

Minimizing Airplane Boarding Time*

Felix J. L. Willamowski^{1†} and Andreas M. Tillmann^{1,2‡}

¹ RWTH Aachen University, Lehrstuhl für Operations Research,
Kackertstr. 7, D-52072 Aachen, Germany

² RWTH Aachen University, Visual Computing Institute,
Ahornstr. 55, D-52074 Aachen, Germany

Abstract. The time it takes passengers to board an airplane is known to directly influence the turn-around time of the aircraft and thus bears a significant cost-saving potential for airlines. Although minimizing boarding time therefore is the most important goal from an economic perspective, previous efforts to design efficient boarding strategies apparently never tackled this task directly. In this paper, we first prove strong NP-hardness of the problem. While this generally justifies the development of inexact solution methods, we demonstrate that all commonly discussed boarding strategies may in fact give solutions that are far from optimal. We complement these theoretical findings by a simple time-aware boarding strategy with guaranteed approximation quality (under very reasonable assumptions) as well as mixed-integer programming formulations. Numerical experiments with structured random data show that for several airplane cabin layouts, provably high-quality or even optimal solutions can be obtained within reasonable time in practice.

1 Introduction and Preliminaries

It is folklore knowledge in the aviation industry that a passenger airplane can only generate revenue while in the air, as ground handling operations and the time a plane spends, e.g., at the gate effectively cost airlines money. Thus, airlines wish to minimize the turn-around times of their airplanes, i.e., the times between the last landing and the next takeoff (alternatively, the time the aircraft spends at the gate after arrival and before the subsequent departure). Although there are many steps involved in turning around an airplane (see, e.g., [1, Figure 1] or [2, Figure 1]), passenger boarding is one of the steps that most affect the turn-around time [3]. This is because all other steps either cannot be significantly shortened (e.g., taxiing the aircraft into its parking position), may run parallel to (and finish during) the boarding process (e.g., baggage handling), or necessarily precede it (like refueling, which is typically not allowed while passengers are on board for safety reasons, or cabin maintenance, avoided for passenger comfort reasons). Moreover, a growing amount of carry-on luggage has been held responsible for an increase of boarding times, further emphasizing the bottleneck-like role of boarding in any time-efficient turn-around process [4].

As revealed by Jaehn and Neumann [5], an average boarding time reduction by just one minute can lead to cost savings of roughly \$50 million per year (or more) for a major airline. Furthermore, a reduced boarding time is also beneficial for passengers and airport operators. For passengers, it results in reduced average individual boarding times, and airport operators are possibly able to offer more flights per day per gate. For a more detailed discussion of benefits and challenges, we refer to the recent survey [5].

In this paper, we study the process of boarding through a jet-bridge, where each passenger has a preassigned seat. Our explicit goal is to minimize the overall boarding time, where we focus on the time elapsing between the first passenger entering the airplane cabin and the last passenger sitting down on his/her seat. In the remainder of this section, we first briefly review the previous literature on airplane boarding strategies, and then formally define the problem treated here and provide an overview of our main contributions. In Section 2, we discuss the computational complexity of the problem under different assumptions, deriving strong NP-hardness results as well as approximation guarantees. Section 3 introduces mixed-integer

* This work was partially funded by the excellence initiative of the German federal and state governments.

† Corresponding author; email: willamowski@or.rwth-aachen.de

‡ Current address: Technische Universität Braunschweig, Institute for Mathematical Optimization, Universitätsplatz 2, 38106 Braunschweig, Germany

programming (MIP) formulations to solve arbitrary instances of the problem to exact optimality. The numerical experiments presented in Section 4 demonstrate the practicality of the proposed approaches and the potential gains compared to several previously known boarding strategies that do not explicitly consider the time-minimization objective; we also assess robustness of the boarding sequences w.r.t. time-data perturbations and disruptive passenger behavior. Finally, some concluding remarks are given in Section 5.

1.1 Related Work

Many different approaches for shortening the boarding time have been presented in the literature; the recent paper by Jaehn and Neumann [5] provides a very detailed discussion and survey of previous works. Although computational intractability had, to the best of our knowledge, not been formally established prior to the present work (see Section 2), the majority of methods that have been proposed are simple heuristic schemes that provide easy-to-implement boarding strategies, including common back-to-front or boarding-group strategies. Since in general, such heuristics do not provide any guarantee regarding the solution quality, and because the boarding process can furthermore be viewed as being of inherently stochastic nature [1], a large body of works employ computer simulations to compare different boarding strategies over large sets of random instance data to identify ones that consistently work better than others. Extensive simulation allows to draw conclusions on the empirical average performance of different strategies (depending on the distribution of random data) and to relatively easily analyze other aspects such as the impact of cabin configuration changes (e.g., multiple doors) in the same fashion. An early such study conducted by Boeing is described in [4], which also included actual passenger loading tests to validate the performance of the developed discrete event simulation tool; one notable result that was subsequently confirmed to varying degrees by other simulation studies is that substantial boarding time reductions can be obtained using the so-called *outside-in* strategy (also known as “WilMA”). Here, as opposed to more traditional row- or block-wise strategies (like filling the airplane from the last to the first row as in the *back-to-front* strategy), passengers with window seats board first, followed by those with middle seats, and aisle seats last. In [6], a broader study and simulation-based comparison of different boarding strategies concluded, in particular, that a *reverse pyramid* scheme leads to good performance as well. Steffen [7] used simulation results to develop a new, empirically superior boarding strategy that became known as the *Steffen method*, along with a modification that is meant to be easier to realize in practice (the Steffen method itself consists of a relatively complicated pattern). A variant of the Steffen method that achieves improvements of the overall boarding time by distributing seat assignments to passengers based on assumed prior knowledge of the number of carry-on items per passenger was discussed in [8]. A ranking of simple boarding strategies including the ones just mentioned appears in [9] (see also [2]); according to this, the smallest average boarding times are achieved by the Steffen method, followed by the (much simpler) outside-in strategy and the reverse pyramid scheme; the modified Steffen method comes in last, even after *random* boarding, in which passengers can just enter the airplane in an arbitrary order.

Some attempts have also been made to improve results by adopting flight-by-flight strategies rather than static ones like outside-in or back-to-front, i.e., boarding sequences that take individual flight/passenger data into account. For instance, the paper [10] proposed an adaptive queueing scheme that mainly targets a reduction of the number of on-board (aisle and seat) interferences, i.e., situations where some passenger is kept from proceeding by another who is, e.g., currently loading carry-on luggage into the overhead bin or sitting in the way to the target seat (cf. [3]). Avoiding such blockages is meant to improve the passengers’ perception of the boarding process (as waiting is a “source of annoyance”), and also has an intuitive tangible impact on the overall boarding time. However, note that an interference-minimal boarding sequence is not automatically also a solution that actually minimizes the boarding time. Moreover, in the last few years, mixed-integer programming (MIP) techniques were applied to some related boarding problems. In [11], the declared goal is to minimize the boarding time, but differing from the general boarding time minimization considered here, a seat-assignment MIP problem based on carry-on luggage information is solved so that the Steffen method will result in short boarding times. A linear MIP to minimize aisle and seat interferences was proposed in [12], similar to an earlier nonlinear MIP (MINLP) introduced in [13]; in the latter work, it is stated that “To make the problem more tractable, we used the minimization of passenger interferences

as our objective in lieu of the minimization of boarding time”. (Moreover, the authors of [13] note their model to be a nonlinear assignment problem, which *in general* is an NP-hard class of problems; however, they do not provide a proof that their concrete problem is still NP-hard.) Another MIP for interference minimization was put forth in [14], along with a genetic algorithm that sometimes achieves better results than commercial MIP solvers under solution time limits.

It is also worth mentioning that robustness of boarding strategies has been treated in the literature as well, even in a MIP context (see [15]), but, to the best of our knowledge, not in the present setting (actual boarding time minimization) and extent (combining time fluctuations, passengers changing positions in the boarding queue, and/or arriving late at the gate). The most extensive work in this direction we are aware of is [3], a broad simulation study involving early and late passengers, yet different from how we treat disruptions (cf. Section 4.4).

For more detailed discussions of the many different (static) boarding strategies and the comparisons and conclusions from the extensive simulation studies, we refer, in particular, to [5,6] and the references therein. Notably, it seems the (economic) main goal of actually minimizing the boarding time has not been tackled explicitly before. While it is understandable that, e.g., passenger boarding comfort is taken into account—an argument for simple, easy-to-implement strategies—it has already been demonstrated (e.g., by the Steffen method and interference-minimizing MIP approaches) that “by seat”-strategies in which each passenger is assigned a specific position in the boarding sequence can lead to greater time savings. Therefore, in this paper, we focus solely on the economic side, i.e., on determining sequences that minimize the boarding time. In principle, such sequences might be realized by roughly dividing the passengers into groups that are asked to gather in certain areas at the gate, where they are then “sorted” by gate agents into the intended order in which to board. Nevertheless, we leave practical details of how to enforce by-seat boarding sequences at the gate for future consideration.

1.2 Problem Definition and Notation

We consider the *airplane boarding problem (ABP)* that asks for a sequence in which to board passengers with given seats to an airplane such that the overall boarding time is minimized. We focus on the setting that is most prevalent in the existing literature: there is one entrance at the beginning of the passenger cabin to be used for boarding, and the cabin consists of a single deck with seats to either side of a single aisle. Moreover, we adopt the common assumption of “single-class boarding”, i.e., we leave out priority boarding (e.g., first class) and other pre-boarding groups (e.g., families with small children) from our considerations. For simplicity, we stick to this setup throughout the paper, but remark that several extensions could easily be incorporated into the methods developed later—some examples are given at the end of this subsection—and that generalization to other cabin layouts (e.g., two doors and/or two aisles) may also be possible without much additional effort.

To formalize things, we need to introduce some notation. The sets of rational, integer and natural numbers are denoted by \mathbb{Q} , \mathbb{Z} and \mathbb{N} , respectively; possible restrictions are indicated by suitable subscripts (e.g., $\mathbb{N} = \mathbb{Z}_{>0}$). For a number $n \in \mathbb{N}$, we write $[n] := \{1, 2, \dots, n\}$. An airplane’s cabin layout parameters are the (ordered) set of rows $\mathcal{R} := [R]$, $R \in \mathbb{N}$ (the row closest to the door being row 1, and all later rows $r \leq R$ being accessible by first passing rows $1, 2, \dots, r-1$), and the collection of seats \mathcal{S} . For each row $r \in \mathcal{R}$, there are two (possibly empty) ordered sets of seats $\mathcal{S}_r^1 := ((r, 1), \dots, (r, k_r^1))$ and $\mathcal{S}_r^2 := ((r, k_r^1+1), \dots, (r, k_r^1+k_r^2))$, with $k_r^1, k_r^2 \in \mathbb{Z}_{\geq 0}$. Each seat $(r, s) \in \mathcal{S}$ is accessible from the aisle by passing seats $(r, k_r^1), \dots, (r, s+1)$ if $s \leq k_r^1$, or $(r, k_r^1+1), \dots, (r, s-1)$ otherwise (i.e., for any r , seats (r, k_r^1) and (r, k_r^1+1) are the aisle seats, and $(r, 1)$ and $(r, k_r^1+k_r^2)$ are the window seats). Moreover, by $\mathcal{P}(r)$ we denote the set of passengers with seats in row r .

An instance of our ABP consists of a given airplane cabin layout (\mathcal{R} and \mathcal{S} as defined above), a set of passengers $\mathcal{P} := [P]$, $P \in \mathbb{N}$ with $P \leq |\mathcal{S}|$, unique passenger-seat assignments given by $\sigma : \mathcal{P} \rightarrow \mathcal{S}$, $p \mapsto (r(p), s(p))$, and for each passenger $p \in \mathcal{P}$, a *settle-in time* $t_p^s \in \mathbb{Q}_{\geq 0}$ (consisting of the time passenger p takes to stow away his/her carry-on luggage, move within row $r(p)$ and finally sit down at the assigned seat $s(p)$) and *moving times* $t_{p,r}^m \in \mathbb{Q}_{\geq 0}$ (for passing row $r \in \mathcal{R}$) for $r \leq r(p) - 1$. We assume that such time data is given; in the context of simulations to evaluate boarding methods, the data is typically drawn

from a random distribution for each instance of many trials to be averaged, whereas in practice, reasonable estimates may be available from real-life observations or based on passenger demographics (cf., e.g., [4]). We will distinguish between different levels of “resolution” regarding the time data in our numerical experiments and, in particular, in Section 2, where we will show that this has an impact on the theoretical difficulty of the ABP.

Furthermore, we presume that each passenger p tries to go directly to his/her seat $\sigma(p)$ without unnecessary “loitering” and that p settles in (in particular, stows away carry-on luggage) at the row $r(p)$ (from which the assigned seat $s(p)$ is accessible). Following common modeling practice, we also assume that overtaking in the aisle is not possible, so a passenger can proceed to some row only if it is not presently occupied by another passenger. More precisely, let p be a passenger who has started some action (moving, waiting, or settling in) at row r at time $t \in \mathbb{Q}_{\geq 0}$. If p passes row r or settles in at it (so $r = r(p)$), he/she occupies the row for the time period $[t, t + t_{p,r}^m]$ or $[t, t + t_p^s]$, respectively. Analogously, in case p has to wait a time period $w \in \mathbb{Q}_{\geq 0}$ at r , he/she blocks it for the time period $[t, t + w]$. (These blockages and resulting waiting periods are what has been referred to as *aisle interferences* in the literature, see the earlier discussion of related work.)

Now, more formally, ABP is the task to find a permutation $\pi : \mathcal{P} \rightarrow [P]$ (i.e., a one-to-one mapping describing the passenger boarding sequence) that minimizes the overall boarding time. Borrowing the common notation for completion times from the scheduling literature (cf., e.g., [16]), we denote the boarding time induced by a sequence π by $C_{\max}(\pi)$ (or simply C_{\max} if π is clear from the context). Thus, we can abstractly express the ABP of interest as $\min\{C_{\max}(\pi) : \pi \in \Pi_P\}$, where Π_P denotes the symmetric group defined over $\mathcal{P} = [P] = \{1, 2, \dots, P\}$.

While there does not appear to exist a closed-form expression for $C_{\max}(\pi)$ in general, we can evaluate the boarding completion time induced by any sequence π recursively as follows.

Lemma 1.1. *Given an ABP instance $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \sigma, \{t_{p,r}^m\}, \{t_p^s\})$ and a permutation $\pi \in \Pi_P$, it holds that $C_{\max}(\pi) = \max_{r \in \mathcal{R}} C(P, r)$, where $C(P, r)$ is defined by the recursion*

$$C(1, r) := \begin{cases} \sum_{r'=1}^r t_{\pi^{-1}(1), r'}^m & \text{if } 1 \leq r \leq r(\pi^{-1}(1)) - 1, \\ \sum_{r'=1}^{r-1} t_{\pi^{-1}(1), r'}^m + t_{\pi^{-1}(1)}^s & \text{if } r = r(\pi^{-1}(1)), \\ 0 & \text{if } r(\pi^{-1}(1)) + 1 \leq r \leq R \end{cases}$$

and for $i \in \{2, 3, \dots, P\}$, with $C(i, 0) := 0$ ($i \geq 2$)

$$C(i, r) := \begin{cases} \max \left\{ \begin{array}{l} C(i-1, r) + t_{\pi^{-1}(i), r}^m \\ C(i, r-1) + t_{\pi^{-1}(i), r}^m \\ C(i-1, r+1) \end{array} \right\} & \text{if } 1 \leq r \leq r(\pi^{-1}(i)) - 1, \\ \max \left\{ \begin{array}{l} C(i-1, r) + t_{\pi^{-1}(i)}^s \\ C(i, r-1) + t_{\pi^{-1}(i)}^s \end{array} \right\} & \text{if } r = r(\pi^{-1}(i)), \\ C(i-1, r) & \text{if } r(\pi^{-1}(1)) + 1 \leq r \leq R. \end{cases}$$

Consequently, $C_{\max}(\pi)$ can be computed in $\mathcal{O}(PR)$ time.

