OR Spectrum (2019) 41:615–639
https://doi.org/10.1007/s00291-018-0525-3

REGULAR ARTICLE

The recoverable robust stand allocation problem: a GRU airport case study

Bert Dijk[1] · Bruno F. Santos[1] · Joao P. Pita[2]

Received: 16 September 2016 / Accepted: 26 June 2018 / Published online: 19 July 2018

Abstract
This paper presents an innovative approach to the tactical planning of aircraft remote and contact-stands allocation at airports. We use the concept of recoverable robustness to obtain a recoverable robust solution to the stand allocation problem, a solution that can be recovered by limited means for the included scenarios. Four objective functions are discussed and tested to assess the efficiency of a stand allocation plan. Namely, the minimization of passengers' walking distance, the minimization of tows, the maximization of the number of passengers allocated to contact-stands, and the maximization of the potential airport commercial revenue. The inclusion of an airport commercial revenue metric in the stand allocation optimization model and the comparison of its performance to the pre-mentioned other objectives is another novelty of this work. The research was developed in collaboration with the Guarulhos International Airport of São Paulo for which the recoverable robust approach was tested for 6 days of operations at the airport. We demonstrate that the solutions obtained with the proposed approach outperform the solutions of a traditional robust approach. In addition, a discussion of the trade-off between the different objectives is provided.

Keywords Airport · Stand allocation · Recoverable robustness · Commercial revenue

✉ Bruno F. Santos
b.f.santos@tudelft.nl

Bert Dijk
dijk.bert@gmail.com

Joao P. Pita
joao.pita@gru.com.br

[1] Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

[2] GRU Airport, Rod. Helio Smidt s/n, Guarulhos, Brazil

1 Introduction

In airport operations, the handling of aircraft movements requires the decision to assign aircraft to specific stands. The stands can be contact-stands, remote boarding stands and parking-only stands. The allocation of aircraft to the stands is usually done 1 day before operations, providing airport gate controllers and airlines with a plan for the coming day. We call the problem of assigning aircraft to stands 1 day in advance the tactical stand allocation problem (TSAP). The literature may refer to the TSAP as the gate assignment problem, in case a 1-to-1 mapping of stands and gates exist.

The assignment of aircraft to stands is a complex problem, complicated by factors as stands classification, multi-aircraft ramp stands (MARS) constraints, immigration rules or scarce resources, like gates or towing vehicles. Another complication to the gate assignment problem is the uncertainty of the arrival and departure times of each aircraft. Airport controllers have to constantly deal with flight schedule changes. Aircraft can arrive early or tardy, and the airport has to provide resources regardless of these deviations. An ideal assignment plan is one that is capable of absorbing these deviations in operations, without significantly compromising efficiency. Thus, to ensure the efficient gate usage, a tactical assignment plan should be flexible to flight times variations, limiting the amount of disruptions caused and improving overall airline and passenger satisfaction.

Due to the flight schedule changes, airports may prefer a robust tactical stand assignment plan—i.e., a plan that remains feasible during operations—relative to a very efficient plan that can be easily jeopardized with small variations in flight times. The robust planning topic has already been widely explored in the academic literature (e.g., the reader is referred to Bertsimas and Sim (2014) and Ben-Tal et al. (2009) for general discussions on robust optimization; and to Dorndorf et al. (2012) and Kumar and Bierlaire (2014) for robust approaches applied to the gate assignment problem). However, traditional approaches usually focus on satisfying the worst-case scenario. This is an over-conservative approach for a planning problem, compromising the best usage of the resources available and usually leading to poor performances during operations.

This paper addresses the TSAP adopting the *recoverable robustness* concept to design flexible allocation plans. This is a rather new concept, introduced by Liebchen et al. (2009), that to our knowledge was never applied to the stand allocation problem. The recoverable robustness concept explores the idea that a planning solution is recovery robust if it can be recovered by limited means in several realization scenarios. The goal is to jointly optimize the stand allocation plan and the strategy for recovery, obtaining a less conservative yet robust solution to the planning problem.

This research was developed in collaboration with Guarulhos International Airport of São Paulo (GRU). The airport is investigating the development of more robust daily stand plans, without compromising the efficiency of these plans. As part of this challenge, the airport wants to explore different objective functions for their stand allocation. In fact, the optimization of a stand allocation plan may depend on the airport strategy and, therefore, can be different from airport to airport. Even within an airport the objectives can differ significantly—for instance, airlines may exclusively used of a set of the stands, managing their gate plan and adopting different objectives.

The airport may decide to focus on passenger-oriented or airport-oriented objectives for their tactical assignment planning (Dorndorf et al. 2007). Therefore, in this paper we considered two passenger-oriented objectives: minimization of passengers walking distance and maximization of the percentage of passengers allocated to a contact-stand, and two airport-oriented objectives: minimization of aircraft tows and maximization of potential airport commercial revenues.

Six days of operations at GRU are used as a case study to illustrate the added value of the recoverable robustness concept. We use the specific case of Terminal 3. This is an international terminal accommodating only long-haul and medium-haul (Latin-American) flights. To illustrate the benefits of the proposed approach, the solutions obtained with the recoverable robust stand allocation model will be compared to the results obtained when using a traditional strict robust stand allocation model, in which recovery actions are not permitted. Furthermore, the case study will be also used to discuss the four objective metrics proposed.

The main innovations of this paper are twofold. On one hand, this is the first application of the recoverable robustness concept to the TSAP. We believe this is a relevant approach for the TSAP because it provides a less conservative concept to the robust stand allocation problem, accommodating multiple scenarios of flight times without compromising overall efficiency of the solutions. On the other hand, this is the first time that maximization of airport commercial revenues is explicitly included as an objective to the TSAP. In an era of airport privatizations, this newly proposed objective has gained relevance. This is particularly the case in Brazil, and at GRU airport.

In this paper, Sect. 2 describes the TSAP, presenting an overview of the literature and describing the mathematical model. Section 3 provides an overview of the recoverable robust stand allocation framework. The GRU case study and the results are discussed in Sect. 4. Finally, conclusions and future research directions are presented in Sect. 5.

2 The tactical stand allocation problem

The stand allocation problem (SAP, also referred to as gate assignment problem in the literature) is commonly formulated as a (mixed) integer or binary linear program, due to the nature of the constraints (Mangoubi and Mathaisel 1985; Dorndorf et al. 2007). Nonetheless, this is not the only possible formulation and other authors have suggested, for instance, multi-commodity network models (Maharjan and Matis 2012) or dynamic programming models (Jaehn 2010) to address the SAP. Initial research on the topic started in the late 1970s and early 1980s with simplistic mathematical models and with basic operational constraints (e.g., Braaksma 1977; Babić et al. 1984; Mangoubi and Mathaisel 1985). Thereafter, research has continued to develop more advanced and more realistic models. For instance, authors started considering potential towing of a long-stay aircraft (e.g., Kumar and Bierlaire 2014; Guépet et al. 2015), stand compatibility (i.e., aircraft can only be serviced at suitable stands) (e.g., Vanderstraeten and Bergeron 1988; Diepen et al. 2012) and MARS constraints, covering physical restrictions of overlapping stands (e.g., Diepen et al. (2012), Guépet et al. (2015), also described as adjacency). A variation of a MARS constraint is the last-in first-out constraint proposed by Kumar and Bierlaire (2014) for airports where two stands are

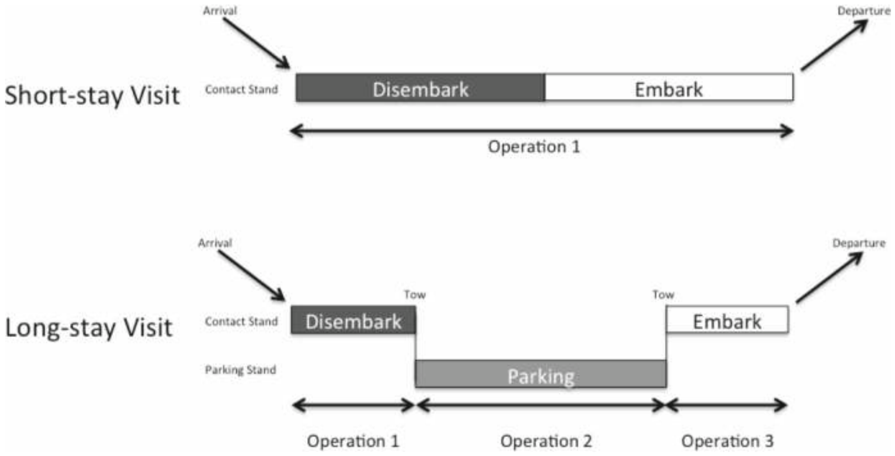


Fig. 1 Example of a short-stay aircraft visit and a long-stay aircraft visit

behind each other. For a comprehensive analysis of the stand allocation literature, please refer to Dorndorf et al. (2007) and Bouras et al. (2014), two recent reviews on the topic.

