak requests:

$$\mu_a := \begin{cases} \frac{\frac{s_a}{S}}{r_a} & \text{if } r_a \neq 0 \\ 1 & \text{if } r_a = 0 \text{ and } s_a = 0, \\ \infty & \text{if } r_a = 0 \text{ and } s_a \neq 0. \end{cases} \tag{10}$$

Based on the peak request proportionality principle, μ_a should be interpreted as follows:

$$\begin{aligned} &\mu_a = 1.0 && \text{airline } a \text{ is fairly treated,} \\ &\mu_a < 1.0 && \text{airline } a \text{ is favoured,} \\ &\mu_a > 1.0 && \text{airline } a \text{ is disfavoured.} \end{aligned}$$

We refer to these indices as *demand-based* fairness indices. Note that it is possible to construct other fairness indices based on the peak request proportionality principle. We use this one as it is similar to indices already used in the literature, and the value of the index has the following easy interpretation: if $\mu_a = q$ then airline a receives q times more (or less) total schedule displacement than it should. We do not claim that this is necessarily the best demand-based fairness index that could be used, but only aim to demonstrate the advantages of demand-based fairness over non-demand-based fairness.

Fairness metrics can be constructed from fairness indices to measure the overall fairness of a schedule. Throughout this paper we use a fairness metric proposed by Zografos & Jiang (2017) called the maximum deviation from absolute fairness (MDA). This is defined as follows:

$$\mu_{MDA} = \max_{a\in\mathcal{A}} |\mu_a - 1|. \tag{11}$$

This metric measures fairness by focusing on the worst case. Again, it is possible to construct other fairness metrics from the airline fairness indices, but this is not the focus of the paper. Two other fairness metrics constructed from fairness indices are proposed in Zografos & Jiang (2017), namely the deviation from average fairness and the Gini coefficient.

Solving the base model will not necessarily yield a schedule which is fair according to the peak request proportionality principle. Fairness can be incorporated into the slot allocation model through the minimization of a fairness metric. In the case of the MDA fairness metric, this is a non-linear function with respect to the decision variables (x_m^t), which could render the optimization model intractable if they were included directly in the objective function. This problem can be circumvented by using these metrics in constraints as these can be linearized as follows:

$$\mu_{MDA} \leq \epsilon \iff |s_a - Sr_a| \leq \epsilon r_a S,\ a \in \mathcal{A}. \tag{12}$$

We call the base model with the above fairness constraints the *ϵ-fairness model.*

In the case where $r_a = 0$, the constraint above becomes $s_a = 0$. The next result shows that constraint (12) is satisfiable in this case, as long as the airline requests arrival and departure slots which are compatible with the required turnaround time.

Proposition 4.6. *Suppose* $(\mathbf{x}^*, \mathbf{y}^*)$ *is an optimal solution to problem* (1)–(9)*. Let* (A_d^t) *and* (D_d^t) *be the arrival and departure saturation indicators for* $\mathbf{y}^*$ *and define* $\widetilde{\mathcal{M}} = \bigcup_{a\in\mathcal{A}:r_a=0}\mathcal{M}_a$*. Assume the following conditions hold:*

(i) For each $(m_1, m_2) \in \mathcal{P}$ we have $\underline{l}_{m_1 m_2} \leq t_{m_2} - t_{m_1} \leq \bar{l}_{m_1 m_2}$,

(ii) For each $(m_1, m_2) \in \mathcal{P}$, if $m_1 \in \widetilde{\mathcal{M}}$ then $D_d^{t_{m_2}} = 0$ for all $d \in \mathcal{D}_{m_2}$,

(iii) For each $(m_1, m_2) \in \mathcal{P}$, if $m_2 \in \widetilde{\mathcal{M}}$ then $A_d^{t_{m_1}} = 0$ for all $d \in \mathcal{A}_{m_1}$.

Then $\mathbf{x}^*$ satisfies $s_a = 0$ for all $a \in \mathcal{A}$ such that $r_a = 0$.

The proof for this proposition can again be found in the Appendix A. Assumption (i) of Proposition 4.6 holds as long as all airlines make requests consistent with the required turnaround times for the aircraft. Assumptions (ii) and (iii) automatically hold if the request series of every request pair are made by the same airline, that is, if $(m_1, m_2) \in \mathcal{P}$ implies that $m_1, m_2 \in \mathcal{M}_a$ for some $a \in \mathcal{A}$. In this case, for $(m_1, m_2) \in \mathcal{P}$ we have $m_1 \in \widetilde{\mathcal{M}}$ if and only if $m_2 \in \widetilde{\mathcal{M}}$.

Since we compare demand-based fairness with the previously proposed non-demand based fairness approach of Zografos & Jiang (2019) in our numerical experiments, we now define the the latter. For an airline a, the *non-demand-based fairness index* ρ_a is defined to be the proportion of schedule displacement allocated to that airline divided by the proportion of requests made by the airline:

$$\rho_a = \frac{\frac{s_a}{S}}{\frac{\sum_{m \in \mathcal{M}_a} |\mathcal{D}_m|}{\sum_{m \in \mathcal{M}} |\mathcal{D}_m|}}. \tag{13}$$

This can be interpreted in the same way as μ_a. Similarly, the non-demand-based MDA fairness metric is defined to be:

$$\rho_{\text{MDA}} = \max_{a \in \mathcal{A}} |\rho_a - 1|.$$

This can be linearized when used in a constraint in the same way as the demand-based MDA metric.

To demonstrate the importance of using demand-based fairness over non-demand-based fairness, we plot the proportions of requests and peak requests for each airline for some real request data (to be described in more detail in Section 7.1) in Figure 2. Due to the very large number of airlines operating at the airport (80), for clarity, we only plot these proportions for airlines who have proportion of peak requests greater than 0.01. This plot shows that for many airlines, the proportions of requests and peak requests can differ greatly, meaning that some airlines would be significantly penalized for their off-peak requests under a non-demand based fairness approach.

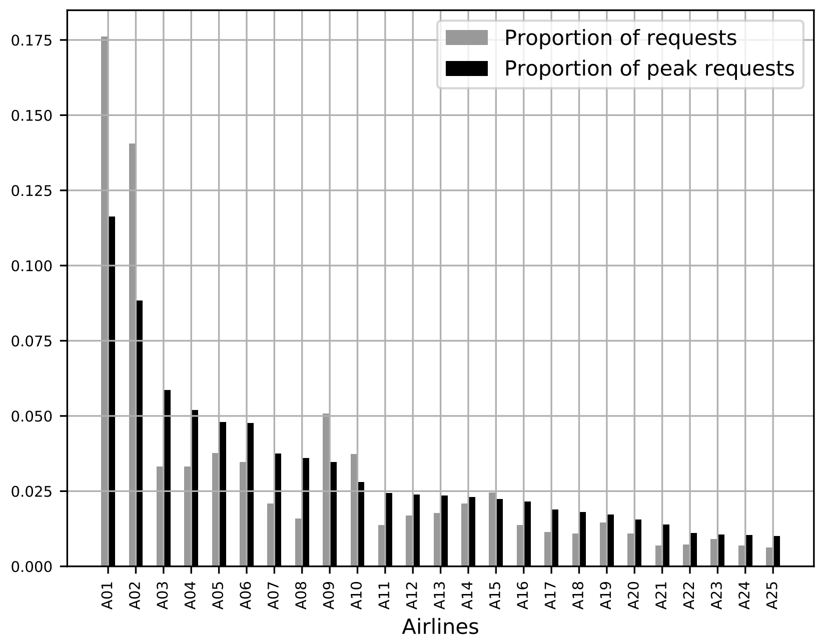


Figure 2: Proportions of requests and peak requests for major airlines at a coordinated airport

4.4 Airline Pareto Optimality

Solving the base slot allocation problem with fairness constraints (12) yields a solution which is (weakly) Pareto optimal with respect to objectives of total displacement (1) and MDA fairness (11). In this section we consider a different type of Pareto optimality.

Suppose $\tilde{\mathbf{x}}$ and $\bar{\mathbf{x}}$ are two feasible schedules. Then, we say that the schedule $\tilde{\mathbf{x}}$ dominates $\bar{\mathbf{x}}$ with respect to airline displacements if:

$$\sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f_m^t \tilde{x}_m^t \leq \sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f_m^t \bar{x}_m^t \qquad \text{for all } a \in \mathcal{A},$$

with at least one inequality strict. That is, one schedule dominates another if the displacements allocated to each airline are less than or equal to those of the other schedule, and where the displacement is strictly less for at least one airline. We say that a feasible schedule is *airline Pareto optimal* if it is not *dominated* with respect to airline displacements by any other feasible schedule. If a schedule is airline Pareto optimal, no airline's displacement can be reduced without increasing it for another.

It is a natural requirement for fairness schemes to yield a solution which is Pareto optimal in this sense (Nash, 1950; Kalai & Smorodinsky, 1975), that is, with respect to the objectives of the participants in the system. A schedule which is not airline Pareto optimal allocates more displacement than necessary to some airline, and so will likely be unacceptable. Unfortunately, the use of both demand and non-demand fairness constraints with the base model will not necessarily yield an airline Pareto optimal schedule. The following toy example demonstrates that fairness constraints which are too tight can yield a schedule which is not airline Pareto optimal.

Example 4.7. *Suppose that we have two airlines each requesting a slot for the same time period, for a single day, and where there is a total slot limit of one for each time period. The optimal solution in this case is to displace one of the airlines' requests by one time period, which results in an MDA fairness value (for both demand and non-demand-based fairness indices) of 1. If we include a fairness constraint where we require the MDA fairness value to be less than one, then the only way to satisfy this is to displace both requests. This is illustrated in Figure 3. Clearly, this is not an airline Pareto optimal solution.*

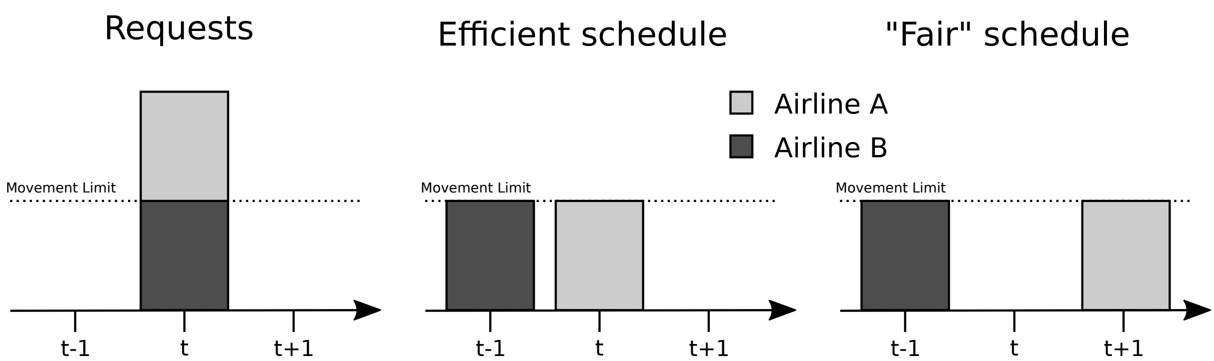


Figure 3: Toy example to illustrate how schedules which are not airline Pareto optimal arise from use of fairness constraints

Fortunately it is easy to check whether a feasible schedule is airline Pareto optimal. The process for doing this is presented in Algorithm 1 below.