Proof. To show correctness, we prove the following assertion by induction over $i \in [P]$: $C(i, r)$ is the time row $r \in \mathcal{R}$ becomes accessible again after being occupied by a passenger $p \in \{\pi^{-1}(1), \dots, \pi^{-1}(i)\}$. Since the very first passenger $\pi^{-1}(1)$ is never blocked, this clearly holds true for $C(1, 1), \dots, C(1, R)$. For the induction step, consider passenger $p := \pi^{-1}(i)$ (and presume the $C(i, r)$ -values are computed in increasing row index order $r = 1, 2, \dots, R$, i.e., following the progression of p through the plane). For $r \leq r(p)$, p can access r only after no predecessor $(\pi^{-1}(i-1), \dots, \pi^{-1}(1))$ any longer occupies it (i.e., at $C(i-1, r)$) and only if p has already reached row r (i.e., at $C(i, r-1)$). In row r , p then spends $t_{p,r}^m$ time moving on to the next row (if $r < r(p)$) or t_p^s time settling in (if $r = r(p)$). In the former case, if row $r+1$ is still occupied by any predecessor $(\pi^{-1}(i-1), \dots, \pi^{-1}(1))$ by the time p would be done in row r , p actually has to wait (in row r) until $r+1$ becomes accessible again (i.e., until $C(i-1, r+1)$). Finally, since p does not traverse rows $r > r(p)$, the times $C(i-1, r)$ for those rows may be preserved (note that for passengers boarding

after p , it is nevertheless ensured by construction that they cannot arrive at a row $r > r(p)$ earlier than at $C(i, r(p))$, so it does not matter if $C(i - 1, r) < C(i, r(p))$. Put together, this concludes the induction, and yields the formula for $C(i, r)$ -values as claimed.

Regarding the running time $\mathcal{O}(PR)$ to obtain $C_{\max} = \max_{r \in \mathcal{R}} C(P, r)$, note that by iterating over passengers (in sequence π) and, for each passenger, over the rows (front to back), every required value is readily available (computed earlier or given data) and computation of any $C(i, r)$ then reduces to a constant number of arithmetic operations. This completes the proof. \square

Finally, let us point out a few optional extensions to the above-defined ABP:

Group boarding: Given an (ordered) group of passengers $\{p_1, \dots, p_k\} \subseteq \mathcal{P}$, it is required that they board consecutively (in the same order), i.e., they are mapped $\pi(p_1) = j_1, \dots, \pi(p_k) = j_k$ with $\{j_1 - \min_{i \in [k]} j_i, \dots, j_k - \min_{i \in [k]} j_i\} = \{0, \dots, k - 1\}$ (and, in the ordered case, $j_{i+1} = j_i + 1$ for $i \in \{0, \dots, k - 1\}$).

Row-dependent order: For each row $r \in \mathcal{R}$ and two passengers p_1, p_2 seated at row $r = r(p_1) = r(p_2)$, it is required that p_1 boards before p_2 ($\pi(p_1) \leq \pi(p_2) - 1$) if p_2 has to pass the seat of p_1 , i.e., $s(p_2) \leq s(p_1) - 1$ if $s(p_1) \leq k_r^1$ or $s(p_1) \leq s(p_2) - 1$ if $s(p_1) \geq k_r^1$. This explicitly forbids the so-called seat interferences, as is the case (implicitly) in the well-known outside-in boarding strategy.

Seat interferences: Instead of disallowing seat interferences entirely, one could introduce waiting times for each passenger p who has to get past already-seated passengers when settling in. Let $\mathcal{P}' \subseteq \mathcal{P}$ be the set of passengers who are also seated in $r(p)$ but board before p and $s(p) \leq s(p') - 1$ if $s(p') \leq k_r^1$ or $s(p') \leq s(p) - 1$ if $s(p') \geq k_r^1$ for all $p' \in \mathcal{P}'$. Then, the settle-in time of p increases by $\sum_{p' \in \mathcal{P}'} \tau_{p'}$, where $\tau_{p'} \in \mathbb{Q}_{\geq 0}$ is the time passenger p' needs to get up again and let p pass (and resettle).

For simplicity, we do not explicitly include such constraints in the main results and models of this paper, but will provide some additional remarks on how to do so later in Section 3.1.4. (Also, note that seat interferences are arguably less important than aisle interferences, cf. [13,6,9].)

1.3 Our Contributions

As alluded to earlier, it appears that previous works on reducing airplane boarding time have not explicitly formulated the task as an optimization problem with the objective to directly minimize the overall boarding time. Existing strategies are either heuristics or, if based on optimization, target other objectives like passenger interference in the aisles (and within seat rows). Indeed, the need for researching ways to find boarding sequences that minimize boarding time was pointed out in a recent extensive survey on boarding methods, see [5, Sect. 7.1.1]. Thus, we introduce the (to the best of our knowledge) first rigorous mathematical optimization models, utilizing mixed-integer programming, for tackling the problem of passenger *airplane boarding time minimization*. Moreover, although the problem has long been recognized as challenging, a formal proof of intractability was lacking. We close this gap by providing several computational complexity results, under different assumptions on the input time data (w.r.t. passengers moving through the plane and settling in at their seats) that reflect varying degrees of information resolution from the “uninformed” use of rough estimates identical for each passenger to “fully informed” knowledge of passenger-individual moving and settle-in times.

We summarize those results in Table 1, along with approximation guarantees for some cases that are also established in this paper. As it turns out, the ABP is NP-hard in the strong sense in almost all cases (for one case, the complexity remains open). Under simplifying assumptions (namely, assuming partially identical, constant time data for all passengers and full plane occupancy), simple boarding strategies can be shown to provide solutions within certain factors of optimality, while for the more general situations, no such guarantees are known. In these cases, our mixed-integer linear programming (MIP) approaches can nevertheless be employed to solve the ABP to optimality. Details on the complexity results follow in Section 2, and the MIP models are introduced in Section 3 afterwards.

Table 1. Main results of this paper on computational complexity, approximation guarantees and algorithmic approaches to the boarding time minimization problem, under different assumptions on the input time data; c^m , c^s , and c_1^m, \dots, c_p^m are constants in $\mathbb{Q}_{\geq 0}$. Here, k is the (maximal) number of seats per row, $H_k := \sum_{i=1}^k (1/i)$, and MIPs are universally applicable exact solution approaches. The polynomial exact or approximation schemes in the first three cases are shown to hold for planes fully booked to capacity.

Case (for all $p \in \mathcal{P}$, $r \in \mathcal{R}$)	Complexity	Approx./Algo.
$t_{p,r}^m = c^m$, $t_p^s = c^s$.	$\mathcal{O}(P)$ (Thm. 2.3)	exact scheme outside-in (Thm. 2.3)
$t_{p,r}^m = c^m$, $t_p^s \in \mathbb{Q}_{\geq 0}$ arbitrary.	strongly NP-hard (Thm. 2.2)	$(1 + \frac{k-1}{k}H_k)$ -approximation max-settle-row (Thm. 2.6)
$t_{p,r}^m = c_p^m$, $t_p^s = c^s$.	?	k -approximation outside-in (Thm. 2.4)
$t_{p,r}^m = c_p^m$, $t_p^s \in \mathbb{Q}_{\geq 0}$ arbitrary.	strongly NP-hard (Thm. 2.2)	MIP
$t_{p,r}^m \in \mathbb{Q}_{\geq 0}$ arbitrary, $t_p^s = c^s$.	strongly NP-hard (Thm. 2.1)	MIP
$t_{p,r}^m \in \mathbb{Q}_{\geq 0}$ arbitrary, $t_p^s \in \mathbb{Q}_{\geq 0}$ arbitrary.	strongly NP-hard (Thm. 2.1, Thm. 2.2)	MIP

2 Computational Complexity and Approximation

The availability of detailed passenger-specific moving and settle-in times pertains to an ideal situation that would allow for the most considerate boarding sequence planning. In practice, such “real” data will hardly be available, but reasonable estimates can be obtained from real-life observations (described in several studies, for instance in [17]; cf. also novel on-plane measurement devices like the sensor network to monitor boarding processes described in [18] and references therein) and used to derive suitable (probabilistic) model assumptions, see, e.g., the simulation approaches in [8,19]. Furthermore, as noted in [5], to actually identify the fastest boarding sequence, simplifying assumptions such as, in particular, identical walking speed for all passengers, would have to be done away with.

In the following, we therefore investigate the theoretical complexity of the boarding time minimization problem (i.e., ABP) under a variety of assumptions on the time data. Namely, we discuss all combinations of fully individual or identical-for-all-passengers settle-in times with fully individual (row-dependent), semi-individual (constant for all rows for any given passenger) and identical-for-all-passengers moving times, cf. Table 1.

2.1 Intractability Results

The most general ABP (with individual moving and settle-in times) turns out to be computationally intractable, even for very simple airplane cabin layouts, as shown by the following two results for the special cases with either constant settle-in times or constant moving times, respectively.

We start off with the case $t_{p,r}^m \in \mathbb{Q}_{\geq 0} \forall p \in \mathcal{P}, r \in \mathcal{R}$ and $t_p^s = c^s \forall p \in \mathcal{P}$ (with $c^s \in \mathbb{Q}_{\geq 0}$ a constant):

Theorem 2.1. *It is NP-complete in the strong sense to decide whether a given ABP instance admits a boarding sequence π that completes boarding by a given time $T \in \mathbb{Q}_{>0}$, even if restricted to instances with one seat per row, all-zero settle-in times, integer moving times, and integer T .*

Proof. For rational time data (moving and settle-in times, and T), containment in NP is easy to see: A “yes”-certificate is given by a permutation π of \mathcal{P} , for which the boarding completion time $C_{\max}(\pi)$ can be computed in polynomial time by Lemma 1.1; thus, it can be verified in polynomial time that $C_{\max}(\pi) \leq T$.

To show hardness, we reduce from $F_3|perm,blocking|C_{\max}$, the permutation flow shop problem with blockages, which is strongly NP-hard by Lemma A.1 in Appendix A. In this problem, we are given a set J of n jobs and three machines m_1, m_2, m_3 ; each job $j \in J$ consists of tasks j_m to be processed on machine m with corresponding processing times $t_{j,m}$. All jobs share the same technological order, i.e., they must be processed by every machine and always in the same machine order (w.l.o.g., $m_1 \rightarrow m_2 \rightarrow m_3$), and furthermore, the job processing order must be identical on all machines. The goal is to minimize the makespan, i.e., the completion time of the last job (task). Thus, let $C \in \mathbb{N}$, $J = [n]$, $M = \{m_1, m_2, m_3\}$ and $\{t_{j,m} \in \mathbb{Z}_{\geq 0} : j \in J, m \in M\}$ be an instance of the decision version of $F_3|perm,blocking|C_{\max}$ asking for a feasible permutation schedule with $C_{\max} \leq C$. By Lemma A.1, we may and do additionally assume that $t_{j,m_2} = 0$ for all $j \in J$.

We construct an instance of ABP as follows: Set $T := C$, $\mathcal{P} := [n+3]$, $\mathcal{R} := [n+3]$, $\mathcal{S}_r^1 := \{(r, 1)\}$ and $\mathcal{S}_r^2 := \emptyset$ for each $r \in \mathcal{R}$, and $t_p^s := 0$ for all $p \in \mathcal{P}$. Moreover, let

$$\sigma(p) := \begin{cases} (3+p, 1), & 1 \leq p \leq n, \\ (p-n, 1), & n+1 \leq p \leq n+3, \end{cases}$$

as well as

$$t_{p,r}^m := \begin{cases} t_{p,m_r}, & 1 \leq p \leq n, 1 \leq r \leq 3, \\ 0, & \text{otherwise.} \end{cases}$$

We claim that this instance admits a boarding sequence π with boarding completion time at most T if and only if the input instance of $F_3|perm,blocking|C_{\max}$ has a feasible solution ρ with makespan at most C . Indeed, π without the three ‘‘dummy’’ passengers $n+1, n+2, n+3$ is in one-to-one correspondence with ρ : Suppose we have a permutation flow shop solution (i.e., job sequence identical for all machines) $\rho = (\rho_1, \dots, \rho_n)$ with $C_{\max} \leq C$. Then, it is easily seen that in the ABP instance constructed above, each task j_m of a job j being processed on machine $m \in M$ with processing time $t_{j,m}$ corresponds exactly to passenger p , $1 \leq p \leq n$, moving within row r , $1 \leq r \leq 3$, for the same amount of time. The machine blockages in the flow shop problem are mirrored in row blockages (aisle interferences) by the corresponding ‘‘job’’ passengers passing the ‘‘machine’’ rows on their way to the assigned seats in the dummy rows $4 \leq r \leq n+3$; the dummy passengers seated in the first three rows have no influence on the completion time and merely serve the purpose of assigning one passenger to each seat (which is not necessary and may be omitted; we included it to include the case of full plane occupancy). Conversely, a boarding sequence π with completion time not exceeding T can be directly mapped to a desired flow shop solution sequence ρ by simply removing the dummy jobs. \square

The next results covers the case $t_{p,r}^m = c^m \forall p \in \mathcal{P}, r \in \mathcal{R}$ (with $c^m \in \mathbb{Q}_{\geq 0}$ a constant) and $t_p^s \in \mathbb{Q}_{\geq 0} \forall p \in \mathcal{P}$:

Theorem 2.2. *It is NP-complete in the strong sense to decide whether a given ABP instance admits a boarding sequence π that completes boarding by a given time $T \in \mathbb{Q}_{>0}$, even if restricted to instances with all-zero moving times, integer settle-in times, and integer T .*

Proof. To show hardness, we reduce from the 3-PARTITION problem, which is well-known to be strongly NP-complete, cf. [20]. In this problem, we are given a finite set $A := \{1, 2, \dots, 3m\} = [3m]$ ($m \in \mathbb{N}$), an integer $B \in \mathbb{N}_{\geq 3}$, and numbers $a_i \in \mathbb{N}$ for $i \in A$ such that $B/4 < a_i < B/2$ and $\sum_{i \in A} a_i = mB$, and we wish to decide whether A can be partitioned into m disjoint three-element sets A_1, A_2, \dots, A_m such that $\sum_{i \in A_j} a_i = B$ for every $j \in [m]$.

Given such an instance of 3-PARTITION, we construct an instance \mathcal{I} of the decision version of ABP with $T := mB$, four rows $\mathcal{R} := \{1, 2, 3, 4\}$, seats in the first and fourth rows only,

$$k_r^1 := \begin{cases} m, & r = 1, \ell = 1, \\ 3m, & r = 4, \ell = 1, \\ 0, & \text{otherwise,} \end{cases}$$

($k_r^2 = 0$ and $\mathcal{S}_r^2 = \emptyset$ for all $r \in \mathcal{R}$), passengers $\mathcal{P} := [4m]$, all-zero moving times $t_{pr}^m = 0$ for all $p \in \mathcal{P}$, $r \in \mathcal{R}$, settle-in times

$$t_p^s := \begin{cases} a_p, & 1 \leq p \leq 3m, \\ B, & 3m + 1 \leq p \leq 4m, \end{cases}$$

and seat assignments,

$$\sigma(p) := (r(p), s(p)) := \begin{cases} (4, p), & 1 \leq p \leq 3m, \\ (1, p - 3m), & 3m + 1 \leq p \leq 4m. \end{cases}$$

(Note that we could have included an arbitrary number of seats in the two middle rows along with suitably many dummy passengers with associated times that do not influence the reduction, but for simplicity, we simply defined these rows to be devoid of any seats.)

We show that the original 3-PARTITION instance is a “yes”-instance if and only if \mathcal{I} admits a boarding sequence that has completion time *exactly* $T (= mB)$. Note that every boarding sequence has completion time *at least* T , since m passengers with settle-in time B have their seats in row 1. (Thus, any sequence with completion time at most T will in fact have completion time equal to T .)

“ \Rightarrow ”: Let A_1, A_2, \dots, A_m be such that $A_1 \dot{\cup} A_2 \dot{\cup} \dots \dot{\cup} A_m = A$, and $|A_j| = 3$ as well as $\sum_{i \in A_j} a_i = B$ for all $j \in [m]$. Furthermore, let $A_j = \{i_1^j, i_2^j, i_3^j\}$ (arbitrarily ordered) for every $j \in [m]$. We set

$$\pi(p) := \begin{cases} 4(j-1) + k, & p = i_k^j \in A_j \subseteq [3m], \\ 4(p-3m), & 3m+1 \leq p \leq 4m. \end{cases}$$

For any $j \in [m]$, the group of four passengers $\pi^{-1}(4j-3), \dots, \pi^{-1}(4j)$ would need time exactly B from passenger $\pi^{-1}(4i-3)$ entering the plane until they have all settled in, if they were not blocked by any other passengers. (Note that, by construction, the two middle rows serve as a “buffer” for passengers waiting to get to the last row.) The passengers $\pi^{-1}(1), \dots, \pi^{-1}(4)$ are indeed not blocked and thus finish settling in at time B . Inductively, since every fourth passenger in the sequence π sits in the first row and thus blocks all subsequent passengers (from getting on board) until having settled in, each passenger group $\pi^{-1}(4j-3), \dots, \pi^{-1}(4j)$ with $2 \leq j \leq m$ has settled in at time jB , respectively. Consequently, the total boarding completion time is exactly mB .

