



**HAL**  
open science

## **SOSTA: an effective model for the Simultaneous Optimisation of airport SloT Allocation**

Paola Pellegrini, Tatjana Bolic, Lorenzo Castelli, Raffaele Pesenti

► **To cite this version:**

Paola Pellegrini, Tatjana Bolic, Lorenzo Castelli, Raffaele Pesenti. SOSTA: an effective model for the Simultaneous Optimisation of airport SloT Allocation. *Transportation Research Part E: Logistics and Transportation Review*, 2017, 99, p34-53. 10.1016/j.tre.2016.12.006 . hal-01466699

**HAL Id: hal-01466699**

**<https://hal.science/hal-01466699>**

Submitted on 7 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SOSTA: An effective model for the Simultaneous Optimisation of airport Slot Allocation

Paola Pellegrini <sup>a,\*</sup>, Tatjana Bolić <sup>b</sup>, Lorenzo Castelli <sup>b</sup>, Raffaele Pesenti <sup>c</sup>

<sup>a</sup> Univ. Lille Nord de France – IFSTTAR, COSYS, LEOST, Villeneuve d'Ascq, France

<sup>b</sup> Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Trieste, Italy

<sup>c</sup> Dipartimento di Management, Università Ca' Foscari di Venezia, Venezia, Italy

---

## ARTICLE INFO

Accepted 12 December 2016

---

### Keywords:

Airport slot allocation  
Air transport  
Integer linear programming  
Grandfather rights  
Case-study  
Capacity-demand imbalances

---

## ABSTRACT

In this paper, we propose SOSTA, an integer linear programming model for optimisation of the airport slot allocation process on the European scale. The main contribution of SOSTA is the simultaneous allocation of slots at all European airports, while applying the existing regulation and practices. Additionally, SOSTA considers aircraft rotations through the turn-around time constraints, which is another novel contribution. In an experimental analysis based on real data, we show the benefits of the simultaneous allocation, and the flexibility and capabilities of SOSTA, along with the extremely good computational performance.

---

## 1. Introduction

The continuous growth of air traffic combined with the constrained airport capacity increase generates imbalances between traffic demand and available capacity, leading to significant delays and associated costs. For instance, in Europe in 2014 the traffic has increased by 2.5% in terms of flight hours controlled with respect to 2013, but the estimated cost of en-route and airport Air Traffic Flow Management delay has increased by 7.6% (EUROCONTROL. PRR 2014, 2015). Since airports often represent bottlenecks in the air transport network, initiatives to increase their capacity are always sought. Infrastructural interventions (such as building new runways) are rarely possible, mainly for economic and environmental concerns. Currently, each major airport may consider the following options to mitigate its congestion: either managing the demand at the strategic level or adjusting the allocation of airport resources to flights at the tactical level. The rationale of the former choice is to smooth peaks of traffic demand by moving arrival and/or departure requests to times of the day with a lower demand for the use of the airport infrastructure. Demand management is usually achieved either through the enforcement of administrative measures or through the use of economic instruments such as congestion pricing or auction mechanisms (see, e.g., Perret, 2015). The rationale of the latter choice is to adapt the use of the infrastructure to improve the utilisation of the airport capacity. For example, dynamic change of runway configurations or the selection of the arrival and departure service rates can be used to that end (Jacquillat and Odoni, 2015). This paper focuses on the administrative-based demand management procedure at the strategic level. It proposes Simultaneous Optimisation of the airport Slot Allocation (SOSTA), a decision support tool capable of optimally coordinating the airports' capacity management at the European level.

---

\* Corresponding author.

E-mail addresses: [paola.pellegrini@ifsttar.fr](mailto:paola.pellegrini@ifsttar.fr) (P. Pellegrini), [tbolic@units.it](mailto:tbolic@units.it) (T. Bolić), [castelli@units.it](mailto:castelli@units.it) (L. Castelli), [pesenti@unive.it](mailto:pesenti@unive.it) (R. Pesenti).

Nowadays the capacity of many airports worldwide is managed through *airport slots*, in accordance with IATA's (International Air Transport Association) Worldwide Slot Guidelines (IATA, 2015). A slot is defined as "a permission given by a coordinator for a planned operation to use the full range of airport infrastructure necessary to arrive or depart at a Level 3 airport on a specific date and time". A Level 3 (or *coordinated*) airport is an airport "where capacity providers have not developed sufficient infrastructure, or where governments have imposed conditions that make it impossible to meet demand. A *coordinator* is appointed to allocate slots to airspace users and other aircraft operators using or planning to use the airport as a means of managing the declared capacity." In addition to Level 3 airports, flight schedules are facilitated at Level 2 (or *facilitated*) airports, which are those "where there is potential for congestion during some periods of the day, week, or season which can be resolved by schedule adjustments mutually agreed between the airspace users and facilitators. A *facilitator* is appointed to facilitate the planned operations of airlines using or planning to use the airport" (IATA, 2015). In Europe, there is one coordinator per country, meaning that a unique national authority manages the primary slot allocation of each Level 3, and schedule facilitation at Level 2 airports of this country (see, <http://www.euaca.org>). The allocation and schedule facilitation is always performed on an airport by airport basis, though. In Europe there are as many as 107 airports designated as Level 3 and another 79 as Level 2, representing 60% and 61% of all Level 3 and Level 2 airports in the world, respectively (see the Appendix 11.2 of IATA (2015)).

In the European Union, the IATA's Worldwide Slot Guidelines are implemented by Council Regulation (EEC) No 95/93 "on common rules for the allocation of slots at Community airport" and its subsequent amendments: EC Regulations No 894/2002, No 1554/2003, No 793/2004, and No 545/2009. The slot allocation process in Europe consists of two main steps: primary slot allocation, and slot exchanges and transfers. The primary slot allocation begins about five months before the start of the season (the winter season starts on the last Sunday of October, the summer season on the last Sunday of March), when the airspace users (e.g., airlines) submit formal requests for slots (and schedule facilitation) to airport coordinators. The requests are submitted in a standardised format, the Standard Schedule Information Manual (SSIM) format (IATA, 2015), containing the flight number, time period of operations (from-to within a season), day of the week, route and arrival or departure time. Airspace users can also indicate the acceptable displacement around the requested slot time. However, "Airport slots are not route, aircraft or flight number specific and may be changed by an airline from one route or type of service to another. Such changes are subject to final confirmation by the coordinator" (IATA, 2015).

At Level 2 airports, when mismatches between capacity and demand exist, an airspace user might be asked to move the scheduled time of an operation, for the minimum necessary amount of time, on a voluntary basis. At the Level 3 airports, the coordinators endeavour to satisfy the requests, under the existing capacity constraints, respecting historical precedence, the so-called *grandfather rights*. An airspace user obtains the grandfather right on a slot, if it operated the corresponding slot at least 80% of the times in the preceding equivalent season. In such a case we refer to this airspace user as an *incumbent*. In addition, the incumbent is allowed to slightly modify the time (w.r.t. the preceding equivalent season) of any of its grandfather slots.

Once all grandfather rights from incumbent airspace users are granted, fifty percent of the remaining slots are allocated to *new entrant* airspace users, and the rest to other airspace users. A new entrant is defined as: "an airline requesting a series of slots<sup>1</sup> at an airport on any day where, if the airline's request were accepted it would hold fewer than five slots at the airport on that day" (IATA, 2015). IATA slot conference takes place after the primary allocations are established, to facilitate negotiations of slot exchanges between airspace users. The aim of the conference is to diminish as much as possible, through negotiations, the difference between the requested and assigned slot times, which is referred to as *schedule displacement* (see, e.g., Pyrgiotis and Odoni, 2014). For example, from the conversations with a coordinator of one of the congested airports, at the beginning of the process, about 40% of requests could be granted as requested, while other requests had an average of 25 min of difference. The subsequent IATA conference negotiations brought the satisfied requests to about 85% with six minutes of average difference for non-satisfied requests. By the start of the season, 97% of requests were satisfied. This shows how the definition of airspace users' schedules is a non-trivial exercise at global level. In fact, the slots an airspace user receives are the outcome of several local allocations and may include different schedule displacements. As such, the received slots may be impracticable with respect to, say, the fleet rotation constraints, or undesirable in terms of departing/arriving times for business purposes. Decisions on which schedule displacements to accept, and their magnitude, may not be straightforward as airspace users may have to deal with many coordinators at the same time, and this may require a significant effort especially from airspace users that fly to/from many Level 3 airports.

The European Commission has financed several studies in the last years to assess the implementation of EC Regulation 95/93 and its amendments (Coopers and Lybrand, 1995; NERA, 2004; Mott MacDonald, 2006; SDG, 2011). These studies acknowledge the inefficient use of capacity at some airports, highlight difficulties that new entrants face to obtain slots, and identify significant differences in coordinators' operations. In this context, Madas and Zografos (2013) propose possible changes in the slot allocation process, including the introduction of economic instruments (e.g., congestion pricing, secondary trading, and auctioning). The design of economic instruments to enhance the efficiency of the slot allocation process has been under investigation for decades as in Rassenti et al. (1982) who proposed to allocate airport slots through a sealed-bid combinatorial auction. Several other auction-based models have been designed and evaluated as in Maldom (2003), Li

<sup>1</sup> Slots are allocated in series, which is a sequence of at "least five slots, requested for the same time on the same day-of-the-week, distributed regularly in the same season." (IATA, 2015). For example, a request for departures from an airport at 10 : 30 for at least five Wednesdays during a season is considered a series of slots.

et al. (2007), and Ball et al. (2007). Recently, ascending-bid multi unit auctions to allocate slots at multiple airports (Sheng et al., 2015) was published. Slot trading has also been extensively studied as, e.g., in Verhoef (2010), or in Pellegrini et al. (2012b) and Fukui (2014) for the European context, and in Kleit and Kobayashi (1996) and Fukui (2010) who analyse the impacts of slot trading in terms of competitiveness and market entry possibilities in the US. Finally, congestion pricing as a tool to manage airport demand has been proposed by several authors such as Brueckner (2009), Basso and Zhang (2010), Czerny (2010), and Czerny and Zhang (2011).

Airport slots also play a pivotal role in shaping airline competition and social welfare. In fact, the possibility for airlines to operate in specific markets may be limited by the availability of slots. These limitations may prevent airlines to expand their international and intercontinental markets and therefore restrict competition, also in liberalised air transportation markets (Li et al., 2010; Adler et al., 2014). Hence, the economic and social impact of airport slot allocation is twofold. On the one hand, the attractiveness of an airport is driven by the portfolio of its destinations, and this affects airport revenues as well as income, employment and tourism effects for the local economy (see, e.g., Gillen and Hensch, 2001). On the other hand, slot availability affects airlines scheduling decisions, which in turn promote economic growth by enhancing the connectivity of a region with other parts of the world, typically in deregulated markets where carriers develop hub-and-spoke networks (Adler, 2001; Adler, 2005) or minimise social costs in subsidised markets to protect remote communities (see, e.g., Pita et al., 2014 for the case of Norway).

This large body of work has contributed to major insights on the economic performance of demand management alternatives. However, the previous research typically considers simplified operational settings and does not capture the complexity and variability of airport operations and of the networks of flights that airspace users operate. Furthermore, major European legacy airspace users strongly oppose any relaxation of the grandfather rights' rule since they claim that it is this very same rule that makes their business sustainable, as it protects their most profitable slots at their main home hubs, and provides stability so that airspace users can make long-term investments. In fact, for airspace users offering a hub-and-spoke network to their passengers, the elimination of grandfather rights could be affordable only in case of monetary compensations (see, e.g., Castelli et al., 2012).