The algorithm first calculates the schedule displacement for each airline and then constructs a new base slot allocation problem where the displacement for each airline is bounded by these values. Optimizing this new model yields a new feasible schedule whose airline displacements do not exceed the previous airline displacements. The algorithm then checks for each airline whether their displacement is less than the previous displacement. If this is the case, then the new solution dominates the previous one, so the original schedule is not airline Pareto optimal and the algorithm returns false. If the new displacement for all airlines is the same as previously, then no feasible schedule which dominates the original exists, so the original schedule is airline Pareto optimal and the algorithm returns true.

4.5 Efficient Frontier and selection of a reference schedule

There is a trade-off between the total displacement and the fairness of a schedule. To examine this trade-off we can calculate the efficient frontier between these two conflicting objectives using the ϵ-

input : $\bar{\mathbf{x}}$ feasible schedule
Construct base slot allocation model;
for $a \in \mathcal{A}$ **do**
  $\Sigma_a \leftarrow \sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f_m^t \bar{x}_m^t$;
  Add constraint $\sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f_m^t x_m^t \leq \Sigma_a$;
Solve slot allocation model and let $\tilde{\mathbf{x}}$ be the optimal solution;
for $a \in \mathcal{A}$ **do**
  if $\sum_{m \in \mathcal{M}_a} \sum_{t \in \mathcal{T}} f_m^t \tilde{x}_m^t < \Sigma_a$ **then**
    return False;
return True;

Algorithm 1: Checks whether a given feasible schedule is airline Pareto optimal

constraint method. After calculating the efficient frontier between total displacement and fairness, one has potentially several schedules from which to choose a reference schedule. Following our discussion in Section 4.4, we first eliminate from consideration all schedules which are not airline Pareto optimal. As explained above, a non-airline Pareto optimal schedule will likely be unacceptable to the participating airlines. In addition, an airline Pareto optimal solution is required to use the displacement budget mechanism described in Section 5.

To select a schedule from the available airline Pareto optimal schedules, we use a simpler scheme which we refer to as the *acceptable price of fairness*. Denote by S^* the optimal value to the base model, and by $S^*(\epsilon)$, the optimal value for the ϵ-fairness problem. Then the *price of fairness* of the optimal solution yielded by the latter problem is defined to be:

$$\frac{S^*(\epsilon) - S^*}{S^*},$$

that is, it is the relative deterioration in the optimal total displacement when introducing an MDA fairness constraint with right-hand side ϵ. For $\alpha > 0$, the α-acceptable price of fairness criterion selects the most fair (airline Pareto optimal) schedule which has a price of fairness of less than α. The parameter α is chosen according to the airport coordinator's and airlines' tolerances to a higher total displacement. This criterion for selecting a fair solution was suggested by Bertsimas *et al.* (2012) who apply it to a different fairness scheme which also depends on the specification of a fairness parameter.

Algorithm 2 describes the overall procedure for calculating the fair reference schedule given the airline requests and airport capacities. It first calculates peak periods, before running a modified version of the ϵ-constraint method which terminates once the price of fairness has been exceeded. Note that at each iteration of this algorithm ϵ is decremented by a factor δ rather than by a fixed value. We choose to use this rule as the optimal total displacement and required solution time become a lot more sensitive to changes in the value of ϵ when ϵ is small.

Although other rules for selecting a reference schedule are possible, the acceptable price of fairness rule has two major advantages. Firstly, unlike a voting process, the procedure does not require any extra input from the airlines. Secondly, the method does not require one to calculate the entire efficient frontier which can save considerable computation time. This is because the ϵ-constraint method used to calculate the efficient frontier consists of solving the ϵ-fairness problem for a decreasing sequence of values of ϵ until the problem becomes infeasible. As the fairness constraints become tighter the optimal total displacement increases. Therefore, once the price of fairness has been exceeded, it will be exceeded for all smaller values of ϵ and so there is no need to solve more fairness problems. As will be seen in the numerical tests in Section 7, the time saved from not having to solve the ϵ-fairness problem for small values of ϵ is particularly large.

5 The displacement budget mechanism

The incorporation of fairness into the slot allocation model ensures that each airline is allocated a reasonable amount of displacement. However, there may be some flexibility in how the displacement assigned to an airline may be distributed amongst its requests. The acceptability of a schedule can therefore be improved by taking into account an airline's preferences regarding which requests should be displaced and

input : airline requests, airport coordination parameters, $0 < \delta < 1$ decrement rate of ϵ, α acceptable price of fairness

output: $\bar{\mathbf{x}}$ reference schedule

Solve base model;
$(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \leftarrow$ optimal solution, $S \leftarrow$ optimal total displacement;
Calculate peak periods (A_d^t) and (D_d^t) from $\bar{\mathbf{y}}$;
Calculate fairness indices and initial MDA fairness ϵ using (10) and (11);
repeat
  $\epsilon \leftarrow \delta\epsilon$;
  Solve ϵ-constraint problem;
  if *Problem infeasible* **then**
    Break;
  else if *Optimal solution found* **then**
    Set $\tilde{\mathbf{x}}, S_\epsilon$ be optimal schedule and optimal solution value;
    if $\frac{S_\epsilon - S}{S} > \alpha$ **then**
      break;
    if $\tilde{\mathbf{x}}$ *is airline Pareto optimal* **then** $\bar{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}$;
return $\bar{\mathbf{x}}$;

Algorithm 2: Calculation of reference schedule

by how much. We propose a mechanism, which we call the *displacement budget mechanism*, which adjusts the reference schedule in order to better satisfy the airlines' displacement preferences. Throughout this section we use $\bar{\mathbf{x}}$ to denote the reference schedule which can be calculated using Algorithm 2.

In Section 5.1 we describe the overall displacement budget mechanism. In Section 5.2 we describe constraints on how airlines specify their preferences. Finally, in Section 5.3 we present a model to facilitate the specification of airline preferences for use in the displacement budget mechanism.

5.1 Overall Model

The fair reference schedule is used to determine the *baseline displacement* for each request in the reference schedule, namely

$$\sigma_m := \sum_{t \in \mathcal{T}} f_m^t \bar{x}_m^t.$$

The *displacement budget* for an airline $a \in \mathcal{A}$ is defined to be the total baseline displacement for all of the airline's requests, that is:

$$\Sigma_a := \sum_{m \in \mathcal{M}_a} \sigma_m$$

The mechanism requires airlines to specify for each request m a *preferred displacement* δ_m, namely an expressed preference that the request be scheduled in the set:

$$\{t \in \mathcal{T} : f_m^t \leq \delta_m\}.$$

The preferences must be specified so as to satisfy the *displacement budget constraint*:

$$\sum_{m \in \mathcal{M}_a} \delta_m \geq \Sigma_a. \tag{14}$$

The displacement budget thus represents the amount of displacement an airline should be allocated, and the mechanism requires airlines to specify their preferences for how this should be allocated amongst its requests. If an airline specifies a preferred displacement less than the baseline displacement for some requests, it must specify greater preferred displacements for others. Besides the displacement budget constraint, other constraints apply to the setting of preferred displacements and these are discussed in Section 5.2. A systematic method that airlines could use for setting preferred costs is presented in Section 5.3.

A request for which an airline specifies a smaller preferred displacement than the baseline displacement is called an *improvement request.* We refer to all other requests as *non-improvement requests.* The aim of the displacement budget mechanism is to adjust the reference schedule to satisfy improvement requests as far as possible.

The mechanism works by solving a modified version of the base model. For each $m \in \mathcal{M}$ we introduce variables π_m to measure the change in displacement with respect to the reference schedule. For improvement requests we also introduce variables w_m which measure the amount of displacement by which preferred displacements are missed.

$$\text{lexmin} \left(\sum_{m \in \mathcal{M}: \delta_m < \sigma_m} w_m, \sum_{m \in M} \sum_{t \in T} f_m^t x_m^t \right) \tag{15}$$

subject to constraints (2)–(9)

$$\pi_m = \sum_m^t f_m^t x_m^t - \sigma_m \text{ for all } m \in \mathcal{M} \tag{16}$$

$$w_m \geq (\sigma_m - \delta_m) + \pi_m \text{ for each } m \in \mathcal{M} \text{ such that } \delta_m < \sigma_m \tag{17}$$

$$\sum_{m \in \mathcal{M}_a} \pi_m \leq \nu \Sigma_a \text{ for each } a \in \mathcal{A} \tag{18}$$

$$\pi_m \leq 0 \text{ for each } m \in \mathcal{M} \text{ such that } \delta_m < \sigma_m \tag{19}$$

$$\pi_m \leq \delta_m - \sigma_m \text{ for each } m \in \mathcal{M} \text{ such that } \delta_m \geq \sigma_m \tag{20}$$

$$w_m \geq 0 \text{ for each } m \in \mathcal{M} \text{ such that } \delta_m < \sigma_m \tag{21}$$

The objective (15) is the lexicographic minimization of the total amount of displacement by which preferred displacements are missed for improvement requests, followed by the total displacement. The second objective is required because minimizing the sum of the w_m variables does not reduce displacement for non-improvement requests. Constraints (16) and (17) respectively encode the definitions of the decision variables (π_m) and (w_m) described above.

Constraints (18) are added in order to ensure that no airline's total displacement increases significantly from this mechanism. The parameter $\nu \geq 0$ here controls the strictness of this requirement. A higher value of ν gives greater flexibility to satisfy improvement requests, but also means airlines may have to accept a greater schedule displacement. These constraints essentially ensure that the displacement budget mechanism is carried out in a fair manner. It may therefore be natural to ask why we do not use the same fairness constraints (12) introduced in Section 4.3, or similar fairness constraints based on how improvements (w_m) are distributed amongst the airlines rather than total displacement. The previous fairness metrics were defined in terms of proportion of schedule displacement as we do not know a priori the exact fair amounts of total schedule displacement that should be allocated to each airline. In the displacement budget mechanism, the reference schedule is by construction a fair baseline, and so considering deviations from this reference is reasonable, and leads to a more computationally tractable optimization problem. In addition, the fairness constraints (12) do not prevent increases in displacement for all airlines.

The constraints (19) and (20) ensure that all requests are assigned a displacement cost at most $\max\{\delta_m, \sigma_m\}$. That is, by participating in this mechanism, an airline will receive no more displacement for a request than its baseline displacement, or its preferred displacement if this is greater.

A key feature of the displacement budget mechanism is that it does not require airlines to reveal any potentially commercially sensitive operating costs.

5.2 Constraints on Airline Preferences

In addition to the displacement budget constraint, we impose other constraints on the preferred displacements specified by the airlines. These constraints forbid airlines to specify preferred displacements which are impossible to satisfy exactly. They also prevent airlines assigning excessive amounts of displacements to individual requests. In this way, they not only help to ensure airlines use their displacement budgets more effectively, but also mitigate against them gaming the mechanism.

Displacement Size: Recall that the displacement of a request series m is given by $|\mathcal{D}_m||t - t_m|$ where t is the time interval of the slots allocated to m. In order to satisfy a preferred displacement δ_m exactly, we must therefore have:

$$\delta_m \bmod |\mathcal{D}_m| = 0. \tag{22}$$

Turnaround constraints: The following proposition states that if there is an upper bound on the turnaround time for a request pair, then the difference between preferred displacements for the request series composing this pair is also bounded.

