Minimizing Airplane Boarding Time

Felix J. L. Willamowski\textsuperscript{1}\textsuperscript{*} and Andreas M. Tillmann\textsuperscript{2}

\textsuperscript{1} RWTH Aachen University, Lehrstuhl für Operations Research, Kackertstr. 7, D-52072 Aachen, Germany
\textsuperscript{2} Technische Universität Braunschweig, Institute for Mathematical Optimization, Universitätsplatz 2, 38106 Braunschweig, 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 rigorously define the problem and prove its NP-hardness. While this generally justifies the development of inexact solution methods, we show 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 a local improvement heuristic and an exact mixed-integer programming (MIP) formulation. Our numerical experiments with simulation data show that for several airplane cabin layouts, provably high-quality or even optimal solutions can be obtained within reasonable time in practice by means of our MIP approach. We also empirically assess the sensitivity of boarding strategies with respect to disruptions of the prescribed boarding sequences and identify robustness against such disruptions as a bottleneck for further improvements.

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 an 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., (Horstmeier and de Haan 2001, Figure 1) or (Ozmec-Ban, Babić, and Modić 2018, Figure 1)), passenger boarding is one of the steps that most affect the turn-around time (Ferrari and Nagel 2005). This is because all other steps either cannot be significantly shortened (e.g., taxing 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 (Marelli, Mattocks, and Merry 1998).

As revealed by Jaehn and Neumann (2015), 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.\textsuperscript{3} 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 fairly recent survey (Jaehn and Neumann 2015) on boarding methods.

\textsuperscript{*} Corresponding author; email: willamowski@or.rwth-aachen.de

\textsuperscript{3} We remark that in this paper, we do not take the present temporary challenges due to the global SARS-CoV-2 pandemic into account, and instead refer to pre-corona air transport operations, as they are widely assumed to reach similar activity levels again after the pandemic is overcome.
Somewhat surprisingly, it appears that previous works on reducing airplane boarding time have never 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 aisles or seat rows. Indeed, the need for research into ways of finding boarding sequences that actually minimize boarding time was pointed out in (Jaehn and Neumann 2015, Sect. 7.1.1).

In this paper, we introduce the (to the best of our knowledge) first rigorous mathematical optimization model, utilizing mixed-integer programming (MIP), for the explicit goal to minimize the overall boarding time. We furthermore introduce a novel heuristic (a time-aware variant of the well-known outside-in strategy) and a local-search post-processing routine to further improve a given boarding sequence. Moreover, although the airplane boarding problem has long been recognized as challenging, a formal proof of intractability was lacking. We close this gap by providing several computational complexity and (in-)approximability 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 (as may be used in practice) to “fully informed” knowledge of individual-passenger moving and settle-in times (as in simulations). With extensive numerical experiments, we demonstrate, in particular, the competitiveness of our heuristics and that our MIP allows to reliably solve (at least) medium-sized instances to exact optimality within reasonable time. Moreover, even if terminating the solution process early, the MIP approach still provides previously unknowable solution quality guarantees and approximation error bounds for heuristics. Indeed, while previous simulation studies to identify “the best” boarding strategies could only deliver relative solution quality assessments (i.e., how well a strategy performs relative to others), our approach allows to quantify solution quality in an absolute sense (i.e., compared to the actual optimum) for what appears to be the first time.

The paper is organized as follows: In the remainder of this section, we first briefly review the previous literature on airplane boarding strategies (Section 1.1), and then formally define the boarding time minimization problem treated in this paper (Section 1.2). In Section 2, we discuss the computational complexity of the problem, deriving strong NP-hardness results as well as approximation guarantees; for readability purposes, all formal proofs have been deferred to an Appendix at the end of the paper. Section 3 introduces a compact mixed-integer programming (MIP) formulation to solve 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. In further experiments, we also assess sensitivity/robustness of the boarding sequences w.r.t. time-data perturbations and disruptive passenger behavior, and (in Section 5) discuss and evaluate extensions that further increase model realism. Finally, concluding remarks are given in Section 6.

1.1 Related Work

Many different approaches for shortening the boarding time have been presented in the literature; the paper by Jaehn and Neumann (2015) 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 (cf. 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 (Horstmeier and de Haan 2001), a large body of works employs computer simulations to compare different boarding strategies over large sets of random instance data to identify those 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 (Marelli, Mattocks, and Merry 1998), 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 (Nyquist and McFadden 2008), 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 (2008) 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 (Milne and Kelly 2014). A ranking of simple boarding strategies including those just mentioned appears in (Kierzkowski and Kisiel 2017) (see also (Ozmec-Ban, Babić, and Modić 2018)); 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, where 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 (Zeineddine 2017) 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 an overhead bin or sitting in the way to the target seat (cf. (Ferrari and Nagel 2005)). 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 (Milne and Salari 2016), the claimed goal is to minimize the boarding time, but differing from the actual 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 (Bazargan 2007), similar to an earlier nonlinear MIP (MINLP) introduced in (van den Briel et al. 2005); 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, van den Briel et al. (2005) 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 itself.) Another MIP for interference minimization was put forth in (Soolaki et al. 2012), along with a genetic algorithm that sometimes achieved 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 (Milne, Salari, and Kattan 2018)), 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 (Ferrari and Nagel 2005), a broad simulation study involving early and late passengers, yet different from how we treat disruptions (cf. Section 4.3).