This motivates the need to enhance the current administrative framework and to make it more efficient. As already pointed out by Zografos et al. (2012), the current slot allocation process is highly inefficient because the management of its complexity (the allocation needs to comply with numerous criteria and rules) is still empirical. Similar conclusions were also provided by Koesters (2007) who analysed the relationship between demand for slots and displacement. To mitigate these inefficiencies, Zografos et al. (2012) formulate an optimisation model that implements EU regulations (and IATA guidelines) and solves the slot allocation for a single airport by minimising the total displacement. Their model is applied on real data from three different Greek airports, and the results show that the schedule displacement can be reduced in a range from 14% to 95%.

In this paper, we introduce SOSTA, a model that optimally and simultaneously allocates the airport slots/requests at all Level 2 and Level 3 airports in Europe. SOSTA draws upon the integer programming model by Zografos et al. (2012), but extends it from one airport to the network of airports. The model takes into account different types of airport capacity constraints and minimises the cost of deviations between what is requested (by the user) and what is allocated (by the coordinator/facilitator). A further novel contribution of the network perspective implemented in SOSTA lies in explicitly linking departure and arrival slots (when both are needed) of each flight and considering aircraft rotations through the introduction of turnaround time constraints. SOSTA can handle all European airport slots/facilitation requests during a busy day, achieving the optimality of the final allocation in reasonable time.

As recently highlighted in the review paper by Zografos et al. (2016), the slot allocation problem was studied in the context of a network of airports by Castelli et al. (2012), Pellegrini et al. (2012a), and Corolli et al. (2014). In particular, Castelli et al. (2012) introduce a much simpler yet more computationally cumbersome model than the one presented in this paper. On the one hand, the problem modelling is too simplified as they do not, for instance, distinguish between interval and hourly capacity constraints (see Section 2.2) or consider turnarounds (Section 2.6). On the other hand, sector capacities are taken into account, which is perhaps a level of detail not needed at the strategic level. In addition, as they adapt the Bertsimas et al. (2011) formulation, which is tailored for the air traffic flow management problem and is unable to deal with very large air traffic samples, the size of the instances that can be solved effectively is significantly smaller than in the current paper (2,200 vs 32,000 requests, see Section 4). Larger instances are solved through a meta-heuristic approach in Pellegrini et al. (2012a), which, however, cannot prove the optimality of the solutions returned. A simultaneous slot allocation is also proposed by Corolli et al. (2014) who extend the Zografos et al. (2012) model by introducing a stochastic programming approach to take into consideration the uncertainty on the capacity availability. However, their model does not capture all the important subtleties of the European allocation process in terms of, for instance, capacity constraints (see again Section 2.2) or presence of grandfather rights, which are ignored. Nevertheless, a relevant feature of their model is the opportunity to perform the slot allocation (a strategic decision) by already encompassing the effects (in terms of estimated delay) of possible capacity reductions that arise at the tactical level only, i.e., on the day of operations. This is achieved through a two-stage stochastic optimisation, the computational burden of which limits the size of manageable instances up to just 9,000 requests.

SOSTA overcomes some limitations of these papers for the benefit of both airport coordinators and airspace users. In fact, SOSTA could ideally be used as a tool to partially replace and shorten the current (lengthy) slot allocation process where users need to interact several times with coordinators and often other airspace users to build and re-build their schedules,

based on the accepted and rejected slot requests. Similarly, it should mitigate the use of the secondary slot exchange when, following the primary allocation, airspace users negotiate and exchange among them the allocated slots, subject to coordinators' approval, to fine-tune their schedules.

Following the introduction of the main assumptions of the problem under investigation (Section 2) and its mathematical formulation (Section 3), we apply SOSTA on the busiest day of 2013. After presenting an overview of the available data (Section 4.1), we show that SOSTA is a valid model that enforces the rules characterising the current system: we apply it to a set of requests for which the final allocation is known and SOSTA returns the same final allocation with only a few exceptions (Section 4.2). Next, we quantify the potential improvement brought by SOSTA's simultaneous slot allocation at all airports. In order to do so, we compare the results and behaviour of SOSTA with the results of current allocation process in the cases in which the ratio between slot requests and slot availability is greater than the current one. This situation is simulated through the reduction of the airports capacities (all airports, uniformly), to avoid adding fictitious slot requests (Section 4.3). Moreover, we assess the sensitivity of SOSTA when the maximum schedule displacement or the maximum additional block time varies (Section 4.4). Then we analyse the effects of moving from linear to quadratic cost functions (Section 4.5) and of loosening grandfather rights (Section 4.6). Finally, we propose two variants of SOSTA to take into account fairness considerations (Section 4.7). We conclude the paper in Section 5.

## 2. SOSTA's key assumptions

A slot is the right to use the airport infrastructure “on a specific date and time”. As an example, the airport coordinator allocates to an airspace user a slot at 10:00 of a given day. To make the use of the infrastructure viable, the slot has a duration associated to it, for instance 10 min. This means that the airspace user has the permission to arrive (or depart) from 10:00 to 10:10. At large airports, where simultaneous operations can be performed, more than one slot can be associated to the same 10:00–10:10 time interval. Hence, we distinguish between the “slot” (the right) and “interval” (when this right is exercised), since there is no one-to-one relationship between them: a time interval is associated with each slot, but more than one slot can be allocated to the same time interval. The distinction between interval and slot, and other features of the European slot allocation system are based on the real data and information available through EUROCONTROL's Slot Coordinator and DDR2 databases, and other data sources (see Section 4.1 for further details). Relying on these data, the SOSTA model is developed in accordance with the following assumptions.

### 2.1. Characteristics of slots

**Assumption 1.** A slot is the right to use the airport facilities for take-off or landing within a time *interval*. An interval is characterised by a *start time* and a *length*. Lengths may vary across airports, but all intervals at the same airport have the same length. Most frequent values for lengths are 5, 10, 15 and 20 min. The start time of an interval can occur only at the minute 0, 5, 10, . . . , 55 of an hour.

Note that slots are required for operations at Level 3 airports, while for Level 2 airports an agreement with the facilitator is sufficient. For the sake of modelling terminology, from now on, both the slots for Level 3, and agreements for Level 2 airports will be termed slots.

We call *subinterval* the time period equal to the difference of two consecutive intervals' start times. It may be shorter than the interval length. As an example, Fig. 1 shows intervals of ten minute length, and subintervals of five (left hand side) and ten minute (right hand side) length. We have *overlapping* intervals in the former situation, and *sequential* intervals in the latter.

**Assumption 2.** The length of all subintervals at an airport is constant and it is either of five minutes or equal to the airport's interval length. The start time of a subinterval is always equal to the preceding subinterval start time plus the subinterval length: subintervals never overlap. Hence, the start time of a subinterval can occur only at the minute 0, 5, 10, . . . , 55 of an hour, and coincides with the start time of an interval.

When an airspace user requests a slot, the start time of the associated interval is specified. Similarly, the coordinator allocates a slot by identifying the start time of a specific interval. Since for each interval there is always a subinterval that shares the same start time, we may associate to every slot the subinterval that starts at the same time of that slot's interval. We name as *slot time* the start time of the corresponding subinterval.

**Assumption 3.** A *movement* is defined as a flight arrival (*arrival movement*) or as a flight departure (*departure movement*). There is a one-to-one correspondence between performed movements and allocated slots: an airspace user must have a slot allocated to perform any (arrival or departure) movement at a Level 2 or Level 3 airport. Hence, a movement can be scheduled to occur only at the minute 0, 5, 10, . . . , 55 of an hour (*movement time*), and is uniquely associated to a subinterval.

We will refer to slots associated to departure (arrival) movements as *departure (arrival) slots*. Analogously, we will refer to the pair of slots associated to the two movements to be performed by the same aircraft as *coupled slots*.

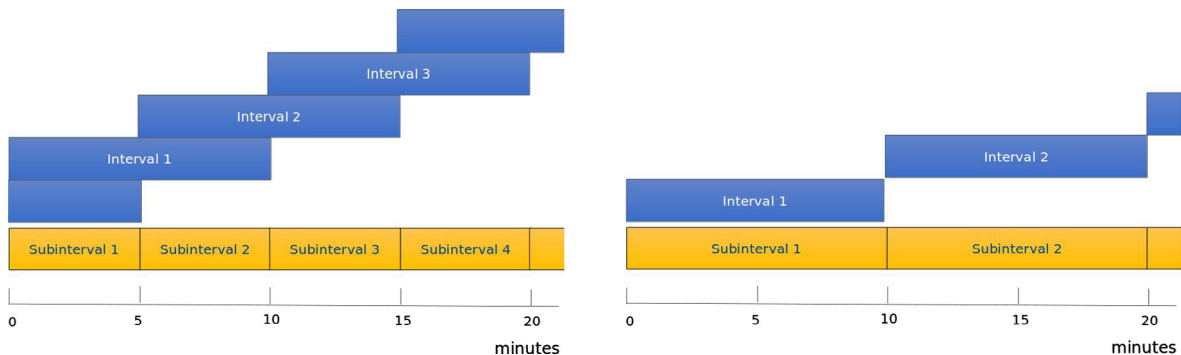


Fig. 1. Overlapping (left) and sequential (right) intervals.

## 2.2. Capacity constraints

Authorities and/or airport coordinators may regulate or otherwise limit the maximum number of slots at each airport for different reasons, including safety, airside and/or ground side infrastructural limitations, or noise reduction. Such capacity constraints are usually defined for a time period, i.e., time periods are typically one hour or one interval long.

**Assumption 4.** An airport has a capacity distribution profile in terms of arrival, departure, and total movements, and hence in terms of arrival, departure and total slots per time period. Capacity in terms of number of total movements means that any movement is taken into account, without distinction between a departure and an arrival movement. Such value is not necessarily equal to the sum of departure and arrival slots in the same time period.

As an example, Table 1 shows capacity constraints throughout the day at the Arlanda airport in Stockholm, Sweden. We see that the maximum number of allowed arrivals per hour is equal to 42 throughout most of the day (third row in Table 1), and varies during the night and early morning. The maximum number of allowed departures per hour varies throughout the day between 20 and 42. The total number of allowed slots vary between 26 and 84 (last column).

### 2.2.1. Hourly capacity constraints.

The bound on the number of slots is given for periods of one hour, and it may be applied either on a rolling basis (*Type I hourly capacity constraints*) or sequentially (*Type II hourly capacity constraints*). In the former case, the beginning of each hourly period is equal to the beginning of the previous one plus five minutes, whereas in the latter case it is equal to the end of the previous one. Depending on the airport, both types of hourly capacity constraints may be imposed.

### 2.2.2. Interval capacity constraints.

The interval capacity represents the maximum number of slots that can be allocated to a specific interval. If intervals are sequential (see Fig. 1), the sum of movements allocated to the unique subinterval composing the interval has to be lower than or equal to the interval capacity. In the case of overlapping intervals:

- The sum of movements over all subintervals of an interval has to be lower than or equal to the interval capacity;
- A movement consumes one unit of capacity of all the intervals that contain the corresponding subinterval.