Proposition 5.1. *Suppose* $(m_1, m_2) \in \mathcal{P}$ *and for the preferred displacements* δ_{m_1} *and* δ_{m_2} *there exist* $T_{m_1}, T_{m_2} \in [1, T]$ *such that* $f_{m_1}^{T_{m_1}} = \delta_{m_1}$, $f_{m_2}^{T_{m_2}} = \delta_{m_2}$, *and* $T_{m_2} - T_{m_1} \leq \bar{l}_{m_1 m_2}$. *Then it follows that*

$$|\delta_{m_1} - \delta_{m_2}| \leq 2|\mathcal{D}_{m_1}|\bar{l}_{m_1 m_2}. \tag{23}$$

Proof. Suppose that there exist T_{m_1}, T_{m_2} satisfying the conditions of the proposition for given $(m_1, m_2) \in \mathcal{P}$. It then follows that

$$||T_{m_1} - t_{m_1}| - |T_{m_2} - t_{m_2}|| \leq |T_{m_1} - T_{m_2}| + |t_{m_1} - t_{m_2}| \leq 2\bar{l}_{m_1 m_2}.$$

Multiplying through by $|\mathcal{D}_m|$ then yields that

$$|f_{m_1}^{T_{m_1}} - f_{m_2}^{T_{m_2}}| = |\delta_{m_1} - \delta_{m_2}| \leq 2|\mathcal{D}_m|\bar{l}_{m_1 m_2},$$

as required. □

Displacement upper bound: It is possible that in order to achieve a reduced delay for some of its requests, an airline may assign a large portion of its total baseline delay to a few requests which it does not value. If these particular requests are made for off-peak periods, then specifying large preferred displacements may not even lead to the requests being displaced more since the displacement budget mechanism will not increase displacement unnecessarily. To avoid airlines gaming the system in this way we must impose restrictions on how this displacement is allocated.

If a request pair is made for unsaturated periods (with respect to the reference schedule) and the requested times are consistent with bounds on turnaround time, then the request pair has been allocated its preferred slots. Moreover, since there is already slack capacity at these time periods, displacing this request pair will not immediately help the coordinator satisfy other improvement requests. Therefore, the airline should not be allowed to use its displacement budget on this request pair. More generally, the preferred displacement allocated to a request pair should not be greater than the displacement from the closest feasible pair of slots in unsaturated time periods with respect to the reference schedule.

The minimum displacement from inserting the request pair $(m_1, m_2) \in \mathcal{P}$ into the reference schedule $\bar{\mathbf{x}}$ is the optimal value to the following optimization problem. This problem is just the base model restricted to allocating slots to (m_1, m_2), and where the capacities have been updated using the aggregate reference schedule $\bar{\mathbf{y}}$.

$$\text{minimize}_{x_{m_1}^t, x_{m_2}^t} \sum_{t \in \mathcal{T}} f_{m_1}^t x_{m_1}^t + f_{m_2}^t x_{m_2}^t \tag{24}$$

$$\text{subject to} \sum_{t \in \mathcal{T}} x_m^t = 1, \; m \in \{m_1, m_2\} \tag{25}$$

$$\sum_{t=s}^{s+c-1} (x_{m_1}^t + \bar{y}_{dt}^A) \leq \alpha_c^{dt}, \; c \in \mathcal{C}, \; d \in \mathcal{D}_{m_1}, \; s \in [0, T-c] \tag{26}$$

$$\sum_{t=s}^{s+c-1} (x_{m_2}^t + \bar{y}_{dt}^D) \leq \beta_c^{dt}, \; c \in \mathcal{C}, \; d \in \mathcal{D}_{m_2}, \; s \in [0, T-c] \tag{27}$$

$$\sum_{t=s}^{s+c-1} (x_{m_1}^t + x_{m_2}^t + \bar{y}_{dt}^A + \bar{y}_{dt}^D) \leq \gamma_c^{dt}, \; c \in \mathcal{C}, \; d \in \mathcal{D}_1, \; s \in [0, T-c] \tag{28}$$

$$\underline{l}_{m_1, m_2} \leq \sum_{t \in \mathcal{T}} t x_{m_1}^t - \sum_{t \in \mathcal{T}} t x_{m_2}^t \leq \bar{l}_{(m_1, m_2)} \tag{29}$$

$$x_{m_1}^t, x_{m_2}^t \in \{0, 1\}, \; \text{for all } t \in \mathcal{T} \tag{30}$$

We call the above formulation the *minimum insertion problem* and the optimal value the *minimal insertion displacement*. Denoting the minimum insertion displacement to this problem by $B_{m_1 m_2}$, we then impose the following constraints on the preferred displacements:

$$\delta_{m_1} + \delta_{m_2} \leq B_{m_1 m_2} \qquad \text{for all } (m_1, m_2) \in \mathcal{P}, \tag{31}$$

To find the minimum insertion displacement $B_{m_1m_2}$, we do not need to solve the minimum insertion problem by integer programming techniques as it can be calculated efficiently via smart enumeration. Details are given in Appendix B.

While (22) and (23) hold immediately for the baseline displacements (i.e. if we set $\delta_m = \sigma_m$), it is not necessarily true that $\sigma_{m_1} + \sigma_{m_2} \leq B_{m_1m_2}$. The following result gives mild conditions under which this does hold.

Proposition 5.2. *Suppose that the reference schedule is airline Pareto optimal and that for each $(m_1, m_2) \in \mathcal{P}$ there exists $a \in \mathcal{A}$ such that $m_1, m_2 \in \mathcal{M}_a$. It follows that*

$$\sigma_{m_1} + \sigma_{m_2} \leq B_{m_1m_2} \qquad \text{for each } (m_1, m_2) \in \mathcal{P}.$$

Proof. We suppose that there exist $(m_1, m_2) \in \mathcal{P}$, both requests arising from airline a, for which $\sigma_{m_1} + \sigma_{m_2} > B_{m_1m_2}$ and obtain a contradiction. Let $T^*_{m_1}, T^*_{m_2}$ be optimal allocated times for the above minimum insertion problem for (m_1, m_2). Now modify the reference schedule so that requests m_1 and m_2 are allocated slots $T^*_{m_1}$ and $T^*_{m_2}$ respectively. This schedule remains feasible (for the base model) and achieves a lower displacement for airline a. This contradicts the airline Pareto optimality of the reference schedule and concludes the proof. □

Remark 5.3. *The assumption that the reference schedule is airline Pareto optimal is already guaranteed by Algorithm 2. Although it is possible that arrival and departure request series in a request pair can be made by different airlines, this rarely occurs in practice.*

Forbidden time periods: It is possible that for some time periods a rolling capacity constraint will have a zero right-hand side. This may occur because of a curfew on flights during the night, or when the slot pool is updated during a hierarchical application of the slot allocation mechanism (see Section 6).

A zero on the right-hand side of an arrival or total movement capacity constraint prevents the allocation of arrival slots for all time periods active in that constraint. This similarly applies to the allocation of departure slots if there is a departure or total capacity constraint with a zero right-hand side.

Denote by $\mathbf{0}$ the *empty schedule*. The set of time periods for which no arrival slots can be allocated consists of those which are arrival-saturated with respect to the empty schedule. We call these *arrival-forbidden periods*. We can similarly define *departure-forbidden periods*. Indicator functions which characterize arrival and departure forbidden periods, which we denote (F^A_{dt}) and (F^D_{dt}) respectively, can thus be defined as follows:

$$F^A_{dt} := A^t_d(\mathbf{0}), \qquad F^D_{dt} := D^t_d(\mathbf{0})$$

Since an airline cannot be allocated a slot in a forbidden period, we require that they do not specify preferred displacements which imply receiving a slot in such a time period. That is, for an arrival request series $m \in \mathcal{M}^A$ we have:

$$\delta_m \neq |D_m|h \text{ if } F^A_{d,(t_m+h)} = 1 \text{ and } F^A_{d,(t_m-h)} = 1 \text{ for any } d \in \mathcal{D}_m, \tag{32}$$

and for a departure request series $m \in \mathcal{M}^D$ we have:

$$\delta_m \neq |D_m|h \text{ if } F^D_{d,(t_m+h)} = 1 \text{ and } F^D_{d,(t_m-h)} = 1 \text{ for any } d \in \mathcal{D}_m. \tag{33}$$

5.3 Calculation of Airline Preferences

Airlines are able to specify preferred displacements in an arbitrary way as long as the displacement budget constraint, together with the constraints in Section 5.2 are satisfied. However, it may not be clear to an airline how best to distribute its displacement in a way which matches its priorities. In this section we propose a model for distributing an airline's displacement. Although our primary motivation here is to be able to simulate an airline's preferences in order to test the displacement budget mechanism, this model could be used as a decision support tool for the airlines.

The model is formulated and solved for each airline a, and assumes for each $m \in \mathcal{M}_a$ and $t \in \mathcal{T}$, a cost c^t_m is incurred to the airline when request m is allocated slot time t. We call these *airline displacement costs*. Preferences are then calculated by solving the following ILP which has binary decision variables $(z^t_m)^{t\in\mathcal{T}}_{m\in\mathcal{M}_a}$ which, like the (x^t_m) variables in the slot allocation model, select a slot time for each request series.

$$\text{minimize} \sum_{m\in\mathcal{M}_a}\sum_{t\in\mathcal{T}} c_m^t z_m^t \tag{34}$$

$$\text{subject to} \sum_{m\in\mathcal{M}_a}\sum_{t\in\mathcal{T}} f_m^t z_m^t \geq \Sigma_a \tag{35}$$

$$\underline{l}_{m_1,m_2} \leq \sum_{t\in\mathcal{T}} t z_{m_1}^t - \sum_{t\in\mathcal{T}} t z_{m_2}^t \leq \bar{l}_{(m_1,m_2)},\ (m_1,m_2)\in\mathcal{P} \tag{36}$$

$$\sum_{t\in\mathcal{T}} f_{m_1}^t z_{m_1}^t + \sum_{t\in\mathcal{T}} f_{m_2}^t z_{m_2}^t \leq B_{m_1 m_2} \text{ for all } (m_1,m_2)\in\mathcal{P} \tag{37}$$

$$z_m^t = 0 \qquad \text{for all } m\in\mathcal{M}_a^A \text{ and } t\in\mathcal{T} \text{ such that } F_{dt}^A = 0 \text{ for some } d\in\mathcal{D}_m \tag{38}$$

$$z_m^t = 0 \qquad \text{for all } m\in\mathcal{M}_a^D \text{ and } t\in\mathcal{T} \text{ such that } F_{dt}^D = 0 \text{ for some } d\in\mathcal{D}_m \tag{39}$$

$$\sum_{t\in\mathcal{T}} z_m^t = 1 \qquad \text{for all } m\in\mathcal{M}_a \tag{40}$$

$$z_m^t \in \{0,1\} \qquad \text{for all } m\in\mathcal{M}_a,\ t\in\mathcal{T},$$

The objective (34) seeks to minimize the total airline displacement costs if all of the airline's displacement preferences are exactly met. The preferred displacements, which will be used as input to the displacement budget mechanism, are then the displacements corresponding to the optimal solution $\mathbf{z}^*$ to the model above, that is:

$$\delta_m = \sum_{t\in\mathcal{T}} f_m^t z_m^{t*}. \tag{41}$$

We now explain how this approach ensures that all constraints on preferred displacements detailed in Section 5.2 hold. The constraint (22) holds directly from the definition of (41) which (along with (40) and the binary constraints) ensures that δ_m is of the form $f_m^t = |\mathcal{D}_m||t - t_m|$ for some $t \in T$. The turnaround constraints (23) hold by constraints (36) and the application of Proposition 5.1. The constraint (35) and definition (41) enforce the displacement budget constraint (14). Similarly, (37) and (41) ensures that the displacement upper bound constraint (31) holds. Finally, the constraints (38) and (39) prevent the airline from specifying forbidden periods for their preferred slots and so ensure that (32)–(33) hold.