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 (Jaehn and Neumann 2015, Nyquist and McFadden 2008) 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 boarding order. Nevertheless, we leave practical details of how to enforce by-seat boarding sequences at the gate for future consideration.

Finally, we remark that minimizing airplane boarding time bears some resemblance to certain (permutation) flow shop scheduling problems with makespan (i.e., completion time) minimization objective; we refer to (Brucker 2007, Pinedo 2016) for details on machine scheduling, and will occasionally give pointers to related results at the appropriate places throughout the paper.

1.2 Problem Definition and Notation

We consider the airplane boarding problem (ABP) that asks for a sequence in which to board passengers with preassigned 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 (through a jet-bridge), and the cabin consists of a single deck with seats to either side of a single aisle. By boarding time, we mean the time between the first passenger entering the airplane cabin and the last passenger sitting down on their given seat. 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 rounds in our considerations. 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., $\mathbb{N} = \mathbb{Z}_{\geq 1}$). For a number $n \in \mathbb{N}$, we write $[n] := \{1, 2, \ldots, n\}$. 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 $1, 2, \ldots, r-1$), and the collection $\mathcal{S}$ of seats (given in the form $(r, s)$ for each $r \in \mathcal{R}$ with seat numbers $s$). We further require a set of passengers $\mathcal{P} := [\mathcal{P}]$, $\mathcal{P} \in \mathbb{N}$ with $\mathcal{P} \leq |\mathcal{S}|$, 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 their 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 $r \leq r(p) - 1$.

Now, the task is to find a permutation $\pi \in \Pi_{\mathcal{P}}$ (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., (Brucker 2007)), we denote the boarding time induced by a sequence $\pi$ by $C_{\text{max}}(\pi)$ (or simply $C_{\text{max}}$ if $\pi$ is clear from the context). Thus, we can abstractly express the ABP of interest as:

**Definition 1.1 (Airplane Boarding Problem (ABP)).** Given an airplane cabin layout $([\mathcal{R}], \mathcal{S})$, a finite set of passengers $\mathcal{P}$ with seat assignments $\sigma: \mathcal{P} \rightarrow \mathcal{S}$ and moving and settle-in times $t_{p,r}^{m}, t_{p}^{s} \in \mathbb{Q}_{\geq 0}$, find a feasible permutation $\pi$ of the passengers with minimum $C_{\text{max}}(\pi)$.

It remains to clarify what feasibility of a permutation means. First, consider the cabin layout: For each row $r \in \mathcal{R}$, there are two (possibly empty) ordered sets of seats $\mathcal{S}_{1}^{r} := ((r, 1), \ldots, (r, k_{1}^{r}))$ and $\mathcal{S}_{2}^{r} := ((r, k_{1}^{r} + 1), \ldots, (r, k_{1}^{r} + k_{2}^{r}))$, with $k_{1}^{1}, k_{2}^{1} \in \mathbb{Z}_{\geq 0}$; thus, $\mathcal{S} = \bigcup_{r \in \mathcal{R}} (\mathcal{S}_{1}^{r} \cup \mathcal{S}_{2}^{r})$. Each seat $(r, s) \in \mathcal{S}$ is accessible from the aisle by passing seats $(r, k_{1}^{r})$, $(r, s + 1)$ if $s \leq k_{1}^{r}$, or $(r, k_{1}^{r} + 1), \ldots, (r, s - 1)$ otherwise (i.e., for any $r$, seats $(r, k_{1}^{r})$ and $(r, k_{r}^{r} + 1)$ are the aisle seats, and $(r, 1)$ and $(r, k_{r}^{r} + k_{2}^{r})$ are the window seats). Moreover, by $\mathcal{P}(r)$ we will denote the set of passengers with seats in row $r$.

Furthermore, we presume that each passenger $p$ tries to go directly to their 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, or blocks, the row for the time period $[t, t + t_{p}^{s}]$ or $[t, t + t_{p}^{m}]$, 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 $[t, t + w]$.

To conclude this section, a few remarks on the ABP definition and instance data appear in order.
First, while there does not appear to exist a general closed-form expression for $C_{\text{max}}(\pi)$, we can evaluate the boarding completion time induced by any sequence $\pi$ recursively as follows. (The recursion is similar to that for the makespan of a permutation schedule in flow shops with limited intermediate storage, cf. (Pinedo 2016, p. 162); a proof by induction is provided in Appendix A).