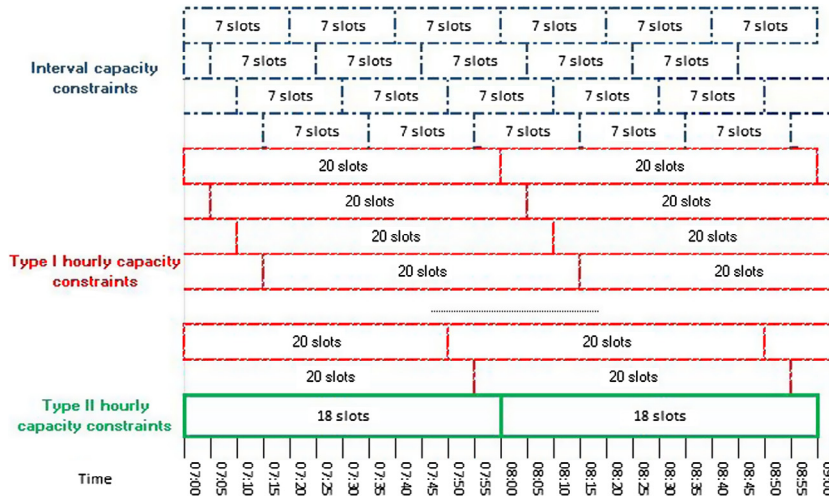
Therefore at one airport up to seven types of capacity constraints may coexist: Interval constraints in terms of total number of slots, and Type I and Type II hourly capacity constraints, in terms of number of departure and arrival slots, and in terms of total number of slots. Fig. 2 shows an example of Interval, Type I and Type II capacity constraints, with overlapping intervals. For the sake of clarity, Fig. 2 does not distinguish between Type I and Type II constraints in terms of number of departure and arrival slots, but treats such capacities only in terms of total number of allocated slots. We consider an airport with the interval and subinterval length equal to 20 and 5 min, respectively. Capacity is set to 7 slots per interval, and 20 (Type I) and 18 (Type II) slots per hour. Hence, the capacity constraints which hold at this airport are:

1. *Interval capacity constraints.* At most 7 slots can be allocated to the 20-min interval starting at 07:00, at most 7 slots to the 20-min interval starting at 07:05, and so on. This means that at most 7 slots can be allocated to any subinterval. Hence, to be able to allocate a slot to the subinterval starting, say, at 07:15, the capacity constraints of four intervals (starting at 07:00, 07:05, 07:10, 07:15, respectively) need to be met.
2. *Type I hourly capacity constraints.* These constraints specify the maximum number of slots (arrival, departure or total) in a 60-min period starting at each multiple of 5 min. In the figure, we see the capacity of 20 slots imposed in the periods 07:00–08:00, 07:05–08:05, 07:10–08:10, and so on.



**Table 1**Maximum number of allowed slots (capacity) at Arlanda airport (url: <http://acs.slot.se.loopiadns.com/Docs/ARNCapacity.pdf>).

Row	Start time	End time	Departures	Arrivals	Total
1	00:00:00	02:59:00	20	23	26
2	03:00:00	05:59:00	20	20	28
3	06:00:00	22:59:00	42	42	84
4	23:00:00	23:59:00	20	23	26

**Fig. 2.** Example of airport capacity constraints: 20-min intervals with capacity of 7 slots imposed based on overlapping intervals (subintervals of length 5 min); Type I hourly capacity constraints of 20 slots per hour; Type II hourly capacity constraints of 18 slots per hour.

3. *Type II hourly capacity constraints.* They define the maximum number of slots (arrival, departure or total) that can be allocated in a 60-min period starting at the beginning of each hour: 18 slots between 07:00 and 08:00, 08:00 and 09:00, 09:00 and 10:00, and so on.

Several airports in our databases have simultaneously active interval and Type II constraints. The rationale is to smooth the demand while considering operational, environmental and other limitations. For instance, relying on the example depicted in Fig. 2, if only Type II constraints exist, it would be possible to concentrate all allowed 18 movements in a very short time period, e.g., from 7:00 to 7:20, and leave empty the remaining part of the hour (from 7:20 to 8:00). However, this situation could be operationally unmanageable by the airport, which imposes capacity constraints on shorter periods (interval constraints). Hence, the 18 movements must be spread over the hour. Vice versa, interval constraints without Type II constraints may lead to an unsustainable traffic level over the longer period.

Type I constraints impose an additional limitation that is not captured by the joint combination of Type II and interval constraints. For instance, the allocation: 4 slots from 7:00 to 7:20, 7 slots from 7:20 to 7:40, 7 slots from 7:40 to 8:00, 7 slots from 8:00 to 8:20, 7 slots from 8:20 to 8:40 and 4 slots from 8:40 to 9:00 satisfies all Type II and interval constraints, but not Type I constraints starting at 7:20, 7:25, 7:30, 7:35, and 7:40. Type II constraints in some cases reflect artificially imposed limits to the airport capacity (like curfew restrictions during the night, as in Table 1). Type I constraints instead reflect the operational capabilities of the airport over a period that lasts one hour. When both Type I and Type II constraints are present (as in London Heathrow, for instance), the capacity values of the latter are always strictly less than those of the former. Differently, Type II constraints would become redundant.

According to the available data, airports may impose Type I and Type II constraints in terms of number of arrival and departure slots, and in terms of total number of slots. Instead, interval capacity constraints are defined in terms of total number of slots. However, to the best of our knowledge, no formal rule forbids airports to define interval capacity constraints for just arrival or departure slots. In any case, SOSTA can be trivially extended to include interval capacity constraints for arrival and/or departure slots.

### 2.3. Schedule displacement

Once airspace users request slots at a given airport, they already know that these very precise requests might not be accepted but alternative slots may be offered instead. The time difference between the requested and allocated slots is termed *schedule displacement*, and distinguishes between *forward* (allocated - requested time is positive) and *backward* (allo-

cated – requested time is negative) displacement. Airspace users are unlikely to accept a (too) large displacement as this may disrupt passenger connections and fleet rotation (e.g., in a hub-and-spoke network) and lead to economically inefficient flight schedule.

**Assumption 5.** For each requested slot, a maximum value for both forward and backward displacement is set.

#### 2.4. Missed allocations

If an airspace user requests a slot at a very congested airport and/or time, it may happen that its request cannot be fulfilled within the available maximum (forward and backward) displacement. In such a case, the only option is not to schedule this flight (*missed allocation*).

**Assumption 6.** Missed allocations cannot be ruled out in case of severe demand-capacity imbalances. Hence, SOSTA adopts a lexicographic approach: first, the number of missed allocations is minimised and then, within the remaining requests, the total cost of displacement is minimised (see Sections 3.3 and 4.3).

#### 2.5. Block time

The instant an aircraft leaves (arrives at) the gate is termed *off-block* (*in-block*) time. For every flight, the difference between off- and in-block times is called *block time*.

**Assumption 7.** Given a pair of coupled slots, the *requested block time* is equal to the difference between the end time of the requested arrival slot and the start time of the requested departure slot.

We assume that airspace users are rational agents and request coupled slots compatible with their desired block times. Considering that more than one route may be available for a pair of origin-destination airports, SOSTA allows the requested block time to be shortened by introducing the minimum flight block time (Constraints 11). The maximum block time is also introduced because airspace users would never accept, at SOSTA’s strategic flight planning phase, unrealistically long routes (Constraints 12).

#### 2.6. Turnaround slots

**Assumption 8.** Two slots associated to two movements to be operated sequentially by the same aircraft, at the same airport, performing two different flights are named *turnaround slots*. The displacement of two turnaround slots can never imply a turnaround time shorter than a predefined value.

During the turnaround time, between the arrival and departure of an aircraft, passengers disembark, aircraft is serviced (i.e., cleaned, re-fuelled, etc.), and new passengers embark.

### 3. SOSTA formulation

In this section, we formulate SOSTA as an integer linear programming model.

#### 3.1. Notation

We use the following notation:

$H$	scheduling horizon (for example one day)
$R$	set of requested slots (or <i>requests</i> ) at Level 2 and 3 European airports during $H$
$A$	set of Level 2 and 3 European airports for which requests in $R$ are expressed
$A_3 \subseteq A$	set of Level 3 European airports for which requests in $R$ are expressed
$U$	set of airspace users that express requests in $R$
$U_{new}^a, U_{inc}^a \subseteq U$	sets including new entrant, and incumbent airspace users at airport $a \in A$
$RA \subseteq R$	set of requested arrival slots
$RD \subseteq R$	set of requested departure slots
$R_u \subseteq R$	set of requests expressed by user $u \in U$
$R_a^a \subseteq R$	set of requests expressed by user $u \in U$ at airport $a \in A$
$R_t^c \subseteq R \times R$	set of pairs of requested turnaround slots, i.e., pairs $(r, r')$ of a requested arrival slot $r$ and a successive departure slot $r'$ associated to two movements to be operated sequentially by the same aircraft at the same airport
$R_c^2 \subseteq R \times R$	set of pairs of requested coupled slots, i.e., pairs $(r, r')$ of a requested departure slot $r$ and a successive arrival slot $r'$ associated to two movements to be performed by the same aircraft
$I$	set of intervals at all airports in $A$ for the requests in $R$
$I^a \subseteq I$	set of intervals at airport $a \in A$ for the requests in $R$



$S_i$	set of subintervals included in interval $i \in I$ (if the lengths of intervals and subintervals coincide $S_i = \{i\}$ )
$S^r$	set of subintervals to which request $r \in R$ can be allocated within the allowed maximum displacement
$t_s, te_s$	start time, and end time of subinterval $s \in S_i$ , for $i \in I$
$H^a$	set of time periods in which the scheduling horizon $H$ is partitioned to ensure grandfather rights at airport
$a \in A$	$H^a = \{h_k^a, k = 1 \dots, K\}$ s.t. $\bigcup_{k=1, \dots, K} h_k^a = H$ and $h_k^a \cap h_{k'}^a = \text{fork}^{-1} = k'$ , where each $h_k^a$ may include one or more consecutive time intervals $i \in I^a$
$c_i^a$	maximum number of slots that can be allocated at airport $a \in A$ in interval $i \in I^a$
$c_m^{a-I}, ca_m^{a-I}, cd_m^{a-I}$	maximum number of total, arrival, and departure slots, per hour that can be allocated at airport $a \in A$ starting from minute $m$ , when the one-hour periods are considered on a rolling basis (Type I hourly capacity constraints)
$c_m^{a-II}, ca_m^{a-II}, cd_m^{a-II}$	maximum number of total, arrival, and departure slots, per hour, that can be allocated at airport $a \in A$ starting from minute $m$ , when the one-hour periods are considered sequentially (Type II hourly capacity constraints)
$g_{u,h}^a$	minimum number of slots granted to user $u \in U$ at airport $a \in A$ during time horizon $h \in H^a$
$a_r$	airport concerned by request $r \in R$
$\hat{t}_r$	requested slot time for request $r \in R$
$dc_r(s)$	displacement cost for allocating request $r \in R$ to subinterval $s \in S^r$ : $dc_r(s)$ is a linearly increasing function of $t_s - \hat{t}_r$ if $t_s \geq \hat{t}_r$ ; it is a linearly increasing function of $\hat{t}_r - t_s$ otherwise (Cook and Tanner, 2011)
$bt_{r,r'}, \underline{bt}_{r,r'}, \overline{bt}_{r,r'}$	requested, minimum feasible, and maximum feasible block time for coupled requested slots $(r, r') \in R_c^2$ , respectively
$btc_{r,r'}(s, s')$	cost of block time difference for an actual block time $t$ resulting from the allocation to coupled slot requests $(r, r') \in R_c^2$ to subintervals $s$ and $s'$ , respectively: $btc_{r,r'}(s, s')$ is a linearly increasing function of $t - bt_{r,r'}$ if $t \geq bt_{r,r'}$ ; it is a linearly increasing function of $bt_{r,r'} - t$ otherwise (Cook and Tanner, 2011)
$tat_{r,r'}$	minimum turnaround time for turnaround slot requests $(r, r') \in R_t^2$
$Q_r$	cost for the missed allocation of requested slot $r \in R$