Our research is focused on the stand allocation on the day before operations, referred by us as the TSAP.

2.1 Input data

The input data for the TSAP can be divided into flight schedule (aircraft visits) and airport layout (stands)-related data.

The flight schedule dataset is composed of the list of aircraft visits in the flight schedule. For the sake of modeling towing operations, we converted aircraft visits into operations. First, we divided the visits in short-stay and long-stay visits. The long-term visits were converted into a set of three operations per flight: arrival, parking and departure, while short-stay visits were converted into a single operation combining arrival and departure (Fig. 1). This approach enables the towing of long-term visits to remote boarding stands or parking stands, reducing the contact-stand occupancy time and increasing the availability of these stands. To keep track of the operational sequence that belongs to the same aircraft visit, we utilize the formulation of the successor as found in Guépet et al. (2015). The successor is found based on the turnaround schedule provided by the airport. Potential conflicts between stand allocations are dealt with by defining a set of overlapping operations, according to the flight times.

The airport layout data consist of the set of available stands at the airport and the list of compatible stands for the aircraft type used in each airport operation. An additional set is defined to handle the MARS that can be used either by a single large aircraft or by two small aircraft. In Fig. 2, we give an example of this situation. For this case, Stand 1 will have two overlapping stands, Stand 1L and Stand 1R. Stand 1 cannot be used if any of these two latter stands are used.

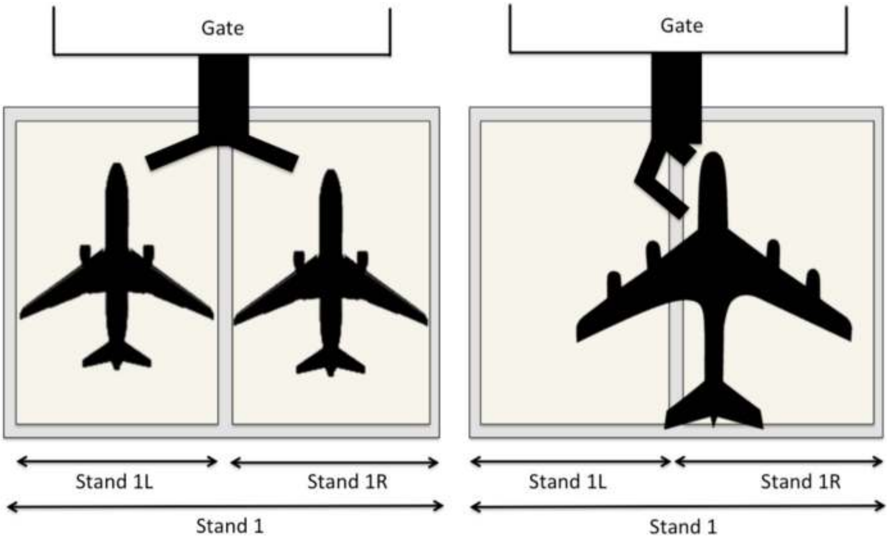


Fig. 2 Example of overlapping MARS: Stands 1 can be used either by one wide-body aircraft (right) or by two narrow-body aircraft, occupying Stands 1L and 1R (left)

The sets used in the TSAP formulation are summarized in the following list:

- **V**: set of aircraft visits as found in the flight schedule
- **O**: set of operations (i.e., arrival, parking and departure) for all visits from set **V**
- $\mathbf{U_i}$: set of successors for operations i: $\mathbf{U_i}$ is the follow-up operation from the split of long-term aircraft visits
- $\mathbf{OV_i}$: set of overlapping operations, which is a subset of **O** with all operations that overlap with operation i—following the strengthened formulation as defined by Guépet et al. (2015)
- **S**: set of stands available at the airport, with location and classifications
- $\mathbf{S_i}$: set of compatible stands, which is a subset of **S** with the stands that can be used by the aircraft type of operation i
- $\mathbf{Q_j}$: set of MARS overlapping with stand j.

In order to define these sets, several parameters need to be defined by the user:

- **Minimum time to tow**: minimum stay time of a visit specified by the airport to allow a towing operation, used to divide aircraft visits in short-stay and long-stay visits.
- **Minimum embark and disembark times**: time required to (dis)embark the passengers, used to determine minimum length of the arrival and departure operations for long-stay visits.
- **Minimum buffer time**: time between two consecutive operations to allow for short stand servicing and to handle small schedule deviations, used to determine overlapping operations.

2.2 Decision variables

For the TSAP, the following decision variables are defined:

- $x_{i,j}$ binary variable equal to 1 if operation i is allocated to stand j, and equal to 0 otherwise
- y_i binary variable equal to 1 if operation i is towed, and equal to 0 otherwise.

2.3 TSAP mathematical formulation

The TSAP is formulated as a 0, 1 integer linear program, influenced by the strengthened formulation as presented by Guépet et al. (2015). The goal is to minimize a given objective function, guaranteeing that all airport operations are allocated either to a contact-stand, to a remote boarding stand or to a parking stand, and that no operations are overlapping. The mathematical formulation of our TSAP is defined as:

$$\underset{x,y}{\text{Minimize:}}\ f(x_{i,j}; y_i) \tag{1}$$

Subject to:

$$\sum_{j \in S_i} x_{i,j} = 1, \quad \forall\, i \in O, \tag{2}$$

$$\sum_{k \in OV_i} x_{k,j} \leq 1, \quad \forall\, i \in O, v\forall\, j \in S_i, \tag{3}$$

$$x_{i,j} - x_{U_i,j} \leq y_i, \quad \forall\, i \in O, v\forall\, U_i \neq 0, \quad \forall\, j \in S_i, \tag{4}$$

$$x_{i,j} + \sum_{k \in OV_i} x_{k,j'} \leq 1, \quad \forall\, i \in O, \quad \forall\, j \in S, \quad \forall\, j' \in Q_j, \tag{5}$$

$$x_{i,j} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \tag{6}$$

The objective function in Eq. 1 describes the minimization of an objective metric that should be written as a function of the allocation decisions (to be discussed next). Equation 2 covers the allocation constraints: each operation needs to be allocated to one and only one stand. The stand capacity constraints are covered by Eq. 3: a stand can only be used by a single overlapping operation. Constraints 4 guarantee that the towing decision variable is activated for long-stay visits when sequential operations of the visit are allocated to a different stand. Equation 5 covers constraints for the MARS: if operation i is allocated to stand j, operation k cannot be allocated to stand j' at the same time. Finally, constraints 6 ensure the decision variables to be binary.

2.4 Objective function

Several objectives have been used to address the TSAP in the literature. Most common is the minimization of walking distance (used, e.g., in Braaksma 1977; Babić et al. 1984; Mangoubi and Mathaisel 1985; Bihr 1990; Cheng et al. 2012; Drexl and Nikulin

2008), but other metrics have been proposed. For instance, the maximization of the utilization of contact-stands (Guépet et al. 2015), the minimization of non-allocated visits (Vanderstraeten and Bergeron 1988; Drexl and Nikulin 2008; Kumar and Bierlaire 2014), the maximization of allocation preferences (Dorndorf et al. 2007; Jaehn 2010; Kumar and Bierlaire 2014) or the minimization of towing operations (Dorndorf et al. 2007; Kumar and Bierlaire 2014; Guépet et al. 2015). A recent trend in the stand allocation literature is the utilization of multi-objective approaches (e.g., Guépet et al. 2015; Kumar and Bierlaire 2014; Dorndorf et al. 2012). However, the selection of appropriate weights to level the objectives can be complex. This discussion is considered to be outside the scope of this research. Instead, we rather want to contribute to the discussion of the impact of choosing different objectives and to analyze the influence of each objective in the metrics of the other objectives.

According to Dorndorf et al. (2007), the optimization objectives used for the TSAP can be divided into passenger-oriented objectives and airport-oriented objectives. In this paper, we also use this division and consider four different objective functions: minimization of passenger walking distance and maximization of percentage of passengers allocated to a contact-stand as passenger-oriented objectives, and minimization of number of tows and maximization of potential commercial revenue as airport-oriented objectives. These objectives were defined for this work following discussions with GRU airport.

In this section, we explain how we formulated each of the four objectives proposed.