**Lemma 1.1.** Given an ABP instance $(P, R, S, \sigma, \{t_p^m\}, \{t_p^s\})$ and a permutation $\pi \in \Pi_P$, it holds that $C_{\text{max}}(\pi) = \max_{r \in R} C(P, r)$, where $C(P, r)$ is defined by the following recursion over the times $C(i, r)$ at which a row $r$ becomes accessible again after being blocked by a passing or settling-in passenger $p \in \{\pi^{-1}(1), \ldots, \pi^{-1}(i)\}$:

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

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

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

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

**Remark 1.1.** In the context of simulations to evaluate boarding methods, the time data $(t_p^m, t_p^s)$ is typically drawn from a random distribution for each instance of many trials to be averaged. 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. Note that in practice, highly detailed time data is likely hard or even impossible to come by, and one has to work with simplifications and estimates that may be available from real-life observations, based on passenger demographics (cf., e.g., (Marelli, Mattocks, and Merry 1998)), or recent machine learning predictions (Schultz and Reitmann 2019). Nevertheless, the distinction between different resolutions of time data is still crucial to evaluate the empirical (simulation) performance of boarding methods based on simplified data versus the true optimal values achievable with fine-resolution data. Thus, the complexity results for different resolutions (Section 2) indicate, among other aspects, the necessary computational effort to obtain these optimal values, and our MIP formulation (Section 3) will provide a means to do so in practice, even for hard cases corresponding to the most realistic simulation data.

**Remark 1.2.** The aforementioned blockages and resulting waiting periods are what has been referred to as *aisle interferences* in the literature, see the earlier discussion of related work. For simplicity, throughout most of the paper, we do not explicitly incorporate further aspects such as *seat interferences* (which are arguably less important than aisle interferences, cf. (van den Briel et al. 2005, Nyquist and McFadden 2008, Kierzkowski and Kisiel 2017)) or *groups of passengers wishing to board together* (e.g., couples, or families with small children). Nevertheless, we will later show how to integrate such additional constraints into our exact ABP model and assess the impact on the exact (MIP) and some heuristic boarding schemes in numerical experiments, cf. Section 5.

## 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
Table 1. Overview of main theoretical and algorithmic contributions. 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; \( c^m, c^s, c^1, \ldots, c^m_P \) are constants in \( Q \geq 0 \). Here, \( k \) is the number of seats per row, \( H_k := \sum_{i=1}^{k} (1/i) \), and MIP refers to our universally applicable exact solution approach. The results for the polynomial schemes in the first three cases are shown to hold for planes fully booked to capacity.

<table>
<thead>
<tr>
<th>Case (for all ( p \in \mathcal{P}, r \in \mathcal{R} ))</th>
<th>Complexity</th>
<th>Approx./Algo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^* = c^* )</td>
<td>( \mathcal{O}(P) )</td>
<td>exact scheme</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>(Thm. 2.3)</td>
<td>outside-in (Thm. 2.3)</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>strongly NP-hard</td>
<td>(1 + ( k^{-1}H_k ))-approximation</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^* = c^* )</td>
<td>(Thm. 2.2)</td>
<td>max-settle-row (Thm. 2.6)</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m )</td>
<td>?</td>
<td>k-approximation</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>strongly NP-hard</td>
<td>outside-in (Thm. 2.4)</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>strongly NP-hard</td>
<td>MIP</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>strongly NP-hard</td>
<td>MIP</td>
</tr>
<tr>
<td>( t_{p,r}^m = c^m ), ( t_p^s ) arbitrary.</td>
<td>(Thm. 2.1, Thm. 2.2)</td>
<td>MIP</td>
</tr>
</tbody>
</table>

available. Reasonable estimates can be obtained from real-life observations (described in several studies, for instance in (van Ladeghem and Beuselinck 2002); cf. also novel on-plane measurement devices like the sensor network to monitor boarding processes described in (Schultz and Schmidt 2018) and references therein) and used to derive suitable (probabilistic) model assumptions, see, e.g., the simulation approaches in (Milne and Kelly 2014, Budesca, Juan, and Fonseca i Casas 2014). Nonetheless, as noted in (Jaehn and Neumann 2015), 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. After all, one can only actually judge how well heuristics using simplified time data really work if one can compare to optimal solutions based on detailed data, in simulations.

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. We remind that the technical proofs are all deferred to the Appendix.

2.1 Intractability Results

In general, the ABP 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 \( t_{p,r}^m \in Q \geq 0 \ \forall p \in \mathcal{P}, r \in \mathcal{R} \) and \( t_p^s = c^s \ \forall p \in \mathcal{P} \) (with \( c^s \in Q \geq 0 \) a constant):

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

In the above setting, the ABP is similar to permutation flow shop scheduling with blockages, which we exploit in the proof (see Appendix B). A different proof (cf. Appendix C) yields the companion result for the case \( t_{p,r}^m = c^m \ \forall p \in \mathcal{P}, r \in \mathcal{R} \) (with \( c^m \in Q \geq 0 \) a constant) and \( t_p^s \in Q \geq 0 \ \forall p \in \mathcal{P} \):
Theorem 2.2. It is NP-complete in the strong sense to decide whether a given ABP instance admits a boarding sequence \( \pi \) 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 \( T \).

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 \( P := |P| = |S| \). It is, however, easy to show that NP-hardness persists in case some seats are left empty or were filled earlier:

Corollary 2.1. The strong NP-hardness results of Theorems 2.1 and 2.2 remain valid in their respective settings even if \( P < |S| \), i.e., after pre-boarding and/or if the aircraft is not fully occupied.

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., (Garey and Johnson 1979) 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 reflects the inherent combinatorial 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 \( \pi \) with completion time \( C_{\text{max}}(\pi) \), should some \( t^m \)- or \( t^s \)-values actually be smaller than the data used to obtain \( \pi \), the resulting actual boarding completion time of \( \pi \) still is at most \( C_{\text{max}}(\pi) \).