Given that the above model used airline costs (or disutilities), a natural question concerns why we do not incorporate these into the base model (1)–(9) to directly construct a schedule which incorporates airline preferences. This suggestion is problematic for several reasons: firstly these data are likely to be commercially sensitive, and the airlines may not want to reveal them. Secondly, it would require a consistent definition of cost across all airlines which may not be possible due to the different operating models of airlines. Thirdly, given that the IATA WSG allows airlines to change details of their requests, such as aircraft type, after the allocation of slots, it may be possible for an airline to game the system by overstating its costs of displacement.

The model above implicitly assumes that the value of a slot for one flight is independent of another. In reality the value of some slots may be linked. For example, for pairs of flights with a potentially large number of connecting passengers, pairs of slots which have compatible connection times will be more valuable than those which do not. Although the development of a model reflecting this is beyond the scope of the current work, we note that existing models in the aviation literature could be adapted for this purpose. For instance, Pita *et al.* (2013) proposed an optimization model for airline scheduling which takes into account expected passenger demand and airline cooperation and competition to select slots in a way which maximizes expected profits.

Facilitation of specification of airline preferences: The displacement budget mechanism requires airlines to specify preferred displacements for potentially many requests subject to several sets of constraints. This process of selecting these preferences could be facilitated by decision support software.

Such software could consist of a graphical interface which allows the user to manually adjust the preferred displacement allocated to each request series (initially the baseline displacements) and an optimization tool, which constructs a set of preferred displacements given cost functions for each movement. The graphical interface could also be used to adjust displacements using output from the optimization tool.

The graphical interface should display all request pairs whose displacement can be adjusted from the baseline. For each of these request pairs the software should display which displacements are feasible. After an adjustment is made, the allowable displacements for other request pairs, which are subject to the displacement budget constraint, should be updated.

The optimization tool could use the model presented in Section 5.3. This requires the user to specify displacement cost functions for each request series. This information would only be used in the optimization, and would not be shared with any other parties. To simplify the process of constructing these functions, the optimization tool could make available some parametric forms. For example, in the numerical tests in Section 7 we use airline displacement costs given by:

$$c_m^t = R_m |\mathcal{D}_m| |t - t_m|^\eta$$

where R_m is a weight representing the relative importance of m, and $\eta > 0$ represents the airline's aversion to large displacements.

6 Hierarchical application of slot allocation mechanism

The mechanism described in this paper can take account of priority classes by allocating slots to each group in a separate stage. That is, we allocate slots for the highest priority class first, update the slot pool, then allocate slots to the next highest class, and so on until slots have been allocated for all priority classes. We call this *hierarchical slot allocation*, and for each priority group we run the whole slot allocation mechanism to construct the schedule.

Updating the slot pool consists of updating the capacity constraints. If $\tilde{\mathbf{y}}$ is the aggregate schedule for an allocation of slots for the previous priority class, then the right-hand sides of the capacity constraints are updated as follows:

$$\begin{aligned}
\alpha_c^{dt} &\leftarrow \alpha_c^{dt} - \sum_{\tilde{s}=s}^{s+c-1} \tilde{y}_{d\tilde{s}}^A \\
\beta_c^{dt} &\leftarrow \beta_c^{dt} - \sum_{\tilde{s}=s}^{s+c-1} \tilde{y}_{d\tilde{s}}^D \\
\gamma_c^{dt} &\leftarrow \gamma_c^{dt} - \sum_{\tilde{s}=s}^{s+c-1} \left(\tilde{y}_{d\tilde{s}}^A + \tilde{y}_{d\tilde{s}}^D\right)
\end{aligned}$$

The lexicographic approach to incorporate priority classes into slot allocation, as deployed by Ribeiro *et al.* (2018) and described in Section 1.2, cannot be used within our mechanism since we use a two-stage approach to allocate slots to each priority level rather than solve a single optimization problem.

7 Numerical tests

In this section we test our slot allocation mechanism using data for a coordinated airport. The purpose of these tests is to demonstrate how the mechanism performs for a realistic slot allocation problem, and to also investigate how this performance depends on the different parameters of the mechanism. Testing performance while varying mechanism parameters is problematic when solving the problem hierarchically. This is because when the problem is solved for lower priority groups, as well as depending on the mechanism parameters, performance will also depend on the slot pool, and these effects cannot easily be disentangled. In order to experiment with the mechanism settings we therefore solve the problem non-hierarchically. Recall, that solving the problem non-hierarchically means disregarding the priority groups and allocating slots to all requests simultaneously. Using the best settings found in these experiments, we will then demonstrate the mechanism using the full hierarchical approach.

This section is organised as follows. In Section 7.1 we describe our general experimental set-up. In Section 7.2 we present our experiments for the non-hierarchical approach, and in Section 7.3 we demonstrate our mechanism using the hierarchical approach. Finally in Section 7.4 we discuss and compare the results of these two approaches.

7.1 Set-up

7.1.1 Problem Data

We solve the slot allocation problem for a medium-sized airport. The rolling capacity parameters for this problem are given in Table 2. Since the shortest duration of a rolling capacity constraint is 15 minutes, we use this as the length of our coordination time intervals. There are thus 96 time intervals in each day, and since there are 212 days in this scheduling season, there are a total of 20352 time periods in the scheduling season.

Arrivals	Departures	Total	
60 minutes	60 minutes	15 minutes	60 minutes
8	12	5	20

Table 2: Rolling capacity constraints for airport

The number of requests series and individual requests for each priority group is shown in Table 3. Note that due to lack of historical request times for the "changes to historic" requests, we amalgamate the "historic" and "changes to historic" requests into a single priority group.

Priority Class	Request Series	Requests
Historics	748	14034
New entrants	76	2680
Others	1290	16328
Total	2114	33042

Table 3: Request data

The required turnaround time for an aircraft depends on the aircraft model, and airport services, but in general a turnaround time of one hour is sufficient for most aircraft. However, there are request pairs in our data set where the requested arrival and departure times are separated by less than this period. In order to use demand-based fairness, Proposition 4.6 requires that the requested slot times of request pairs are compatible with minimum turnaround times. We therefore use the following formula to set the minimum turnaround time:

$$\underline{l}_{m_1 m_2} = \min\{4, t_{m_2} - t_{m_1}\}$$

where 4 coordination time intervals corresponds to one hour. Since it is not strictly necessary, and we do not have the required data, we do not enforce upper bounds on turnaround time.

7.1.2 Airline displacement costs

Airline displacement cost data are not available, and therefore they have been constructed synthetically. For the purposes of these numerical tests, we assume that the airline displacement costs are given by

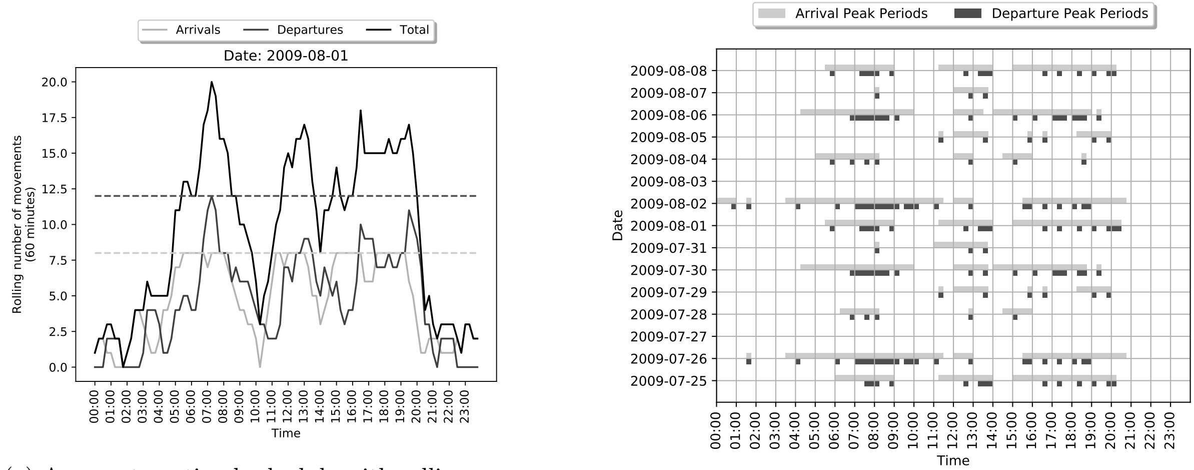
$$c_m^t = R_m |\mathcal{D}_m| |t - t_m|^\eta.$$

This is an increasing function with respect to deviation from the requested slot time. A superlinear function is used to represent the airlines' costs to reflect the fact that airlines are much more sensitive to larger displacements. A superlinear cost function was also used in the slot allocation model in the paper by Castelli *et al.* (2012).

To capture the relative importance of each request, this function is weighted by $R_m|\mathcal{D}_m|$ where R_m is the normalised passenger capacity of the aircraft. In particular, $R_m = \frac{N_m}{456}$ where N_m is the number of seats of the aircraft for which the request is made, and 456 is the maximum aircraft capacity over all requests. Passenger-based costs were used for a similar purpose in the context of a ground-holding problem in Vossen & Ball (2006).

7.1.3 Computer and solver set-up

The integer linear programs are solved using Gurobi 8.10 (Gurobi Optimization, 2017), on a desktop computer that has an Intel(R) Core(TM) i5-4690 processor with 4 cores. Many of the rolling capacity constraints (5)–(7) will be redundant and binding only for periods of high slot demand. Therefore, to solve



(a) Aggregate optimal schedule with rolling period of one hour

(b) Peak periods

Figure 4: Rolling aggregate schedule and peak periods for non-hierarchical problem

the problem more efficiently, we use a branch-and-cut scheme, where the rolling capacity constraints are added as lazy constraints through a solver callback. To do this, the Gurobi LazyConstraints parameter is set to true.

The most computationally demanding part of the slot allocation mechanism is the calculation of the efficient frontier. The use of fairness constraints can increase the required solution time by orders of magnitude. Whereas the base model for the non-hierarchical problem takes around five to ten minutes to solve, the fairness model can take hours, and is sometimes not solved at all within a reasonable time. We therefore limit the solution time for each problem to two hours using the Gurobi TimeLimit parameter.