“ \Leftarrow ”: Let π be a boarding sequence for \mathcal{I} with completion time $C_{\max}(\pi) = mB$. We may and do assume w.l.o.g. that $\pi(3m+1) < \pi(3m+2) < \dots < \pi(4m)$, and for notational convenience, we define $\mathcal{P}' := \{p \in \mathcal{P} : \pi(p) < \pi(3m+1)\}$ and $\mathcal{P}'_j := \{p \in \mathcal{P} : \pi(3m+j) < \pi(p) < \pi(3m+j+1)\}$ for $j \in [m-1]$. (Thus, \mathcal{P}' describes the passengers that, according to π , board before passenger $3m+1$, and \mathcal{P}'_j those that board before passenger $3m+j+1$ but after passenger $3m+j$.)

First, we note that the very last passenger to board the plane, $\pi^{-1}(4m)$, must be one of the passengers with seat in row 1—otherwise, the last passenger would be blocked by the last first-row passenger while the latter settles in, and the completion time would trivially be greater than mB (which coincides with the sum of settle-in times of all first-row passengers).

We proceed by contraposition: Assume that either $|\mathcal{P}'| \geq 4$ or $|\mathcal{P}'_j| \geq 4$ for at least one $j \in [m-1]$. Then, there exists an $\ell \in [m]$ such that passenger $3m+\ell$ can *begin* settling in no earlier than at time

$$S := \min\{a_p : p \in \mathcal{P}'\}$$

if $\ell = 1$, or

$$S_\ell := (\ell-1)B + \min\{a_p : p \in \mathcal{P}'_{\ell-1}\}$$

if $2 \leq \ell \leq m$, respectively. (The term $(\ell-1)B$ reflects the sum of settling-in times of all first-row passengers that have boarded before passenger $3m+\ell$.) Now, in case $\ell = 1$, passenger $3m+1$ completes settling in no earlier than at $S+B$, and since $S > 0$ and the remaining first-row passengers $3m+2, \dots, 4m$ each take B time to settle in, the total boarding time exceeds mB . This contradicts the fact that $C_{\max}(\pi) = mB$.

Similarly, in case $2 \leq \ell \leq m$, since $S_\ell > (\ell - 1)B$, it follows that $S_\ell + (m - \ell + 1)B > mB$. Because $S_\ell + (m - \ell + 1)B$ is a lower bound on the completion time (analogously to the case $\ell = 1$), this is again a contradiction to $C_{\max}(\pi) = mB$. Thus, it follows immediately that $|\mathcal{P}'| \leq 3$ and $|\mathcal{P}'_j| \leq 3$ for all $j \in [m - 1]$. Moreover, if any of these sets had cardinality strictly smaller than 3, a simple counting argument would yield the existence of another set with cardinality exceeding 3, which we have just ruled out. Thus, we have in fact shown that

$$|\mathcal{P}'| = |\mathcal{P}'_1| = \dots = |\mathcal{P}'_{m-1}| = 3.$$

Next, assume (again by contraposition) that either

$$B < T' := \sum_{p \in \mathcal{P}'} a_p$$

or, for at least one $j \in [m - 1]$,

$$B < T'' := \sum_{p \in \mathcal{P}'_j} a_p.$$

Then, there exists an $\ell \in [m - 1]$ such that passenger $3m + \ell + 1$ can begin settling in no earlier than at time

$$S := \ell B + \begin{cases} T' - B, & \text{if } \ell = 1, \\ T'' - B, & \text{if } 2 \leq \ell \leq m - 1, \end{cases}$$

and it holds that $S > \ell B$. By similar arguments as used before, $S + (m - \ell)B$ is an obvious lower bound on the completion time of π , but since $S + (m - \ell)B > mB$, we arrive at a contradiction to the presupposition $C_{\max}(\pi) = mB$. Thus, the settle-in times associated with any three-element set $\mathcal{P}', \mathcal{P}'_1, \dots, \mathcal{P}'_{m-1}$ sum to at most B ; in fact, since $\sum_{p \in [3m]} a_p = mB$, these sums are all exactly equal to B .

This shows that we have identified a “yes”-certificate of the original 3-PARTITION instance by setting

$$A_j := \mathcal{P}'_j \text{ for } 1 \leq j \leq m - 1, \quad A_m := \mathcal{P}',$$

which completes the proof of NP-hardness.

It remains to note that the reduction is clearly polynomial (in particular, retaining boundedness in the problem dimension of all occurring numbers as well as their respective encoding lengths and thus preserving the “in the strong sense” assertion of NP-hardness), and that containment in NP can be obtained analogously to the proof of Theorem 2.1. \square

The above two core intractability results for the ABP implicitly (by virtue of the respective reductions) assume that the plane is fully booked, i.e., that $P := |\mathcal{P}| = |\mathcal{S}|$. It is, however, easy to see that NP-hardness persists in the case that some seats are left empty:

Corollary 2.1. *The strong NP-hardness results of Theorems 2.1 and 2.2 remain valid in their respective settings even if $P < |\mathcal{S}|$, i.e., the aircraft is not fully occupied.*

Proof. Since the seat assignments σ are part of an ABP instance, we may simply extend the respective reductions in the NP-hardness proofs by, e.g., appending a row at the back of the aircraft containing at least one seat to which no passenger will be assigned in the constructed instance. (In the instance from the proof of Theorem 2.2, we could also place unbooked seats in the two middle rows.) Clearly, this has no further implications w.r.t. the rest of those proofs. \square

Remark 2.1. Recall that NP-hardness *in the strong sense* implies that, unless $P = NP$, not only can there not be a polynomial-time exact solution algorithm, but also neither a pseudo-polynomial exact algorithm nor a fully polynomial-time approximation scheme (FPTAS), see, e.g., [20] for details. In particular, the “strong sense” assertion may be viewed as evidence that a problem’s intractability is not due to possible ill-conditioning of problem data, but inherently combinatorial in nature.

2.2 Approximability Results

In view of the fact that in practice, one will have to work with estimates of the passengers' moving and settle-in times, the following result justifies the use of *pessimistic* such estimates to retain a certain planning robustness:

Proposition 2.1. *For an optimal ABP solution π with completion time $C_{\max}(\pi)$, should some t^m - or t^s -values actually be smaller than the data used to obtain π , the resulting actual boarding completion time of π still is at most $C_{\max}(\pi)$.*

Proof. This can easily be seen by considering the boarding sequence π and propagating the changes in t^m - and/or t^s -values: Decreasing any such value either yields no change or leads to earlier termination of some moving and settling-in operations in the given sequence, but never incurs delays that were not present before. Thus, the previously optimal sequence π is a feasible (but not necessarily optimal) solution of the modified ABP instance, which has completion time $C'_{\max} \leq C_{\max}(\pi)$. \square

Moreover, it can make sense to consider certain simplifications of ABP (more precisely, of its data model), in particular in combination with Proposition 2.1. In the most simple case, we may somewhat neglect passenger differences and simply assume identical moving times and identical settle-in times for all passengers (i.e., $t_{p,r}^m = c^m \forall p \in \mathcal{P}, r \in \mathcal{R}$ and $t_p^s = c^s \forall p \in \mathcal{P}$, with constants $c^m, c^s \in \mathbb{Q}_{\geq 0}$). In that case, and assuming for simplicity that each row has the same number of seats (i.e., $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$) and that the aircraft is booked to capacity (every seat is occupied), ABP becomes solvable in polynomial time by the simple boarding strategy known as *outside-in* (cf., e.g., [5]), defined below. Throughout, we will focus on outside-in boarding and a novel variant of it, as various simulation studies have already demonstrated outside-in to be (one of) the best boarding heuristics, cf. Section 1.1.

Definition 2.1 (outside-in strategy). *Let the passengers board the plane in groups of size R (one per row per group), ordered first by increasing seat numbers (identifying the groups) and then by decreasing row number (within each group). Assuming w.l.o.g. that $k_r^1 \geq k_r^2 \forall r \in \mathcal{R}$, the position in the outside-in boarding sequence of a passenger $p \in \mathcal{P}$ can be directly expressed as*

$$\pi(p) = \begin{cases} (2s(p) - 1)R + 1 - r(p), & \text{if } s(p) \leq k_r^2, \\ (k_r^2 + s(p))R + 1 - r(p), & \text{if } k_r^2 < s(p) \leq k_r^1, \\ (k - s(p) + 1)R + 1 - r(p), & \text{if } s(p) \geq k_r^1 + 1. \end{cases} \quad (1)$$

(Recall that in our notation, seats are numbered $1, 2, \dots, k_r^1$ from window to aisle on one side, and then continue from aisle to window as $k_r^1 + 1, k_r^1 + 2, \dots, k_r^1 + k_r^2$ on the other side, in any row r ; also, here, $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k$ implies $k_r^2 = k - k_r^1$.)

Theorem 2.3. *A minimum-boarding-time solution to ABP instances with $P = |\mathcal{S}|$, $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, $t_p^s = c^s \in \mathbb{Q}_{\geq 0}$ for all $p \in \mathcal{P}$, and $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$ can be computed in time $\mathcal{O}(P)$ by the outside-in strategy.*

For the proof, we make use of the following two auxiliary results providing upper and lower bounds on the ABP objective.

Lemma 2.1. *For ABP instances with $P = |\mathcal{S}|$ and $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, every boarding strategy producing a sequence π such that*

$$\mathcal{P}(R - i + 1) = \{\pi^{-1}(i), \pi^{-1}(i + R), \dots, \pi^{-1}(i + (k - 1)R)\} \quad \forall i \in \mathcal{R} \quad (2)$$

(i.e., *generalized outside-in with arbitrary passengers per row per group, not necessarily according to seating order from window to aisle*) yields a boarding completion time $C_{\max}(\pi)$ of at most

$$\sum_{i=0}^{k-1} \max_{j \in \mathcal{R}} \left\{ \sum_{r=1}^{r(\pi^{-1}(j+iR))-1} \left(\max_{\ell \in \{0, \dots, R-r-1\}} t_{\pi^{-1}(j+iR-\ell), r+\ell}^m \right) + t_{\pi^{-1}(j+iR)}^s \right\}. \quad (3)$$

Proof. The first passenger $p_1 := \pi^{-1}(1)$ in the boarding sequence needs time $\sum_{r=1}^{R-1} t_{p_1,r}^m + t_{p_1}^s$ to finish boarding. The second passenger $p_2 := \pi^{-1}(2)$ can pass a row $r \leq R - 2$ either directly in time $t_{p_2,r}^m$, or in time at most $t_{p_1,r+1}^m$ if he/she has to wait (in r) while the next row $r + 1$ is still blocked by the preceding passenger p_1 currently passing it. Pursuing this idea, and using (2), we can upper-bound the time that passenger $\pi^{-1}(i)$ in position $i \leq R$ needs to finish boarding as follows:

$$\sum_{r=1}^{r(\pi^{-1}(i))-1} \max \left\{ t_{\pi^{-1}(i),r}^m, t_{\pi^{-1}(i-1),r+1}^m, \dots, t_{\pi^{-1}(1),R-1}^m \right\} + t_{\pi^{-1}(i)}^s.$$

With the obvious indexing adjustments, this bound can be adapted straightforwardly to upper bounds for the time that passengers of the other “groups” of R passengers (of which there are k) needs for boarding. Clearly, the next group of R passengers can begin boarding *at the latest* when *all* passengers of the previous group have settled in, so the sum of the respective upper bounds over all k groups gives an upper bound on the total boarding time. Combining this summation with maximization of the passenger-specific boarding completion bound within each group yields the claimed upper bound (3) on $C_{\max}(\pi)$. \square

Lemma 2.2. *For arbitrary ABP instances, lower bounds on the minimum boarding time are given by*

$$\max_{p \in \mathcal{P}} \left\{ \sum_{r=1}^{r(p)-1} t_{p,r}^m + t_p^s \right\}, \quad (4)$$

$$\max_{r \in \mathcal{R}} \left\{ \sum_{\substack{p \in \mathcal{P}: \\ r(p) \geq r+1}} t_{p,r}^m + \sum_{p \in \mathcal{P}(r)} t_p^s + \min_{\substack{p \in \mathcal{P}: \\ r(p) \geq r}} \sum_{r'=1}^{r-1} t_{p,r'}^m \right\}, \quad (5)$$

$$\frac{1}{\min\{R, P\}} \sum_{p \in \mathcal{P}} \left(t_p^s + \sum_{r=1}^{r(p)-1} t_{p,r}^m \right). \quad (6)$$

Proof. Each passenger p has to pass rows $1, \dots, r(p) - 1$ before eventually settling in at row $r(p)$. If a passenger is not blocked along the way, he/she needs time exactly $\sum_{r=1}^{r(p)-1} t_{p,r}^m + t_p^s$ to complete boarding. Thus, maximization over all passengers yields lower bound (4).

Passengers p with seat in row $r(p) \geq r + 1$ have to pass row r and all passengers with seat in row r have to settle in at r . None of these events can happen at the same time. Thus, row r is blocked for at least $\sum_{p:r(p) \geq r+1} t_{p,r}^m + \sum_{p \in \mathcal{P}(r)} t_p^s$. Additional time is required for the first passenger who will block row r to pass rows $1, \dots, r - 1$ (on the way to row r), which can be lower bounded by the minimum of the associated moving time sums. Finally, maximizing over all rows gives lower bound (5).

To see (6), note that in a best case, the sum of all settle-in and moving times is distributed evenly over the maximal number of actions that can take place in parallel. Since this number of actions is obviously bounded from above by $\min\{R, P\}$, (6) indeed provides a lower bound on the minimum achievable boarding time. \square

Proof (of Theorem 2.3). With $t_{p,r}^m = c^m$ and $t_p^s = c^s$ for all p, r , it can easily be seen that the terms in the maximum in the lower bound (5) all reduce to $(P - k)c^m + kc^s$, which hence is the value of this boarding time lower bound itself. Similarly, the upper bound (3) (which is applicable as the outside-in strategy clearly satisfies the requirement (2) from Lemma 2.1) reduces to

$$\sum_{i=0}^{k-1} \max_{j \in \mathcal{R}} \left\{ c^s + \sum_{r=1}^{r(\pi^{-1}(j+iR))-1} c^m \right\} = kc^s + \sum_{i=0}^{k-1} \sum_{r=1}^{R-1} c^m = kc^s + k(R-1)c^m.$$

(To see the first equality, recall that by (2), every group of R passengers who board consecutively includes one person with seat in the last row, so for every $i = 0, 1, \dots, k - 1$, the maximum is attained for j such that

$r(\pi^{-1}(j+iR)) = R$.) Since $P = |\mathcal{P}| = |\mathcal{S}| = kR$, this upper bound coincides with the lower bound (5), which immediately shows optimality of the outside-in boarding sequence. For the running time bound, note that we can compute the position of any passenger $p \in \mathcal{P}$ in the outside-in sequence directly with Equation (1). Therefore, iterating over all passengers once is sufficient to obtain the full boarding sequence. \square

Remark 2.2. It should be noted that in Theorem 2.3, we explicitly assume full plane occupancy, i.e., $P = |\mathcal{S}|$. If some seats are unoccupied, we currently do not have a quality guarantee for outside-in boarding. On the other hand, it is also not implied by Theorems 2.1 or 2.2 that ABP remains NP-hard in the setting of Theorem 2.3 (with $P < |\mathcal{S}|$ instead of $P = |\mathcal{S}|$): While the intractability results can be extended straightforwardly to include seat vacancies, cf. Corollary 2.1, the data assumptions $t_{p,r}^m = c^m$, $t_p^s = c^s$ of Theorem 2.3 are still more restrictive and might therefore, in principle, yield easier special cases. Similarly, it is unclear if it might help to use dummy passengers with all-zero moving and settle-in times to occupy empty seats (to restore the assumption $P = |\mathcal{S}|$), because such dummy passengers would still be able to block others just like real passengers and can therefore induce unforeseen waiting times.