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, one may neglect passenger differences and simply assume identical moving times and identical settle-in times for all passengers (i.e., \( t^m_p = c^m \forall p \in P, r \in R \) and \( t^s_r = c^s \forall p \in P \), 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., \( |S^1_r| + |S^2_r| = k \in \mathbb{N} \) for all \( r \in R \)) 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., (Jaehn and Neumann 2015)), formally defined below. Throughout, we will focus on outside-in boarding and a novel variant of it, since 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 window-to-aisle (left-right alternating) seat numbers (identifying the groups) and then by decreasing row number (within each group).

Assuming w.l.o.g. that \( k^1_r \geq k^2_r \forall r \in R \), the position in the outside-in boarding sequence of a passenger \( p \in P \) can be directly expressed as

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

(1)

Recall that in our notation, seats are numbered 1, 2, ..., \( k^1_r \) from window to aisle on one side, and then continue from aisle to window as \( k^1_r + 1, k^1_r + 2, ..., k^1_r + k^2_r \) on the other side, in any row \( r \); also, here, \( |S^1_r| + |S^2_r| = k \) implies \( k^1_r = k - k^2_r \), and the sequence of alternating seat numbers then is 1, \( k^2_r + 1 \), \( k^2_r + 2 \), ..., \( k^2_r + k^1_r \).

Theorem 2.3. A minimum-boarding-time solution to ABP instances with \( P = |S|, |S^1_r| + |S^2_r| = k \in \mathbb{N} \) for all \( r \in R \), \( t^m_r = c^m \in \mathbb{Q}_{\geq 0} \) for all \( p \in P \), and \( t^m_{p,r} = c^m \in \mathbb{Q}_{\geq 0} \) for all \( (p,r) \in P \times R \) can be computed in time \( O(P) \) by the outside-in strategy.
Lemma 2.1. For ABP instances with \( P = |S| \) and \(|S^1| + |S^2| = k \in \mathbb{N}\) for all \( r \in \mathcal{R} \), every boarding strategy producing a sequence \( \pi \) such that

\[
P(R - i + 1) = \{\pi^{-1}(i), \pi^{-1}(i + R), \ldots, \pi^{-1}(i + (k - 1)R)\} \quad \forall i \in \mathcal{R}
\]

(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

\[
\sum_{i=0}^{k-1} \max_{j \in \mathcal{R}} \left\{ \sum_{r=1}^{(r-1) \mathcal{P}} \left( \frac{r_{(p)} - 1}{r_{(p)} + 1} \right) + \frac{t^m_{p,r} + t^s_{p}}{r_{(p)} \mathcal{P}} + \frac{t^s_{p} + \min_{r_{(p)} \mathcal{P}} \sum_{r=1}^{r_{(p)} - 1} t^m_{p,r}}{r_{(p)} \mathcal{P}} \right\}.
\]

Lemma 2.2. For arbitrary ABP instances, the minimum boarding time is bounded from below by

\[
\max_{r \in \mathcal{R}} \left\{ \sum_{p \in \mathcal{P}} \left( \sum_{r_{(p)} \mathcal{P}} t^m_{p,r} + t^s_{p} + \frac{t^s_{p} + \min_{r_{(p)} \mathcal{P}} \sum_{r=1}^{r_{(p)} - 1} t^m_{p,r}}{r_{(p)} \mathcal{P}} \right) \right\}.
\]

Remark 2.2. It should be noted that in Theorem 2.3, we explicitly assume full plane occupancy, i.e., \( P = |S| \). In case some seats are unoccupied (or filled earlier, e.g., during priority boarding), it is currently still unknown whether the quality guarantee for outside-in boarding remains intact. On the other hand, Theorems 2.1 or 2.2 do not imply that ABP remains NP-hard in the setting of Theorem 2.3 with \( P \leq |S| \) either: While the intractability results can be extended straightforwardly to include seat vacancies, cf. Corollary 2.1, the data assumptions \( t^m_{p,r} = c^m, t^s_{p} = c^s \) of Theorem 2.3 are 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 \( P = |S| \)), because such dummy passengers would still be able to block others just like real passengers and could 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 \( P = |S| \) and \(|S^1| + |S^2| = k \in \mathbb{N}\) for all \( r \in \mathcal{R} \), the boarding time of any boarding sequence \( \pi \) with property (2) is at most a factor \( \beta \) worse than the optimal boarding time, where

\[
\beta \coloneqq \frac{(3)}{\text{max}\{(4), (5), (6)\}}.
\]

The main drawback of the general approximation ratio (7) is its dependence on instance-specific time values \( t^m_{p,r} \) and \( t^s_{p} \). 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 factor-k approximation (see Appendix I for the proof):

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

**Remark 2.3.** For the first case considered in Theorem 2.4, i.e., for $t^{m}_{p,r} = c^m_p \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in P \times R$ and $t^s_p = c^s_p \in \mathbb{Q}_{\geq 0}$ for all $p \in P$, one can also obtain the approximation ratio $1 + (\max p\epsilon c^m_p - \min p\epsilon c^m_p) / \min p\epsilon c^m_p$ 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 (proved in Appendix I) 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 $|S^1_r| + |S^2_r| = k$ for all $r \in R$, and $t^{m}_{p,r} = c^m_p \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in P \times R$, such that the outside-in boarding time is at least a factor $(2k - \varepsilon)$ worse than the optimal boarding time.