2.4.1 Passenger-oriented objectives

Minimization of passengers' walking distance: the most common objective for the TSAP. It focuses on increasing passenger satisfaction at the airport. We formulated this objective metric according to the following expression:

$$\min f(x_{i,j}; y_i) = \sum_{i \in O} \sum_{j \in S} \mathrm{WD}_j * \mathrm{Pax}_i * x_{i,j} \tag{7}$$

in which WD_j represents the walking distance from the entrance of the pier to stand j and Pax_i is the number of passengers associated with operation i (note: for parking operations, the number of passengers is assumed to be zero). In our work, due to the layout of the GRU airport and the fact that we will focus on an international terminal, transfer passengers are not considered separately in this objective metric. It is assumed that each passenger starts or ends the journey at the terminal in the security control point at the entrance of the pier. For TSAP formulation examples considering connecting passengers, please refer to Drexl and Nikulin (2008) and Cheng et al. (2012).

Maximization of passengers allocated to contact-stands: the goal is to allocate as many passengers as possible to a contact-stand, reducing required remote boarding of passengers. The objective is only relevant for airports with remote boarding possibilities. Less remote boarding will likely increase passenger satisfaction due to easier access to the aircraft. The objective was formulated as:

$$\min f(x_{i,j}; y_i) = 1 - \sum_{i \in O} \sum_{j \in S_c} \mathrm{Pax}_i * x_{i,j} / \sum_{i \in O} \mathrm{Pax}_i \tag{8}$$

in which S_c represents the set of contact-stands, a subset of $\mathbf{S}$.

2.4.2 Airport-oriented objectives

Minimization of tows: the goal is to reduce the usage of towing tractors at the airport. The same is to say that we try to maximize the number of long-stay visits with operations allocated to the same stand. Since the decision variable y_i indicates a required tow, the objective metric was simply described as:

$$\min f(x_{i,j}; y_i) = \sum_{i \in O} y_i \tag{9}$$

Maximization of potential commercial revenue: this objective addresses the fact that non-aeronautical revenues are becoming increasingly important for airports due to privatization and competition between hubs (Graham 2009). Commercial experts at GRU airport expressed the interest in exploiting the fact that passengers from some flights are more consumption oriented than others and that the allocation of flights to stands may influence the shopping behavior of passengers (Geuens et al. 2004; Lin and Chen 2013). The objective metric was formulated as:

$$\min f(x_{i,j}; y_i) = \sum_{i \in O} \sum_{j \in S} -A_{i,j} * \mathrm{Pax}_i * x_{i,j} \tag{10}$$

where $A_{i,j}$ represents the potential revenue generated by the combination of flight operation i and stand j. For parking operations, the revenue was considered to be equal to zero. This potential revenue needs to be estimated based on historical non-aeronautical revenue data of specific flights per store. The matrix is expressed in dollars per passenger of a given flight when allocated to a specific gate at the terminal. The assumption is that high values of $A_{i,j}$ reflect a high potential of generating revenues per passenger from a flight i and that the revenue generated by allocating a flight i to a stand j is depending on the set of existing stores between the stand and the entrance of the terminal (thus, on the location of the stand).

This metric was developed according to the layout of GRU Terminal 3. However, it is important to state that this metric can easily be extended to other terminal layouts, as long as the sales per store and per flight can be estimated. A major problem could be the lack of detailed or sufficient data, which will limit the validity of such an approach. Nevertheless, it is common for airport authorities to use new technologies nowadays, such as Wifi and Bluetooth sensors, to collect massive amounts of data on passengers flows, consumer behavior and airport operations. Thus, we believe that this objective metric will become popular in future works when addressing the planning of operations at airports.

3 Recoverable robust stand allocation model

Solving the TSAP model, as explained before, for the flight schedule of a day results in a stand allocation plan for that day. The plan is, however, not necessarily flexible enough to accommodate changes during the day of operations. Flight times are subject to uncertainty and disruptions are common, jeopardizing the benefits of having an optimized stand allocation plan. Therefore, we aim to effectively handle flight time uncertainty by directly addressing plan robustness in our stand allocation model.

In the literature, robustness has been usually included in the SAP by introducing buffer time constraints [e.g., in Yan and Huo (2001) and Yan et al. (2002)] or by considering robustness objectives [e.g., in Bolat 2000, Dorndorf et al. (2008), Dorndorf et al. (2012), Diepen et al. (2012) and Kumar and Bierlaire (2014)]. Flight-specific buffers were considered by Kumar and Bierlaire (2014), in which the 95th percentile of historical actual flight times were considered to define the buffers. Other methodologies to include robustness in the SAP involve, for instance, the stochastic optimization of stand allocation for a specified number of scenarios (Yan and Tang 2007; Seker and Noyan 2012) or the reduction in the probability for gate conflicts (Lim and Wang 2005). More recent works on SAP can be find in Dorndorf et al. (2017) and Xu et al. (2017). The first authors proposed a graph theoretical approach to solve the gate allocation problem with deterministic and stochastic objectives, while the later work presented an *alpha*-reliable approach to the same problem.

In this paper, we propose a new concept to address robustness in the TSAP: the recoverable robustness concept. Recoverable robustness is based on the foundations of robust optimization and stochastic programming, as explained by Liebchen et al. (2009). The aim of the recoverable robustness in the stand allocation context is to obtain a recoverable robust solution that is feasible for the standard TSAP and can be recovered (if necessary) by limited means for multiple flight times scenarios. It provides the airport controllers with an efficient solution that, with limited required actions, will likely remain feasible during normal daily operations. Furthermore, information on the recovery for the tested scenarios and critical flights can be provided to the airport controllers to support decisions during operations.

The recoverable robustness concept consists of three main steps: an original optimization problem (Step O), imperfection of the information (via scenarios, Step S) and limited allowed recovery actions (Step R). The recovery actions are considered to fix the original optimized plans under the realization of different scenarios. Recoverable robustness has already been successfully applied to air transportation problems, including the timetabling problem (D'Angelo et al. 2011) and the tail assignment problem (Froyland et al. 2013). Nonetheless, to our knowledge, this is the first application of the recoverable robustness concept in the aircraft stand allocation context.

To implement the recoverable robustness concept, we developed a recoverable robust tactical stand allocation model (RRTSAP). The model includes the pre-described TSAP (Step O), a set of generated scenarios based on historical airport data (Step S) and a model extension to include the recovery actions (Step R). In this section, we will describe Step S and the model extension for Step R.

3.1 Step S: scenario generation

Multiple techniques can be used to generate test scenarios for our recoverable robustness approach—please refer to Löhndorf (2016) for a recent discussion on scenario generation methods and on the error that arises from using a small set of scenarios. The typical approach to generate scenarios in the SAP literature assumes fixed distributed delay profiles with no relations between aircraft visits [e.g., a triangular distribution (Seker and Noyan 2012) or a normal distribution (Genç et al. 2012)].

The proposed method to model the relation between aircraft visits is based on the Monte Carlo sampling scheme, using historical data to capture delay relations between flights. It is divided into four steps, which will be sequentially described in this subsection.

The *first step* is to determine the flight times approximate distribution function for every flight number. Several continuous distributions (including Gaussian, exponential, gamma and Weibull) are utilized and tested, in order to find the best fit to the available data.

The *second step* in the scenario generation method is to incorporate relations between the flight time deviations of different aircraft visits. This step focuses only on arrival times and considers how a sequence of delayed flights or early arriving flights can strongly influence the feasibility of the stand plan. Departure times are assumed to be uncorrelated. Ideally, one would compute flight delays correlations at the flight number level. However, in most cases it is hard to prove flight time causality or correlations at such detail level (as it proved to be the case in our case study). Therefore, we suggest to analyze flight time deviations relations in terms of arrival time deviation sign (i.e., positive if arrived early or negative if arrived late) for flights arriving from the same origin region. This way, we try to capture delay effects at the origin airports. In addition, we also consider the deviation sign relations between pairs of flights arriving within a time window of 1 h, according to the schedule. With this, we aim to capture consecutive delays caused by adverse weather conditions or air traffic management congestion near the airport of study.

Both deviation sign relations are expressed in probabilities. The probabilities are then used to estimate the probability of a flight time deviation having a specific sign. This is done by following three assumptions:

1. the influence of flight time deviations is only observed if there were at least three related flights arriving earlier in the previous hour;
2. flight time deviation signs are primarily influenced by delays observed in previous flights from the same region only, which were scheduled to arrive at the airport of study in the previous hour;
3. if no influence was observed within fights of the same region, then the flight can be influenced by any other flight scheduled to arrive at the airport of study in the hour before.

We require a minimum of three relevant flights to ensure that the arrival time deviation sign is not too dependent on a single previous visit. If for an aircraft visit insufficient comparable aircraft visits are found, an independent deviation is computed based on the historical arrival time distribution.