As described in Section 4.5, we use the α-acceptable price of fairness criterion to select the fair reference schedule. Using this approach, we can terminate our ϵ-constraint algorithm once the price of fairness has been exceeded. However, we need not solve an ϵ-fairness problem to optimality to verify that it yields a schedule which exceeds the acceptable price of fairness. The solution algorithm can be terminated once the current lower bound on the optimal value exceeds the acceptable price of fairness. We do this by setting the Gurobi parameter BestBdStop appropriately. All other Gurobi parameters are left with their default values.

7.2 Non-hierarchical results

In this section we allocate slots using a non-hierarchical approach, that is, we allocate all slots simultaneously, disregarding priority classes.

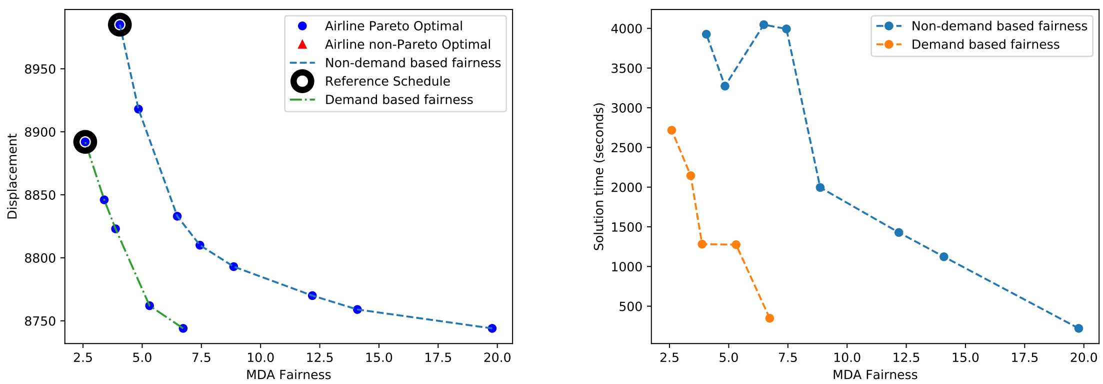
7.2.1 Peak Periods and Fairness

To calculate demand-based fairness we first need to solve the base model and calculate the peak periods. Figure 4a shows the aggregate hourly schedule of an optimal solution for the base model on the busiest day in the scheduling season. For a specified time, this gives the numbers of arrivals, departures, and total movements which take place over the next hour. The dashed lines represent the hourly capacities for each type of constraint. The arrival and departure peak periods corresponding to this optimal schedule are shown in Figure 4b for a 15-day period including the busiest day. This figure shows that departure peak periods are more intermittent than arrival peak periods, which is due to the fact that arrival capacity limits are significantly lower than departure capacity limits. This difference in peak periods for arrivals and departures emphasises the importance of treating these two types of requests separately.

In Figure 5a we have plotted the total displacement-MDA fairness efficient frontiers. Although we use a reference schedule constructed using demand-based fairness, for the purposes of comparison we also calculate the efficient frontier for non-demand-based fairness. The points encircled by black lines correspond to the schedules which would be selected under the acceptable price of fairness rule. We use a decrement rate of $\delta = 0.8$ for these calculations. Note that in this figure we plot the actual values of MDA fairness found by solving these problems, rather than the values of ϵ.

Recall that the smaller the value of the MDA fairness metric, the fairer the schedule. Figure 5a shows that the trade-off for displacement versus fairness is better for demand-based fairness, that is, we can construct fairer schedules with smaller values of total displacement. This suggests that for demand-based fairness, it is easier to redistribute schedule displacement to airlines with a high proportion of peak requests rather than to airlines with a high proportion of total requests.

The solution of the ϵ-fairness problems is by far the most computationally expensive component of the proposed slot allocation mechanism. For both demand-based and non-demand-based fairness, the ϵ-constraint algorithm terminated because the time limit, rather than the acceptable price of fairness, was exceeded. In Figure 5b we plot the solution time for each ϵ-fairness problem. This shows that the tighter the fairness constraints, the more difficult the problem becomes to solve. This suggests that the acceptable price of fairness criterion is therefore quite appropriate as it massively reduces the required computation time to calculate the fair reference schedule. This plot also shows that the solution of the non-demand-based fairness problem is more difficult to solve than the demand-based version. Again, this may be because it is easier to redistribute displacement to airlines with a high proportion of peak requests.



(a) Displacement-MDA fairness efficient frontiers

(b) Solution times for ϵ-fairness problems

Figure 5: Comparison of demand-based and non-demand based fairness schemes

7.2.2 Displacement budget mechanism

We now study the performance of the displacement budget mechanism. We denote by $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ the reference schedule and aggregate reference schedule, and by $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ the solutions returned from the displacement budget formulation in (15)– (21). Since the displacement budget mechanism only takes a few seconds to run, we do not present the run times.

Distribution of displacement adjustments: We first study the distribution of schedule adjustments. Although we only present results when the mechanism is run for parameters $\nu = 0.1$, and $\eta = 1.8$, we have observed the same trends when the mechanism is run under different settings. The results are plotted in Figure 6. This shows the number of requests for every possible combination of requested adjustments ($\frac{\sigma_m - \delta_m}{|\mathcal{D}_m}|$) and actual adjustments ($\frac{\sum_{t\in\mathcal{T}} f^t_m \tilde{x}^t_m - \sigma_m}{|\mathcal{D}_m|}$).

This shows that for the majority of requests (around 75%), the airlines ask for and receive no adjustment to their baseline displacement. We also observe that the range of actual adjustments is much narrower than the range of demanded adjustments. Although a significant number of requests are made for large improvements, up to 16 coordination time intervals (4 hours), no request has its displacement reduced by more 6 coordination time intervals (1.5 hours), and the majority of actual improvements are less than or equal to 2 coordination time intervals (0.5 hours). This suggests that a better strategy for airlines participating in the displacement budget mechanism is to demand smaller adjustments.

We observe that although there are many requests whose displacement can be increased (those with negative requested adjustments), only a fraction (around 11%) of these actually receive extra displacement. Interestingly, some of the requests receive a greater reduction in displacement than demanded,

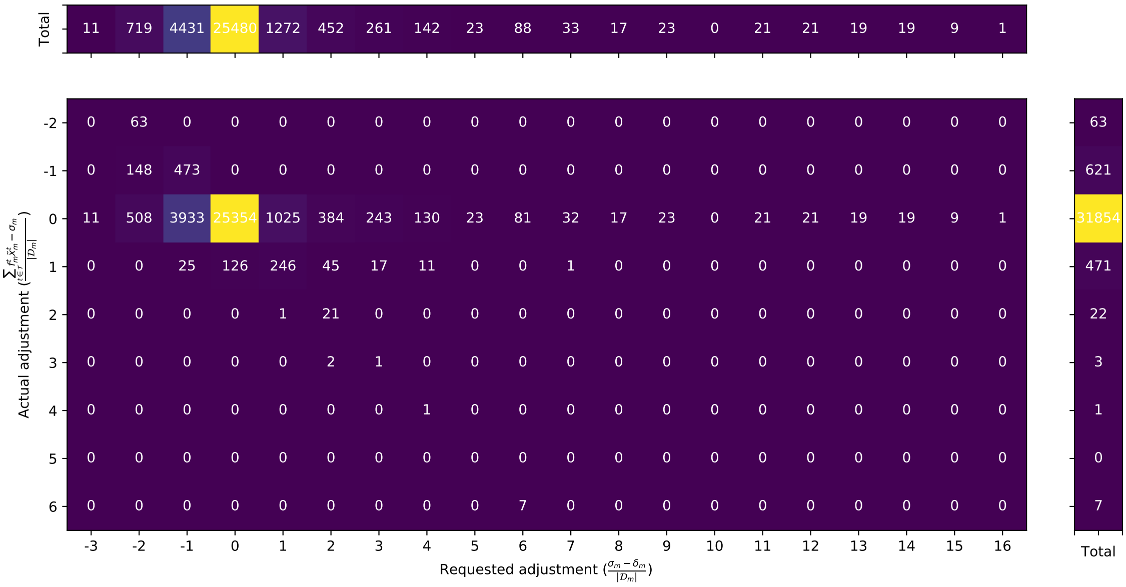


Figure 6: Distribution of adjustments from displacement budget mechanism

including even a small number of requests for which no or a negative adjustment was asked for. This is because when some the mechanism increases displacement for some requests in order to satisfy improvement requests, more capacity is created than necessary, and the displacement of other requests can be reduced.

Sensitivity analysis with respect to ν: We now study the performance of the methodology with respect to the parameter ν which controls the proportion of extra displacement each airline can be allocated. We run the displacement budget mechanism for $\nu = 0.0, 0.1, \ldots, 0.5$ with $\eta = 1.8$ as in the previous test. We look at three different attributes of performance. The first is the total improvements, $\sum_{m \in \mathcal{M}: \delta_m < \sigma_m} \tilde{w}_m$, the second is the schedule displacement, $\sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} f_m^t \tilde{x}_m^t$, and the third is the total airline cost $\sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} c_m^t \tilde{x}_m^t$. The results are shown in Figure 7.

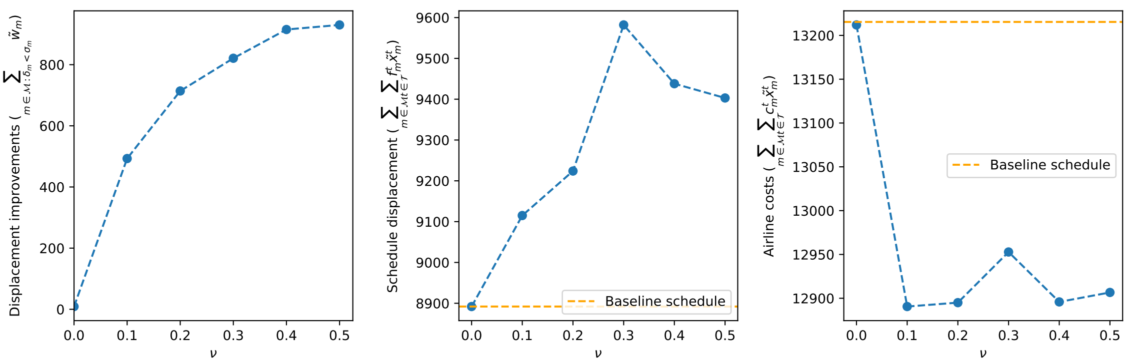


Figure 7: Performance of displacement budget mechanism for different values of ν

This shows that when no increase in displacement is allowed ($\nu = 0$) no schedule improvements can be made. Since no improvements are made for $\nu = 0$, the schedule displacement and airline costs do not change from the baseline. As ν increases, the total amount of schedule improvements increases.

Interestingly, although the schedule displacement increases initially with ν, for larger ν it then decreases. This means that for higher values of ν, increases in displacement are being redistributed to fewer airlines. This unfair redistribution of displacement for higher ν could mean that the size of the largest displacements for some airlines is increasing. Since large displacements contribute more to airline costs, this may explain why airline costs seem to flatline for higher values of ν even though the total schedule improvements are increasing.