In fact, Proposition 2.3 can be generalized to all (non-random) strategies that do not explicitly take settle-in times into account. 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 by the following result (proved in Appendix J).

**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.

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 $r \in R$. The i-th group contains a passenger $p$ of each row with the i-th longest settle-in time of their 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 (Milne and Kelly 2014), which performs seat assignment based on number of carry-on items, targeting fast boarding with the Steffen method (Steffen 2008) applied afterwards. However, although the number of hand-luggage pieces naturally correlates 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 \( t^m_{p,r} = c^m \in \mathbb{Q}_{\geq 0} \) for all \( p \in \mathcal{P} \), the max-settle-row strategy reduces to outside-in boarding. Consequently, in this case, for ABP instances with \( P = |S| \) and \( |S^1_r| + |S^2_r| = k \in \mathbb{N} \) for all \( r \in \mathcal{R} \), max-settle-row gives an optimal solution if \( t^m_{p,r} = c^m \in \mathbb{Q}_{\geq 0} \) for all \( (p,r) \in \mathcal{P} \times \mathcal{R} \), and is a \( k \)-approximation algorithm if \( t^m_{p,r} = c^m \in \mathbb{Q}_{\geq 0} \) 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 \( P = |S| \), \( |S^1_r| + |S^2_r| = k \in \mathbb{N} \) for all \( r \in \mathcal{R} \), and \( t^m_{p,r} = c^m \in \mathbb{Q}_{\geq 0} \) 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 \( H_k := \sum_{i=1}^{k} (1/i) \).

The proof can be found in Appendix K. 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 the above theorems, the asymptotic behavior of both max-settle-row and outside-in boarding reflects the growing influence of moving times on the total boarding time in case of longer airplane cabins, eventually outweighing settle-in time contributions entirely:

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

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 mixed-integer programming to optimally solve ABP in general.

### 3 Exact MIP Formulation

In this section, we describe our compact exact linear MIP model for the ABP. We model the objective by using the index ordering corresponding to the boarding sequence in an inverse fashion; therefore, we abbreviate the MIP given below by (8)–(16) as **IPA** (inverted position assignment). It is worth mentioning that in a preliminary version of this paper, Willamowski and Tillmann (2019), we also experimented with alternative formulations using direct (non-inverted) assignments, predecessor relations, and a time-expanded multi-commodity flow network, but IPA turned out to be computationally superior to all of those. Thus, we only present IPA here. (We note also that IPA is conceptually related to, but different from, MIP models put forth for permutation flow shop scheduling problems, see, e.g., Wilson (1989), Jessin, Madankumar, and Rajendran (2020).)

The following variables are used to model a boarding sequence \( \pi \) (which corresponds to a permutation, or linear ordering, cf. (Coudert 2016)) and the associated boarding completion time:

- 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) \( \pi \), i.e., \( x_{p,i} = 1 \) if \( \pi(p) = 1 \) and \( x_{p,i} = 0 \) otherwise.
- Let \( t^a_{i,r} \in \mathbb{Q}_{\geq 0} \) be the arrival time variable representing the time at which passenger \( \pi^{-1}(i), i \in \{P\} \), arrives at row \( r \in \mathcal{R} \), and let \( t^f_{i,r} \) be the finishing time variable representing the time at which passenger \( \pi^{-1}(i), i \in \{P\} \), finishes their action at row \( r \in \mathcal{R} \).
Finally, we introduce the completion time variable \( C_{\text{max}} \in \mathbb{Q}_{\geq 0} \) 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 it (\( \tau_{p,r} := t_{p,r}^f \) for \( r = r(p) \)). For rows \( r \) that a passenger \( p \) needs not to visit at all, i.e., those behind their assigned seat, we may and do assume that \( \tau_{r,r} = 0 \). 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. (Nevertheless, note that times spent walking through jet-bridge could be incorporated easily as lower bounds on \( t_{i,1}^A \).

Naturally, every passenger has to be assigned precisely one slot in the sequence, yielding constraints (9) and (10) in IPA. We work directly with indices of the output sequence \( \pi := (1, 2, \ldots, P) \) and formulate the remaining constraints in terms of the \( i \)-th passenger in this sequence, \( i \in [P] \). Incorporating the assignment decisions, we can thus model the objective (8) (minimizing total boarding time) via (11), the requirement that rows are traversed consecutively by each passenger via (12) and (13), and that passengers cannot be overtaken 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. (Nevertheless, note that times spent walking through jet-bridge could be incorporated easily as lower bounds on \( t_{i,1}^A \).)

Note that (12) and (13) also ensure 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 time due to working with the output 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 (13) 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

\[
M_{p,r,i} := \max_{p' \subseteq \{p' \in \mathcal{P} | r(p') > r\}, \sum_{p' \in \mathcal{P}'} \left( \sum_{r' = r + 1}^{r(p') - 1} t_{p',r'}^m \right) + t_{p,r}^f.
\]

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 (11) is formulated w.r.t. the last row \( R \) (since we cannot directly access the index \( r(\pi^{-1}(i)) \) of the row in which the \( i \)-th passenger in the output sequence is seated).