It is worth mentioning that Lemmas 2.1 and 2.2 provide general instance-dependent approximation ratios for boarding sequences obeying (2), at least for fully occupied airplanes:

Proposition 2.2. *For ABP instances with $P = |\mathcal{S}|$ and $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, the boarding time of any boarding sequence π with property (2) is at most a factor β worse than the optimal boarding time, where*

$$\beta := \frac{(3)}{\max\{(4), (5), (6)\}}. \quad (7)$$

Proof. Let $C_{\max}(\pi)$ and C_{\max}^* be the boarding completion time of sequence π and the minimum boarding time, respectively. By Lemmas 2.1 and 2.2, it holds that

$$C_{\max}(\pi) \leq (3) \quad \text{and} \quad C_{\max}^* \geq \max\{(4), (5), (6)\},$$

whence $C_{\max}(\pi)/C_{\max}^* \leq \beta$, which proves the claim. \square

The main drawback of the general approximation ratio (7) is its dependence on instance-specific time values $t_{p,r}^m$ and t_p^s . This is unfortunate for two reasons: First, the ratio can conceivably become quite large when an instance contains some settle-in time that is much larger than the moving times—a situation that seems very natural, as walking past some row should take no more than a few seconds but stowing away carry-on luggage and taking a seat may take up to several minutes. Moreover, passenger time data changes for every instance (and is not even known exactly) while aircraft cabin parameters (k and R) are shared by whole fleets of airplanes. This makes it desirable to have approximation bounds that only depend on these instance-size parameters, thus enabling one to judge the quality of a boarding strategy regardless of the actual instance-specific passenger data. In the remainder of this section, we will demonstrate that such “almost constant-factor” approximation results can indeed sometimes be obtained.

For the first such result, we consider generalizing w.r.t. the moving or the settle-in times (i.e., relaxing from identical constant moving times for all rows and passengers to individual moving times, or from constant to arbitrary settle-in times, respectively). Then, the outside-in strategy is no longer necessarily optimal, but nevertheless, for a fixed airplane model, it still provides a constant-factor approximation:

Theorem 2.4. *For ABP instances with $P = |\mathcal{S}|$, $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, and either $t_{p,r}^m = c_p^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$ and $t_p^s = c^s \in \mathbb{Q}_{\geq 0}$ for all $p \in \mathcal{P}$, or $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$ and $t_p^s \in \mathbb{Q}_{\geq 0}$ (arbitrary) for all $p \in \mathcal{P}$, the outside-in boarding strategy is a k -approximation algorithm.*

Proof. First consider the case $t_{p,r}^m = c_p^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$, $t_p^s = c^s \in \mathbb{Q}_{\geq 0}$ for all $p \in \mathcal{P}$. In this setting, the upper bound (3) on the completion time of the outside-in boarding strategy collapses to

$$kc^s + \sum_{i=0}^{k-1} \max_{\substack{p \in \mathcal{P}: \\ iR+1 \leq \pi(p) \leq (i+1)R}} (r(p) - 1)c_p^m,$$

which can be further estimated from above by $k(c^s + \max_{p \in \mathcal{P}}(r(p) - 1)c_p^m)$. The latter corresponds to exactly k times the lower bound (4), which shows (analogous to (7) and its proof) that the outside-in boarding time cannot be worse than k times the minimum boarding time.

Now suppose $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$, $t_p^s \in \mathbb{Q}_{\geq 0}$ (arbitrary) for all $p \in \mathcal{P}$. In this case, similarly to what we just saw, the upper bound (3) can be further bounded from above by $k \max_{p \in \mathcal{P}}\{(r(p) - 1)c^m + t_p^s\}$, which is again equal to k times the lower bound (4). \square

Remark 2.3. For the first case considered in Theorem 2.4, i.e., for $t_{p,r}^m = c_p^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$ and $t_p^s = c^s \in \mathbb{Q}_{\geq 0}$ for all $p \in \mathcal{P}$, one can also obtain the approximation ratio $1 + (\max_p c_p^m - \min_p c_p^m) / \min_p c_p^m$ by simplifying (7) (further upper bounding (3) and lower bounding (6)); we omit the proof. While this bound depends on instance time data rather than the size parameters, it may nevertheless be expected to be fairly small, possibly even better than k when k is relatively large. Moreover, if in fact all moving times are identical for all passengers, this results provides an alternative proof of optimality (cf. Theorem 2.3).

As with Theorem 2.3, it is unclear whether the above approximation guarantees continue to hold if the plane is not fully occupied, cf. Remark 2.2. Moreover, notwithstanding the general approximation bound (7), the next result exhibits that outside-in boarding could lead to rather bad results in even more general settings, and regardless of whether all seats are taken.

Proposition 2.3. *For every $k \in \mathbb{N}_{\geq 2}$ and $\varepsilon \in (0, k(k-1)/(3k-2)]$, there exists an ABP instance with $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k$ for all $r \in \mathcal{R}$, and $t_{p,r}^m = c_p^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$, such that the outside-in boarding time is at least a factor $(2k - \varepsilon)$ worse than the optimal boarding time.*

Proof. First suppose $P = |\mathcal{S}|$. Let $k \in \mathbb{N}_{\geq 2}$ and $0 < \varepsilon \leq k(k-1)/(3k-2)$, let $c^m, c^s \in \mathbb{Q}_{>0}$ be constants to be defined later, and consider the following ABP instance with $R \geq 2k$ (R will also be specified later): For ease of presentation, assume w.l.o.g. that all seats are on the same side of the aisle ($(r(p), s(p)) \in \mathcal{S}_{r(p)}^1 \forall p \in \mathcal{P}$). Only the passengers seated in the last row have positive moving times, namely $t_{p,r}^m = c^m$ for all $(p, r) \in \mathcal{P}(R) \times \mathcal{R}$, only k passengers—one per row for the first k rows—have positive settle-in times, namely $t_p^s = c^s$ for $p \in \{\sigma^{-1}(i, i) : i \in [k]\}$ (i.e., the window-seat passenger in the first row, the next-to-the-window passenger in the second row, “diagonally” continued to the aisle-seat passenger in row k), and all other t^m - and t^s -values are zero.

Due to aisle interferences (passengers trailing behind the last-row passengers), a passenger with positive settle-in time in one of the first k rows only ever reaches his/her target row once the preceding last-row passenger has reached row R . Thus, it is easily verified that for this instance, the boarding time of the outside-in strategy is exactly $k(R-1)c^m + kc^s$. On the other hand, if all passengers from the last row were to board first, followed by the passengers with nonzero settle-in times from the first k rows in decreasing order of their row numbers (back-to-front fashion), and finally all other passengers in arbitrary order, then the boarding time would be only be $\max\{(R-1)c^m + (k-1)c^m, (2k-1)c^m + c^s\}$. (The second summand in the first term amounts to the cumulative waiting times of the last-row passengers, and the first summand of the second term to those of the passengers with nonzero settle-in times.)

As the latter gives an upper bound on the actual optimal boarding time, the completion time of the outside-in boarding strategy is at least

$$\frac{k(R-1)c^m + kc^s}{\max\{(R-1)c^m + (k-1)c^m, (2k-1)c^m + c^s\}} \quad (8)$$

times the optimal completion time. Let $R := \lceil k(k-1)/\varepsilon \rceil - k + 2$, $c^m := 1$, and $c^s := (R - k - 1)$. Then, $(R-1)c^m + (k-1)c^m = c^s$ and term (8) collapses to

$$k + \frac{k(R-1)}{(R-1) + (k-1)} = 2k - \frac{k(k-1)}{\lceil k(k-1)/\varepsilon \rceil} \geq 2k - \varepsilon.$$

Observing that to extend the result to the case $P < |\mathcal{S}|$, we may simply add, e.g., an additional $(R+1)$ -th row with unoccupied seats at the rear-end of the plane (either omitting them from the outside-in sequence, or including dummy passengers with all-zero time data to be “seated” there) completes the proof. \square

The following result generalizes Proposition 2.3 to all (non-random) strategies that do not explicitly consider settle-in times. In particular, all the classical boarding strategies mentioned in Section 1 (back-to-front, reverse pyramid, etc.) rely solely on the seat assignments and are thus covered.

Theorem 2.5. *For every $R, k \geq 2$, there exists an ABP instance for which any deterministic boarding strategy that disregards individual settle-in times leads to a boarding time at least twice as long as the optimal boarding time.*

Proof. First, suppose that the given strategy π boards passengers in back-to-front fashion, i.e., for any $(p_1, p_2) \in \mathcal{P}^2$ with $r(p_1) < r(p_2)$, it holds that $\pi(p_1) > \pi(p_2)$. In particular, this implies that all passengers with seats in row R board first. Suppose that $t_{\sigma^{-1}(\pi^{-1}(1), R)}^s = T = t_{\sigma^{-1}(\pi^{-1}(P), r(\pi^{-1}(P)))}^s$ (i.e., the very first and the very last passenger according to the considered boarding strategy have settle-in time T , for an arbitrary constant $T > 0$), and that all other t^m - and t^s -values are zero. Then, the strategy results in boarding time $2T$, since row $R - 1$ is blocked by another passenger $p \in \mathcal{P}(R)$ while passenger $\pi^{-1}(1)$ is settling in, so (even if $R = k = 2$) the last passenger necessarily has to wait T time units before possibly starting to settle-in, which then again takes T time. Obviously, an optimal boarding strategy would let the two passengers with nonzero settle-in times settle in simultaneously and thus has boarding time T .

Now assume there are two passengers p_1, p_2 with $r(p_1) < r(p_2)$ but $\pi(p_1) < \pi(p_2)$, i.e., the considered strategy lets passenger p_1 board before passenger p_2 , but p_1 has his/her seat in an earlier row than p_2 . Suppose that $t_{p_1}^s = t_{p_2}^s = T$ ($T > 0$ arbitrary), and that all other t^m - and t^s -values are zero. Then, the strategy yields boarding time $2T$, since passenger p_2 is blocked by passenger p_1 (aisle interference) and the two cannot settle in simultaneously. Again, an optimal boarding strategy clearly yields boarding time T .

It remains to note that in these constructions, one can obviously always find some passenger p (with $t_{p,r}^m = t_p^s = 0$ for all r) whose removal does not change the described aisle interference behavior, so the arguments go through for both plane occupancy cases $P = |\mathcal{S}|$ and $P < |\mathcal{S}|$. \square

Let us now introduce a modification of the outside-in boarding strategy:

Definition 2.2 (max-settle-row strategy). *Let the passengers board the plane in groups of size R , each group with exactly one passenger with seat in row r for all $r \in \mathcal{R}$. The i -th group contains a passenger p of each row with the i -th longest settle-in time of his/her row, breaking ties in outside-in fashion (seats furthest from the aisle first), and is ordered by decreasing row number.*

Besides generalizing outside-in boarding, the max-settle-row strategy is also related to the method proposed in [8], which performs seat assignment based on number of carry-on items, targeting fast boarding with the Steffen method [7] applied afterwards. However, although the number of hand-luggage pieces is naturally correlated with the settle-in times, there is no direct correspondence, and furthermore, here, the seat assignments are fixed a priori.

We first note that when settle-in times are assumed to be identical for all passengers, max-settle-row reduces to outside-in and thus immediately inherits the solution quality guarantees from Theorems 2.3 and 2.4:

Corollary 2.2. *If $t_p^s = c^s \in \mathbb{Q}_{\geq 0}$ for all $p \in \mathcal{P}$, the max-settle-row strategy reduces to outside-in boarding. Consequently, in this case, for ABP instances with $P = |\mathcal{S}|$ and $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, max-settle-row gives an optimal solution if $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$, and is a k -approximation scheme if $t_{p,r}^m = c_p^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$.*

As it turns out, the max-settle-row strategy gives an improved approximation result for the case in which we have constant moving times but arbitrary settle-in times (and a full aircraft).

Theorem 2.6. *For ABP instances with $P = |\mathcal{S}|$, $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, and $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p, r) \in \mathcal{P} \times \mathcal{R}$, the max-settle-row strategy is an $(1 + H_k(k-1)/k)$ -approximation algorithm, where $H_k := \sum_{i=1}^k (1/i)$.*

Proof. Let $t_p^s \in \mathbb{Q}_{\geq 0}$ be arbitrary, and let π be the max-settle-row boarding sequence. To upper-bound the approximation quality, consider the quotient of the upper bound (3) and the lower bound (5), which, in the present setting, reads

$$\frac{\sum_{i=0}^{k-1} \max_{\substack{p \in \mathcal{P}: \\ iR+1 \leq \pi(p) \leq (i+1)R}} ((r(p)-1)c^m + t_p^s)}{\max_{r \in \mathcal{R}} \left((k(R-r) + (r-1))c^m + \sum_{p \in \mathcal{P}(r)} t_p^s \right)} =: \frac{U}{L}. \quad (9)$$

For the remainder of the proof, we may assume that $L \neq 0$, because otherwise, it would necessarily hold that $c^m = 0$ and $t_p^s = 0$ for all $p \in \mathcal{P}$, so any sequence π would in fact be optimal.

Since the passengers from each row are sequenced in non-increasing order of their settle-in times, the settle-in time of each passenger in boarding group i is bounded from above by $(1/i) \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s$. Hence, using that $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in \mathcal{P} \times \mathcal{R}$, we obtain

$$U \leq k(R-1)c^m + H_k \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s.$$

Furthermore, the denominator of (9) can be bounded from below by

$$L \geq \max \left\{ k(R-1)c^m + \min_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s, (R-1)c^m + \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s \right\}.$$

Consequently, we have

$$\begin{aligned} \frac{U}{L} &\leq \frac{k(R-1)c^m + H_k \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s \left(+ H_k(R-1)c^m - H_k(R-1)c^m \right)}{\max \left\{ k(R-1)c^m + \min_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s, (R-1)c^m + \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s \right\}} \\ &\leq \frac{H_k(R-1)c^m + H_k \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s}{(R-1)c^m + \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s} + \frac{(k-H_k)(R-1)c^m}{k(R-1)c^m + \min_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s} \\ &\leq H_k + \frac{(k-H_k)(R-1)c^m}{k(R-1)c^m} = H_k + 1 - \frac{H_k}{k}, \end{aligned}$$

which establishes the claimed approximation ratio $1 + \frac{k-1}{k}H_k$. \square

Note that $1 + (k-1)H_k/k < k$ for all $k \geq 2$, so Theorem 2.6 (max-settle-row) indeed gives a stronger guarantee than Theorem 2.4 (outside-in).

Interestingly, in the setting of these theorems, the asymptotic behavior of both max-settle-row and outside-in boarding reflects the growing influence of the moving times on the total boarding time in case of longer airplane cabins, eventually outweighing the settle-in time contributions entirely:

Proposition 2.4. *For ABP instances with $P = |\mathcal{S}|$, $|\mathcal{S}_r^1| + |\mathcal{S}_r^2| = k \in \mathbb{N}$ for all $r \in \mathcal{R}$, $t_{p,r}^m = c^m \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in \mathcal{P} \times \mathcal{R}$, and (arbitrary) $t_p^s \in (0, T]$, $T < \infty$, both the outside-in strategy and the max-settle-row strategy are asymptotically optimal. More precisely, as $R \rightarrow \infty$, the respective boarding times of their solutions tend to the optimal boarding time.*

Proof. From the proof of Theorem 2.4, we know that

$$k \max_{p \in \mathcal{P}} \{(r(p)-1)c^m + t_p^s\} \leq k(R-1)c^m + k \max_{p \in \mathcal{P}} t_p^s \leq k(R-1)c^m + kT \quad (10)$$

is an upper bound on the outside-in boarding time. Since Lemma 2.1 also applies to max-settle-row, the bound (10) is actually valid for both boarding strategies. The same can be said about the lower bound from the proof of Theorem 2.6,

$$\max \left\{ k(R-1)c^m + \min_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s, (R-1)c^m + \max_{r \in \mathcal{R}} \sum_{p \in \mathcal{P}(r)} t_p^s \right\} \geq k(R-1)c^m. \quad (11)$$

Thus, for both outside-in and max-settle-row boarding, the approximation ratio is upper-bounded by

$$\frac{k(R-1)c^m + kT}{k(R-1)c^m} = 1 + \frac{T}{(R-1)c^m} \xrightarrow{R \rightarrow \infty} 1,$$

which completes the proof. \square

The discrepancies between the positive and negative results presented in this section exhibit possibly significant potential for improvements to be gained by solving ABP to optimality. Although Theorems 2.1 and 2.2 do not directly imply NP-hardness for all the restricted ABP versions (cf. Table 1 and Remark 2.2), there is also no immediate strategy to exactly solve these problems in polynomial time (nor to obtain results with *guaranteed* approximation bounds in all cases). Therefore, in the following section, we turn to general (mixed-)integer programming approaches to optimally solve ABP.