To exemplify the computation of the signs probabilities, let us consider the case of an inbound flight i scheduled to arrive at the airport of study at a given moment in the day. Before this flight, less than three flights from the same region were scheduled to arrive in the hour before. Therefore, flight i is considered to be independent from previous flights from the same region. There were, however, three flights from other regions that were scheduled to arrive at the airport in the previous hour. These three preceding flights, noted by i^1, i^2 and i^3 in order of arrival, influence the time deviation sign of flight i. For the scenario generation, two types of indicators are considered:

- the historical percentage of days where both flight i and the preceding flights i^k had the same time deviation sign (positive/negative), denoted by $P(S)_{i^k}$;
- and, in the current scenario, if flight i^k was generated as arriving early or late.

Assuming that the flight time deviations of the three preceding flights were already computed and in the current scenario these flights are arriving *early*, *early* and *late*, respectively, the probability of flight i arriving late can be computed by:

$$P(\text{Negative})_i = (1 - P(S)_{i^1}) * (1 - P(S)_{i^2}) * P(S)_{i^3} \tag{11}$$

In the same way, the probability of flight i arriving early can be computed by:

$$P(\text{Positive})_i = P(S)_{i^1} * P(S)_{i^2} * (1 - P(S)_{i^3}) = 1 - P(\text{Negative})_i \tag{12}$$

This formulation can be extended to a generic case of flight i being influenced by $|N_l|$ preceding flights arriving late and $|N_e|$ preceding flights arriving early:

$$P(\text{Negative})_i = \prod_{k \in N_l} P(S)_{i^k} * \prod_{p \in N_e} (1 - P(S)_{i^p}) \tag{13}$$

The value of $P(\text{Positive})_i$ can be then estimated by subtracting $P(\text{Negative})_i$ from the unit.

The *third step* concerns the computation of a random number, between 0 and 1, which determines whether the aircraft visit arrives early or late, by comparing the random number to the calculated probabilities. In the case the flight was considered to be independent from previous flights, the third step is skipped and the complete historical arrival time distribution is used.

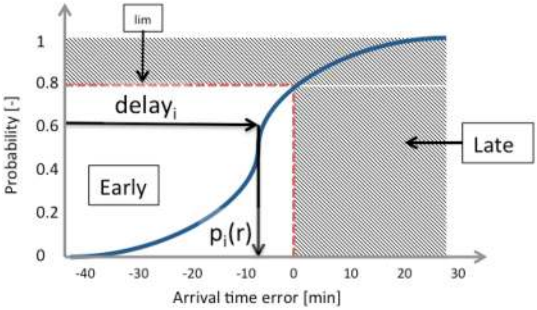
The *fourth step* is to determine the quantity of the arrival time deviation for each flight, based on the computed historical arrival time deviation distributions and deviation sign. As previously stated, the historical arrival time distributions are based upon actual flight data. For each dependent flight i, the arrival time deviation (expressed by $delay_i$) is estimated by verifying the following equality:

$$P(X \leq \text{delay}_i) = p_i(r) \tag{14}$$

where $p_i(r)$ is generated as follows:

$$p_i(r) = \begin{cases} \lim * r, & \text{if flight } i \text{ is late} \\ (1 - \lim) * r + \lim, & \text{otherwise} \end{cases} \tag{15}$$

Fig. 3 Determination of arrival time deviation for an early dependent visit



in which *r* is a [0, 1] random number and lim is the probability of having an arrival time deviation less or equal to zero. The visualization of the determination of the arrival time deviation quantity for a dependent flight *i*, given an early arrival, is provided in Fig. 3. In this example, only the part of 0–0.8 probability is utilized to determine the quantity of the arrival time deviation—due to the determined sign "early" for this specific flight.

For independent flights, $p_i(r)$ is considered over the full arrival time distribution. That is, no deviation sign determination is utilized to determine the quantity of arrival time deviation and the complete distribution function (i.e., from 0 to 1) is used.

To conclude the generation of the scenarios, the corresponding time deviation is added to the scheduled flight arrival or departure time. The deviations per scenario are then coupled to a visit of an aircraft, which is consequently translated into operations, as explained in Sect. 2.1.

3.2 Step R: RRTSAP

The recovery actions in a recoverable robust approach are used to try to solve a conflict by limited means and to recover the feasibility of a plan for a given scenario. We considered three possible recovery actions:

- **Action 1—Waiting:** to let an aircraft wait until its allocated stand is free—this action should be limited by a *maximum waiting time* defined by the airport.
- **Action 2—Re-allocate to a free stand:** to allocate an aircraft to an alternative compatible stand, in the case the conflict time is larger than the maximum waiting time and if the alternative stand is free.
- **Action 3—Tow a long-stay parking operation:** to tow an aircraft to a remote (parking-only) stand, in the case conflict time is larger than the maximum waiting time, there is no alternative stand free and if the aircraft to be towed is involved in a long-stay visit.

Ideally, the airport would solve conflicts only using Action 1. If not possible, Action 2 is considered. Action 3 is only considered by the airport to solve a conflict in the case the two previous actions are not sufficient.

These recovery actions are considered in our approach by extending the TSAP to a RRTSAP. To do so, we define the following additional (sub-)sets:

- **W**: set of scenarios denoted by w
- $\mathbf{OV_{i,w}}$: set of overlapping operations, which is a subset of **O**, with all operations that overlap with operation i in scenario w.

The last set disregards all flights that have a time overlap of less than a *maximum waiting limit* specified by the airport. This way, we indirectly model **Action 1—Waiting**: all conflicts involving the overlap of two operations by no more than this *maximum waiting limit* are considered to be solved by making the aircraft associated with the latter operation wait. Two new sets of decision variables are also defined for the RRTSAP:

- $\text{add}_{i,j,w}$ binary variable equal to 1 if operation i is added to stand j in scenario w; and equal to 0 otherwise, relative to a scheduled solution $x_{i,j}$;
- $\text{rem}_{i,j,w}$ binary variable equal to 1 if operation i is removed from stand j in scenario w; and equal to 0 otherwise, relative to a scheduled solution $x_{i,j}$;

The mathematical formulation of the RRTSAP is defined as:

$$\underset{x}{\text{Minimize:}}\ f(x_{i,j}) \tag{16}$$

Subject to:

$$(2)-(6)$$

$$\sum_{j\in S_i}(x_{i,j} - \text{rem}_{i,j,w} + \text{add}_{i,j,w}) \geq 1, \quad \forall\, i \in O, \quad \forall\, w \in W, \tag{17}$$

$$\begin{aligned}&\sum_{k\in OV_{i,w}} x_{k,j} - \sum_{k\in OV_{i,w}} \text{rem}_{i,j,w} + \sum_{k\in OV_{i,w}} \text{add}_{i,j,w} \leq 1,\\ &\forall\, i \in O, \forall\, j \in S_i, \forall\, w \in W,\end{aligned} \tag{18}$$

$$\begin{aligned}&x_{i,j} - \text{rem}_{i,j,w} + \text{add}_{i,j,w} + \sum_{k\in OV_{i,w}} x_{k,j'} - \sum_{k\in OV_{i,w}} \text{rem}_{k,j',w}\\ &+ \sum_{k\in OV_{i,w}} \text{add}_{k,j',w} \leq 1, \quad \forall\, i \in O, \forall\, j \in S, \forall\, j' \in Q_j, \forall\, w \in W,\end{aligned} \tag{19}$$

$$\sum_{j\in S} \text{add}_{i,j,w} - \sum_{j\in S} \text{rem}_{i,j,w} = 0, \quad \forall\, i \in O, \forall\, j \in S, \forall\, w \in W, \tag{20}$$

$$x_{i,j} \geq \text{rem}_{i,j,w}, \forall\, i \in O, \forall\, j \in S, \quad \forall\, w \in W, \tag{21}$$

$$\sum_{i\in O}\sum_{j\in S} \text{add}_{i,j,w} \leq \lambda, \quad \forall\, w \in W, \tag{22}$$

$$\text{add}_{i,j,w} \in \{0, 1\}, \quad \text{rem}_{i,j,w} \in \{0, 1\} \tag{23}$$