4 Computational Experiments

We now turn to an experimental evaluation of the MIP formulation IPA and select boarding strategies. We compare the outside-in heuristic and Steffen’s method (Steffen 2008), our novel max-settle-row strategy as well as random boarding and a randomized variant of outside-in with our exact model. The first two methods had emerged in previous simulation studies as often being superior to other classical strategies that also do not explicitly take passenger moving and settle-in times into account; 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. Similarly, randomized outside-in boarding more closely mirrors how outside-in boarding is currently implemented in practice by several airlines (e.g., Lufthansa) by replacing the deterministic back-to-front row order by a random one. For any instance, we take the best one of 1000 random realizations as the respective outputs of the random boarding strategies (thus, the stated results for randomized schemes are, in fact, more optimistic than their average behavior).

Moreover, 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 \( \pi \) 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 \( \pi \) 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).

We implemented the boarding heuristics in Python 3.6, which was also used as a scripting language and to generate test instances. The MIP model (IPA) was solved with the commercial solver Gurobi (Gurobi Optimization, LLC 2019); 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.

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

4.1 Construction of Test Instances

For our test instances, we considered four different cabin layouts: \((R, k) \in \{(10, 2), (20, 2), (20, 4), (30, 6)\}\). The settle-in times were generated 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. The moving times were independently sampled from \{1, 2, 3\} with respective probabilities \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}. We created three instance sets \(mp-sp\), \(m-sp\), and \(mp-s\), each with 10 instances for each cabin layout (for a total of 120 instances). These 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 \(t_{m,r}^p \in \mathbb{Q}_{\geq 0}\) (i.e., only \(t_{m,r}^p = c_{p}^m\) and \(t_{m,r}^m = c_{r}^m\) are used).

With these instances, we evaluate the algorithm performances in the different settings, see Section 4.2. 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 4.3. 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 passengers disregarding their assigned positions, we let a total of 10% of passengers take different places within the boarding sequence. For this, we generate a collection of old/new position tuples (uniformly at random), delete the passengers in descending order of the old positions (w.r.t. the original boarding sequence) and reinsert them in increasing order of their respective new positions. 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 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}}, and 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 displacements, and perturbed time data—separately as well as all together (passenger displacements performed before repositioning late passengers), the latter representing the most real-world oriented scenario.

Furthermore, in light of Proposition 2.1, we also study 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.

### 4.2 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 10 s 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 MIP solutions). The column labeled “% impr.” gives the resulting average improvement percentages $(100\% \cdot (old \ value \ - \ new \ value)/(old \ value)$, 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 IPA model, respectively. The columns “% gap” and “# opt” provide the average optimality gap (100\% \cdot (best \ lower \ bound \ - \ best \ upper \ bound)/(best \ upper \ bound), rounded to two decimals) and the (absolute) number (out of 40) of instances that Gurobi/IPA solved to exact optimality within a time limit of two hours (7200 s). 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 difference between (deterministic) outside-in and max-settle-row boarding are notable but not overwhelmingly large, across all scenarios and 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. The
randomized outside-in strategy turns out to be much worse than its deterministic counterpart, with boarding times roughly three times larger before 2-opt post-processing and still nearly twice as large afterwards. This contrast alone emphasizes the drastic improvements that may be possible by actually enforcing by-seat boarding in practice, even if sticking to simple, structured strategies. Indeed, random outside-in boarding is apparently only slightly better than allowing a completely random boarding sequence (and sometimes even marginally worse after local search refinements). Overall, random boarding performs worst of all strategies, 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 performed), but the end results are still clearly inferior.

Regarding approximation 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 on the optimal boarding times, with which we can compute (average) empirical approximation ratios as $1 + (\text{heur. } C_{\text{max}})/(\text{IPA lower bound})$. For instance, on the m-sp testset (which appears to yield the easiest MIPs, cf. Table 3), max-settle-row is very close to optimal even without local search post-processing, with empirical average approximation ratio estimate of 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 bounds derived from the IPA optimality gap in Table 4, the approximation ratio is just about 1.09 empirically.

Let us now turn to the MIP formulation, IPA. From the tables, it becomes clear that IPA/Gurobi generally can only marginally improve over the best heuristic (usually max-settle-row + 2-opt) solutions, and apparently (for the larger instance) has 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 may nevertheless seem 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 by no means wasted, as the MIP lower bounds provide a hitherto unavailable way to gauge the effectiveness of boarding heuristics. Indeed, the
Table 4. Experimental results on testset mp-s. Numbers are average values over 40 instances; 10 each for 4 airplane sizes.