### 3.1.1. Remarks

- Requests  $r \in R$  may not belong to a pair of coupled requested slots in  $R_c^2$ :  $R$  includes all requests that either depart from or arrive at a Level 2 or Level 3 European airport. Hence, it also considers flights that may start or end at a Level 1 (slots are not needed) or at an extra-European airport.
- Current rules on grandfather rights allow for a flexibility in schedules from one season to another. In fact, it is not required that grandfather slots of an incumbent airspace user are allocated exactly at the same times as in the previous equivalent season. IATA's guidelines state that for a new season the incumbent users can request slot times (slightly) different from the historical slot times (IATA, 2015). To allow this flexibility, we partition the schedule horizon  $H$  through the introduction of the sets  $H^a$ , for  $a \in A$ . If an airspace user holds grandfather rights on a number of slots in a period  $h \in H^a$ , then it has the right of obtaining the same number of slots in the corresponding period of the next equivalent season, even if the requested slot times in the two seasons are not exactly the same. Consider a user that had a slot within the time interval 10:00–10:10 in the previous season at the airport  $a$ . If  $H^a$  includes, e.g., the period  $h = [10 : 00, 11 : 00]$ , then the same user may take advantage of its grandfather rights in requesting a slot within the time interval 10:50–11:00 for the current season.
- For incumbent users,  $g_{u,h}^a$  (the minimum number of slots granted to user  $u \in U$  at airport  $a \in A$  during time horizon  $h \in H^a$ ) is equal to the number of slots held in the previous equivalent season during the corresponding interval  $h \in H^a$ , for which a movement was regularly operated. For new entrant users,  $g_{u,h}^a$  is a fraction of the number of slots remaining after the allocation of grandfather slots. More precisely, half of the remaining slots are equally distributed among all new entrants requiring them, as specified in IATA guidelines. If the number of slots resulting from this distribution is higher than the number of slots requested by  $u$  in  $h \in H^a$ , then  $g_{u,h}^a$  is equal to this latter (smaller) number. Finally, in accordance with IATA guidelines, the other half of the remaining slots can be allocated to any other unassigned request, either from an incumbent or a new entrant, without further constraints.
- For all the requests  $r \in R$ , the set  $S^r$  includes all the subintervals that are within the maximum displacement. The only exception to this rule occurs when an airspace user makes a request  $r \in R$  associated with a grandfather right. In this case,  $S^r$  only includes the subintervals that are within the interval of the requested slot.
- The limits imposed by the maximum displacement implicitly imply a maximum turnaround time. If  $(r, r')$  are a pair of turnaround slot requests for a given aircraft at a given airport, and  $\hat{t}_r$  and  $\hat{t}_{r'}$  are the associated requested slot times, the earliest possible arrival time for the aircraft is  $\hat{t}_r$  - backward maximum displacement. Since the latest departure time is at  $\hat{t}_{r'}$  + forward maximum displacement, the aircraft maximum turnaround time is  $\hat{t}_{r'} - \hat{t}_r$  + forward maximum displacement + backward maximum displacement.

### 3.2. Decision variables

SOSTA aims to identify the optimal values of the following binary variables:

$$y_r = \begin{cases} 1 & \text{if the requested slot } r \text{ is not allocated} \\ 0 & \text{otherwise} \end{cases} \quad \forall r \in R,$$

$$x_{r,s} = \begin{cases} 1 & \text{if the requested slot } r \text{ is allocated to subinterval } s \\ 0 & \text{otherwise} \end{cases} \quad \forall r \in R, \forall s \in S^r.$$

### 3.3. Objective function

SOSTA minimises the overall airspace users' costs, that is:

$$\min \sum_{r \in R} Q_r y_r + \sum_{r \in R: \exists r': (r,r') \in R_c^2 \vee (r',r) \in R_c^2} \sum_{s \in S^r} dc_r(s) x_{r,s} + \sum_{(r,r') \in R_c^2} \sum_{s \in S^r} \left[ dc_r(s) x_{r,s} + btc_{r,r'} \left( \sum_{s \in S^r} s x_{r',s}, \sum_{s \in S^r} s x_{r,s} \right) \right]. \quad (1)$$

The first term of (1) penalises the missed allocation of requested slots. The second term penalises the displacement of requests not belonging to a pair of coupled requested slots in  $R_c^2$ . Finally, the third term considers all requests belonging to a pair of coupled requested slots in  $R_c^2$ : it penalises the displacement of the requested departure slot and the block time difference due to the allocation. The costs of the missed allocation of a requested slot  $Q_r$  and the form of the cost functions  $dc_r(s)$  and  $btc_{r,r'}(s, s')$  must be given as an input to the model. Since missed allocations represent the last resort,  $Q_r$  are given a much higher value than  $dc_r(s)$  and  $btc_{r,r'}(s, s')$ , thus mimicking a lexicographic optimisation (see also [Assumption 6](#)). In this sense,  $Q_r$  represents the relative importance of missed allocations with respect to displacements, rather than an actual economic cost.

### 3.4. Constraints

SOSTA requires the following sets of constraints to be satisfied:

- Two coupled slot requests are either both allocated or none of them is allocated:

$$y_r = y_{r'} \quad \forall (r, r') \in R_c^2 \cup R_c'^2. \quad (2)$$

- Each requested slot is allocated to exactly one subinterval unless it is subject to a missed allocation:

$$\sum_{s \in S^r} x_{r,s} = 1 - y_r \quad \forall r \in R. \quad (3)$$

- All the airport capacity constraints are respected:

$$\sum_{r \in R: a_r = a} \sum_{s \in S^r: t_s \in S_i} x_{r,s} \leq c_i^a \quad \forall a \in A, i \in I^a; \quad (4)$$

$$\sum_{r \in R: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq c_m^{a-I} \quad \forall a \in A, m = 0, 5, 10, \dots, 1435; \quad (5)$$

$$\sum_{r \in RA: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq ca_m^{a-I} \quad \forall a \in A, m = 0, 5, 10, \dots, 1435; \quad (6)$$

$$\sum_{r \in RD: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq cd_m^{a-I} \quad \forall a \in A, m = 0, 5, 10, \dots, 1435; \quad (7)$$

$$\sum_{r \in R: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq c_m^{a-II} \quad \forall a \in A, m = 0, 60, 120, \dots, 1380; \quad (8)$$

$$\sum_{r \in RA: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq ca_m^{a-II} \quad \forall a \in A, m = 0, 60, 120, \dots, 1380; \quad (9)$$

$$\sum_{r \in RD: a_r = a} \sum_{s \in S^r: t_s \in [m, m+55]} x_{r,s} \leq cd_m^{a-II} \quad \forall a \in A, m = 0, 60, 120, \dots, 1380. \quad (10)$$

Specifically, Constraints (4) impose the interval capacity constraints: each constraint sets the limit on the number of slots that can be allocated to subintervals belonging to an interval. Constraints (5)–(7) and Constraints (8)–(10) impose Type I and Type II hourly capacity constraints, respectively. Each triplet consists of the setting of the limit for the total number of allocated slots, the number of arrival slots and the number of departure slots, in this order. They are set for all airports and one-hour periods for which a capacity value is present in the input data. Since in one day there are 1440 min and, as introduced in Section 2.2, Type I hourly capacity constraints refer to as a “60-min period starting at each multiple of 5 min”,

the last possible starting time in a day is at minute 1435 for Constraints (5)–(7). Similarly, since Type II hourly capacity refer to as a “60-min period starting at the beginning of each hour” the last possible starting time in a day is at minute 1380 for Constraints (8)–(10).

- Coupled slot requests are allocated to subintervals respecting the minimum and the maximum block time:

$$\sum_{s \in S^r} t_{s,r} x_{r,s} - \sum_{s \in S^r} t_{s,r} x_{r,s} \leq \overline{bt}_{r,r'} \quad \forall (r, r') \in R_c^2; \quad (11)$$

$$\sum_{s \in S^r} t_{s,r} x_{r,s} - \sum_{s \in S^r} t_{s,r} x_{r,s} \geq \underline{bt}_{r,r'} \quad \forall (r, r') \in R_c^2. \quad (12)$$

- Turnaround slot requests are allocated to subintervals respecting the minimum turnaround time:

$$\sum_{s \in S^r} t_{s,r} x_{r,s} - \sum_{s \in S^r} t_{s,r} x_{r,s} \geq \overline{tat}_{r,r'} \quad \forall (r, r') \in R_t^2. \quad (13)$$

- At each Level 3 airport, each user is allocated a number of slots at least equal to the minimum number of its granted slots in each time period  $h \in H^a$ :

$$\sum_{r \in R_{inc}^a} \sum_{s \in S^r: t_{s,r} \in h} x_{r,s} \geq g_{u,h}^a \quad \forall a \in A_3, h \in H^a, u \in U_{inc}^a \cup U_{new}^a. \quad (14)$$

Other constraints may also be considered. For instance, we can impose constraints very similar to (13) to allow coupling between the arrival and departure slots of connecting flights.

## 4. Computational experiments

A non trivial issue that always arises when dealing with problems concerning air traffic over the European skies is the collection of real data. Works present in the literature usually either consider just a small number of airports/sectors (e.g., Zografos et al., 2012) or generate artificial data (e.g., Bertsimas et al., 2011; Castelli et al., 2012; Corolli et al., 2014). In this section, we focus on the computational experiments carried out with the real data that were accessible to us. To this aim, we first describe the data sources and the data items used, and then we present the experimental analysis run on these data. The computational experiments show that SOSTA captures the existing regulations and best practices, and can be used in practice as an effective tool to evaluate the impact of (large) imbalances between capacity and requests. This second part of the analysis shows that computational performances do not worsen significantly when capacity constraints become tighter. In the computational analysis, we also assess the sensitivity of the performance of SOSTA as a function of the maximum schedule displacement allowed for each request. Finally, we show how the objective function may be changed for capturing the need for fairness in the displacements of requests of different users.

### 4.1. Data sources

Two key data items are needed to run SOSTA: airport capacities and airspace users' requests for each European Level 2 and Level 3 airport.

**Airport capacity** We mainly rely on two data sources.

1. Airport coordinators' websites. Some airport coordinators publish on their websites coordination parameters that describe the airport infrastructure capacity. Unfortunately, such information is publicly available for a limited set of airports only (the list of worldwide coordinators can be found at <http://www.wwacg.org>).
2. Airport capacity figures from Demand Data Repository 2 (DDR2), a database containing airspace network and traffic data. It is developed and maintained by EUROCONTROL (the European Organisation for the Safety of Air Navigation) and contains the capacity figures for different elements of the airspace network, including airports.

For a few airports (23 out of 152 in our dataset) both of the above options were unsuccessful. In such cases, we conservatively assumed the airport capacity equal to the maximum number of movements actually observed per time period, making a distinction between night and daylight hours.