The model assumes the same objective function and is subject to all constraints (2–6) defined in TSAP. In addition to those, Eqs. 17 and 18 cover the re-allocation of an operation in one of the scenarios. Constraints 19 cover the MARS constraints, now including the add/rem variables. Constraints 20 ensure that if an operation is added to a stand, it is also removed from another. Constraints 21 ensure that the removed

```
Algorithm 1: Add/Rem Algorithm
Result: Add limited add/rem variables to RRTSAP
Solve TSAP for Scenario 0 - solution x0
w=0
while $w \leq len(W)$ do
    Find conflicts > limit in $w$ based on x0 ;
    for each conflict do
        Step 1: ;
        if Add/Rem variables for conflict-causing operations in Scenario $w$ not added then
            Add add/rem variables for both conflict-causing operations in Scenario $w$
        end
        ;
        Step 2: ;
        for $OV_i$ of conflict-causing operation $i$ in Scenario $w$ do
            if Add/Rem variables for $OV_i$ in Scenario $w$ not added then
                if $S_i$ for Operation $OV_i \leq 2$ then
                    add add/rem variables for Operation $OV_i$ and its overlapping operations in
                    Scenario $w$
                end
            end
        end
    end
    ;
    $w = w + 1$
end
```

Fig. 4 Add/rem algorithm pseudo-code

decision variable for an operation aligns with the scheduled allocation (described in $x_{i,j}$). Constraints 22 ensure that the amount of changes in a scenario is limited to a maximum number of re-allocations λ. Finally, constraints 23 ensure the decision variables to be binary. Note that no additional towing variables (y) are included in the new constraints, since it is considered that the number of recovery tows does not have an impact on the objective function or any other constraint of the model.

3.3 Solving the RRTSAP

By transforming the TSAP into a multi-scenario model with the flexibility to recover, the RRTSAP becomes a complex problem to solve. Initial tests with the RRTSAP indicated a long computation time when add/rem variables are added for all operations in all scenarios. In a test case of 34 stands (including 18 MARS) and 60 aircraft visits, we have a problem of the order of 650 thousand columns and 950 thousand rows. It took us 7 h to solve the problem up to a gap of 11.8%. After 10 h, the gap was still of about 9.0%. To satisfy the requirement of a reasonable runtime for the airport (around 2 h), methodologies to limit the runtime were investigated.

To reduce the computation time of the RRTSAP, we propose a column-generation inspired algorithm. The idea is that most of our $\text{add}_{i,j,w}$ and $\text{rem}_{i,j,w}$ variables will be zero in the final solution. Thus, we want to only add decision variables that are relevant to solve our model (i.e., that may assume a value of one in the end). The pseudo-code of the algorithm is described in Fig. 4.

In the algorithm, decision variables are added according to two steps. Step 1 focuses on the operations that, based on the scheduled solution ($x_{i,j}$), would cause conflicts in one of the scenarios. For these operations, *add/rem* variables are added to the RRTSAP. This step allows the re-allocation of operations to free stands (**Action 2—Re-allocate to a free stand**) and the swap of stands between the two conflicting operations.

Step 2 considers that, in some cases, there are not enough free stands to solve a conflict and that a set of overlapping operations may have to swap allocations in

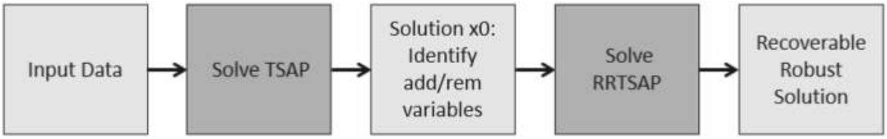

Input Data
Solve TSAP
Solution x0: Identify add/rem variables
Solve RRTSAP
Recoverable Robust Solution

Fig. 5 Solution workflow with adjusted RRTSAP

order to solve a single conflict (**Action 3—Tow a long-stay parking operation**). The step adds new swapping possibilities by adding *add*/*rem* variables for all operations overlapping with the conflict operations that do not have a minimum buffer of free stands available (considered to be two stands). The number of free stands available is analyzed by comparing the compatible stands and the minimum occupancy of these stands looking at all operations that have (limited) time overlap in the scenario under analysis.

This algorithm allows us to solve the RRTSAP in the described test case with 45% less rows and 77% less columns, considerably speeding up the computation of a solution.

3.4 Resulting RRTSAP model

According to the proposal described before, the RRTSAP needs to be solved in a sequential approach (Fig. 5). First the original TSAP is solved. The solution is then used in Algorithm 1 to evaluate which *add*/*rem* variables to add, including the respective constraints that bound these variables. Afterward, the RRTSAP is solved with a limited number of added variables and constraints.

For the sake of completion, we provide the formulation of the RRTSAP with limited *add*/*rem* variables. We define the following additional (sub-)sets:

- $\mathbf{O_w}$: subset of operations set O, for which *add*/*rem* variables are included in the model for scenario w
- $\mathbf{OV'_{i,w}}$: subset of overlapping operations for scenario w for which *add*/*rem* variables were added to the model.

The resulting RRTSAP can then be formulated as follows:

$$\underset{x}{\text{Minimize:}}\ f(x_{i,j}) \tag{24}$$

Subject to:

$$\sum_{j \in S_i} x_{i,j} = 1, \quad \forall\, i \in O, \tag{25}$$

$$\sum_{k \in OV_i} x_{k,j} \leq 1, \quad \forall\, i \in O, \quad \forall\, j \in S_i, \tag{26}$$

$$x_{i,j} - x_{U_i,j} \leq y_i, \quad \forall\, i \in O, \quad \forall\, U_i \neq 0, \quad \forall\, j \in S_i, \tag{27}$$

$$x_{i,j} + \sum_{k \in OV_i} x_{k,j'} \leq 1, \quad \forall\, i \in O, \quad \forall\, j \in S, \quad \forall\, j' \in Q_j, \tag{28}$$

$$\sum_{j \in S_i} (x_{i,j} - \text{rem}_{i,j,w} + \text{add}_{i,j,w}) \geq 1, \quad \forall\, i \in O_w, \quad \forall\, w \in W, \tag{29}$$

$$\begin{aligned} &\sum_{k \in OV_i} x_{k,j} - \sum_{k \in OV'_{i,w}} \text{rem}_{i,j,w} + \sum_{k \in OV'_{i,w}} \text{add}_{i,j,w} \\ &\leq 1, \quad \forall\, i \in O, \quad \forall\, j \in S_i, \quad \forall\, w \in W, \end{aligned} \tag{30}$$

$$\begin{aligned} &x_{i,j} - \text{rem}_{i,j,w} + \text{add}_{i,j,w} + \sum_{k \in OV_{i,w}} x_{k,j'} - \sum_{k \in OV'_{i,w}} \text{rem}_{k,j',w} \\ &+ \sum_{k \in OV'_{i,w}} \text{add}_{k,j',w} \leq 1, \quad \forall\, i \in O_w, \quad \forall\, j \in S, \quad \forall\, j' \in Q_j, \forall\, w \in W, \end{aligned} \tag{31}$$

$$\begin{aligned} &x_{i,j} + \sum_{k \in OV_{i,w}} x_{k,j'} - \sum_{k \in OV'_{i,w}} \text{rem}_{k,j',w} \\ &+ \sum_{k \in OV'_{i,w}} \text{add}_{k,j',w} \leq 1, \quad \forall\, i \in O \setminus O_w, \quad \forall\, j \in S, \quad \forall\, j' \in Q_j, \forall\, w \in W, \end{aligned} \tag{32}$$

$$\sum_{j \in S} \text{add}_{i,j,w} - \sum_{j \in S} \text{rem}_{i,j,w} = 0, \quad \forall\, i \in O_w, \quad \forall\, j \in S, \quad \forall\, w \in W, \tag{33}$$

$$x_{i,j} \geq \text{rem}_{i,j,w}, \quad \forall\, i \in O_w, \quad \forall\, j \in S, \quad \forall\, w \in W, \tag{34}$$

$$\sum_{i \in O_w} \sum_{j \in S} \text{add}_{i,j,w} \leq \lambda, \quad \forall\, w \in W, \tag{35}$$

$$x_{i,j} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \quad \text{add}_{i,j,w} \in \{0, 1\}, \quad \text{rem}_{i,j,w} \in \{0, 1\} \tag{36}$$

It can be seen that, in comparison with the original RRTSAP definition, the MARS constraints are split into two sets, per scenario w: in Eq. 31, involving the operations for which *add*/*rem* variables were added (set O_w), and in Eq. 32, involving the other operations (set $O \setminus O_w$). Constraints 29 are only in place for the operations that have *add*/*rem* variables added in the scenario. Moreover, constraints 33 and 34 are only included for operations part of O_w.