<table>
<thead>
<tr>
<th>method</th>
<th>objective</th>
<th>w/ 2-opt</th>
<th>% impr.</th>
<th>% gap</th>
<th># opt</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>1250.7</td>
<td>750.0</td>
<td>40.03</td>
<td>—</td>
<td>—</td>
<td>14.1</td>
</tr>
<tr>
<td>Steffen</td>
<td>628.4</td>
<td>584.9</td>
<td>6.93</td>
<td>—</td>
<td>—</td>
<td>9.4</td>
</tr>
<tr>
<td>rand. outside-in</td>
<td>1207.0</td>
<td>750.1</td>
<td>37.85</td>
<td>—</td>
<td>—</td>
<td>12.5</td>
</tr>
<tr>
<td>outside-in / max-settle-row</td>
<td>421.8</td>
<td>396.3</td>
<td>6.06</td>
<td>—</td>
<td>—</td>
<td>8.6</td>
</tr>
<tr>
<td>IPA</td>
<td>395.7</td>
<td>—</td>
<td>&lt;0.01</td>
<td>8.46</td>
<td>16</td>
<td>1007.1</td>
</tr>
</tbody>
</table>

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. This holds especially on the m-sp testset as 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, 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. For the hardest instances, corresponding to the largest cabin sizes, the average IPA optimality gap is 14.11%, and the resulting empirical approximation ratios for outside-in and max-settle-row with (without) 2-opt post-processing are still only 1.20 (1.36) and 1.17 (1.25), respectively.

Moreover, Gurobi/IPA can solve a relevant portion of the instances to provable optimality, albeit mostly for the smaller instances. Fixing either moving or settle-in times to the same constant value for all passengers renders 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, and the mean runtimes as well as the optimality gaps are significantly smaller. For completeness, we remark that on small instances, the aforementioned alternative MIP formulation based on predecessor-relations gives slightly better results than IPA on average, but on instances where it fails—in particular, for the (arguably more interesting) large instances—the performance is significantly worse, cf. Willamowski and Tillmann (2019).

Constant moving times (m-sp) appear to generally give the easiest-to-handle instances for the MIP model (or for Gurobi, applied to solve it, more precisely). This observation may have positive practical consequences: Moving times are small compared to settle-in times, unavoidable as they (unlike settling-in operations) cannot be “parallelized”, and should not differ drastically for different passengers. Thus, their influence on the overall boarding time is less significant, so if expectedly small fluctuations are neglected (e.g., taking a “pessimistic estimate” standpoint in view of Proposition 2.1), one may work with constant moving times and more quickly obtain better solutions or similarly accurate empirical approximation bounds for heuristics by solving IPA on m-sp data rather than on the most realistic mp-sp data.

Finally, it is worth mentioning that, on average, the IPA 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 IPA (12.37% average gap).

Thus, to summarize, 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” (deterministic) 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 formulation for the ABP and the resulting computational lower bounds on optimal boarding times. IPA generally provides highly useful lower bounds, and the fact that MIP can often only marginally improve the best heuristic solutions gives further indication that they often are likely already very close to optimal.
Table 5. Results of robustness experiments on testset mp-sp. Considered are disruptions caused by late passengers, passenger displacements, and for perturbed moving and settle-in times as well as all three combined. Numbers are average values over 40 instances; 10 each for 4 airplane sizes.

<table>
<thead>
<tr>
<th>method</th>
<th>obj.</th>
<th>late</th>
<th>displ.</th>
<th>pert. obj.</th>
<th>comb.</th>
<th>ref. obj.</th>
<th>% comb. loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>random + 2-opt</td>
<td>747.4</td>
<td>891.7</td>
<td>1000.5</td>
<td>833.1</td>
<td>1151.4</td>
<td>679.8</td>
<td>[60.7, 64.9]</td>
</tr>
<tr>
<td>Steffen + 2-opt</td>
<td>660.8</td>
<td>812.8</td>
<td>928.9</td>
<td>740.0</td>
<td>1077.2</td>
<td>644.6</td>
<td>[58.0, 62.5]</td>
</tr>
<tr>
<td>rand. outside-in + 2-opt</td>
<td>750.7</td>
<td>891.5</td>
<td>1002.4</td>
<td>838.1</td>
<td>1155.6</td>
<td>683.8</td>
<td>[60.9, 65.1]</td>
</tr>
<tr>
<td>outside-in + 2-opt</td>
<td>462.9</td>
<td>615.5</td>
<td>726.7</td>
<td>513.4</td>
<td>888.2</td>
<td>467.7</td>
<td>[49.1, 54.5]</td>
</tr>
<tr>
<td>max-settle-row + 2-opt</td>
<td>454.6</td>
<td>609.3</td>
<td>711.7</td>
<td>497.5</td>
<td>874.8</td>
<td>467.8</td>
<td>[48.3, 53.8]</td>
</tr>
<tr>
<td>IPA</td>
<td>452.7</td>
<td>607.4</td>
<td>711.0</td>
<td>500.5</td>
<td>880.5</td>
<td>464.1</td>
<td>[48.7, 54.1]</td>
</tr>
</tbody>
</table>

4.3 Robustness Experiments

For the experiments discussed in the following, we focus on the mp-sp data model (recall that even though such data is generally not available in practice, it allows for the most accurate simulation-based estimation of heuristic approximation ratios, via the MIP lower bounds). Since our 2-opt local search routine improved the output of all heuristics, here, we consider only the post-processed heuristic solutions, and of course the exact method (Gurobi applied to) IPA.