**Airspace users' slot requests** To receive the slots for a specific airport, airspace users have to send a request to the national slot coordinator this airport refers to. A few online tools exist to ease this process (such as the Online Coordination System, [www.online-coordination.com](http://www.online-coordination.com)), but consisting of a limited selection of airports available. We have instead access to EUROCONTROL's Slot Coordinator data, and this is the main data source used in this study. The information used, for each individual flight at European Level 2 and Level 3 airports, comprises of: origin and destination airport, requested and assigned departure and arrival slot times and dates. Furthermore, the distribution of allocated slots per airport, and the user of each slot, at each airport is available. However, there is no explicit mention of the grandfather

rights, or of the slot series. It is also worth noting that only a small percentage of flight records (about 1%) show the displacement between the requested and allocated slots. The reason is that the Slot Coordinator shows only the data that remains in the slot coordinator database at the end of the period but does not reflect the situation at any of the previous stages of the slot allocation process. Hence the small number of slots holding a time different from the requested one occurs because many airspace users have already either cancelled their unwanted slots or adjusted their schedules to these not ideal times. In fact, the aim of the Slot Coordinator is the post-analysis, mainly the slot usage monitoring.

Because of the information available in the Slot Coordinator, SOSTA respects regulations, constraints and good practices in terms of, e.g., airport capacities, flight block times, aircraft turnaround times, airspace users' historic rights, and allocation to new entrants. The model is introduced and mathematically formulated in Sections 2 and 3, respectively, relying on the Slot Coordinator data. The resulting instances include about 145,000 binary variables and 243,000 constraints.

#### 4.2. Validation

In the previous subsection, we pointed out that the real data available presents only a small percentage of flight records with displacement between the requested and allocated slots, probably because the users have already ex-post optimised their flights with respect to the allocated slots. Here, to validate SOSTA, we used the above data on the requested slots as an input. Then, if the assumptions on the structure of the model constraints, on the objective function, and on the values of the model parameters are correct, SOSTA should provide a slot allocation very similar to the current practice. If this is the case, we can claim that a centralised model as SOSTA can reach in short time the same, or even better, result in terms of feasibility and cost reduction as the decentralised IATA procedure, which requires first the solution of many local allocation problems and then a significant effort to coordinate the local allocations and obtain a final feasible global solution.

Specifically, we tested SOSTA on the slot allocation that occurred at Level 2 and Level 3 European airports on June, 28th, 2013, that is, on the day with the greatest number of movements in 2013. In particular, following some data cleaning, we consider 32,665 out of 33,006 slot requests, as recorded in the Slot Coordinator database for this day. The data in the Slot Coordinator contains, as mentioned in Section 4.1, both the requested and the allocated slot times. For each requested slot  $r$ , the coupled request is indicated, if it exists. For coupled requests, we deduced the requested block time by subtracting the requested departure from the requested arrival slot.

Based on conversations with airspace user representatives and airport coordinators, we consider a maximum displacement of 30 min (positive or negative) for each request. This value is a model parameter and can be changed, as we show in Section 4.4.1. The minimum block time is set equal to the requested block time, as, in the available data, airspace users most often request coupled slots coherent with the fastest route between origin and destination. Since for each request, the coordinator receives the information on airline, aircraft type and OD pair, the minimum block time could be determined in such a way. The maximum block time is always set equal to the minimum block time plus 15 min. Nevertheless, the sensitivity of the solution with respect to the maximum additional block time is investigated in Section 4.4.2. The minimum turnaround time is set to 90 min for wide-body aircraft and 30 min for other aircraft types, in accordance with the information available to us from traffic data and airport planning characteristics ([Airbus Aircraft Characteristics Airport and Maintenance Planning, 2016](#); [Boeing Airplane Characteristics for Airport Planning, 2016](#); [Van Landeghem and Beuselinck, 2002](#)). This threshold is respected in 97.7% of turnarounds available in our dataset. In accordance to [Assumption 6](#), which states that any displacement (within the maximum allowed displacement) is preferable to an additional missed allocation, we set  $Q_r = 30,000$ . This value guarantees that the cost of a missed allocation is significantly higher than the total displacement cost and hence ensures the lexicographic optimisation of the number of missed allocations over the total displacement cost. Note that from a lexicographic perspective the precise value of  $Q_r$  has no impact on the optimisation result as far as it is sufficiently high. Displacement costs are represented by a linear monotone function: the cost of an allocation is equal to the number of minutes of the displacement it implies ([Cook and Tanner, 2011](#)).

In accordance with IATA guidelines ([IATA, 2015](#)) that describe the allocation of slots to incumbent airspace users on the basis of historical precedence (see Section 1), grandfather rights for different airspace users at a given airport are deduced by observing the slots allocated to each airspace user in the previous equivalent season (i.e., summer 2012 for summer 2013), in the same week, and on the same day-of-the-week. To consider the flexibility allowed by airport coordinators to airspace users in the exploitation of grandfather rights, we set four time horizons  $h \in H^a$  at each coordinated airport: 0:00–9:00, 9:00–15:00, 15:00–19:00, and 19:00–24:00.

We ran SOSTA on a computer with eight Intel Xeon 3.5 GHz processors and 128 GB RAM, under Linux Ubuntu distribution version 14.04, using CPLEX 12.6 as integer linear programming solver. CPLEX found the proven optimal solution in 125 CPU seconds.

The slot allocation proposed by SOSTA matches the actual slot allocation for all but six requests: 32659 out of 32665 requests. In all the cases, the displacement proposed differs from the actual one for 10 min at the most. The slightly different combination allows reducing the total displacement for 5 min.

The final slot allocation leads to the saturation of some interval capacity constraints at several airports. Specifically, at least one interval is saturated in 120 airports of 23 European nations. Even many minor airports, above all in Spain and Greece, had some saturated intervals due to the beginning of the tourist season and relatively low capacity of these airports.

In addition, some night intervals (between 11:00 pm and 6:00 am) are also saturated at 31 airports, due to airports' capacity reductions because of night curfew restrictions.

Furthermore, SOSTA slot allocation results in the saturation of hourly capacities (Type I hourly capacity constraints) at only 15 airports. Most of them are the major European airports, which reportedly have capacity problems. Only two airports have a significant number of saturated hours: Heathrow (UK), and Linate (Italy). Heathrow is reportedly (SDG, 2011) the most congested European airport, while Linate has very binding capacity constraints.

These results are consistent with the findings reported in SDG (2011). Many European airports experience some peaks of the number of requests in relatively short periods. However, under normal operating conditions as the ones considered in the slot allocation phase, the capacity of very few airports is saturated for long periods.

In summary, the slot allocation returned by SOSTA almost perfectly matches the one resulting from the actual slot allocation process. SOSTA appears to reproduce the IATA slot regulations and best practices.

In the following, we propose further experimental analyses. In Section 4.3, we test SOSTA in conditions of severe imbalance of demand and capacity. In Section 4.4, we assess the sensitivity to the values of two parameters: the maximum displacement and the maximum additional block time. In Section 4.5, we consider quadratic shape of the displacement cost function, and in Section 4.6 we analyse the impact of the presence of grandfather rights and of their flexibility. Finally, in Section 4.7, we propose two variants of SOSTA which take into account the fairness of the penalisation of airspace users in the optimisation.

### 4.3. Analysis of demand and capacity imbalances

In the following, we describe the testing of the behaviour of SOSTA, using the real data, when slot requests exceed the available capacities. As we did not want to mix real and artificial data, we opted for reducing the airports' capacities instead of adding new fictitious slot requests. In particular, we considered worst case scenarios in which the reduction of capacity occurs uniformly all over Europe. The aim of these tests was to verify whether a simultaneous allocation of slots across all airports produces significant benefits with respect to the current approach to slot allocation (*Current* scenario), which is the airport by airport slot allocation. In particular, we analyse different levels of imbalance between slot requests and capacity, and show to what extent SOSTA makes it possible to reduce the number of missed allocations and the displacement cost with respect to the *Current* scenario. Relying again on the 28th June 2013 instance, we consider uniform reductions of airport capacities of 5, 10, 15, 20, 25 and 30 per cent over all European airports. In all tests, the aim is first to allocate as many slot requests as possible within the limits imposed by the constraints on the maximum feasible displacements, and then minimise the total displacement cost. The *Current* slot allocation is simulated through a two-step process:

1. To mimic the problem addressed by airport coordinators during the primary allocation phase, a separate slot allocation for each individual airport is performed using a simplified version of SOSTA, i.e., without considering turnaround and coupled slot requests (constraints (2), (11)–(13)). Therefore at each airport, first, the number of missed request allocations and, second, the total displacement costs, are minimised.
2. Once an airspace user is aware of all the slots allocated to it at each different airport (following Step 1), it optimises their exploitation by minimising, first, the number of its missed request allocations and, second, its total displacement cost. In this individual optimisation, the airspace user freely decides how to exploit its allocated slots: no constraint is imposed on the flight which should use them. However, no possibility is given to use slots allocated to other users in Step 1. Hence, no conflicts can arise at this stage. The individual airspace user optimisation can be performed using, again, SOSTA, which considers just one element  $u \in U$  at a time and takes into account both coupled and turnaround constraints. The number of missed request allocations and total displacement costs resulting with this paired individual airport and individual airspace user optimisations represent the performance achieved by the *Current* scenario.

Table 2 presents the main results. In particular, it shows the number of missed request allocations and total displacement cost in the SOSTA and *Current* optimal solutions for different levels of airport capacity reductions. It appears the simultaneous slot/request allocation at all airports significantly outperforms the current allocation process: the number of missed request allocations is from 86% (minor capacity reduction – 5%) to 59% (major capacity reduction – 30%) lower. The total displacement cost is on average 5% lower when using SOSTA, with a maximum decrease of 18% (minor capacity reduction – 5%) and a maximum increase of 1% (major capacity reduction – 30%). Since the minimisation of the number of missed allocation is the first objective of SOSTA, it always chooses a solution with fewer missing allocations, even if this may imply a higher displacement cost. Finally, the last column of Table 2 displays the CPU times needed by the SOSTA optimisation, which indeed exhibits very good computational performance.

As an example, in the case with 20% reduction of airport capacities, the solution proposed by SOSTA identifies 860 missed allocations of requested slots at 60 European airports, 94 of which are associated with coupled requests. Hence, some of the missed allocations at minor airports are a consequence of missed allocation of the coupled requests at major, more congested airports. Furthermore, this analysis highlights where and to what extent congested airports are distributed across Europe. In fact, it appears that more than 200 missed allocations occur at Heathrow (UK), more than 50 at Charles De Gaulle (France), Munich (Germany), Frankfurt (Germany) and Dusseldorf (Germany), and more than 25 at Gatwick (UK), Zürich (Switzerland), Linate (Italy) and Vienna (Austria). Only 5 other airports, typically of capital cities, suffer from more than 10 missed

**Table 2**

Number of missed request allocations and total displacement cost in the SOSTA and *Current* optimal solutions for different levels of airport capacities reductions, and corresponding computational times for SOSTA.