3 Exact (MIP) Approaches

Let us now consider exact approaches for solving ABP. We begin with compact linear mixed-integer programming (MIP) formulations in Section 3.1 and present a linear time-discretized (layered) multicommodity network flow model later in Section 3.2. In this section, we state and explain the MIP models in what we felt are their best-interpretable forms, and defer the discussion of respective reductions (such as variable eliminations) and implementation details to Section 4.1.

3.1 Compact MIP Formulations

For our compact linear mixed-integer programs, the following variables are convenient to model a linear ordering π [21] and the associated boarding completion time:

- Let $w_{p,i} \in \{0,1\}$ be the binary *set assignment* variable representing whether a passenger $p \in \mathcal{P}$ is among the first $i \in [P]$ passengers who board, i.e., $w_{p,i} = 1$ if $\pi(p) \leq i$ and $w_{p,i} = 0$ otherwise.
- Let $x_{p,i} \in \{0,1\}$ be the binary *position* variable representing whether a passenger $p \in \mathcal{P}$ is placed at position $i \in [P]$ in the linear order (boarding sequence) π , i.e., $x_{p,i} = 1$ if $\pi(p) = i$ and $x_{p,i} = 0$ otherwise.
- Let $y_{p_1,p_2} \in \{0,1\}$ be the binary *predecessor* variable representing whether passenger $p_1 \in \mathcal{P}$ is a (not necessarily direct) predecessor of $p_2 \in \mathcal{P} \setminus \{p_1\}$ in the boarding order π , i.e., $y_{p_1,p_2} = 1$ if $\pi(p_1) \leq \pi(p_2) - 1$ and $y_{p_1,p_2} = 0$ otherwise.
- Let $t_{p,r}^A \in \mathbb{Q}_{\geq 0}$ be the *arriving time* variable representing the time at which passenger $p \in \mathcal{P}$ arrives at row $r \in \mathcal{R}$, and let $t_{p,r}^F$ be the *finishing time* variable representing the time at which passenger $p \in \mathcal{P}$ finishes his/her action at row $r \in \mathcal{R}$.
- Analogously, let $t_{i,r}^A \in \mathbb{Q}_{\geq 0}$ be the *arriving time* variable representing the time at which passenger $\pi^{-1}(i)$, $i \in [P]$, arrives at row $r \in \mathcal{R}$, and let $t_{i,r}^F$ be the *finishing time* variable representing the time at which passenger $\pi^{-1}(i)$, $i \in [P]$, finishes his/her action at row $r \in \mathcal{R}$.
- Finally, we introduce the *completion time* variable $C_{\max} \in \mathbb{Q}_{\geq 0}$ representing the boarding completion time of the computed linear order/boarding sequence.

Regarding the input ABP time data, it is helpful to introduce further notation as well: For every passenger $p \in \mathcal{P}$ and each row $r \in \mathcal{R}$, we are given times $\tau_{p,r} \in \mathbb{Q}_{\geq 0}$ that passenger p *must* spend at row r either moving past it ($\tau_{p,r} := t_{p,r}^m$ for $r \leq r(p) - 1$) or settling in in it ($\tau_{p,r} := t_p^s$ for $r = r(p)$). For rows r that a passenger p needs not to visit at all, i.e., those behind his/her assigned seat, we may and do assume that $\tau_{p,r} = 0$. Similarly, we neglect what happens before the first passenger has arrived at the first row; by minimizing the final completion time, it is thus automatically ensured that the first passenger will start with the first row action at time zero.

3.1.1 Predecessor Assignment Formulation (MIP_{PR})

Our first MIP formulation for the ABP, (12)–(19) (summarily referred to as MIP_{PR}), is based on the predecessor variables $y_{p_1,p_2} \in \{0,1\}$ for every pair of distinct passengers $p_1, p_2 \in \mathcal{P}$ ($p_1 \neq p_2$). Constraint (13) establishes a predecessor relation between every such pair, and (14) ensures transitivity of these relations. Constraint (15) ensures that all time/row-requirements are met while allowing for more time to be spent at each row in case passengers have to wait behind others who block passage or row access. Incorporating the assignment decisions, we can directly model the objective (12) (minimizing total boarding time) via (18), the requirement that rows are traversed consecutively by each passenger via (16) (also ensuring that no times remain uncaptured by equating the finishing times for any one row with the respective arrival times for the subsequent one), and that passengers cannot be overtaken until they are seated via (17).

$$\min C_{\max} \tag{12}$$

$$\text{s.t. } y_{p_1,p_2} + y_{p_2,p_1} = 1 \quad \forall (p_1, p_2) \in \mathcal{P}_{\neq}^2 \tag{13}$$

$$y_{p_1,p_2} + y_{p_2,p_3} - y_{p_1,p_3} \leq 1 \quad \forall (p_1, p_2, p_3) \in \mathcal{P}_{\neq}^3 \tag{14}$$

$$t_{p,r}^F - t_{p,r}^A \geq \tau_{p,r} \quad \forall p \in \mathcal{P}, r \in [r(p)] \tag{15}$$

$$t_{p,r}^F - t_{p,r+1}^A = 0 \quad \forall p \in \mathcal{P}, r \in [r(p) - 1] \tag{16}$$

$$t_{p_2,r}^A + (1 - y_{p_1,p_2})M_r \geq t_{p_1,r}^F \quad \forall (p_1, p_2) \in \mathcal{P}_{\neq}^2, r \in [\min\{r(p_1), r(p_2)\}] \tag{17}$$

$$C_{\max} \geq t_{p,r(p)}^F \quad \forall p \in \mathcal{P} \tag{18}$$

$$y \in \{0,1\}^{\mathcal{P}_{\neq}^2}; t^A, t^F \in \mathbb{Q}_{\geq 0}^{\mathcal{P} \times \mathcal{R}}; C_{\max} \in \mathbb{Q}_{\geq 0}, \tag{19}$$

where we abbreviated $\mathcal{P}_{\neq}^{\ell} := \{(p_1, \dots, p_{\ell}) \in \mathcal{P}^{\ell} : \{p_1, p_2, \dots, p_{\ell}\} \text{ pairwise distinct}\}$. The big-M constant in (17) must be sufficiently large to effectively “turn off” the constraint if $y_{p_1,p_2} = 0$; it can be chosen, e.g., as $M_r = \sum_{p \in \mathcal{P}, r \in \mathcal{R}} \tau_{p,r}$.

3.1.2 Set Assignment Formulation (MIP_{SA})

The second MIP (20)–(25), MIP_{SA} for short, additionally employs set assignment variables $w \in \{0,1\}^{\mathcal{P} \times [P]}$. Constraint (21) ensures that for each $i \in [P]$, exactly i passengers are assigned to the set of the first i passengers, and (22) establishes (upwards-) monotonicity for each passenger (i.e., if p is among the first i passengers who board, then he/she is also among the first j passengers, for any $j > i$). Constraint (23) couples the set assignment with the predecessor variables $y \in \{0,1\}^{\mathcal{P}_{\neq}^2}$. The objective (20) and the constraints listed in (24) are adopted from the previous model MIP_{PR}, where the big-M constant in (17) can again be chosen as described earlier in Section 3.1.1.

$$\min C_{\max} \tag{20}$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} w_{p,i} = i \quad \forall i \in [P] \tag{21}$$

$$w_{p,i} \leq w_{p,i+1} \quad \forall p \in \mathcal{P}, i \in [P] \tag{22}$$

$$y_{p_1,p_2} \geq w_{p_2,i} - w_{p_1,i} \quad \forall i \in [P], (p_1, p_2) \in \mathcal{P}_{\neq}^2 \tag{23}$$

$$(13), (15), (16), (17), (18) \quad (24)$$

$$w \in \{0, 1\}^{\mathcal{P} \times [P]}; y \in \{0, 1\}^{\mathcal{P}^2}; t^A, t^F \in \mathbb{Q}_{\geq 0}^{\mathcal{P} \times \mathcal{R}}; C_{\max} \in \mathbb{Q}_{\geq 0} \quad (25)$$

3.1.3 Position Assignment Formulations (MIP_{DPO}, MIP_{IPO})

The next two MIPs both use position assignment variables $x \in \{0, 1\}^{\mathcal{P} \times [P]}$. The first formulation, (26)–(31) (MIP_{DPO} for short), models the objective directly via the passengers, whereas the second one, (32)–(40) (abbreviated MIP_{IPO}) does so implicitly via the boarding sequence/index ordering.

Direct Formulation (MIP_{DPO}): Naturally, every passenger has to be assigned to precisely one slot in the sequence, yielding constraints (27) and (28). Constraint (29) couples the position assignments with the predecessor variables $y \in \{0, 1\}^{\mathcal{P}^2}$: If $\pi(p_1) < \pi(p_2)$, the right-hand side is greater than zero (in fact, ≥ 1 and at most $P - 1$), forcing $y_{p_1, p_2} = 1$; otherwise, the right-hand side becomes negative (≤ -1), so y_{p_1, p_2} may become zero (the constraint is redundant), and indeed it does, because of (13) (as the constraint (29) with p_1 and p_2 interchanged now forces $y_{p_2, p_1} = 1$). Again, the objective (26) and the constraints listed in (30) are identical to the previous models MIP_{PR} and MIP_{SA}, including the choice of a suitable big-M constant for (17).

$$\min C_{\max} \quad (26)$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} x_{p, i} = 1 \quad \forall i \in [P] \quad (27)$$

$$\sum_{i \in [P]} x_{p, i} = 1 \quad \forall p \in \mathcal{P} \quad (28)$$

$$(P - 1)y_{p_1, p_2} \geq \sum_{i \in [P]} i(x_{p_2, i} - x_{p_1, i}) \quad \forall (p_1, p_2) \in \mathcal{P}^2_{\neq} \quad (29)$$

$$(13), (15), (16), (17), (18) \quad (30)$$

$$x \in \{0, 1\}^{\mathcal{P} \times [P]}; y \in \{0, 1\}^{\mathcal{P}^2}; t^A, t^F \in \mathbb{Q}_{\geq 0}^{\mathcal{P} \times \mathcal{R}}; C_{\max} \in \mathbb{Q}_{\geq 0} \quad (31)$$

Implicit Formulation (MIP_{IPO}): Again, every passenger is to be assigned precisely one slot in the sequence, yielding constraints (33) and (34) in MIP_{IPO}. For this model, we work directly with indices of the output sequence $\pi = (1, 2, \dots, P)$ and formulate the remaining constraints in terms of the i -th passenger in this sequence, $i \in [P]$. Incorporating the assignment decisions, we can directly model the objective (32) (minimizing total boarding time) via (35), the requirement that rows are traversed consecutively by each passenger via (36) and (37) (also ensuring that no times remain uncaptured by equating the finishing times for any one row up to the one in which the respective passenger's seat is located with the respective arrival times for the subsequent one), and that passengers cannot be overtaken until they are seated via (38).

$$\min C_{\max} \quad (32)$$

$$\text{s.t. } \sum_{p \in \mathcal{P}} x_{p, i} = 1 \quad \forall i \in [P] \quad (33)$$

$$\sum_{i \in [P]} x_{p, i} = 1 \quad \forall p \in \mathcal{P} \quad (34)$$

$$C_{\max} \geq t_{iR}^F \quad \forall i \in [P] \quad (35)$$

$$t_{i, r+1}^A - t_{i, r}^F \geq 0 \quad \forall i \in [P], r \in [R - 1] \quad (36)$$

$$t_{i, r+1}^A - t_{i, r}^F \leq \sum_{p \in \mathcal{P}: r(p) \leq r} M_{p, r, i} x_{pi} \quad \forall i \in [P], r \in [R - 1] \quad (37)$$

$$t_{i+1, r}^A \geq t_{i, r}^F \quad \forall i \in [P - 1], r \in \mathcal{R} \quad (38)$$

$$t_{i,r}^F - t_{i,r}^A \geq \sum_{p \in \mathcal{P}} \tau_{p,r} x_{pi} \quad \forall i \in [P], r \in \mathcal{R} \quad (39)$$

$$x \in \{0, 1\}^{\mathcal{P} \times [P]}; t^A, t^F \in \mathbb{Q}_{\geq 0}^{\mathcal{P} \times \mathcal{R}}; C_{\max} \in \mathbb{Q}_{\geq 0} \quad (40)$$

The big-M constants in (37) are used to disable equating $t_{i,r+1}^A$ with $t_{i,r}^F$ for rows r beyond the one in which the i -th passenger is eventually seated; this becomes necessary due to working with the *output* index sequence rather than passenger (input) indices, so that we do not know a priori the data and seat belonging to the i -th passenger *in the boarding sequence* and therefore let *all* passengers fictitiously pass through to the last row. Here, $\tau_{p,r} = 0$ for $r > r(p)$ ensures no extra time is added anywhere, and the big-M term in (37) ensures that passengers that are only “virtually” present in some row after the one in which they are actually settled in cannot block “real” passengers around them. It is not hard to see that the big-M constants can be chosen, for instance, as

$$M_{p,r,i} := \max_{\substack{\mathcal{P}' \subseteq \{p' \in \mathcal{P} \setminus \{p\} : r(p') > r\}, \\ |\mathcal{P}'| \leq i-1}} \sum_{p' \in \mathcal{P}'} \left(\sum_{r'=r+1}^{r(p')-1} t_{p',r'}^m \right) + t_{p'}^s.$$

This is, for each tuple $(p, r, i) \in \mathcal{P} \times \mathcal{R} \times [P]$, an upper bound on the time passenger p assigned to position i has to (virtually) wait in row r , since only the $i-1$ predecessors of p can induce this (virtual) waiting by blocking some rows $r' \geq r+1$.

Furthermore, note that because of having all passengers fictitiously arrive at the last row eventually, constraint (35) is formulated w.r.t. the last row R (since we cannot directly access $r(\pi^{-1}(i))$, the index of the row in which the i -th passenger in the output sequence is seated).

3.1.4 Extensions and Optional Constraints

For the sake of simplicity, we have not included other possible aspects of the ABP in the theoretical results and MIP models in this paper, nor will we do so in the computational experiments later. To nevertheless demonstrate that the MIP approach allows to incorporate further desiderata with relative ease, we now briefly discuss the two examples mentioned at the end of Section 1.2—seat interferences and inseparable passenger groups—in the context of MIP_{IP0}.

Seat interferences: Suppose we wish to extend our model to consider possible delays prompted by a passenger having to get up again to let another move to his/her seat further from the aisle. One possibility would be to prohibit such situations altogether by enforcing an outside-in pattern: For any pair of passengers $(p_1, p_2) \in \mathcal{P}_{\neq}^2$ seated in the same row $r = r(p_1) = r(p_2)$, w.l.o.g. with $s(p_1) < s(p_2) \leq k_r^1$, this can be ensured by adding the constraints

$$1 - x_{p_1, i_1} \geq x_{p_2, i_2} \quad \forall (i_1, i_2) \in [P]^2 : i_1 > i_2. \quad (41)$$

to MIP_{IP0}, i.e., to the model (32)–(40); analogously for passenger pairs on the other side of the aisle. Note that even with these restrictive constraints, the MIP still allows for more general boarding sequences than that obtained by the outside-in heuristic.