Sensitivity analysis with respect to η The parameter η controls how averse airlines are to larger displacements: the larger the value of η the more averse they are. Although in reality a coordinator would have no control over η, some useful insights can be gained about how the mechanism performs with respect to the airlines' preferences by varying this parameter. In this experiment we run the displacement budget mechanism for $\eta = 0.0, 0.2, \ldots, 2.6$, with fixed $\nu = 0.1$. For each value of η, we plot the relative reduction in total airline costs with respect to the reference schedule, that is

$$\frac{\sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} c_m^t \bar{x}_m^t - \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} c_m^t \tilde{x}_m^t}{\sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}} c_m^t \bar{x}_m^t}$$

where, recall, $c_m^t = |\mathcal{D}_m| R_m |t - t_m|^\eta$. The results are plotted in Figure 8.

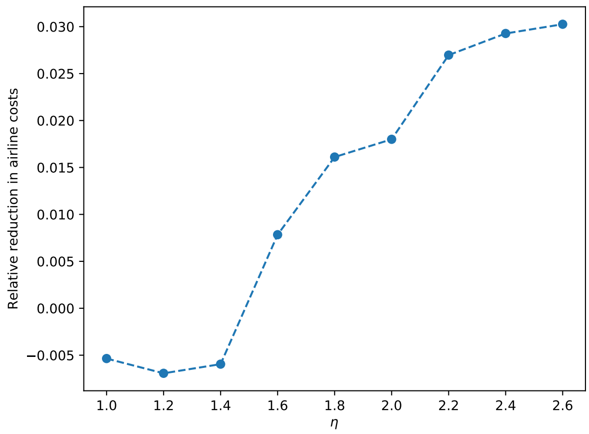


Figure 8: Relative reduction in total airline costs for different values of η

This figure shows that for small values of η, the displacement budget mechanism results in a schedule with greater total airline costs. This is because the displacement budget mechanism increases the overall schedule displacement, and for small values of η, the total airline costs and schedule displacement as functions have a similar shape. For higher values of η, the reduction in airline costs increases, and reaches around 3% for $\eta = 2.6$. This is likely due to the mechanism reducing the largest displacements, which for high values of η, contribute much more to the total airline costs than small displacements. These results suggest that the more averse the airlines are to large displacements, the greater the benefit of the displacement budget mechanism.

7.3 Hierarchical results

For this experiment we solve the slot allocation problem using the hierarchical approach described in Section 6. The peak periods and efficient frontiers for each priority group are shown in Figure 9. In the case of historical and new entrant requests, the ϵ-constraint algorithm terminates when the acceptable price of fairness is exceeded, while for the other requests it terminates due to the time limit. A summary of the results of the displacement budget mechanism at each stage is shown in Table 4.

Historic requests From Figure 9a we can see that the peak periods seem to be largely confined to Tuesday and Friday morning. By solving the efficient frontier, we were able to improve the MDA fairness from around 6.0 (by solving the base model) to 1.0 after which the price of fairness is exceeded.

In the displacement budget mechanism although there a total of 243 individual requests for improvement, no improvements are made. This is to be expected as only a very small number of the historical requests receive any displacement, which means that the opportunities for slot exchange are very small.

There is a very small change in the total airline costs due to the displacement budget mechanism returning a different schedule from the reference schedule.

New entrants There are relatively few new entrants request series (76 out of 2114), and so the peak periods for these requests largely coincide with the peak periods for the historic request series. Since the slots for the historical request series have already been allocated at this point, these periods do not have any slack capacity and are therefore considered to be forbidden for the purposes of the displacement budget mechanism. Solving the base model yields a schedule which is already very fair with MDA fairness of around 1.0. Although fairer schedules can be found, none of these are airline Pareto optimal. For the displacement budget mechanism, there are 28 individual improvement requests and the mechanism is able to make total improvements of 27. This is done without increasing the schedule displacement for new entrants, and the new schedule has significantly lower airline costs.

Others The peak periods for the other requests resemble those which were calculated for the solution of the non-hierarchical problem, but here many of the periods are forbidden. For the calculation of the efficient frontier, the value of the MDA fairness can be improved from an initial MDA fairness value of around 3.3 to 1.1 while the solutions remain airline Pareto optimal. In the displacement budget mechanism, a total displacement improvement of 384 out of 2120 individual improvement requests, which leads to an improvement in airline costs of 1.2%.

	Historics	New Entrants	Others
Baseline schedule displacement	459	291	12867
New schedule displacement	459	291	13174
Number of improvement requests	243	28	2128
Total displacement improvements	0	27	384
Baseline airline costs	335.80	138.39	18395.04
New airline costs	336.39	98.45	18179.16

Table 4: Summary of displacement budget mechanism for hierarchical slot allocation

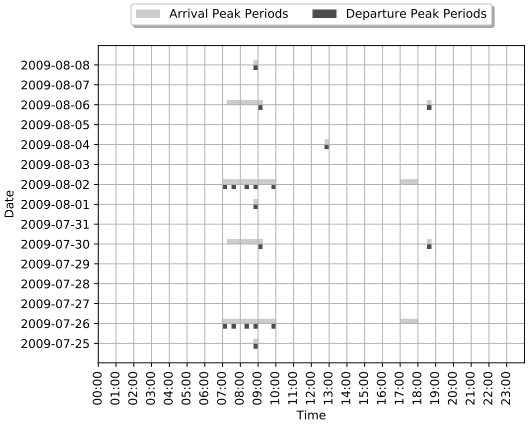
7.4 Discussion

We note that for both the non-hierarchical and hierarchical approaches, the fairness of a schedule can in almost all cases be greatly improved with only small increases in the total displacement of a schedule. Most ϵ-fairness problems which were solved yielded an airline Pareto optimal schedule. The only exceptions to this were schedules with MDA fairness less than 1. Inspection of airline Pareto optimal solutions with MDA fairness near 1 reveals that there are a number of airlines with a very small proportion of peak requests, but which have zero displacement, which means the deviation from absolute fairness of these airlines' fairness indices is 1. When $\epsilon < 1$ these airlines must be assigned some displacement, and the solutions become non-airline Pareto optimal. We surmise that this is because, although these airlines have some requests during peak periods, no request in these periods needs to be displaced. This further motivates the need for the development of a more sophisticated measure of peak periods as discussed at the end of Section 4.2.

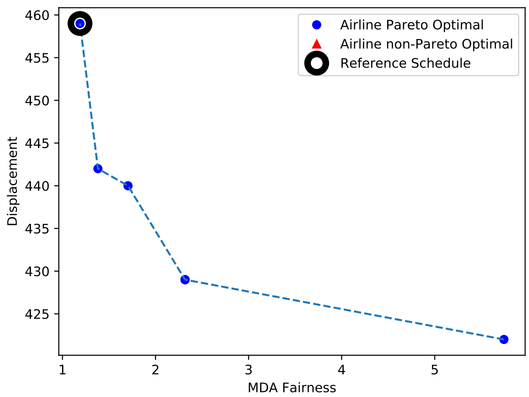
In terms of schedule improvements and reduced airline costs, the displacement budget seems to be slightly more effective for the non-hierarchical than the hierarchical approach. This is to be expected as allocating slots to higher priority groups greatly restricts the slots which can be allocated to lower priority groups, and also allocating slots to smaller groups of requests reduces the opportunities for exchanging slots. This reasoning suggests that the mechanism may be more effective for larger and more congested airports, where there would be more opportunities for slot exchange. Also, if we could relax the requirement for all requests in a request series to be allocated slots at the same time, we would expect the overall performances of displacement budget to improve due to the greater flexibility in scheduling.

8 Conclusions

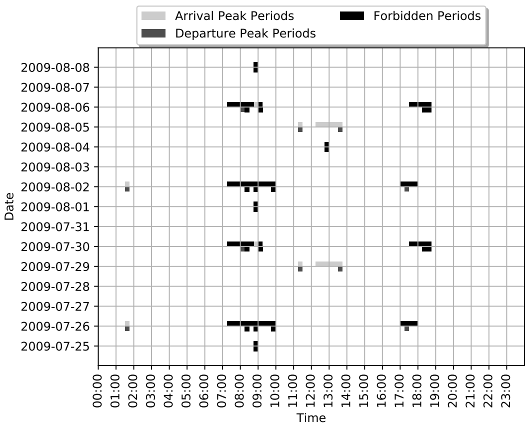
In this paper we propose a mechanism for allocating slots at congested airports which incorporates efficiency, fairness and airline preferences. The mechanism consists of constructing a fair reference schedule, and then adjusting this to better match the participating airlines' priorities. The construction of the fair



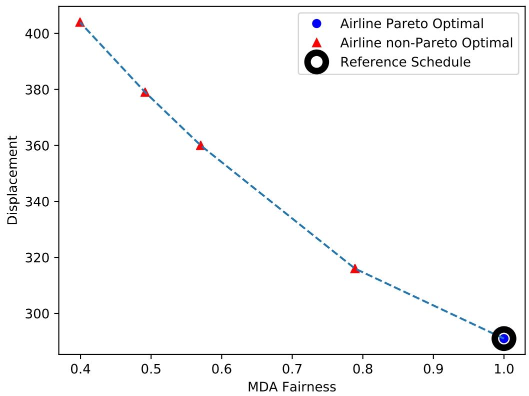
(a) Historics peak periods



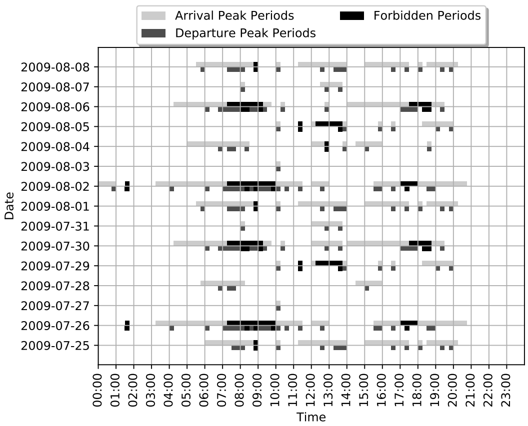
(b) Historics efficient frontier



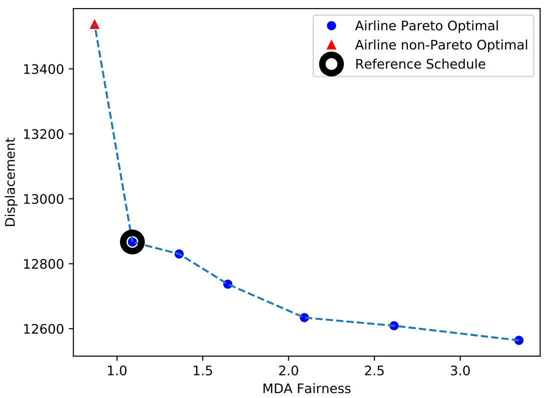
(c) New entrants peak periods



(d) New entrants efficient frontier



(e) Others peak periods



(f) Others efficient frontier

Figure 9: Peaks periods and efficient frontiers for hierarchical slot allocation

reference schedule uses a new fairness measure which was developed on the principle that the proportion of schedule displacement an airline receives should depend on the number of requests it makes during peak demand periods. The approach was tested using request data from a coordinated airport. Our fairness approach outperformed the previous fairness metric of Zografos & Jiang (2019) in terms of trade-off with schedule displacement, and required computation time. The displacement budget mechanism was shown to be able to make a significant number of improvements requested by airlines without large increases in schedule displacement.