Capacity reduction	No. missed allocations		Displacement cost		Computational time SOSTA (CPU sec)
	SOSTA	<i>Current</i>	SOSTA	<i>Current</i>	
5%	31	229	1,591	1,949	88
10%	188	667	4,072	4,447	102
15%	477	1,432	7,079	7,095	96
20%	860	2,586	10,648	10,634	140
25%	1,302	3,427	13,988	14,764	705
30%	2,029	4,942	18,381	18,214	840

allocations. Fig. 3 displays the number of missed allocations over all Europe using SOSTA. A colour and a size scale defines the number of missed allocations at each airport: the larger the bullet, the higher the number of missed allocations at the corresponding airport. Red bullets represent more than 100 cancellations, blue bullets between 30 and 100 cancellations, and green bullets between 10 and 30 cancellations. Many of the minor airports experience very few or no missed allocations, except those induced by a missed allocation at a major airport. Differently, Heathrow and few other major airports have little or no spare capacity. In particular, the area around London is particularly congested: some missed allocations occur at each of its five main airports (Heathrow, Gatwick, Stansted, City, and Luton). The benefits of the simultaneous slot allocation over the current one, at European level, can be clearly seen when comparing Fig. 3 with Fig. 4. Fig. 4 shows the allocations under the *Current* scenario, using the same colour and size scales.

#### 4.4. Sensitivity analysis

As input data, SOSTA needs the definition of some parameters by the airspace users. They are likely to have an impact on the final slot allocation and on the computational performance. In this section, we propose some sensitivity analyses to quantify this impact. In particular, we focus on the value of the maximum displacement and of the maximum additional block time.

##### 4.4.1. Sensitivity to maximum displacement

A set of input data which SOSTA needs for each request is the maximum displacement which can be accepted by the user. If a solution implying a displacement smaller than or equal to this threshold does not exist, the request is added to the set of missed allocations. In this section, we observe the sensitivity of the allocation and of SOSTA computational performance with respect to this threshold.

To deal with an instance in which some imbalance between slot offer and demand exist, we consider the case with 20% airport capacity reduction from Section 4.3. These results are representative of the ones obtained with the other percentages of reduction.

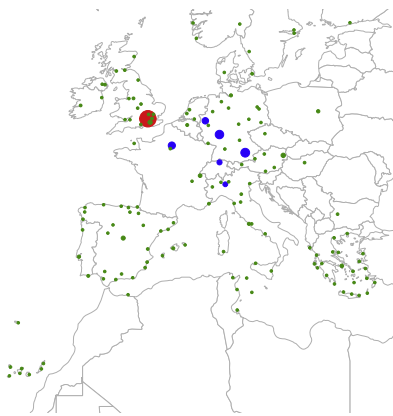
Table 3 reports the number of missed allocations, the total displacement and the CPU time corresponding to the optimisation with maximum displacement of 15, 30, 45 or 60 min. These results show that the maximum displacement has an impact on the optimal solution found by SOSTA. As expected, this impact is positive: the higher the maximum displacement, the fewer the number of missed allocations. The counterweight of this smaller number is the increase of the total displacement cost. We finally observe that the computational time does not significantly increase when the maximum displacement reaches the very large value of 60 min. Even if in this case the size of the instance becomes quite large (280,000 binary variables and 378,000 constraints), the optimisation proves the optimality of the solution in less than 15 min CPU. Hence, we can conclude that the result of the slot allocation is sensitive with respect to the value of the maximum displacement for each request. However, SOSTA can deal with rather large values without a remarkable impact on its computational performance.

##### 4.4.2. Sensitivity to maximum additional block time

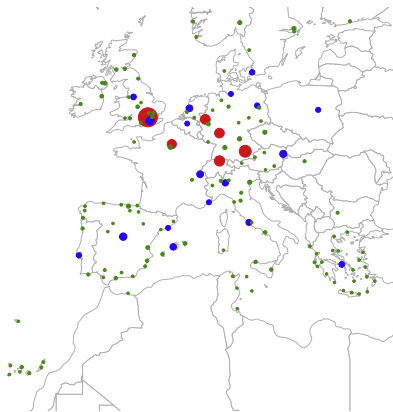
Another set of input data which SOSTA needs for each request is the maximum additional block time which can be accepted by the concerned user. As for the maximum displacement, if a solution implying an additional block time smaller than or equal to this threshold does not exist, the request is added to the set of missed allocations. Paralleling the analysis performed in Section 4.4.1, here we describe the sensitivity of SOSTA's allocation and computational performance with respect to this threshold. As in 4.4.1, we consider the 20% airport capacity reduction from Section 4.3.

Table 4 reports the number of missed allocations, the total displacement cost and the CPU time corresponding to the optimisation with maximum additional block time of 10, 15 or 20 min. As an average flight duration in Europe is between an hour and a half and two hours, we deem the additional block times higher than 20 min unrealistic (EUROCONTROL and FAA, 2012).





**Fig. 3.** Missed allocations induced by 20% reduction of airport capacities using SOSTA.



**Fig. 4.** Missed allocations induced by 20% reduction of airport capacities in the *Current* scenario.

These results show that, as expected, the higher the allowed additional block time, the better the solution. In particular, by passing from 10 to 20 min allowed, the number of missed allocation decrease of almost 4%, while the total displacement cost increases slightly. The computational times remain on the order of a few minutes.

As in the case of maximum displacement, SOSTA is sensitive to the specific values chosen for the additional block time allowed in terms of solution returned but not in terms of computational performance.

#### 4.5. Displacement cost function

The slot allocation produced by SOSTA also depends on the displacement cost function associated with each slot request. In the experiments reported in the rest of this paper, this function is linear, following [Cook and Tanner \(2011\)](#). However, to simulate costs that become gradually steeper as the displacement increases, in this section we study the impact of the shape of this function on SOSTA. In particular, we reproduce the analysis reported in Section 4.3, but consider costs equal to the square value of the displacement. [Table 5](#) reports the results of the two sets of experiments, with the linear and quadratic displacement cost functions.

The results show that the shape of this function does not bring the modification of the number of missed allocation. As a matter of fact, the minimisation of this number is the first objective of SOSTA's lexicographic optimisation and it does not depend on the displacement cost function.

The total displacement cost of course changes, since with the linear function (third column of [Table 5](#)) it is equal to the number of minutes of displacement, while with the quadratic function (sixth column of [Table 5](#)) it is equal to the sum of the square of the minutes of displacement of each request. As expected, by moving from linear to quadratic displacement cost function, the number of requests which suffer a short displacement increases and the opposite holds for the number of those for which the displacement is long. [Fig. 5](#) illustrates the number of requests displaced for each amount of minutes for the

**Table 3**

Number of missed request allocations and total displacement cost in the SOSTA optimal solutions for different values of maximum displacement per request, and corresponding computational times. Instance with 20% airport capacity reduction.

Maximum displacement	No. missed allocations	Displacement cost	Computational time (CPU sec)
15	998	5,661	111
30	860	10,648	140
45	792	14,442	776
60	759	17,108	834

**Table 4**

Number of missed request allocations and total displacement cost in the SOSTA optimal solutions for different values of maximum additional block time per request, and corresponding computational times. Instance with 20% airport capacity reduction.

Maximum additional block time	No. missed allocations	Displacement cost	Computational time (CPU sec)
10	862	10,507	133
15	860	10,648	140
20	854	10,972	547

**Table 5**

Utilization of linear and quadratic cost displacement function: Number of missed request allocations and total displacement cost in the optimal solutions for different levels of airport capacities reductions, and corresponding computational times.

Capacity reduction	Linear cost function			Quadratic cost function		
	No. missed allocations	Disp. cost	CPU time (sec)	No. missed allocations	Disp. cost	CPU time (sec)
5%	31	1,591	88	31	21,607	96
10%	188	4,072	102	188	54,601	504
15%	477	7,079	96	477	104,114	110
20%	860	10,648	140	860	152,265	290
25%	1,302	13,988	705	1,302	185,938	1,088
30%	2,029	18,381	840	2,029	260,412	961

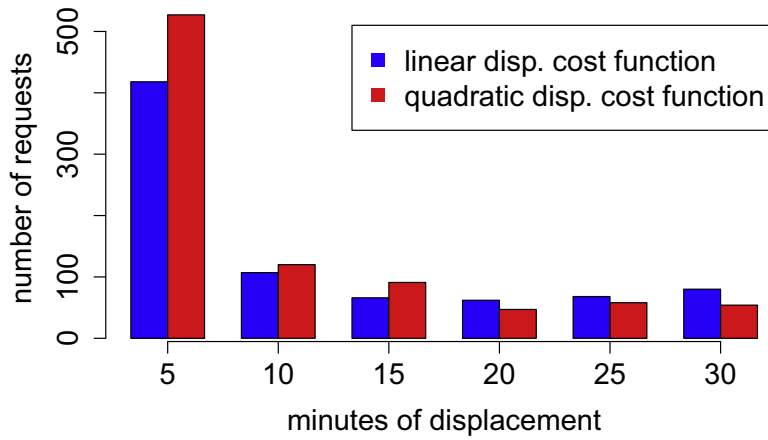
case of 20% of airport capacity reduction. Here, the trend toward shorter displacements is clear: for instance, 527 requests are displaced for five minutes when the cost function is quadratic, instead of the 418 obtained with the linear one. Finally, the computational times are always small.

#### 4.6. Grandfather rights' relaxation

In this section, we assess the impact of the flexibility and the presence of grandfather rights on SOSTA: first, we vary the duration of the time horizons  $h \in H^a$  by allowing a full flexibility, i.e., we assume only one time horizon of 24 h per airport; second, we completely eliminate grandfather rights. Table 6 reports the results obtained by SOSTA in these two cases.

The effects of the flexibility permitted to airspace users to exploit grandfather rights are shown by comparing SOSTA's results in Table 2 and in columns 2–4 of Table 6. We see that a higher flexibility has a twofold impact on the model. On the one hand, at each coordinated airport, the number of constraints imposing a specific number of slots for airspace users is reduced: from four constraints (four time horizons) to one (24-h horizon). On the other hand, the number of grandfather rights guaranteed may increase: in the four time horizon case, if an airspace user had a slot in the first time horizon in the preceding equivalent season and now it requests a slot in the second time horizon, no grandfather right is guaranteed. Instead, in the 24 h horizon case, this right is maintained. This twofold effect implies that the model flexibility does not necessarily improve. In fact, the results in terms of the number of missed allocations show that with all levels of capacity reduction but 5% and 10%, the allocation with full flexibility is slightly better than the one reported in Table 2. The computational times are comparable across the two grandfather rights flexibility options, being slightly higher in the case of full flexibility in a few cases.

A more intuitive relation is found between the cases of presence (Table 2) and absence of grandfather rights. This latter set of results are reported in the last three columns of Table 6. Here, the elimination of grandfather rights strongly improves the final slot allocation: the number of missed allocations decreases between 55% and 90% with respect to Table 2's corresponding capacity reductions. Instead, the computational times increase significantly, being quite large even for the easiest instance (5% capacity reduction). This is due to the elimination of several constraints and hence to the much larger number of feasible solutions. However, the analysis of the results show that the wall-clock time needed to solve the instances is larger than one hour only in the most difficult case of a 30% capacity reduction. However, since SOSTA is aimed at replacing a process which takes months in its current implementation, a computation of a few hours is acceptable.



**Fig. 5.** Distribution of the number of requests displaced for the case of 20% airport capacity reduction, with the linear and the quadratic displacement cost function.

**Table 6**

Number of missed request allocations, total displacement cost and corresponding computational times for SOSTA when full flexibility is permitted for grandfather rights and when no grandfather rights are guaranteed.