4 GRU case study

The recoverable robust stand allocation framework was tested in a case study at Guarulhos International Airport of São Paulo (GRU). The focus of the case study is on the international terminal of the airport. This is a linear terminal, with almost 500 m of length, 10 wide-body contact-stands and the security control at one side of the terminal. Besides, these contact-stands, 8 wide-body remote stands, 5 narrow-body remote stands and 16 parking-only stands that are usually saved for international flights were also considered. Any of the wide-body contact-stands can be split into two narrow-body stands. Considering the potential split of the wide-body stands and the previously mentioned stands, a total of 75 stands were included in the model. We do not allow for

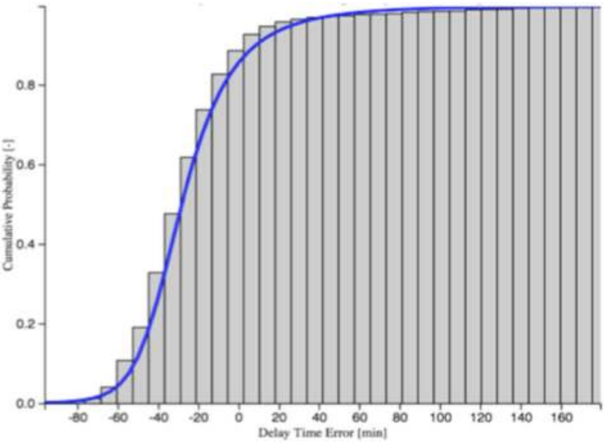


Fig. 6 Cumulative arrival time deviation distribution of one of the airlines

allocation of aircraft to other terminals, although in practice this could occur in case of stand shortage.

Six days of operations were used as cases to test the proposed framework (from the 19th until the 27th of November, 2015). Nevertheless, a full year of operations (from August 2014 to July 2015) was used for the flight data analysis. These data included information about all international flights operated in GRU airport during that year. A historical arrival time deviation distribution was generated for every flight number with an occurrence higher than thirty. For the flight numbers with less than thirty occurrences in the data, an airline arrival time deviation distribution was used. To avoid generalization of the home carrier of the airport (TAM airlines), the home-carriers arrival time deviation distribution was further split per region. This followed our data analysis, which showed differences in arrival time deviation distributions per region for the home carrier. For each of these cases, the historical arrival time deviations were divided into a high number of bins and a theoretical best fit was determined by the Kolmogorov–Smirnov test, with the evaluation of 23 theoretical distributions available in Python SciPy library. Figure 6 gives an example of a calibrated distribution curve obtained for a particular airline.

In terms of departing times, an initial analysis showed that the delays were in most of the cases minor. With exception of few cases, most of the flights analyzed were either delayed due to a late arrival (in the case of short-stay visits) or their delay could be accommodated within the embark times and buffer times considered. It should be mentioned that a large number of visits in this terminal are long-stay visits, involving a morning arrival flight and an evening departure flight. Therefore, departure delays are easier to mitigate. That being the case, we assumed that the departure times of flights remained equal to the scheduled departure time, unless the turnaround time becomes shorter than the minimum disembarking plus minimum embarking time. In that case, the departure time becomes the arrival time plus the minimum disembarking time plus the minimum embarking time.

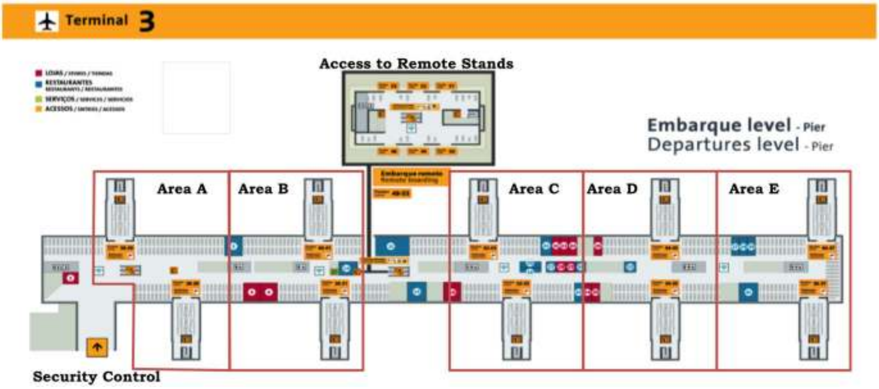


Fig. 7 Airport layout of Terminal 3—the security control and the five contact-stands areas

The parameters for the TSAP were defined together with GRU airport planners. For instance, the buffer time between two consecutive (dis)embarking operations at the same (MARS) was set to 15 min. A towing operation in the TSAP was only allowed if the aircraft visits the airport for a time longer than 3 h, which includes a set scheduled hour of disembarkation and an hour embarkation. For the recovery algorithm, a maximum allowed waiting time of 5 min was considered in the case of a stand conflict (i.e., for the waiting recovery action). It was also decided to not include a maximum number of re-allocations (λ).

To determine the walking distance, the airport terminal was divided into areas, composed of two opposing contact-stands (Fig. 7). For the remote stands, an adjustment was made in the walking distance—a fictitious opposing gate pair was assumed for the remote stands with a walking distance further than the furthest contact-stand pair in the terminal. Please note that due to the scope of the international terminal, transfer passengers are not specifically considered since nearly all passengers need to clear customs upon arrival/before departure. The matrix of potential revenue generated was estimated based on historical non-aeronautical revenue data of specific flights from both terminal stores (i.e., close to security control) and pier stores (i.e., close to the contact-stands) at GRU airport. The matrix was computed per combination flight and terminal area.

The airport set a maximum desired runtime for RRTSAP of 2 h. Consequently, after some tests with the computational times per plan and scenario, it was decided to run the RRTSAP with 30 scenarios with an allowed gap of 1%. All cases were solved with an 8 GB Mac OS computer with an average runtime of 87 min for the walking distance objective. The RRTSAP was implemented in Python and solved with Gurobi LP Solver.

The analysis of the results will be divided in two parts. First, we discuss the results obtained with the RRTSAP proposed and compare them with the results obtained with a traditional robustness approach. Then, we discuss the impact of adopting different objective functions to the recoverable robust TSAP.

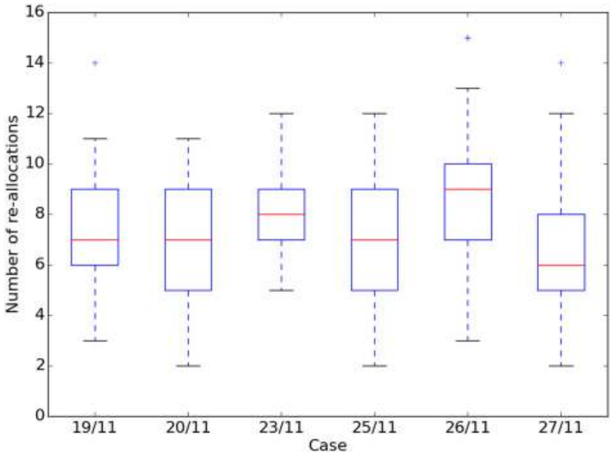


Fig. 8 Number of re-allocations required per scenario, for each case

4.1 Recoverable robustness versus strict robustness

This section aims to provide an overview of the results obtained from the recoverable robust stand allocation framework for six full-day case studies. All the results presented in this section used the minimization of the total walking distance as the objective function.

The results of the recoverable robust stand allocation model are compared with the results obtained by a strict robust stand allocation model. Liebchen et al. (2009) introduced this term to indicate a classical robust approach, in which all established scenarios have to be fully satisfied to return a solution. This strict approach has already been utilized in stand allocation, for example in Seker and Noyan (2012). In the strict robust stand allocation model, recovery actions as described in our recoverable robustness framework are not allowed, and thus, all generated scenarios have to be fully satisfied. This differs from our RRTSAP, where limited schedule recoveries are allowed in the solution. In Fig. 8, we present the variation observed for all analyzed cases. In all combinations of scenario cases, at least two re-allocations were observed. The maximum number of re-allocations was fifteen, observed for one scenario in the 26/11 case. The number of required changes for most combinations was smaller than 10% of the amount of operations in the analyzed cases.

The results for the 6 days are presented in Table 1. For the computation of the strict robust results, we utilized an extended TSAP formulation, including the scenarios from Step S. The recoverable robust solution (RR WD) is, on average, 17.80% better than the strict robust solution (strict WD). Forcing a strict robust approach, we obtain a resource allocation that is, at least, 16.79% worse than the resource allocation obtained with the RRTSAP. This is the price of the conservatism of such classical robust approaches.