There are numerous possible extensions to our approach. Firstly, the base model could be modified to model the IATA WSG to a higher fidelity. For example, although the airport on which we tested our model had more than sufficient capacity to allocate slots to all requests, in general this will not be true. Therefore the model should be extended to allow some requests to be rejected (as is done in (Ribeiro *et al.* , 2018)). In the case where there are likely to be a lot of rejected requests, there may also be a need to enforce the fair distribution of rejections to airlines in the model.

Another important extension to this work would be the development of a peak indicator which not only indicates whether or not a period has peak demand, but also measures the severity of that demand. This would allow one to distinguish between requests made in periods with different levels of demand. As discussed at Section 7.4, such an indicator may lead to improved fairness outcomes. Such an indicator could also be used to improve the displacement budget mechanism, by making it more expensive for airlines to reduce displacement during highly peak periods. By encouraging airlines to make more attainable requests, more improvement requests could be satisfied.

Finally, there is a need to develop faster algorithms for solving our slot allocation problems, particularly for the ϵ-fairness problem whose solution is the most time-consuming part of the proposed slot allocation mechanism. This will be necessary in order to run the mechanism for larger airports.

Acknowledgements

The work in this paper has been supported by the Engineering and Physical Sciences Research Council (EPSRC) through the Programme Grant EP/M020258/1 "Mathematical Models and Algorithms for Allocating Scarce Airport Resources" (OR-MASTER). Thanks also to Rob Shone for his detailed feedback on a draft version of this manuscript. Due to confidentiality concerns, supporting data cannot be made openly available. Additional details relating to the data are available from Lancaster University data archive at `https://dx.doi.org/10.17635/lancaster/researchdata/295`.

A Proofs of Lemmas and Theorems

To prove Proposition 4.5 we make use the following lemma.

Lemma A.1. *Let $\nu(\mathcal{M})$ denote the optimal solution value for problem* (1)–(9) *with respect to the set of request series $\mathcal{M}$. If m' is a request series such that $m' \notin \mathcal{M}$ then $\nu(\mathcal{M} \cup \{m'\}) \geq \nu(\mathcal{M})$.*

Proof. It suffices to show that for every feasible solution with respect to $\mathcal{M} \cup \{m'\}$ there is a corresponding feasible solution for the set $\mathcal{M}$ whose objective is no greater than the first solution. Let $\tilde{\mathbf{x}}$ be a feasible schedule with respect to the requests $\mathcal{M} \cup \{m'\}$. Then let $\bar{\mathbf{x}}$ be the restriction of this solution to $\mathcal{M}$. This clearly is a feasible solution for the slot allocation with respect to $\mathcal{M}$ and we have:

$$\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M} \cup \{m'\}} f_m^t \tilde{x}_m^t = \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} f_m^t \bar{x}_m^t + \sum_{t \in \mathcal{T}} f_m^t \bar{x}_{m'}^t.$$

Since $f_{m'}^t \geq 0$ for all $t \in \mathcal{M}$ we therefore must have

$$\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M} \cup \{m'\}} f_m^t \tilde{x}_m^t \geq \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} f_m^t \bar{x}_m^t,$$

as required. □

We now prove Proposition 4.5.

Proof of Proposition 4.5. We prove the result for the case of arrival sensitivity. The proof for the case of departure sensitivity is directly analogous. Our proof is via negation, namely we shall demonstrate that if a period is not arrival saturated with respect to some optimal solution for the base problem then it is not arrival sensitive.

Let $\mathbf{x}^*$ be an optimal solution to the base slot allocation problem for the set of request series $\mathcal{M}$. Suppose that under $\mathbf{x}^*$ the period (d', t') is not saturated. Let m' be a new arrival request series with $\mathcal{D}_{m'} = \{d'\}$, $t_{m'} = t'$. It is enough to show that $\nu(\mathcal{M}) = \nu(\mathcal{M} \cup \{m'\})$ where as above, we use $\nu(\cdot)$ for the optimal value of the base problem with the set of request series $\cdot$. It will then follow that the period (d', t') is not arrival sensitive which will conclude the proof.

By Lemma A.1 we have that $\nu(\mathcal{M}) \leq \nu(\mathcal{M} \cup \{m'\})$ so it is enough to construct a feasible solution for the base problem with solution value $\nu(\mathcal{M})$. We construct a new feasible solution $\tilde{\mathbf{x}}$ for the base slot allocation problem with respect to $\mathcal{M} \cup \{m'\}$ as follows:

$$\tilde{x}_m^t = \begin{cases} x_m^{t*} & \text{for all } m \in \mathcal{M},\ t \in \mathcal{T} \\ 1 & \text{for } m = m',\ t = t_{m'} \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

We now show that $\tilde{\mathbf{x}}$ is a feasible solution for the base slot allocation problem with respect to $m \in \mathcal{M} \cup \{m'\}$.

The feasibility of the solution $\mathbf{x}^*$ with respect to the request series $\mathcal{M}$ immediately implies the assignment, turnaround and binary constraints hold for $\tilde{\mathbf{x}}$ with respect to $\mathcal{M} \cup \{m'\}$. The addition of an arrival request will also not affect the departure capacity constraints, and so it only remains to verify the arrival and total capacity constraints still hold.

Let $\mathbf{y}^*$ and $\tilde{\mathbf{y}}$ represent the aggregate schedules corresponding to $\mathbf{x}^*$ and $\tilde{\mathbf{x}}$ respectively. Note that by (42) we have the following relation:

$$\tilde{y}_{dt}^A = \begin{cases} y_{dt}^{*A} + 1 & \text{for } d = d',\ t = t_{m'}, \\ y_{dt}^{*A} & \text{otherwise.} \end{cases} \tag{43}$$

Given that the aggregate number of arrivals only changes for the period (d', t'), we only need to consider arrival capacity constraints which involve aggregate allocations for this period. For constraints of duration $c \in \mathcal{C}$, the affected constraints are those with date index $d = d'$, and time index $t \in [\max\{t' - c + 1, 0\}, t']$. But since the period (d', t') is not saturated with respect to $\mathbf{x}^*$, we have that

$$\min_{t \in [max(t'-c+1,0),t']} \left(\alpha_c^{d't} - \sum_{s=t}^{t+c-1} y_{d's}^{*A} \right) \geq 1.$$

Hence, for all $c \in \mathcal{C}$ and $t \in [\max(t' - c + 1, 0), t']$ we have that

$$\begin{aligned} \alpha_c^{d't} - \sum_{s=t}^{t+c-1} \tilde{y}_{d's}^A &= \alpha_c^{d't} - \sum_{s=t}^{t+c-1} y_{d's}^{*A} - 1 \\ &\geq 0. \end{aligned}$$

Hence, all arrival capacity constraints hold for $\tilde{\mathbf{x}}$. It can be similarly be shown that all total capacity constraints still hold for $\tilde{\mathbf{x}}$ and so we can conclude that $\tilde{\mathbf{x}}$ is a feasible solution for the slot allocation with respect to the set of request series $\mathcal{M} \cup \{m'\}$.

It now remains to evaluate the objective function value for $\tilde{\mathbf{x}}$:

$$\begin{aligned} \sum_{m \in \mathcal{M} \cup \{m'\}} \sum_{t \in \mathcal{T}} f_m^t \tilde{x}_m^t &= \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} f_m^t \tilde{x}_m^t + \sum_{t \in \mathcal{T}} f_{m'}^t \tilde{x}_{m'}^t \\ &= \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} f_m^t x_m^{t*} \qquad \text{since } f_{m'}^{t_{m'}} = 0 \\ &= \nu(\mathcal{M}) \end{aligned}$$

as required. □

The following is a counter-example to the converse of Proposition 4.5.

Example A.2. *Suppose we have a set of two requests $\mathcal{M} = \{m_1, m_2\}$ which must be scheduled for a single day, for both requests we have $t_{m_1} = t_{m_2} = t$ and there is a total capacity limit of one for each time period.*

An optimal solution for this problem would be to displace m_2 to time $t + 1$. This is shown on the left-hand side of Figure 10. Time $t + 1$ is saturated with respect to this optimal solution. However, as is shown on the right-hand side of Figure 10, an extra request m' such that $t_{m'} = t + 1$ can be scheduled for $t + 1$ without increasing the total displacement, and so time $t + 1$ is not sensitive.

The following proof is for Proposition 4.6 which relates to the satisfiability of fairness constraints.

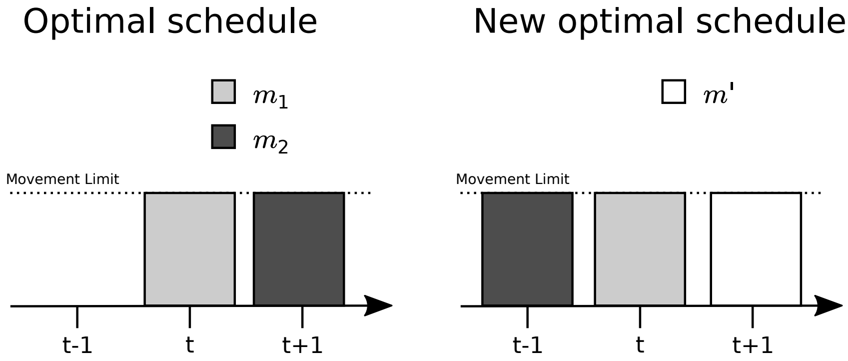


Figure 10: Counter-example to converse of Proposition 4.5

Proof of Proposition 4.6. The proof is by contradiction. Suppose there exists a movement $m_1 \in \widetilde{\mathcal{M}}$ such $\tilde{x}_{m_1}^{t_{m_1}} \neq 1$, that is, where the request m_1 is displaced. Consider the case where m_1 is an arrival, and there exists a departure $m_2 \in \mathcal{M}$ such that $(m_1, m_2) \in \mathcal{P}$. Define a new schedule as follows:

$$\hat{x}_m^t = \begin{cases} x_m^{t*} & \text{if } m \notin \{m_1, m_2\}, \\ 1 & \text{if } m = m_1 \text{ and } t = t_{m_1}, \\ 1 & \text{if } m = m_2 \text{ and } t = t_{m_2}, \\ 0 & \text{otherwise.} \end{cases}$$

By assumption (i) this schedule satisfies turnaround constraints. The periods (d, t_{m_1}) are off-peak for all $d \in \mathcal{D}_{m_1}$ since $r_a = 0$, and by assumption (ii) the periods (d, t_{m_2}) are off-peak for all $d \in \mathcal{D}_{m_2}$. Hence, the movements m_1 and m_2 can be rescheduled at t_{m_1} and t_{m_2} respectively without breaking any capacity constraints, and so this new schedule is feasible. However, this schedule has smaller total displacement than (x_m^{t*}) which contradicts the fact that it is an optimal solution to the base model. The case where m_1 is a departure can be similarly shown to yield contradictions. $\square$

B Minimum Insertion Displacement

We describe in this appendix how the minimum insertion problem (24)–(30) can be solved efficiently by enumerating over pairs of time intervals for the arrival and departure slots.

In particular, we enumerate possible arrival-departure time intervals. To feasibly allocate a pair of arrival-departure slots, this pair must satisfy turnaround time constraints, and the allocation of the additional slots must not break any of the capacity constraints.