Capacity reduction	Flexible GFR			No GFR		
	No. missed allocations	Disp. cost	CPU time (sec)	No. missed allocations	Disp. cost	CPU time (sec)
5%	32	1,591	70	3	1,892	6,628
10%	188	4,080	500	67	4,493	6,430
15%	474	7,082	88	153	15,405	12,802
20%	855	10,702	669	334	26,981	19,107
25%	1,289	14,025	645	543	40,598	64,009
30%	2,025	18,347	1,158	904	60,734	505,797

In conclusion, we can state that SOSTA can effectively deal with the simultaneous slot allocation problem even if a full flexibility is permitted for the grandfather right exploitation, or if no grandfather rights exist. In the latter case, it finds much better slot allocations, at the cost of higher computational time.

#### 4.7. Fairness concern

The model proposed in Section 3 minimises the number of missed allocations and the total displacement cost imposed on users. As it optimises from a system's perspective, some users might be particularly penalised in the allocation. Indeed, fairly allocating the available slots in case of capacity shortage is not an easy task, since the definition of fairness in this case is not necessarily commonly agreed. As the aim of SOSTA is the application of IATA guidelines at the network level, finding novel strategies to ensure the rules are adhered to is out of the scope of this paper. However, a number of schemes for fair distribution of missed allocations and displaced requests among the users can be thought of.

To this aim, here we consider two possible variants of SOSTA to take into account the fairness of the result. In particular, in the first variant, we adopt as possible measures of fairness the maximum cost of missed allocations over the users, and the maximum total displacement cost over the users, represented by two new variables  $M_{missed}$  and  $M_{cost}$ , respectively. To set the values of these variables, we include the following constraints into the model:

- The cost of the missed allocations for each user is smaller than or equal to  $M_{missed}$ :

$$\sum_{r \in R_u} Q_r y_r \leq M_{missed} \quad \forall u \in U. \quad (15)$$

- The sum of the displacement costs of each user is smaller than or equal to  $M_{cost}$ :

$$\sum_{r \in R_u: \exists r': (r, r') \in R_c^2 \vee (r', r) \in R_c^2} \sum_{s \in S^r} dc_r(s) x_{r,s} + \sum_{(r, r') \in R_c^2 \cap R_u \times R_u} \sum_{s \in S^r} \left[ dc_r(s) x_{r,s} + btc_{r,r'} \left( \sum_{s \in S^{r'}} s x_{r',s} + \sum_{s \in S^r} s x_{r,s} \right) \right] \leq M_{cost} \quad \forall u \in U. \quad (16)$$

Including these new variables, the objective function is formulated as:

$$\begin{aligned} \min w_m (M_{missed} + M_{cost}) \\ + \frac{1}{|U|} \left\{ \sum_{r \in R} Q_r y_r + \sum_{r \in R: \exists r': (r, r') \in R_c^2 \vee (r', r) \in R_c^2} \sum_{s \in S'} dc_r(s) x_{r,s} + \sum_{(r, r') \in R_c^2} \sum_{s \in S'} \left[ dc_r(s) x_{r,s} + btc_{r,r'} \left( \sum_{s \in S'} s x_{r',s}, \sum_{s \in S'} s x_{r,s} \right) \right] \right\} \end{aligned} \quad (17)$$

where  $|U|$  is the number of users and  $w_m$  is the weight assigned to the maximum cost of missed allocations and to the maximum total displacement cost per user. Then, we can interpret the objective function (17) as minimising a linear combination of the costs paid by the average and by the most penalised user. By setting an appropriate value of  $w_m$ , the modeller can choose how to weigh the costs paid by the most penalised users with respect to the ones paid by the average users. In a sense, the higher  $w_m$ , the fairer the system: it is always possible to set a value high enough to ensure that SOSTA does not sacrifice a single user to minimise the overall costs.

With Constraints (15) and (16), and objective function (17), SOSTA considers the costs imposed on each user independently of the number of requests the user submits. To differently weigh the number of missed allocations assigned to a small (i.e., making a few requests) compared to a large (i.e., making tens or hundreds requests) user, we introduce a second variant of SOSTA. This variant is based on a different definition of fairness that takes into account the size of the users. The objective function of this second variant considers again the costs of both the average and the most penalised users. However, it normalises the costs to take into account users' size. Specifically, the costs of each user are normalised by dividing by the number of the users' requests. The new constraints and objective function are:

$$\frac{1}{|R_u|} \sum_{r \in R_u} Q_r y_r \leq M_{missed} \quad \forall u \in U, \quad (18)$$

$$\frac{1}{|R_u|} \left\{ \sum_{r \in R_u: \exists r': (r, r') \in R_c^2 \vee (r', r) \in R_c^2} \sum_{s \in S'} dc_r(s) x_{r,s} + \sum_{(r, r') \in R_c^2 \cap R_u \times R_u} \sum_{s \in S'} \left[ dc_r(s) x_{r,s} + btc_{r,r'} \left( \sum_{s \in S'} s x_{r',s}, \sum_{s \in S'} s x_{r,s} \right) \right] \right\} \leq M_{cost} \quad \forall u \in U, \quad (19)$$

$$\begin{aligned} \min w_m (M_{missed} + M_{cost}) \\ + \frac{1}{|U|} \sum_{u \in U} \frac{1}{|R_u|} \left\{ \sum_{r \in R_u} Q_r y_r + \sum_{r \in R_u: \exists r': (r, r') \in R_c^2 \vee (r', r) \in R_c^2} \sum_{s \in S'} dc_r(s) x_{r,s} + \sum_{(r, r') \in R_c^2: r, r' \in R_u} \sum_{s \in S'} \left[ dc_r(s) x_{r,s} + btc_{r,r'} \left( \sum_{s \in S'} s x_{r',s}, \sum_{s \in S'} s x_{r,s} \right) \right] \right\}. \end{aligned} \quad (20)$$

Through the normalisation, we can define an objective function which can be interpreted as minimising a linear combination of the costs per request paid by the average and by the most penalised users. To some extent, in this variant of SOSTA, the larger the number of requests by a user, the less costly the penalisation is with respect to the average user. Hence, the advantage of requesting "useless" slots is at least reduced with respect to the classic SOSTA model, which accurately reproduces the existing IATA guidelines.

In the following, we refer to the model including conditions (15)–(17) as applying the *fair* objective function, and to the one including (18)–(20) as applying the *normalised fair* objective function. Table 7 reports the results of this analysis by comparing the solutions obtained with the *classic* objective function presented in Section 3 to those obtained with the *fair* and the *normalised fair* ones. We report only the results for the case of 20% airport capacity reduction, as they are representative of the ones for all the levels analysed in Section 4.3. In particular, for different values of the weight  $w_m$ , the total number of missed allocations (*tot*) and the maximum number of missed allocations that a single user is subject to (*max*) are shown. On the one hand, as expected, considering fairness in the allocation implies a worse performance at the system level (columns *tot*): the total number of missed allocations increases when passing from classic to fair or normalised fair, for all the values of  $w_m$  considered. In particular, the higher  $w_m$ , the worse the overall results. As mentioned in the description of the fair conditions, this is intuitive: the more we weigh the penalisation of a single user with respect to the average one, the higher the number of missed allocations for the latter. On the other hand, the maximum number of missed allocations per single user significantly decreases with respect to the classical solution. In the normalised fair allocation, this value is expressed as a percentage because its contribution to the objective function is a quantity representing the portion of missed allocations per user.

The computational time needed for SOSTA to prove the optimality of the solution increases when adding fairness, always remaining at an acceptable level. Indeed, the time is longer with the fair objective function, since a much higher number of equivalent solutions exist with respect to the case of the normalised fair objective. Even in this case, the wall-clock time for the optimisation remains under thirty minutes.

These results show that SOSTA allows a noticeable flexibility in terms of the constraints and objective functions considered. In particular, if fairness considerations are taken into account, SOSTA permits to do so without a significant loss of per-

**Table 7**

Total and maximum number (assigned to a single user) of missed request allocations in the SOSTA optimal solution with the classic, fair and normalised fair objective function for the case of 20% of airport capacities reduction and for different values of  $w_m$ , and corresponding computational times.

$w_m$	Fair			Normalised fair		
	No. missed allocations		Comp. time	No. missed allocations		Comp. time
	<i>tot</i>	<i>max</i>	(CPU sec)	<i>tot</i>	<i>max</i>	(CPU sec)
1	1,006	26	3,428	1,131	7.21%	270
2	1,062	22	8,720	1,144	6.62%	300
4	1,064	20	22,144	1,164	6.57%	297
16	1,098	18	5,872	1,166	6.51%	453
Classic	860	211	140			

formance from a computational point of view. The approach we used can be extended if other specific fairness definitions were to be adopted.

## 5. Conclusions

The current airport slot allocation process is a complex process in which optimisation is not really used. In the literature, the need for introducing an optimisation-based decision-aid tool is unanimously recognised, but an agreement on the characteristics of this tool is not reached yet.

In this paper, we propose SOSTA, an integer linear programming model to simulate the slot allocation process currently performed in Europe. Its main original features consist of capturing most of the regulations and best practices currently applied, and in solving European-size real instances in a very short computational time. Moreover, it allocates slots at all airports simultaneously, taking into account all the network effects like connections between departure and arrival slots, and aircraft turnarounds.

We validated and tested SOSTA on real data: we considered all requests on the busiest day of 2013. The available data are those that remained in the database when airspace users had already either cancelled their unwanted slots or adjusted their schedules to non ideal times. Hence, SOSTA could not provide a large improvement with respect to the available (and almost completely agreed) allocation. However, even if applied in this late phase of the process, SOSTA identified an improved allocation with respect to the actually implemented one, showing that optimisation can bring significant advantages at any moment of the process. Indeed, if fed with data of the very first airspace users' requests, SOSTA could give a greater contribution in performing the slot allocation process effectively. Furthermore, the good computational performance in the experiments where airport capacities are significantly decreased suggests that the slot allocation through SOSTA will not suffer from large imbalances between demand and capacity. Similarly, SOSTA performance will not significantly worsen if the maximum displacement cost or the additional block time varies, or if a different objective function is considered. In particular, we showed how fairness considerations may be included in the allocation process.

The use of SOSTA may improve the current slot allocation process under two perspectives. First, through the optimisation, it allocates slots based on user requests, using a predefined objective function. In this paper, we propose the minimisation of the number of requests that cannot be satisfied, followed by the minimisation of the sum of the displacement costs of all airspace users. The objective function, as well as some of the constraints, like the maximum displacement can be easily changed if needed.

Second, in the current process, airspace users build their desired flight schedule and request the slots necessary to perform it. Then, they might need to re-build this schedule to match the slots they are actually allocated at different airports. SOSTA, an optimisation model that can assign slots and schedule requests at all Level 2 and 3 airports in Europe can significantly decrease the efforts needed in the current system. In principle, such a model could be used as a one-stop-shop for the seasonal slot allocation. In practice, it is more likely that any slot allocation would be subject to ex-post negotiation. The relatively short computation times needed by SOSTA (on the order of hundreds of CPU seconds) allow for quick analysis of different scenarios. Also, SOSTA would allow for iterations to fine tune the model parameters (e.g., costs, intervals) with the users and to manage the requests the airspace users are not content with the proposed slot assignments. Of course, in this latter situation, the discontent and the additional iterations should be reasonable, and within the established IATA guidelines. As an example, one could assume that, in the subsequent iterations, the slot assignments that satisfy the users' requests remain fixed, whereas different parameter values are set to find a different slot assignment for the remaining requests.