Alternatively, and more generally, one could model additional waiting times for the case that a passenger p_2 boards before a passenger p_1 with $r(p_2) = r(p_1)$ (and again, w.l.o.g. $s(p_1) < s(p_2) \leq k_r^1$). To that end, suppose we are given data $\tau_p \in \mathbb{Q}_{\geq 0}$ ($p \in \mathcal{P}$) for the time passenger p needs to get up (to let someone pass) and settle in again. Let $z_{i,r,p} \in \mathbb{Q}_{\geq 0}$ for all $i \in [P]$, $r \in \mathcal{R}$ and $p \in \mathcal{P}(r)$ be variables to capture additional waiting times connected to passengers having to get up (and resettle) to let passenger p pass to his/her seat in row r , conditioned on p being the i -th one to board. The constraint

$$M_r (x_{p,i} - 1) + \sum_{\substack{p' \in \mathcal{P}(r) : \\ s(p') < s(p)}} \sum_{i' < i} \tau_{p'} x_{p',i'} \leq z_{i,r,p} \quad \forall i \in [P], r \in \mathcal{R}, p \in \mathcal{P}(r) : s(p) < k_r^1 \quad (42)$$

ensures that $z_{i,r,p}$ is at least the sum of these seat interference waiting times if $x_{p,i} = 1$, and becomes redundant otherwise (if $x_{p,i} = 0$) for a sufficiently large big-M constant, e.g., $M_r := \sum_{p \in \mathcal{P}(r)} \tau_p$. Then, constraint (39) in MIP_{IPO} can be modified to

$$t_{ir}^F - t_{ir}^A \geq \sum_{p \in \mathcal{P}} (\tau_{p,r} x_{pi} + \mathbb{1}_{\{r=r(p)\}} z_{i,r,p}) \quad \forall i \in [P], r \in \mathcal{R}, \quad (43)$$

where $\mathbb{1}_{\{r=r(p)\}} := 1$ if $r = r(p)$, and 0 otherwise. The overall time minimization then brings each $z_{i,r,p}$ down to the bound from (42) (if positive; to zero otherwise). Naturally, analogons of constraints (42) and (43) can straightforwardly be set up for the other side of the aisle as well. Moreover, one could also generalize the waiting times to be row-dependent ($\tau_{p,r}$); they might differ in practice, e.g., for emergency exit rows.

Groups boarding together: It is also conceivable that some groups of passengers wish to board together, which can be accounted for by incorporating constraints that ensure such groups are sequenced consecutively.

Suppose first that the order of passengers within a group is fixed a priori (e.g., say, a group of children in some arbitrary but fixed order with chaperones at the front and at the back). Then, such constraints take a quite simple form: For an ordered passenger group $\{p_1, \dots, p_\ell\} \subseteq \mathcal{P}$, we ensure that its members will be mapped as $p_1 \mapsto i_1, \dots, p_\ell \mapsto i_\ell$ with $i_{j+1} = i_j + 1$ ($j = 1, \dots, \ell - 1$) by enforcing

$$x_{p_{j+1}, i_{j+1}} = x_{p_j, i_j} \quad \forall i \in \{0, \dots, P - k\}, j \in [k - 1]. \quad (44)$$

In case the order within a group $G \subseteq \mathcal{P}$ is itself also still to be optimized, let $\mu_i^G \in \{0, 1\}$, for $i \in \{0, \dots, P - |G|\}$, be decision variables indicating whether the group starts boarding at position $i + 1$ or not. The following constraints then ensure that the group boards together:

$$\sum_{i=0}^{P-|G|} \mu_i^G = 1, \quad (45)$$

$$\frac{1}{|G|} \sum_{p \in G} \sum_{j=1}^{|G|} x_{p, i+j} \geq \mu_i^G \quad \forall i \in \{0, \dots, P - |G|\}. \quad (46)$$

Indeed, note that exactly one μ -variable must be 1 (45) and that $\mu_i^G = 1$ forces $|G|$ consecutive x -variables (associated with the group G) to 1 (46).

3.2 Time-Expanded Multicommodity Flow Formulation

Our final model is a time-expanded integer programming (IP) formulation, for which we discretize the time and build on a (layered) multicommodity flow network model. Suppose we have an upper bound $T \in \mathbb{Z}_{\geq 0}$ on the total boarding time (i.e., there exists a feasible boarding sequence completing before or at time T). We consider points in time $t \in \mathcal{T} := \{0, 1, 2, \dots, T\}$; w.l.o.g. (by the discretization), we may assume that $t_{p,r}^m, t_p^s \geq 1$ and that all these values are integral. Some further notation will be convenient:

We define the set of possible points in time when a passenger $p \in \mathcal{P}$ can start moving within a row $r \leq r(p) - 1$ as

$$\mathcal{T}_{p,r}^m := \left\{ t \in \mathcal{T} : \sum_{r'=1}^{r-1} t_{p,r'}^m \leq t \leq T - t_p^s - \sum_{r'=r}^{r(p)-1} t_{p,r'}^m \right\},$$

that of those when he/she can start settling in (at row $r = r(p)$) as

$$\mathcal{T}_p^s := \left\{ t \in \mathcal{T} : \sum_{r'=1}^{r(p)-1} t_{p,r'}^m \leq t \leq T - t_p^s \right\},$$

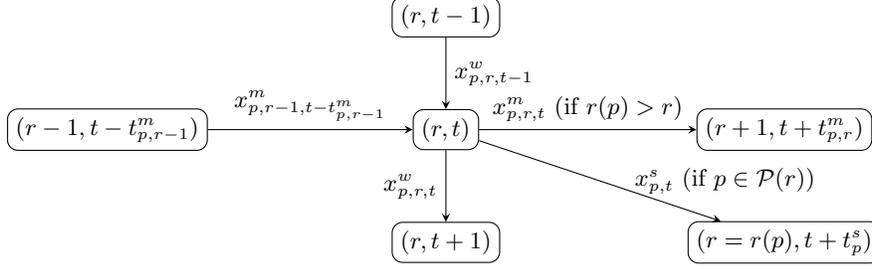


Fig. 1. Multicommodity flow network configuration around an inner node (r, t) from the view point of a passenger (commodity) $p \in \mathcal{P}$.

and that of those when he/she can start waiting (for at least one unit of time) in a row $2 \leq r \leq r(p) - 1$ as

$$\mathcal{T}_{p,r}^w := \left\{ t \in \mathcal{T} : \sum_{r'=1}^{r-1} t_{p,r'}^m \leq t \leq T - t_p^s - 1 - \sum_{r'=r}^{r(p)-1} t_{p,r'}^m \right\}.$$

Furthermore, for each row $r \in \mathcal{R}$, we define the set of points in time at which some action can take place in r as

$$\mathcal{T}_r := \bigcup_{\substack{p \in \mathcal{P}: \\ r(p) < r}} \mathcal{T}_{p,r}^m \cup \bigcup_{p \in \mathcal{P}(r)} \mathcal{T}_p^s.$$

(Note that $\mathcal{T}_{p,r}^w \subseteq \mathcal{T}_{p,r}^m$ for all $(p, r) \in \mathcal{P} \times [r(p) - 1]$.)

The variables for the IP are, for every passenger $p \in \mathcal{P}$, $x_{p,r,t}^m \in \{0, 1\}$, indicating whether p starts moving in row $1 \leq r < r(p)$ at time $t \in \mathcal{T}_{p,r}^m$ (arriving at row $r+1$ at time $t+t_{p,r}^m$), $x_{p,r,t}^w \in \{0, 1\}$, indicating whether p starts waiting in row $2 \leq r < r(p)$ from time $t \in \mathcal{T}_{p,r}^w$ to $t+1$, $x_{p,t}^s \in \{0, 1\}$, indicating whether p starts settling in at his/her seat in row $r(p)$ at time $t \in \mathcal{T}_p^s$ (finishing at time $t+t_p^s$), and finally $C_{\max} \in \mathbb{Z}_{\geq 0}$, the time when boarding is completed. Figure 1 depicts an excerpt of the underlying time-layered network around an “inner node” (r, t) , for which all depicted variables (associated with network arcs) exist.

Using the indicator functions $\mathbb{1}_p^w : \mathcal{R} \times \mathcal{T} \rightarrow \{0, 1\}$ with $\mathbb{1}_p^w(r, t) = 1$ if and only if $2 \leq r < r(p)$ and $t \in \mathcal{T}_{p,r}^w$, we can now state the full time-discretized ABP model (abbreviated as IP_{TE} henceforth):

$$\min C_{\max} \tag{47}$$

$$\text{s.t. } C_{\max} \geq \sum_{t \in \mathcal{T}_{p,r(p)-1}^m} (t + t_{p,r(p)-1}^m + t_p^s) x_{p,r(p)-1,t-t_{p,r(p)-1}^m}^m \quad \forall p \in \mathcal{P} : r(p) \geq 2 \tag{48}$$

$$C_{\max} \geq \sum_{t \in \mathcal{T}_p^s} (t + t_p^s) x_{p,t}^s \quad \forall p \in \mathcal{P}(1) \tag{49}$$

$$\sum_{t \in \mathcal{T}_{p,1}^m} x_{p,1,t}^m = 1 \quad \forall p \in \mathcal{P} : r(p) \geq 2 \tag{50}$$

$$\sum_{t \in \mathcal{T}_p^s} x_{p,t}^s = 1 \quad \forall p \in \mathcal{P}(1) \tag{51}$$

$$x_{p,r-1,t-t_{p,r-1}^m}^m + \mathbb{1}_p^w(r, t-1) x_{p,r,t-1}^w = x_{p,r,t}^m + \mathbb{1}_p^w(r, t) x_{p,r,t}^w \quad \forall p \in \mathcal{P}, 2 \leq r < r(p), t \in \mathcal{T}_{p,r}^m \tag{52}$$

$$\begin{aligned} & \sum_{\substack{p \in \mathcal{P}: \\ r(p) > r}} \sum_{\substack{t' \in \mathcal{T}_{p,r}^m: \\ t-t_p^m < t' \leq t}} x_{p,r,t'}^m + \sum_{p \in \mathcal{P}(r)} \sum_{\substack{t' \in \mathcal{T}_p^s: \\ t-t_p^s < t' \leq t}} x_{p,r(p)-1,t'-t_{p,r(p)-1}^m}^m \\ & + \mathbb{1}_{\{r=1\}} \sum_{p \in \mathcal{P}(1)} \sum_{\substack{t' \in \mathcal{T}_p^s: \\ t-t_p^s < t' \leq t}} x_{p,t'}^s + \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}_{p,r}^w} \mathbb{1}_p^w(r, t) x_{p,r,t}^w \leq 1 \quad \forall r \in \mathcal{R}, t \in \mathcal{T}_r \end{aligned} \tag{53}$$

$$x^m, x^w \in \{0, 1\}^{\mathcal{P} \times \mathcal{R} \times \mathcal{T}}; x^s \in \{0, 1\}^{\mathcal{P} \times \mathcal{T}}; C_{\max} \in \mathbb{Z}_{\geq 0} \quad (54)$$

The meaning of the constraints in the above IP model IP_{TE} is as follows: (48) and (49) ensure correctness of the objective to minimize total boarding time, and (50) and (51) ensure that each passenger enters the plane. The multicommodial flow conservation constraints (52) represent that every passenger $p \in \mathcal{P}$ with respective seat in row $r(p) \geq 2$ must traverse rows $r = 1, \dots, r(p) - 1$ to reach the target row (otherwise, passengers could settle in immediately after entering the plane). Obviously, we only need such flow conservation for $r \leq r(p) - 1$ and $t \in \mathcal{T}_{p,r}^m$, respectively. Finally, (53) ensures that only one passenger can act (wait, move, or settle in) in any row at any time step. (Note that our model also contains all variables with overlap w.r.t. a given time step t , though it would suffice to use only those associated to starting at t .)

4 Computational Experiments

We now turn to an experimental evaluation of the MIP formulations and select boarding strategies. We compare the outside-in heuristic and Steffen’s method [7], our novel max-settle-row strategy as well as random boarding with our exact models. The first two methods had emerged in previous simulation studies as often being superior to other classical strategies that also do not explicitly take account of passenger moving and settle-in times; we complement these findings by contrasting the behavior of these schemes with the “time-aware” methods proposed in this paper. (Random boarding is included as a reference for what would happen if no care at all were to be taken regarding the sequence in which passengers are asked to board. For any instance, we take the best one of 1000 randomly generated permutations as the output of the random boarding strategy.)

In particular, we will also evaluate a simple time-aware improvement heuristic that can be applied to any boarding method as a post-processing routine:

Definition 4.1 (2-opt local search). *Given a boarding sequence π of \mathcal{P} with completion time $C_{\max}(\pi)$ for a given ABP instance, repeatedly perform swaps of the positions of a pair of entries in π that lead to a (strict) reduction in the total boarding time, until no further improvements can be achieved by such swaps (one iteration traverses all pairs in lexicographical order of indices, performing all improving swaps along the way).*

In the following, we first describe our implementation (especially, some tweaks to the MIP models) in Section 4.1 and describe our test instances (Section 4.2) before presenting and discussing several computational experiments in Sections 4.3 (main comparison of the considered boarding methods for different levels of “resolution” w.r.t. the passenger time data) and 4.4 (gauging robustness of boarding sequences obtained with inexact data w.r.t. random disruptions).

4.1 Implementation Details

We implemented the boarding heuristics in Python 3.6, which was also used as a scripting language and to generate test instances. The (M)IP models were solved with the state-of-the-art commercial solver Gurobi 8.1.0 [22]; for each instance, we provided Gurobi with the best solution obtained by any of the heuristics (plus local search) as a primal incumbent (starting solution). All experiments were carried out on a cluster of 64 Linux machines with Xeon L5630 Quad-Core CPUs (2.13 GHz) and 16 GB memory (except for IP_{TE} , which we put on otherwise identical machines with 128 GB memory).

The MIP models described in the previous section all allow for some refinements that we did not incorporate in the respective model statements for the sake of readability, but that were (mostly) adopted in our implementation.

MIP_{PR} : Obviously, one does not need both variables y_{p_1, p_2} and y_{p_2, p_1} for all $(p_1, p_2) \in \mathcal{P}_{\neq}^2$, since they are complementary by constraint (13). Thus, we kept only those y_{p_1, p_2} with $p_1 < p_2$, omitting (13) and replacing y_{p_2, p_1} by $1 - y_{p_1, p_2}$. Similarly, one does not need all constraints (14), only those for ordered tuples; see [23] for details. Moreover, as mentioned in [24], it can be beneficial to add the constraints (14)

on the fly, in a cutting-plane fashion. We indirectly take this approach by adding the constraints to our Gurobi model as so-called lazy constraints, letting Gurobi choose automatically which constraints are used when during the solution process. This indeed provided some speed-up compared to always including all constraints of type (14) to begin with. Furthermore, we eliminated the variables $t_{p,r}^A$ by resolving constraints (16). Other reductions that we did not implement are the elimination of $t_{p,r(p)}^F$, incorporating C_{\max} and (18) into (15), and removing (17) for $r = 1$.

MIP_{SA}: The variables $w_{p,P}$ are clearly not needed, so we omitted them in the implementation. For variables and constraints shared with MIP_{PR}, the reductions described above were applied here as well.

MIP_{DPO}: Again, reductions for variables and constraints this model shares with MIP_{PR} were applied as well, see above for the details.

MIP_{IPO}: Variables $t_{i,1}^A$ and $t_{i,R}^F$ are not needed (in (38), the latter could be replaced by the corresponding lower bound from (39), and in (39) it could be replaced by C_{\max}), and since the first-boarding passenger is never blocked, the variables $t_{1,r}^A$ for all $r \in \mathcal{R}$ are also not needed. Besides eliminating these variables, one could also omit constraints (36) and (37) for $i = 1$. We tried out these modifications, but ultimately did *not* keep them in our implementation because they provided no improvement. Furthermore, it is worth mentioning that Gurobi’s presolve routines removed constraints (38) for $r = 1$, as well as constraints (39); we do not currently know why this is apparently possible.

IP_{TE}: Gurobi presolve removes many variables and constraints, but we did not see a way to derive reduction rules from these observations.

Moreover, in the experiments discussed in the following, we also report results of a MIP model that combines MIP_{PR}, MIP_{SA} and MIP_{DPO} (i.e., the three MIPs that share several variables and constraints already, including the above-described reductions); we will refer to this combined formulation as MIP_{comb}.

4.2 Construction of Test Instances

For our instance sets, we considered 4 different cabin layouts: $(R, k) \in \{(10, 2), (20, 2), (20, 4), (30, 6)\}$. The settle-in times were obtained as $\max\{\min\{\lfloor z \rfloor, 120\}, 1\}$, with $z \sim \mathcal{N}(60, 20)$ drawn independently at random from the normal distribution with mean 60 and standard deviation 20, where $\lfloor \cdot \rfloor$ denotes rounding to the nearest integer (breaking ties in favor of the nearest even integer). The moving times were independently sampled from $\{1, 2, 3\}$ with respective probabilities $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$. We created 3 instance sets *mp-sp*, *m-sp*, and *mp-s*, each with 10 instances for each cabin layout (for a total of 120 instances). The instance sets differ in which times are constant and which are fluctuant: “m” stands for moving and “s” for settle-in times, and a trailing “p” represents that the corresponding times are passenger-dependent (otherwise, the respective times are the same for all passengers). For simplicity, we do not consider passenger- and *row*-dependent moving times $t_{p,r}^m \in \mathbb{Q}_{\geq 0}$ (i.e., only $t_{p,r}^m = c_p^m$ and $t_{p,r}^m = c^m$ are used).