We further analyze the difference between the results of both approaches by looking at their performance per scenario. The strict WD has a constant performance for all scenarios—the solution is feasible in all scenarios and it results in the same average

Table 1 Average passenger walking distances of the recoverable robust solution (RR WD), the strict robust solution (strict WD) and the average value of the recoverable robust solution over the 40 scenarios tested (AVG RR WD)

Case	Visits	RR WD	Strict WD	$\Delta_{1,2}$	AVG RR WD	$\Delta_{3,2}$
		(m) (1)	(m) (2)	(%)	(m) (3)	(%)
19/11	70	164.6	200.9	– 18.07	190.9	– 4.97
20/11	66	163.9	197.7	– 17.10	187.7	– 5.06
23/11	68	163.0	195.9	– 16.79	194.9	– 0.52
25/11	60	160.7	195.7	– 17.91	186.5	– 4.70
26/11	70	165.3	204.4	– 19.11	197.2	– 3.50
27/11	64	165.6	201.6	– 17.83	196.0	– 2.75

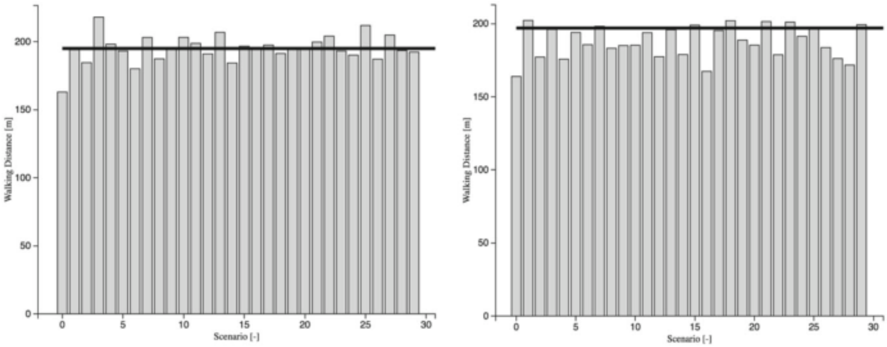


Fig. 9 Comparison of the average passenger walking distance for the recoverable robust solution (bars) and the strict robust solution (black line) per scenario, for 23/11 (left) and 20/11 (right)

walking distance value. In the case of the recoverable robustness framework, the recovery algorithm is activated to solve some conflicts and the final objective function value varies per scenario. The results are summarized in Fig. 9, for the most and least differential day between the two approaches (20/11 and 23/11, respectively), and the average values for all scenarios are presented in Table 1.

The comparison further highlights the capability of the recoverable robust stand allocation framework to provide a robust yet less conservative solution. It can be concluded that the results from the recoverable robust model are often better than the ones obtained with the strict robust model. The RRTSAP provided solutions that were, on average over the scenarios, 3.6% better than the strict TSAP model.

To further highlight the performance of the recoverable robust framework, we provide in Fig. 10 the analysis of the same cases but now with the objective to maximize the percentage of passengers allocated to a contact-stand, one of the alternative objective functions. For the 23/11 case, only five scenarios resulted in a slightly lower percentage of passengers allocated to a contact-stand. Most scenarios resulted in a higher percentage for the recoverable robust solution. In the 20/11 case, all scenarios, except 1, maintained a higher percentage of passengers allocated to a contact-stand. It can therefore be concluded that the relative cost of robustness for the recoverable

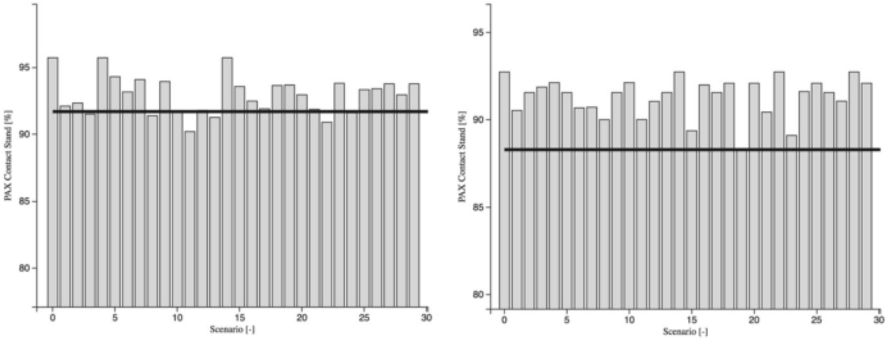


Fig. 10 Comparison of percentage of passengers allocated to a contact-stand for the recoverable robust solution (bars) and the strict robust solution (black line) per scenario, for 23/11 (left) and 20/11 (right)

robust solutions is lower compared to the strict robust solutions, not only for the walking distance objective function but also for the objective of percentage of passengers allocated to a contact-stand.

4.2 Objective function comparison

In this subsection, we discuss the impacts of adopting different objectives. As explained in Sect. 2.4, we considered four objectives: the minimization of passengers' walking distance (WD), the maximization of potential commercial revenue (PCR), the minimization of number of tow operations (TO) and the maximization of the percentage of passengers allocated to a contact-stand (PC). The metrics included in the comparison of the objective functions are: average walking distance over the scenarios (AVG WD), average potential commercial revenue over the scenarios (AVG PCR), the average number of tows in the scenarios (AVG TO) and the average percentage of passengers allocated to a contact-stand (AVG PC). The values represent the average of each metric over the set of 30 scenarios.

We used the previous results, with the minimization of WD for the 6 days of operations, as reference results. We solved the RRTSAP again for the different objectives. By changing the objective function, the new results are compared with those reference results. In Table 2, we present the range of the metrics variation between the reference results and the results obtained when using the different objective functions. The objectives used in the model are indicated in rows and the impact on each of the metrics used is indicated in columns.

As can be seen, the AVG WD for passengers can increase with more than 100% if we consider the objective of minimizing towing operations. For the objective of maximizing of passengers allocated to a contact-stand, this increase can be larger than 42%. Although the impact of considering the minimization tows seems logical, since to minimize the towing operations several long-stay visits are assigned to remote stands, the results for the maximization of PC are somehow surprising. At first sight, both objectives seem equivalent or similar. Nevertheless, the solutions for minimization of WD resulted in a high occupancy of contact-stands associated with short WD (either

Table 2 Ranges of relative variations in walking distance (WD), towing operations (TO), percentage of passengers allocated to a contact-stand (PC) and potential commercial revenue (PCR) when using different objectives, comparing with the results for minimization of walking distance

Objective	Δ AVG WD (%)	Δ AVG TO (%)	Δ AVG PC (%)	AVG PCR (%)
Min TO	[89.7, 111.9]	[− 82.4, − 72.4]	[− 54.7, − 42.7]	[− 55.0, − 44.8]
Max PC	[22.0, 42.1]	[− 0.3, 0.2]	[3.9, 5.6]	[0.2, 2.3]
Max PCR	[4.1, 9.7]	[− 0.1, 0.4]	[− 0.7, 1.5]	[0.7, 3.3]

with short-stay visits or with short disembark and embark operations from long-stay visits of large aircraft) and the solutions for maximization of PC resulted in an almost equal use of all contact-stands. This is due to the fact that for the latter there is no preference within the contact-stands. Especially during the off-peak periods, in which capacity is available, the PC solution allocates much more operations to contact-stands associated with long walking distances. This results in a high AVG PC, but also in a higher WD for a significant number of passengers.

When the goal is to maximize PC, the AVG PC increases with 3.9–5.6 percent points and the AVG PCR increases (up to 2.3%), relative to the AVG WD objective. On the other hand, if the selected objective is to minimize the number of towing operations, these operations are reduced by around 75% points. However, this happens at the expense of reducing AVG PCR (with more than 44.8%), reducing AVG PC (with more than 42.7%) and increasing AVG WD. It has to be noted that the objective to minimize tows degrades the performance of the RRTSAP for some cases. This is due to the strict binary requirements and the impossibility for Gurobi to distinguish the relative importance of tows. For some cases, this resulted in computational times of more than 6 h to obtain to an optimality gap of 10%.

One of the innovations of this work is the inclusion of a commercial revenue objective in the TSAP. The results suggest that the trade-off between WD and PCR is rather small—GRU airport can increase the AVG PCR by up to 3.3% if it accepts an increase in the AVG WD of passengers with up to 10%. However, these results should be carefully analyzed. First, because the results reflect the linearity of Terminal 3 at GRU airport with a major store at the entrance of the pier. Most of the commercial revenue generated at this terminal comes from sales in the main terminal store. Thus, although there are differences between different contact-stands areas, all stand areas are mainly influenced by the this store and the relative differences between the areas are smaller than the common influence of the main terminal store. Second, because in this case study to minimize WD, is, to a certain extend, similar to the maximization of PCR, as the most promising stand areas in terms of commercial revenue are the ones located at the first half of the terminal (with limited walking distance). This is again a consequence of GRU's Terminal 3 layout and the way the stores are located along the pier. The differences in PCR are much larger when we compare solutions from other objectives—for instance, the best solutions for minimization of TO are on average nearly 50% worse in terms of PCR than the best solutions for maximization of PCR. A third important and relevant point is the fact that AVG WD and AVG PCR metrics are expressed in different units. Variations on the same scale in these metrics do have different significance. For instance, an increase in AVG WD of 10.0% corresponds to an increase in AVG WD of around 20 m per passenger. In a large airport like GRU, this might not have a large impact in passengers' satisfaction. On the other hand, an increase in AVG PCR of 3.3% can easily result in an increase in PCR by several thousands of dollars a day. Therefore, even for small percentage gains in terms of AVG PCR, the airport might be interested in considering the trade-off between these two objectives.