Turnaround constraints can be ensured by restricting the enumeration to pairs slots which satisfy these. Arrival and departure capacity constraints can be efficiently verified by using saturation indicators. On the other hand, the total capacity constraints must be verified by explicitly checking all affected constraints. Algorithm 3 below uses these ideas to efficiently calculate the minimum insertion displacement of a request.

input : $\tilde{\mathbf{y}}$ feasible aggregate schedule, (m_1, m_2) request pair, arrival, departure and total capacities α, β, γ, and capacity durations $\mathcal{C}$
output: $B_{m_1 m_2}$ minimum insertion displacement

1 $B_{m_1 m_2} \leftarrow 2 * T$;
2 **foreach** $t \in \mathcal{T}$ **do** $a_t \leftarrow \bigvee_{d \in \mathcal{D}_{m_1}} A^d_t(\tilde{\mathbf{y}})$;
3 **foreach** $t \in \mathcal{T}$ **do** $d_t \leftarrow \bigvee_{d \in \mathcal{D}_{m_2}} D^d_t(\tilde{\mathbf{y}})$;
4 **foreach** $T_a \in t_{m_1}, \ldots, T - \underline{l} - 1$ **do**
5   **if** a_{T_a} **then** continue;
6   **if** $|t_{m_1} - T_a| > B_{m_1 m_2}$ **then** break;
7   **foreach** $T_D \in T_a + \underline{l}, \ldots, \min\{T - 1, T_A + \bar{l}\}$ **do**
8   **if** d_{T_D} **then** continue;
9   **if** $|T_A - t_{m_1}| + |T_D - t_{m_2}| < B_{m_1 m_2}$ **then**
10   **if** check_insertion$(T_A, T_D, \mathcal{C}, \gamma, \tilde{\mathbf{y}})$ **then**
11                 $B_{m_1 m_2} \leftarrow |T_A - t_{m_1}| + |T_D - t_{m_2}|$;
12   **if** $T_D \geq t_{m_2}$ **then**
13   break;
14 **foreach** $T_a \in t_{m_1} - 1, \ldots, 0$ **do**
15   **if** a_{T_a} **then** continue;
16   **if** $|T_a - t_{m_1}| > B_{m_1 m_2}$ **then** break;
17   **foreach** $T_D \in T_A + \underline{l}, \ldots, \min\{T_A + \bar{l}, T - 1\}$ **do**
18   **if** d_{T_D} **then** continue;
19   **if** check_insertion$(T_A, T_D, \mathcal{C}, \gamma, \tilde{\mathbf{y}})$ **then**
20             $B_{m_1 m_2} \leftarrow |T_A - t_{m_1}| + |T_D - t_{m_2}|$;
21   **if** $T_D \geq t_{m_2}$ **then**
22   break;
23 **return** $|\mathcal{D}_{m_1}| B_{m_1 m_2}$

Algorithm 3: Calculate minimum insertion displacement

This algorithm begins by calculating in line 2 a Boolean indicator which tells us for which time periods we can allocate an extra arrival slot without breaking any capacity constraints. This is done by checking the arrival peak indicator for each day the arrival request applies. Similarly, a feasibility indicator is constructed for the departure request series in line 3. In lines 4-13, we search through all pairs of arrival and departure slot times where the arrival slot time is greater than or equal to the request arrival time t_{m_1} for the pair with the smallest feasible displacement. We loop over all pairs of time intervals with feasible turnaround time. The order of pairs of arrival and departure times searched is such that the displacement is strictly increasing when $T_D \geq t_{m_2}$. This means that once a feasible pair of slots is found and $T_D \geq t_{m_2}$, we can break out of the search loop. For each of these pairs, we check that the slots can be allocated without breaking the arrival and departure constraints by using the feasibility indicators calculated in lines 2 and 3. Finally, if these are feasible, and the pair of slots has small displacement than the current best, we check whether the the pairs of slots can be allocated without breaking capacity constraints by using the check_insertion function which is presented in Algorithm 4. This is the most computationally expensive part of the procedure which is why it is the final feasibility check we carry out. Once all pairs with arrival time greater than or equal to t_{m_1} have been searched, we search the remaining possible pairs of requests in lines 14-22. This search is carried out in the same way as in lines

4-12.

input : Arrival and departure times T_A and T_D, capacity constraint durations $\mathcal{C}$, total capacities γ, aggregate schedule $\tilde{\mathbf{y}}$

output: True or False

foreach $c \in \mathcal{C}$ **do**

  if $T_D - T_A| \geq c$ **then** continue;

  foreach $t \in \max\{0, T_D - c + 1\}, \ldots, T_A$ **do**

    if $\sum_{s=t}^{t+c-1} \tilde{y_{ds}} + 2 > \gamma_c^{dt}$ **then return** False;

return True

Algorithm 4: Check whether arrival and departure slots can be allocated to a request pair without breaking any total capacity constraints

References

Ball, M., Donohue, G. L., & Hoffman, K. 2005. Auctions for the Safe, Efficient, and Equitable Allocaiton of Airspace System Resources. *Chap. 20 of: Combinatorial Auctions*. MIT Press.

Barnhart, C., Bertsimas, D., Caramanis, C., & Fearing, D. 2012. Equitable and Efficient Coordination in Traffic Flow Management. *Transportation Science*, **46**(2), 262–280.

Benlic, U. 2018. Heuristic search for allocation of slots at network level. *Transportation Research Part C: Emerging Technologies*, **86**, 488 – 509.

Bertsimas, D., & Gupta, S. 2015. Fairness and collaboration in network air traffic flow management: an optimization approach. *Transportation Science*, **50**(1), 57–76.

Bertsimas, D., Farias, V. F., & Trichakis, N. 2012. On the efficiency-fairness trade-off. *Management Science*, **58**(12), 2234–2250.

Castelli, L., Pellegrini, P., & Pesenti, R. 2012. Airport slot allocation in Europe: economic efficiency and fairness. *International Journal of Revenue Management*, **6**(1–2), 28–44.

Commission, European. 1993a. *Council Regulation (EEC) No. 95/93 of 18 January 1993 on common rules for the allocation of slots at Community airports.*

Commission, European. 1993b. *Regulation (EC) No. 793/2004 of the European Parliament and of the Council of 21 April 2004 amending Council Regulation (EEC) No 95/93 on common rules for the allocation of slots at Community airports.*

Corolli, L., Lulli, G., & Ntaimo, L. 2014. The time slot allocation problem under uncertain capacity. *Transportation Research Part C: Emerging Technologies*, **46**, 16–29.

Dal Sasso, V., Fomeni, F. D., Lulli, G., & Zografos, K. G. 2019. Planning efficient 4D trajectories in Air Traffic Flow Management. *European Journal of Operational Research*. Forthcoming.

Evans, A., Vaze, V., & Barnhart, C. 2016. Airline-Driven Performance-Based Air Traffic Management: Game Theoretic Models and Multicriteria Evaluation. *Transportation Science*, **50**(1), 180–203.

Fairbrother, J., & Zografos, K. G. 2018. *On the development of a fair and efficient slot scheduling mechanism at congested airports.* Presented at 97th Transportation Research Board Annual Meeting, last accessed February 2019.

Gurobi Optimization, Inc. 2017. *Gurobi Optimizer Reference Manual.*

International Air Transport Association. 2017. *Worldwide Slot Guidelines.* `http://www.iata.org/policy/infrastructure/slots/Pages/slot-guidelines.aspx`, accessed 25 April 2017.

Jacquillat, A., & Odoni, A.R. 2015. An integrated scheduling and operations approach to airport congestion mitigation. *Operations Research*, **63**(6), 1390–1410.

Jacquillat, A., & Vaze, V. 2018. Interairline equity in Airport Scheduling Interventions. *Transportation Science.*

Kalai, E., & Smorodinsky, M. 1975. Other solutions to Nash's Bargaining Problem. *Econometrica*, **43**(3).

Morrison, S.A., & Winston, C. 2007. Another look at airport congestion pricing. *The American Economic Review*, **97**(5), 1970–1977.

Nash, J.F. 1950. The Bargaining Problem. *Econometrica*, **18**(2).

Pellegrini, P., Castelli, L., & R., Pesenti. 2012. Secondary trading of airport slots as a combinatorial exchange. *Transportation Research Part E: Logistics and Transportation Review*, **48**(5), 1009–1022.

Pellegrini, P., Bolić, T, Castelli, L., & Pesenti, R. 2017. SOSTA: An effective model for the Simultaneous Optimisation of aiport SloT Allocation. *Transportation Research Part E: Logistics and Transportation Review*, 34–53.

Pilon, N., Cook, A., Ruiz, S., Bujor, A., & Castelli, L. 2016. Improved Flexibility and Equity for Airspace Users During Demand-capacity Imbalance. *In: 6th SESAR Innovation Days*. Last accessed May 2018.

Pita, J. P., Barnhart, C., & Antunes, A. P. 2013. Integrated Flight Scheduling and Fleet Assignment Under Airport Congestion. *Transportation Science*, **47**(4), 477–492.

Rassenti, S.J., Smith, V.L., & Bulfin, R.L. 1982. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, 402–417.

Ribeiro, N. A., Jacquillat, A., Antunes, A.P., Odoni, A.R., & Pita, J.P. 2018. An optimization approach for airport slot allocation under {IATA} guidelines. *Transportation Research Part B: Methodological*, **112**, 132 – 156.

Swaroop, P., & Ball, M. O. 2013. Consensus-Building Mechanism for Setting Service Expectations in Air Traffic Flow Management. *Transportation Research Record*, **2325**(1), 87–96.

Vaze, V., & Barnhart, C. 2012. Airline frequency competition in airport congestion pricing. *Transportation Research Record: Journal of the Transportation Research Board*, 69–77.

Vossen, T., & Ball, M. O. 2006. Slot Trading Opportunities in Collaborative Ground Delay Programs. *Transportation Science*, **40**(1), 29–43.

Vossen, T., Ball, M. 0., Hoffman, R., & Wambsganss, M. 2003. A general approach to equity in traffic flow management and its application to mitigating exemption bias in ground delay programs. *Air Traffic Control Quarterly*, **11**(4), 277–292.

Wambsganss, M. 1996. Collaborative Decision Making Through Dynamic Information Transfer. *Air Traffic Control Quarterly*, **4**(2).

Zografos, K. G., Salouras, Y., & Madas, M. A. 2012. Dealing with the efficient allocation of scarce resources at congested airports. *Transportation Research Part C: Emerging Technologies*, **21**(1), 244–256.

Zografos, K. G., Madas, M. A., & Androutsopoulos, K. N. 2017. Minding the Gap: Optimizing Airport Schedule Displacement and Acceptability. *Transportation Research Part A: Policy and Practice*.

Zografos, K.G., & Jiang, Y. 2016. Modelling and solving the airport slot scheduling problem with efficiency, fairness, and accessibility considerations. *In: TRISTAN Symposium*. Last accessed May 2018.

Zografos, K.G., & Jiang, Y. 2017. *Modelling fairness in slot scheduling decisions at capacity-constrained airports*. Presented at 96th Transportation Research Board Annual Meeting, last accessed February 2019.

Zografos, K.G., & Jiang, Y. 2019. A Bi-objective Efficiency-Fairness Model for Scheduling Slots at Congested Airports. *Transportation Research Part C: Emerging Technologies*. In press.