The current experiments are performed on the data set containing all requests at the European Level 2 and Level 3 airports on the busiest day of 2013. We are aware that considering one day for the slot allocation is limiting, as slot allocation process gives priority to the assignment of slot series. However, the slot allocation problem at European level for the entire season could be heuristically solved by iteratively calling SOSTA. First, the slot allocation problems would be solved for the busiest days of a season. Then, the solutions of these first problems would be used to derive constraints on the allocation of the slot series during the remaining days of the season.

Some other questions regarding the simultaneous slot assignment are often raised by the airspace users. Most often, the questions regard the schedule displacement. How is minimisation performed, are the airspace users' indications taken into account? In the paper, SOSTA is presented with the same value of maximum displacement for all airspace users and requests. It would be trivial to introduce the displacement interval for each request, thus taking into account the airspace users' needs, if they choose to share this information (which they already do in some cases). Another question that is raised is how are the grandfather rights taken into account, and can they be displaced or not? Here, SOSTA also gives airspace users some flexibility. If they choose, the slot either is allocated as requested or is rejected. Otherwise, it can be moved using the constraints provided by user. In this paper, this feature is not demonstrated, as we did not have access to the initial slot requests containing the flexibility data.

Next, let us turn to the confidentiality issues the use of SOSTA might raise. It is true that today, airspace users do not submit coupled requests, as they submit requests for one airport at the time. However, the initial slot requests can and often do contain the information on the turnaround times, as the slot request message contains the flight numbers of arriving and departing flights, the times of arrival and departure, even the routes the flight is supposed to take arriving and departing from the given airport. SOSTA does not require the change of this process, as in order to run it, just the collection of the requests sent to different airports is needed. Of course this can always raise issues of trust, i.e., would airspace users trust the entity that runs SOSTA to handle all this information in a confidential way? However, we see that there is a trend of more and more coordinators using centralised services for providing access to their schedule data, like [www.online-coordination.com](http://www.online-coordination.com). Thus, we believe that this particular confidentiality matter would not be a problem for airspace users.

Further, let us discuss the policy implications of using SOSTA, especially from the point of view of users trying to trick the system (i.e., gaming). The current slot allocation process is criticised for lack of fairness, barriers to entry (due to grandfather rights), and inefficient use of airport capacity. The whole process relies heavily on grandfather rights. The airspace users are putting a lot of effort to retain the grandfather right slots, as those slots provide the stability in the market and predictability when the investments are needed (e.g., new aircraft). In the current system, the incentives for truthfulness lie in the relationship between airspace user and coordinator. The airspace user can try to game the system by asking for a slot it does not need or does not intend to use. In case it does not use the slot for more than 80% of the time, however, then it loses the right on this slot, which returns to the pool of free slots. With SOSTA, we do not intend to change these rules, as these are subject to regulations or IATA guidelines (where local regulations still do not exist), in almost the same way all around the world. As acceptance of IATA guidelines was built over the decades, we do not believe the radical changes to the guidelines/regulations likely. Hence, SOSTA is designed to simultaneously assign slots, within the current regulation framework. The good computational performance of SOSTA suggests that it may be used in a scenario analysis aiming to assess the possible changes of the current slot allocation process, as shown by the impact assessment of the removal of grandfather rights and of the imposition of fairness considerations. In complex network analysis, as the one proposed by [Adler et al. \(2014\)](#), we think that the advantages of using SOSTA for analysing different scenarios can be expected to be even more relevant.

## Acknowledgments

Contributions from Tatjana Bolić and Lorenzo Castelli to this work are part of the ACCESS project and as such are co-financed by EUROCONTROL acting on behalf of the SESAR Joint Undertaking (SJU) and the European Union as part of the SESAR Exploratory Research programme. Opinions expressed in this work reflect the authors views only. EUROCONTROL and/or the SJU shall not be considered liable for them or for any use that may be made of the information contained herein.

## References

- Adler, N., 2001. Competition in a deregulated air transportation market. *Eur. J. Oper. Res.* 129 (2), 337–345.
- Adler, N., 2005. Hub-spoke network choice under competition with an application to western Europe. *Transp. Sci.* 39 (1), 58–72.
- Adler, N., Fu, X., Oum, T.H., Yu, C., 2014. Air transport liberalization and airport slot allocation: the case of the Northeast Asian transport market. *Transp. Res. Part A: Policy Pract.* 62, 3–19.
- Airbus Aircraft Characteristics Airport and Maintenance Planning, 2016. <<http://www.airbus.com/support/maintenance-engineering/technical-data/aircraft-characteristics/>> (accessed November 2016).
- Ball, M.O., Ausubel, L.M., Berardino, F., Cramton, P., Donohue, G., Hansen, M., Hoffman, K., 2007. Market-based alternatives for managing congestion at New York's La Guardia Airport. In: *Proceedings of the AirNeth Annual Conference*.
- Basso, L., Zhang, A., 2010. Pricing vs. slot policies when airport profits matter. *Transp. Res. Part B – Methodol.* 44, 381–391.
- Bertsimas, D., Lulli, G., Odoni, A.R., 2011. An integer optimization approach to large-scale air traffic flow management. *Oper. Res.* 54 (1), 406–422.
- Boeing Airplane Characteristics for Airport Planning, 2016. <[http://www.boeing.com/commercial/airports/plan\\_manuals.page](http://www.boeing.com/commercial/airports/plan_manuals.page)> (accessed November 2016).
- Brueckner, J.K., 2009. Price vs. quantity-based approaches to airport congestion management. *J. Public Econ.* 93 (5/6), 681–690.
- Castelli, L., Pellegrini, P., Pesenti, R., 2012. Airport slot allocation in Europe: economic efficiency and fairness. *Int. J. Revenue Manage.* 6 (1/2), 28–44.
- Cook, A., Tanner, G., 2011. European airline delay cost reference values. In: *Final Report Prepared for the Performance Review Unit, EUROCONTROL, Brussels, University of Westminster, London, UK*.
- Coopers and Lybrand. The Application and Possible Modification of Council Regulation 95/93 on Common Rules for the Allocation of Slots at Community Airports. Report for the European Commission, 1995.
- Corolli, L., Lulli, G., Ntaimo, L., 2014. The time slot allocation problem under uncertain capacity. *Transp. Res. Part C – Emerg. Technol.* 46, 16–29.
- Czerny, A., 2010. Airport congestion management under uncertainty. *Transp. Res. Part B – Methodol.* 44, 371–380.
- Czerny, A., Zhang, A., 2011. Airport congestion pricing and passenger types. *Transp. Res. Part B – Methodol.* 45, 595–604.
- EUROCONTROL. PRR 2014, 2015. Performance Review Report – An Assessment of Air Traffic Management in Europe during the Calendar Year 2014. Brussels, Belgium.
- EUROCONTROL and FAA, 2012. 2010 U.S./Europe Comparison of ATM-Related Operational Performance. Brussels, Belgium, and Washington, DC.



- Fukui, H., 2010. An empirical analysis of airport slot trading in the U.S. *Transp. Res. Part B – Methodol.* 44, 330–357.
- Fukui, H., 2014. Effect of slot trading on route-level competition: evidence from experience in the UK. *Transp. Res. Part A – Policy Pract.* 69, 124–141.
- Gillen, D., Hinsch, H., 2001. Measuring the economic impact of liberalization of international aviation on Hamburg airport. *J. Air Transp. Manage.* 7 (1), 25–34.
- IATA, 2015. *Worldwide Slot Guidelines - 7*. Technical Report 7, International Air Transport Association - Joint Slot Advisory Group.
- Jacquillat, A., Odoni, A.R., 2015. An integrated scheduling and operations approach to airport congestion mitigation. *Oper. Res.* 63 (6), 150–165.
- Kleit, A.N., Kobayashi, B.H., 1996. Market failure or market efficiency? Evidence on airport slot usage. *Res. Transp. Econ.* 4, 1–32.
- Koesters, D., 2007. Study on the Usage of Declared Capacity at Major German Airports. Study in cooperation with the Eurocontrol Performance Review Unit.
- Li, L., Donahue, G., Chun-Hung, C., 2007. Auction based slot allocation for traffic demand management at hartsfield atlanta international airport: a case study. *Transp. Res. Board: J. Transp. Res. Board* 1888, 50–58.
- Li, Z.-H., Lam, W.H.K., Wong, S.C., Fu, X., 2010. Optimal route allocation in a liberalizing airline market. *Transp. Res. Part B: Methodol.* 44 (7), 886–902.
- Mott MacDonald, 2006. Study on the Impact of the Introduction of Secondary Trading at Community Airports. Report for the European Commission, Mott MacDonald.
- Madas, M.A., Zografos, K.G., 2013. Implementation of airport demand management strategies: a European perspective. In: Zografos, K.G., Andreatta, G., Odoni, A.R. (Eds.), *Modelling and Managing Airport Performance*. Wiley, pp. 233–258.
- Maldom, D., 2003. Auctioning capacity at airports. *Utilities Policy* 11 (1), 47–51.
- NERA, 2004. Study to Assess the Effects of Different Slot Allocation Schemes. Report for the European Commission, DG TREN, National Economic Research Associates.
- Pellegrini, P., Castelli, L., Pesenti, R., 2012a. Metaheuristic algorithms for the simultaneous slot allocation problem. *IET Intel. Transport Syst.* 6 (4), 453–462.
- Pellegrini, P., Castelli, L., Pesenti, R., 2012b. Secondary trading of airport slots as a combinatorial exchange. *Transp. Res. Part E – Logistics Transp. Rev.* 48, 1009–1022.
- Perret, B., 2015. China will Let Markets Play Role in Airline Slot Allocation. *Aviation Week & Space Technology*. December.
- Pita, J., Adler, N., Antunes, A., 2014. Socially-oriented flight scheduling and fleet assignment model with an application to Norway. *Transp. Res. Part B: Methodol.* 61, 17–32.
- Pyrgiotis, N., Odoni, A.R., 2014. On the impact of scheduling limits: a case study at Newark Liberty International Airport. *Transp. Sci.* 50 (1), 150–165.
- Rassenti, S., Smith, V., Bulfin, R., 1982. A combinatorial auction mechanism for airport time slot allocation. *Bell J. Econ.* 13 (2), 402–417.
- SDG, 2011. Impact Assessment of Revisions to Regulation 93/95. Report for the European Commission, DG MOVE, Steer Davies Gleave.
- Sheng, D., Li, Z.-C., Xiao, Y.-B., Fu, X., 2015. Slot auction in an airport network with demand uncertainty. *Transp. Res. Part E – Logistics Transp. Rev.* 82, 79–100.
- Van Landeghem, H., Beuselinck, A., 2002. Reducing passenger boarding time in airplanes: a simulation based approach. *Eur. J. Oper. Res.* 142 (2), 294–308.
- Verhoef, E.T., 2010. Congestion pricing, slot sales and slot trading in aviation. *Transp. Res. Part B – Methodol.* 44 (3), 320–329.
- Zografos, K.G., Salouras, Y., Madas, M.A., 2012. Dealing with the efficient allocation of scarce resources at congested airports. *Transp. Res. Part C – Emerg. Technol.* 21 (1), 244–256.
- Zografos, K.G., Madas, M.A., Androutsopoulos, K.N., 2016. Increasing airport capacity utilisation through optimum slot scheduling: review of current developments and identification of future needs. *J. Sched.* <http://dx.doi.org/10.1007/s10951-016-0496-7>.