5 Conclusion

In this paper, we presented a recoverable robust stand allocation model to address the tactical airport stand allocation problem (TSAP). The model was developed in cooperation with Guarulhos International Airport of São Paulo (GRU). The goal of a more robust approach for the stand allocation problem and our interest in exploring multiple objectives drove this research.

The benefits of adopting a recoverable robust solution were clearly illustrated with the application of our framework to 6 days of operations in GRU. The recoverable robust stand allocation framework was always able to obtain robust solutions that perform better and are less conservative than the ones obtained with a traditional robust model. When comparing the results for different objectives, it was concluded that the passengers' walking distance is highly penalized when the other objectives such as minimizing towing operations or maximize the number of passengers allocated to a contact-stand are adopted. In addition, for the first time the maximization of airport commercial revenue was explicitly used as an objective for the stand allocation problem. The results suggest that the increase in potential commercial revenue for GRU airport is not large in terms of percentage points but that this increase can in practice be several extra thousands of dollars per day of operations. This is achieved by adapting the stand allocation plan, without significantly compromising the results in terms of average walking distance. Given the current trend to privatize airports and the fact that it is getting easier to collect accurate passenger and revenue data at airports, we believe that this objective will become a popular metric in future works addressing the allocation of resources at airports.

This was the first study to consider the concept of recoverable robustness to solve the TSAP. Future steps are needed to consolidate and extend this research. One recommendation is to further improve the computational speed of the proposed model, either via strengthened formulations or the adoption of heuristics. This would allow to explore more scenarios when defining a robust solution, without compromising computational time limitations. Furthermore, comparing the results with other robustness approaches to the TSAP would be recommended. This would provide insights in the performance of the different robustness approaches, in terms of objective function values as well as computational time.

Another extension would be the consideration of a multi-objective approach. This has already been implemented by several other authors under other frameworks (e.g., Guépet et al. 2015; Dorndorf et al. 2012). A multi-objective recoverable robust framework to solve the TSAP would thus be an innovative and challenging line of research. In particular, the traditional recovery algorithm would have to be adapted to an efficient multi-objective recovery algorithm dealing with dominated solutions. Furthermore, the definition of proper trade-offs between objectives would be a major challenge. It is widely recognized that a coherent definition of these trade-offs is hardly if ever found in the decision-makers' minds. Finally, the integration of the (recoverable robust) TSAP with other tactical resource planning problems at an airport would be an area of future research. This has already been proposed by Kim et al. (2013) and Guépet et al. (2015) and could provide useful insights in the context of a robust optimization of airport resources.

References

Babić O, Teodorović D, Tošić V (1984) Aircraft stand assignment to minimize walking. J Transp Eng 110(1):55–66

Ben-Tal A, Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press, Princeton

Bertsimas D, Sim M (2014) The price of robustness. Oper Res 52(1):35–53

Bihr R (1990) A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. J Comput Ind Eng 19(1–4):280–284

Bolat A (2000) Procedures for providing robust gate assignments for arriving aircrafts. Eur J Oper Res 120(1):63–80

Bouras A, Ghaleb MA, Suryahatmaja US, Salem AM (2014) The airport gate assignment problem: a survey. Sci World J. https://doi.org/10.1155/2014/923859

Braaksma J (1977) Reducing walking distances at existing airports. In: Airport Forum, vol 7

Cheng CH, Ho SC, Kwan CL (2012) The use of meta-heuristic for airport gate assignment. Expert Syst Appl 39(16):12,430–12,437

D'Angelo G, Di Stefano G, Navarra A, Pinotti CM (2011) Recoverable robust timetables on trees. In: Proceedings on 3rd international conference on combinatorial optimization and applications, Lecture notes in computer science, vol 5573. Springer, Huangshan, pp 433–446

Diepen G, den Akker JM, Hoogeveen JA, Smeltink JW (2012) Finding a robust assignment of flights to gates at Amsterdam Airport Schiphol. J Sched 15(6):703–715

Dorndorf U, Drexl A, Nikulin Y, Pesch E (2007) Flight gate scheduling: state-of-the-art and recent developments. Omega 35(3):326–334

Dorndorf U, Jaehn F, Pesch E (2008) Modelling robust flight-gate scheduling as a clique partitioning problem. Transp Sci 42(3):292–301

Dorndorf U, Jaehn F, Pesch E (2012) Flight gate scheduling with respect to a reference schedule. Ann Oper Res 194(1):177–187

Dorndorf U, Jaehn F, Pesch E (2017) Flight gate assignment and recovery strategies with stochastic arrival and departure times. OR Spectrum 39(1):65–93

Drexl A, Nikulin Y (2008) Multicriteria airport gate assignment and pareto simulated annealing. IIE Trans 40(4):385–397

Froyland G, Maher SJ, Wu CL (2013) The recoverable robust tail assignment problem. Transp Sci 48(3):351–372

Genç HM, Erol OK, Eksin I, Berber MF (2012) A stochastic neighborhood search approach for airport gate assignment problem. Expert Syst Appl 39(1):316–327

Geuens M, Vantomme D, Brengman M (2004) Developing a typology of airport shoppers. Tour Manag 25(5):615–622

Graham A (2009) How important are commercial revenues to today’s airports? J Air Transp Manag 15(3):106–111

Guépet J, Acuna-Agost R, Briant O, Gayon JP (2015) Exact and heuristic approach to the airport stand allocation problem. Eur J Oper Res. https://doi.org/10.1016/j.ejor.2015.04.040

Jaehn F (2010) Solving the flight gate assignment problem using dynamic programming. Z Betr 80(10):1027–1039

Kim SH, Feron E, Clarke JP (2013) Gate assignment to minimize passenger transit time and aircraft taxi time. J Guid Control Dyn 36(2):467–475

Kumar P, Bierlaire M (2014) Multi-objective airport gate assignment problem in planning and operations. J Adv Transp 48(7):902–926

Liebchen C, Lübbecke M, Möhring R, Stiller S (2009) The concept of recoverable robustness, linear programming recovery, and railway applications. In: Ahuja RK, Möhring RH, Zaroliagis CD (eds) Robust and online large scale optimization: models and techniques for transportation systems, vol 5868. Lecture notes in computer science. Springer, Berlin, pp 1–27

Lim A, Wang F (2005) Robust airport gate assignment. In: Proceedings of the 17th IEEE international conference on tools with artificial intelligence. IEEE, Hong Kong, pp 74–81

Lin YH, Chen CF (2013) Passengers’ shopping motivations and commercial activities at airports—the moderating effects of time pressure and impulse buying tendency. Tour Manag 36(June):426–434

Löhndorf N (2016) An empirical analysis of scenario generation methods for stochastic optimization. Eur J Oper Res 255(1):121132

Maharjan B, Matis T (2012) Multi-commodity flow network model of the flight gate assignment problem. Comput Ind Eng 63(4):1135–1144

Mangoubi R, Mathaisel D (1985) Optimizing gate assignments at airport terminals. Transp Sci 19(2):173–188

Seker M, Noyan N (2012) Stochastic optimization models for the airport gate assignment problem. Transp Res 48(2):438–459

Vanderstraeten G, Bergeron M (1988) Automatic assignment of aircraft to gates in a terminal. Comput Ind Eng 14(1):15–25

Xu L, Zhang C, Xiao F, Wang F (2017) A robust approach to airport gate assignment with a solution-dependent uncertainty budget. Transp Res B Methodol 105:458–478

Yan S, Huo CM (2001) Optimization of multiple objective gate assignments. Transp Res A Policy Pract 35(5):413–432

Yan S, Tang CH (2007) A heuristic approach for airport gate assignments for stochastic flight delays. Eur J Oper Res 180(2):547–567

Yan S, Shieh CY, Chen M (2002) A simulation framework for evaluating airport gate assignments. Transp Res A 36(10):885–898