With these instances, we evaluate the algorithm performances in the different settings, see Section 4.3. Furthermore, we conduct a second set of experiments to assess the robustness of different boarding strategies w.r.t. data perturbations and disruptions of the determined boarding sequences, see Section 4.4. With these simulations, we hope to gauge the influence of using estimated time data as opposed to the “real” time data (that will generally not be available a priori in practice) as well as that of passengers not keeping to the predetermined boarding order or arriving late at the gate, events that are likely to happen in practice. To that end, first, we consider late passengers: for every instance, we independently sample uniformly at random 2, 3, 4, or 5 passengers, respectively, for each cabin layout (the larger the cabin, the more late-comers), and place them at the end of the boarding sequence (in random order). Second, for the purpose of assessing the impact of position swaps (within the boarding sequence), we divided the (given/computed) order into groups of 10 (consecutive) passengers and swap two passengers within each group, for a total of 10% of passengers changing places; the passengers are sampled independently uniformly at random within each group. (The partition into small groups of size 10 is motivated by the possible way to realize by-seat strategies like max-settle-row at the gate by dividing the boarding sequence into small groups that could then easily be sorted manually by gate agents.) Third, to examine the relevance of precise knowledge of settle-in and moving times, we evaluate the computed boarding sequences also w.r.t. perturbed times, obtained as random numbers drawn from the normal distribution centered at the original respective time values: For each

Table 2. Experimental results on testset mp-sp (average values over 40 instances; 10 each for 4 airplane sizes).

method	objective	w/ 2-opt	% impr.	% gap	# opt	runtime
random	1383.1	747.4	45.96	—	—	18.3
Steffen	778.2	660.8	15.08	—	—	13.8
outside-in	514.6	462.9	10.04	—	—	12.2
max-settle-row	475.5	454.6	4.40	—	—	9.7
MIP _{PR}	452.1	—	0.33	42.90	18	829.5
MIP _{SA}	452.3	—	0.28	43.71	18	1032.3
MIP _{DPO}	452.6	—	0.20	47.98	14	1639.1
MIP _{comb}	453.0	—	0.13	47.72	12	1699.2
MIP _{IPO}	452.7	—	0.19	12.37	10	2629.3
IP _{TE}	453.3	—	0.06	61.41	8	5286.4

settle-in time t_p^s , a perturbed time is created by sampling $z \sim \mathcal{N}(t_p^s, 10)$, clipping to $\max\{\min\{z, 120\}, 0\}$ and rounding the (not necessarily integral) result to two significant digits. Similarly, for each moving time c_p^m (recall we do not have row-dependency here), we sample $z \sim \mathcal{N}(c_p^m, 1)$, clip so that $z \in [0, 3]$ and round to two significant digits. In the experiments, we evaluate those three disturbances—late passengers, passenger swaps, and perturbed time data—separately as well as all together (swaps performed before repositioning late passengers), the latter representing the most real-world oriented scenario. Furthermore, in light of Proposition 2.1, we also studied the influence of using overestimated time data to obtain the boarding sequence. To that end, we used the perturbed time data and rounded each moving time c_p^m up to 3, and rounded the settle-in times t_p^s up to 30, 75 (if $t_p^s \in (30, 75]$), or 120, respectively. The boarding sequences obtained using these overestimated time values are then evaluated using the original (“real”) values, including the above-described disturbances.

Our instance testset can be downloaded from the project webpage [25].

4.3 Experiments for Different Data Models

The results for the experiments with different boarding strategies under the three data scenarios are summarized in Tables 2, 3, and 4. For each scenario, the tables provide average values over all instances (10 per cabin layout, i.e., 40 instances per table), where all time values are stated in seconds rounded to one significant digit (second, third, and last column). Average values are calculated as arithmetic means except for the algorithm runtimes, where we used the shifted geometric mean (with a 10s shift to reduce the influence of easy instances). The columns labeled “objective” give the average completion times of the boarding sequences obtained with the respective algorithms listed in the first column (“method”), and the column “w/ 2-opt” shows the average completion times after post-processing the heuristically computed boarding sequences with our local search routine from Definition 4.1 (we did not apply this post-processing to any (M)IP solutions). The column labeled “% impr.” gives the resulting average improvement percentages ($100\% \cdot (\text{old value} - \text{new value}) / (\text{old value})$, rounded to two significant digits) due to the 2-opt post-processing routine for the heuristic boarding strategies, and the average improvement over the best (post-processed) heuristic solution achieved by Gurobi applied to the different (M)IP models, respectively. The columns “% gap” and “# opt” provide the average optimality gap ($100\% \cdot (\text{best upper bound} - \text{best lower bound}) / (\text{best upper bound})$, rounded to two decimals) and the (absolute) number (out of 40) of instances that the respective (M)IPs were solved to exact optimality within a time limit of two hours (7200s). Finally, the “runtime” column gives the mean runtimes of the algorithms, including local search (the heuristic boarding strategies themselves take less than a second for all instances).

From Tables 2, 3, and 4, we can draw a variety of conclusions. Starting with the heuristics, we first observe that the max-settle-row strategy yields lower objectives than all other heuristics in all three time-data scenarios, before and after applying the 2-opt local search post-processing routine. The differences between outside-in and max-settle-row boarding are notable but not overwhelmingly large, across all scenarios and

Table 3. Experimental results on testset m-sp (average values over 40 instances; 10 each for 4 airplane sizes).

method	objective	w/ 2-opt	% impr.	% gap	# opt	runtime
random	1352.4	763.6	43.54	—	—	15.4
Steffen	747.1	640.5	14.27	—	—	10.7
outside-in	472.0	434.4	7.97	—	—	10.3
max-settle-row	432.2	430.3	0.44	—	—	7.1
MIP _{PR}	429.1	—	0.12	36.20	19	737.4
MIP _{SA}	429.3	—	0.07	37.06	18	803.2
MIP _{DPO}	429.3	—	0.07	41.38	15	1439.3
MIP _{comb}	429.5	—	0.02	41.19	15	1466.5
MIP _{IPO}	429.3	—	0.06	2.36	21	440.8
IP _{TE}	429.5	—	0.02	50.88	17	1449.6

Table 4. Experimental results on testset mp-s (average values over 40 instances; 10 each for 4 airplane sizes).

method	objective	w/ 2-opt	% impr.	% gap	# opt	runtime
random	1250.7	750.0	40.03	—	—	14.1
Steffen	628.4	584.9	6.93	—	—	9.4
outside-in / max-settle-row	421.8	396.3	6.06	—	—	8.6
MIP _{PR}	395.6	—	0.03	42.81	17	772.6
MIP _{SA}	395.6	—	0.03	42.92	17	820.8
MIP _{DPO}	395.7	—	< 0.01	45.65	17	1411.8
MIP _{comb}	395.7	—	< 0.01	45.30	16	1374.7
MIP _{IPO}	395.7	—	< 0.01	8.46	16	1007.1
IP _{TE}	395.7	—	< 0.01	63.39	10	3809.7

instance sizes. After 2-opt, these differences become even smaller (outside-in boarding times can be reduced further by local search than max-settle-row boarding times); recall also that for constant settle-in times, max-settle-row and outside-in coincide, so we report their results only once in Table 4. Overall, random boarding performs worst of all strategies (even though we pick the best sequence out of 1000 random ones for each instance), and Steffen’s method is also consistently and significantly worse than outside-in and max-settle-row; for both random and Steffen boarding, large improvements are achievable by 2-opt post-processing (which apparently translates to longer runtimes, as more local search sweeps are required), but the end results are still clearly inferior.

Regarding solution quality, we find that the outside-in and max-settle-row strategies empirically behave more benevolently than guaranteed by our theoretical results from Section 2.2. This conclusion is enabled by our MIP lower bounds (especially those from MIP_{IPO}) on the optimal boarding times. For instance, on the m-sp testset (which appears to be easiest for the MIPs, cf. Table 3), max-settle-row is very close to optimal even without local search post-processing, with empirical average approximation ratio estimate about 1.03—much better than the $(1 + H_k(k-1)/k)$ -guarantee from Theorem 2.6 already for $k = 2$. (Note also that for the mp-s instances, where $t_{p,r}^m = c_p^m \in \{1, 2, 3\} \forall (p, r)$, Remark 2.3 gives an approximation ratio guarantee of 3 for max-settle-row/outside-in, which is better than the ratio k of Theorem 2.4 for the larger-instances half of the testset, where $k \in \{4, 6\}$; on average over all instance sizes, again using the lower bound derived from the MIP_{IPO} optimality gap in Table 4, the approximation ratio is just about 1.09 empirically.)

Let us now turn to the MIP formulations. From the tables, it becomes clear that the MIPs generally can only marginally improve over the best heuristic (max-settle-row + 2-opt) solutions, and apparently (for the larger instance) have trouble increasing the dual bounds in order to certify optimality of the best known solution. While this behavior is not atypical for hard MIP instances, it nevertheless is somewhat disappointing. (We also tried out different settings for Gurobi’s MIP focus hyperparameter, putting solver

emphasis on finding better feasible solutions or better lower bounds, but this did not have any relevant impact on the solution progress.) However, the effort is still not wasted, as the MIP lower bounds provide a hitherto unavailable way to gauge the effectiveness of boarding heuristics. Indeed, the optimality gaps provide a *computational proof* that the solutions produced by max-settle-row (and, to a lesser degree, outside-in) are not very far from optimal, especially after 2-opt post-processing. That this holds on the mp-s and (especially) m-sp testsets was discussed in the previous paragraph, but even in the most general case (mp-sp) without local search, max-settle-row is empirically only about 16.5% off (at most), i.e., it is a 1.17-approximation on empirical average. Similarly, the boarding time of the outside-in solution without subsequent 2-opt local search is empirically no worse than an average of about 1.23 times the best possible boarding time on mp-sp instances.

Moreover, the MIPs (in particular, MIP_{IPO}) can solve a relevant portion of the instances to provable optimality, albeit mostly for the smaller instances (where, in fact, MIP_{PR} often performed best). Fixing either moving or settle-in times to the same constant value for all passengers makes all MIPs notably easier: comparing the results in Tables 3 and 4 with those from Table 2, note that for m-sp and mp-s testsets, more instances are solved, the mean runtimes are significantly smaller, and the optimality gaps are smaller—significantly so for MIP_{IPO} —(though slightly worse for IP_{TE} on mp-s instances). Constant moving times (m-sp) appear to give the generally easiest-to-handle instances for the MIP models (or Gurobi applied to them, more precisely).

Overall, it becomes clear that MIP_{IPO} consistently yields the best average optimality gaps by very large margins. For mp-sp instances (Table 2), predecessor-formulations (particularly, MIP_{PR} and MIP_{SA}) find slightly better solutions on average and find exact optima more often (thus running into the time limit less often, which translates to lower average runtimes), but on instances where they fail, the gaps are much larger. On the m-sp testset (Table 3), MIP_{IPO} is clearly the best formulation w.r.t. all three measures—average gap, number of certified optimal solutions, and mean runtime. For mp-s instances (Table 4), MIP_{IPO} is a bit slower on average than the first two predecessor-formulations and reached one less proven optimum, but the much lower average gap tips the scales back in its favor. The time-expanded formulation IP_{TE} is by far the worst in all scenarios, which is likely explained by the comparatively huge model size (due to time discretization) compared to the other, compact formulations.

Among the predecessor-formulations, those MIPs that extend MIP_{PR} with additional variables and constraints (i.e., MIP_{SA} , MIP_{DPO} and MIP_{comb}) are inferior to the pure MIP_{PR} formulation. For the smallest 20 instances, MIP_{PR} even outperforms MIP_{IPO} (especially on mp-sp instances of size $(R, k) = (10, 2)$, the former is much faster), but for the larger instances (which arguably are more interesting, since more is to be gained by optimizing boarding for large planes), MIP_{PR} lower bounds are much worse than those MIP_{IPO} obtains (within the two-hour time limit we enforced), while the primal solution qualities are comparable.

Finally, it is worth mentioning that, on average, only the MIP_{IPO} formulation succeeds in producing lower bounds that are better than what we can obtain from Lemma 2.2: Consistently, the lower bound (5) is significantly better than the other two ((4) and (6)), and yields, for instance, an average value of about 391.33 for the hardest (mp-sp) instances; this implies an associated average optimality gap of 13.56%, which is improved upon by MIP_{IPO} (12.37% average gap) but still far better than any of the other MIP dual bounds.

Thus, in conclusion, the results discussed in this section demonstrate that our max-settle-row boarding strategy is the best of the considered heuristics across all time-data scenarios, closely followed by its “parent” outside-in. The proposed 2-opt post-processing routine can significantly improve all heuristic solutions, and (combined with max-settle-row) yields boarding sequences that are provably close to optimal, by virtue of our MIP formulations for the ABP and the resulting computational lower bounds on optimal boarding times. Among the MIP models, MIP_{IPO} generally performs best, providing highly useful lower bounds even if only marginally improving on the best heuristic solutions.

4.4 Robustness Experiments

For the experiments discussed in the following, the only exact method we include is (Gurobi applied to) MIP_{IPO} , because it significantly outperformed the other models on average in the results seen in the previous

Table 5. Results of robustness experiments on testset mp-sp for disruptions caused by late passengers, passengers swapping positions, and for perturbed moving and settle-in times as well as all three combined (average values over 40 instances; 10 each for 4 airplane sizes).

method	objective	obj. (late)	obj. (swap)	pert. obj.	combined	ref. obj.	% comb. loss
random + 2-opt	747.4	936.1	1064.9	833.1	1209.8	679.8	[62.63, 66.62]
Steffen + 2-opt	660.8	864.4	1049.1	740.0	1201.0	644.6	[62.36, 66.37]
outside-in + 2-opt	462.9	667.3	859.7	513.4	1014.5	467.7	[55.44, 60.19]
max-settle-row + 2-opt	454.6	659.6	821.7	497.5	992.6	467.8	[54.45, 59.31]
MIP _{IPO}	452.7	658.4	825.4	500.5	988.4	464.1	[54.26, 59.14]

subsection. Similarly, our 2-opt local search routine improved the output of all heuristics, whence here, we consider only the post-processed heuristic solutions. Also, because it reflects the most realistic scenario, we focus on the mp-sp data model.

To assess robustness aspects of the boarding schemes w.r.t. different possible disturbances as outlined in Section 4.2, the experiments summarized in Table 5 show the impact on the boarding times (again averaged over 40 instances, consisting of 10 instances each for all four airplane cabin sizes) of placing randomly chosen passengers at the end of the boarding queue (column “obj. (late)”), swapping the positions of random passenger pairs in the predetermined boarding sequence (“obj. (swap)”), and assuming the actual passenger time data are perturbed w.r.t. those with which boarding sequences were computed (“pert. obj.”), respectively, as well as for the combination of all these disruptions (column “combined”). Additionally, the “ref. obj.” column lists the average boarding times achieved by letting the respective algorithms run on the perturbed time data (without late- or swap-disruptions applied afterwards); for each instance, the smallest value here is an upper bound on the respective ideal boarding time (computed with knowledge of the reference perturbed time data and no disturbance of the boarding sequence). Together with the lower bounds from MIP_{IPO}, we can estimate averaged loss intervals w.r.t. the ideal situation as incurred by the data uncertainty and disruptions, stated in the final column “% comb. loss”.

Moreover, Table 6 presents the outcome of experiments in which we evaluate boarding sequences that were computed using “pessimistic” time data (overestimated moving and settle-in times) on the “true” reference time data and after late- and swap-disturbances (the “true” data here is the perturbed time data from the experiments summarized in Table 5, see Section 4.2 for the details).

We begin with a closer look at the computational results of Table 5. It can be observed that, on average across all instance sizes, the boarding sequences computed with the original moving and settle-in times are relatively resilient w.r.t. perturbations of the time data: the boarding times increase only by roughly 10% for the better strategies when evaluating the sequences on the perturbed data. Having passengers swap positions within the boarding order as well as having a few passengers arrive late (and take a place at the end of the otherwise unchanged boarding sequence) has a notably stronger effect. Swaps seem to lead to stronger deterioration of the solution quality than late passengers (although to be fair, the numbers are not directly comparable due to the different numbers of disruptions). Combining time-data perturbation and disruptions, the boarding times for all strategies increase drastically—by well over 50%—with max-settle-row and (naturally) MIP_{IPO} yielding the lowest overall times, between about 54% and just below 60% larger than the ideal baseline/reference solutions on average. Random boarding and Steffen’s strategy are more robust to the disturbances, but nevertheless the final (combined) boarding times are significantly inferior to those of the other three approaches.

Looking at the results in more detail (not directly discernable from Table 5), we observed that the overall deterioration generally increases with cabin size; this of course seems natural, given that more perturbations and/or disturbances are applied the larger the aircraft is. More precisely, for late passengers, the deterioration actually decreases again from $(R, k) = (20, 2)$ to $(20, 4)$ to $(30, 6)$; the larger influence for smaller cabin sizes here can likely be explained by shorter absolute boarding completion times (and a percentage-wise larger number of disruptions). For instance, for the largest instances (with 5 out of 180 late passengers), the boarding time of the MIP_{IPO}-sequence increases by about 22% on average, versus about

Table 6. Results for robustness experiments on testset mp-sp for pessimistic data assumptions and all disturbances (average values over 40 instances; 10 each for 4 airplane sizes).

method	obj. (pess.)	obj. (true)	ref. true obj.	obj. (true, disrupted)	runtime
random + 2-opt	1088.4	908.6	679.8	1258.86	22.0
Steffen + 2-opt	875.7	747.3	644.6	1214.90	18.1
outside-in + 2-opt	578.8	512.5	467.7	1012.22	21.2
max-settle-row + 2-opt	577.5	504.1	467.8	996.83	9.7
MIP _{IPO}	577.4	504.3	464.1	999.34	218.5

31% for the smallest (2 of 20 late), 35% for the second-smallest (3 of 40 late) and 28% for the second-largest (4 of 80 late). For MIP_{IPO}, passengers swaps (10% of passengers in total, swapping one pair within every disjoint group of 10 passengers scheduled to board consecutively) increase the boarding time by about 43%, 52%, 42% and 45%, respectively (from smallest to largest cabin size). Thus, comparing the results for instances of size (10, 2) and (20, 2), i.e., where only the row number R changes, there is a clear increase in the boarding time due to late- and swap-disturbances, while between (20, 2) and (20, 4) (where only the number k of seats per row changes), the numbers actually go down again.

On the one hand, the overall strong increase in the boarding times when exposing some computed boarding sequence to “realistic” changes of environment (misjudged time data and passengers not following the boarding order or arriving very late at the gate) demonstrates that even then, the newly proposed strategies (max-settle-row, the 2-opt local search routine, and MIP_{IPO}) provide significant improvements over the others (outside-in being a close follower). On the other hand, the large differences in boarding times suggest that there is still room for improvement, by explicitly incorporating robustness aspects into boarding strategy design.

Furthermore, the large deterioration of the boarding times after perturbing the time data and performing the other disruptions also raises the question of how much worse things might get if the time data used to compute the boarding sequence is actually quite far off from the true data. In particular, in view of Proposition 2.1, do we fare much worse if we adopt very rough (over-)estimates of the passenger moving and settle-in times to compute the boarding order? This leads to our next, and final, experiment, summarized in Table 6.

In Table 6, the columns “obj. (pess.)”, “obj. (true)”, and “obj. (true, disrupted)” give the boarding times of the sequences computed using the pessimistic (overestimated) time-data, evaluated on that same data, and on the true data (perturbed time values from previous mp-sp experiments) without and with late-passenger and position-swap disruptions, respectively. The “ref. true obj.” column states the boarding times achieved by the respective algorithms when directly using the true data, which serve as the reference ideal-case solutions. The final column provides the (shifted geometric mean) runtimes for computing the boarding sequences (including local search runtime for the heuristics).

We can observe from Table 6 that, somewhat surprisingly, the boarding times using pessimistic time estimates are not terribly far off the reference values, which indicates that disregarding other disruptions, it might not be a huge obstacle that one likely will not have precise knowledge of passenger moving and settle-in times in practice. Indeed, comparing the first two boarding time columns, we see a validation of Proposition 2.1—replacing the pessimistic data by the smaller true time values (but retaining the boarding sequence itself), the boarding times always reduce—and for the better strategies (outside-in, max-settle-row (each post-processed by the 2-opt routine) and MIP_{IPO}), the difference to the reference solution values turns out to be below 10% on average. However, analogously to the experiments from Table 5, once we additionally consider disruptions of the boarding order, the picture turns quite glum again, with final estimates of the “real-life” outcome of the present strategy that are about or more than 50% larger than the ideally achievable boarding times.

Hence, while one (tentative) take-away message from these last experiments is that the boarding planning is not ridiculously sensitive to the accuracy of the utilized time-data, the other large point to be made reiterates the finding of the first set of robustness experiments above: To reduce large fluctuations induced

by disruptive passenger behavior, one must find a way to explicitly robustify the boarding process (or algorithmic boarding schemes).

5 Concluding Remarks

The present paper offered several key contributions to the research on time-efficient airplane boarding methods: We provided an (almost) complete characterization of the airplane boarding problem in terms of computational complexity, proving strong NP-hardness for four out of the six considered passenger moving and settle-in time models, and that one of the two other cases can be solved to optimality in polynomial time by the well-known outside-in boarding strategy. In spite of this positive result for outside-in boarding, we also proved that all deterministic boarding strategies that neglect passenger time information (encompassing all commonly used ones, including outside-in) can yield boarding times that are far from optimal. Moreover, we derived theoretical lower bounds for the optimal boarding time that led to the first approximability guarantees for the ABP, which encompass two data models, including the one not covered by our complexity findings. We also developed what appears to be the first algorithmic approaches that directly tackle the problem of minimizing the airplane boarding time, in contrast to previous work focusing on simulations or related but different goals such as minimizing the number of passenger interferences. In an empirical study, we compared several new MIP models with known and new heuristics; our best models were able to solve almost half of our test instances to optimality, and provide computational certificates of solution quality for the heuristics in any case. Moreover, our computational results showed that the proposed “max-settle-row” boarding strategy outperforms all other heuristics, closely followed by the related outside-in boarding strategy, and that a simple local search can significantly improve heuristic solutions. Finally, we assessed the robustness of the considered boarding methods with respect to perturbations of the passenger time data and passengers disrupting the planned boarding sequence.

These last experiments revealed that there is still a large potential for improvements that might be gained by incorporating robustness aspects more explicitly. Thus, we believe it to be an interesting topic for future research to tackle the boarding time minimization problem from the perspective of robust optimization (see, e.g., [26]). Moreover, it may be worth spending some effort to devise a dedicated branch-and-cut solver for the ABP that incorporates local bounds like those from Lemma 2.2, heuristics, and further MIP techniques (branching rules, cutting planes, etc.) still to be investigated for the present problem. An example for a specialized solver is the column generation approach developed and implemented in the very recent work [27], based on a preliminary version of the present paper, for the time-expanded multicommodity flow model IP_{TE} in an effort to overcome the apparently prohibitively large model sizes. Unfortunately, the computational results with this method were discouraging, as they did not improve over using the model IP_{TE} directly; see [27] for details. Nevertheless, it is conceivable that MIP_{IPO} (or possibly MIP_{PR}) may be solvable more efficiently by a more sophisticated specialized solver, and we intend to pursue this approach further. A more efficient MIP algorithm would furthermore open the door to more extensive simulation studies, involving further heuristics, larger test instance sets, different data models (possibly real-world data obtained from field tests), and including extensions to the boarding problem like the treatment of seat interferences mentioned in Section 3.1.4.

It is also of interest to continue the theoretical investigation of (simple) boarding strategies. Some possible topics for future research in this direction are resolving the question of possible optimality or approximation guarantees of (say) the outside-in strategy in case the plane is not fully occupied, obtaining constant-factor approximation schemes, or answering the open question of the complexity of the ABP in case of passenger-dependent moving times and identical-for-all settle-in times (cf. Table 1). Similarly, given the much better empirical performance of outside-in and max-settle-row compared to the respective theoretical guarantees and the fact that the parameter k (number of seats per row) was relatively small in all test instances, it may be worth looking at the problem from the viewpoint of parameterized complexity (cf. [28,29]), to possibly even find a fixed-parameter tractable special case and corresponding practically efficient (exact) algorithm.

Acknowledgements

The authors would like to thank Marco Lübbecke for hosting inspiring discussions in his gORden and suggesting the time-layered network approach.

References

1. T. Horstmeier, F. de Haan, Influence of ground handling on turn round time of new large aircraft, *Aircraft Engineering and Aerospace Technology* 73 (3) (2001) 266–271.
2. M. Ozmec-Ban, R. Š. Babić, A. Modić, Airplane Boarding Strategies for Reducing Turnaround Time, in: *Proc. 18th International Conference on Transport Science (ICTS)*, 2018.
3. P. Ferrari, K. Nagel, Robustness of Efficient Passenger Boarding Strategies for Airplanes, *Transportation Research Record* 1915 (1) (2005) 44–54.
4. S. Marelli, G. Mattocks, R. Merry, The role of computer simulation in reducing airplane turn time, *Aero Magazine* 1 (1).
5. F. Jaehn, S. Neumann, Airplane boarding, *European Journal of Operational Research* 244 (2) (2015) 339–359.
6. D. C. Nyquist, K. L. McFadden, A study of the airline boarding problem, *Journal of Air Transport Management* 14 (4) (2008) 197–204.
7. J. H. Steffen, Optimal boarding method for airline passengers, *Journal of Air Transport Management* 14 (3) (2008) 146–150.
8. R. J. Milne, A. R. Kelly, A new method for boarding passengers onto an airplane, *Journal of Air Transport Management* 34 (2014) 93–100.
9. A. Kierzkowski, T. Kisiel, The Human Factor in the Passenger Boarding Process at the Airport, *Procedia Engineering* 187 (2017) 348–355.
10. H. Zeineddine, A dynamically optimized aircraft boarding strategy, *Journal of Air Transport Management* 58 (2017) 144–151.
11. R. J. Milne, M. Salari, Optimization of assigning passengers to seats on airplanes based on their carry-on luggage, *Journal of Air Transport Management* 54 (2016) 104–110.
12. M. Bazargan, A linear programming approach for aircraft boarding strategy, *European Journal of Operational Research* 183 (1) (2007) 394–411.
13. M. H. L. van den Briel, J. R. Villalobos, G. L. Hogg, T. Lindemann, A. V. Mulé, America West Airlines Develops Efficient Boarding Strategies, *Interfaces* 35 (3) (2005) 191–201.
14. M. Soolaki, I. Mahdavi, N. Mahdavi-Amiri, R. Hassanzadeh, A. Aghajani, A new linear programming approach and genetic algorithm for solving airline boarding problem, *Applied Mathematical Modeling* 36 (9) (2012) 4060–4072.
15. R. J. Milne, M. Salari, L. Kattan, Robust Optimization of Airplane Passenger Seating Assignments, *Aerospace* 5 (3) (2018) Art. No. 80.
16. P. Brucker, *Scheduling Algorithms*, 5th Edition, Springer, Berlin Heidelberg, 2007.
17. H. van Ladeghem, A. Beuselinck, Reducing passenger boarding time in airplanes: a simulation based approach, *European Journal of Operational Research* 142 (2) (2002) 294–308.
18. M. Schultz, M. Schmidt, Advancements in Passenger Processes at Airports from Aircraft Perspective, *Sustainability* 10 (11) (2018) Art. No. 3877.
19. G. C. Budesca, A. A. Juan, P. Fonseca i Casas, Optimization of aircraft boarding processes considering passengers’ grouping characteristics, in: A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, J. A. Miller (Eds.), *Proc. Winter Simulation Conference, IEEE*, 2014, pp. 1977–1988.
20. M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.
21. D. Coudert, A note on Integer Linear Programming formulations for linear ordering problems on graphs, *Research report hal-01271838*, Inria; I3S; Université Nice Sophia Antipolis; CNRS (2016).
22. Gurobi Optimization, LLC, [Gurobi Optimizer Reference Manual](http://www.gurobi.com) (2019).
URL <http://www.gurobi.com>
23. S. Mallach, Linear ordering based MIP formulations for the vertex separation or pathwidth problem, *Journal of Discrete Algorithms* 52–53 (2018) 156–167.
24. R. Mart, G. Reinelt, *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, 1st Edition, Springer Publishing Company, Incorporated, 2011.
25. F. J. L. Willamowski, A. M. Tillmann, [Instances for minimizing the airplane boarding time](https://www.or.rwth-aachen.de/airplane-boarding) (2019).
URL <https://www.or.rwth-aachen.de/airplane-boarding>

26. A. Ben-Tal, L. El Ghaoui, A. Nemirovski, *Robust Optimization*, Princeton Series in Applied Mathematics, Princeton University Press, 2009.
27. J.-C. Resch, Berechnung eines optimalen Boarding-Prozesses von Flugzeugen mit Hilfe von Spaltengenerierung (Computation of an optimal boarding process for airplanes by means of column generation), Master’s thesis (submitted), in German, Technische Universität Braunschweig (2019).
28. R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.
29. R. Downey, M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.
30. C. H. Papadimitriou, P. C. Kanellakis, Flowshop Scheduling with Limited Temporary Storage, *Journal of the ACM* 27 (3) (1980) 533–549.
31. N. G. Hall, C. Sriskandarajah, A Survey of Machine Scheduling Problems with Blocking and No-Wait in Process, *Operations Research* 44 (3) (1996) 510–525.

A Appendix: An Auxiliary Flow Shop Problem

The following result was used in the proof of Theorem 2.1 and may be of interest in its own right:

Lemma A.1. *The decision version of the three-machine permutation flow shop scheduling problem with blockages, $F_3|perm, blocking|C_{max}$, is NP-complete in the strong sense, even for integer processing times and if all jobs have processing time zero on the second (middle) machine.*

Proof. To show hardness, we begin with an instance of $F_2|b = 1|C_{max}$, a flow shop scheduling problem that was shown to be strongly NP-hard in [30] (for integer processing times). In this problem, we are given a set of n jobs J and two machines m_1 and m_2 ; each job $j \in J$ consists of tasks j_m to be processed on machine m with corresponding processing times $t_{j,m}$. Between the machines, there is a temporary job buffer (i.e., a single job can be temporarily held in buffer storage if the second machine is not yet available). The goal is to minimize the makespan, i.e., the completion time of the last job (task).

Recall that in a flow shop, all jobs share the same technological order, i.e., they must be processed by every machine and always in the same order (w.l.o.g., $m_1 \rightarrow m_2$). Moreover, recall that a so-called permutation flow shop (PFS) additionally requires an identical *job* order on all machines. It is known (see, e.g., [16, Lemma 6.8]) that unrestricted flow shop problems with makespan objective always have an optimal schedule in which the respective job sequences on the first two machines and that on the last two machines are the same. In the presence of blockages (i.e., if a task completed on one machine cannot start being processed on the next machine while another task is still being processed on it), this permutation property of optimal flow shop schedules needs no longer hold in general. However, for two machines, blockages obviously do not provide a way to circumvent the permutation property, i.e., $F_2|b = 1|C_{max}$ is necessarily identical to its PFS variant, $F_2|perm, b = 1|C_{max}$.

To now move to three machines, we note that [31, Lemma 1] observed that $F_2|b = 1|C_{max}$ (and thus, $F_2|perm, b = 1|C_{max}$) can be expressed equivalently as an instance of $F_3|blocking|C_{max}$ in which the processing times on the second machine are all zero, and “blocking” refers to allowing jobs to remain on a machine after processing if the next machine is still busy, and blocking the machine for other jobs while doing so (i.e., the possibility of the aforementioned blockages). In fact, by the above discussion, it is clear that the PFS variant $F_3|perm, blocking|C_{max}$ is equivalent for such instances as well, and therefore also NP-hard in the strong sense.

Finally, containment in NP of the decision version of $F_3|perm, blocking|C_{max}$ (asking whether the makespan does not exceed a given value) is trivial, which completes the proof. \square