To assess robustness aspects of the boarding schemes w.r.t. different possible disturbances as outlined in Section 4.1, the experiments summarized in Table 5 show the impact on the boarding times (again in seconds, averaged over 40 instances, i.e., 10 instances each for all four airplane cabin sizes) of placing randomly chosen passengers at the end of the boarding queue (column “late”), of displacing random passengers to random new positions at odds with the predetermined boarding sequence (“displ.”), 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 “comb.”). 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 displacement-disruptions applied afterwards); for each instance, the smallest value here is an upper bound on the respective average ideal possible boarding time (computed with knowledge of the reference perturbed time data and no disturbance of the boarding sequence). Together with the lower bounds from IPA, we can estimate average loss intervals w.r.t. the ideal situation as incurred by the data uncertainty and disruptions, stated in the final column “% comb. loss”.

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. Displacing passengers 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. Displacements seem to lead to stronger deterioration of the solution quality than late passengers (although 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 close to 50% or more—with max-settle-row and (naturally) IPA yielding the lowest overall times, between about 48% and 54% larger than the ideal baseline/reference solutions (i.e., boarding sequences computed directly with the perturbed time data and without applying lateness- and displacement-disruptions afterwards) on average. Interestingly, in the present experiments, max-settle-row actually seems to be slightly less sensitive to the combined disruptions/perturbations than IPA; however, this effect is probably not significant, given that IPA is initialized with the best heuristic solution (usually from max-settle-row + 2-opt) and only marginally improves it. Random boarding, randomized outside-in and Steffen’s strategy are relatively more robust to the disturbances, but nevertheless the final (combined) boarding times are significantly inferior to those of the other three approaches.

On the one hand, the overall strong increase in the boarding times when exposing a computed boarding sequence to “realistic” changes of environment demonstrates that even then, the newly proposed strategies (max-settle-row, the 2-opt local search routine, and IPA) provide significant improvements over the others.
Table 6. Results for robustness experiments on testset mp-sp for pessimistic data assumptions and all disturbances. Numbers are average values over 40 instances; 10 each for 4 airplane sizes.

<table>
<thead>
<tr>
<th>method</th>
<th>pess.</th>
<th>true</th>
<th>ref. true</th>
<th>true, disr.</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>random + 2-opt</td>
<td>1088.4</td>
<td>908.6</td>
<td>679.8</td>
<td>1234.63</td>
<td>22.0</td>
</tr>
<tr>
<td>Steffen + 2-opt</td>
<td>875.7</td>
<td>747.3</td>
<td>644.6</td>
<td>1172.51</td>
<td>18.1</td>
</tr>
<tr>
<td>rand. outside-in + 2-opt</td>
<td>1102.4</td>
<td>922.4</td>
<td>683.8</td>
<td>1252.44</td>
<td>12.8</td>
</tr>
<tr>
<td>outside-in + 2-opt</td>
<td>578.8</td>
<td>512.5</td>
<td>467.7</td>
<td>980.28</td>
<td>21.2</td>
</tr>
<tr>
<td>max-settle-row + 2-opt</td>
<td>577.5</td>
<td>504.1</td>
<td>467.8</td>
<td>956.64</td>
<td>9.7</td>
</tr>
<tr>
<td>IPA</td>
<td>577.4</td>
<td>504.3</td>
<td>464.1</td>
<td>957.29</td>
<td>218.5</td>
</tr>
</tbody>
</table>

(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 performing the disruptions along with their comparatively less pronounced sensitivity with respect to perturbed time data 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 in this section, summarized in Table 6.

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 displacement-disturbances (the “true” data here is the perturbed time data from the experiments summarized in Table 5, see Section 4.1 for the details). In Table 6, the columns “pess.”, “true”, and “true, disr.” give the boarding times of the sequences computed using the pessimistic (overestimated) time-data, when evaluated on that same data, and on the true data (perturbed time values from previous mp-sp experiments) without or with late-passenger and position-change disruptions, respectively. The “ref. true” column states the boarding times achieved by the respective algorithms when directly using the true data, which serve as the reference ideal-situation best known solutions. The final column provides the (shifted geometric mean) runtimes of 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. Disregarding other disruptions, this indicates that 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 IPA), 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 strategies about or more than 50% larger than the ideal boarding times.

Hence, while a positive take-away message from these last experiments is that boarding planning (using the “right” strategies) is apparently not strongly 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 ABP Extensions

For the sake of simplicity, we have so far not included other possible aspects of the ABP in the theoretical and numerical results in this paper. For the two examples mentioned at the end of Section 1.2—seat inter-
Inseparable passenger groups—we will now briefly discuss how they can easily be incorporated into heuristics and the exact IPA model, and evaluate the algorithms’ performance with some further computational experiments. In particular, although the theoretical approximation bounds from Section 2.2 do not directly carry over to the modified (outside-in and max-settle-row) heuristics, the lower bounds from the MIP solver applied to the modified IPA model still allow us to obtain empirical estimates for the respective solution qualities. (We point out that the extended ABP is also strongly NP-hard in general, since at least the proof of Theorem 2.1 carries over straightforwardly.)

5.1 Incorporating Seat Interferences and Inseparable Passenger Groups

Let us begin by demonstrating how to modify IPA to capture the ABP refinements:

**Seat interferences:** Suppose we wish to take into account possible delays prompted by a passenger having to get up again to let another one move to a seat in the same row (but further from the aisle). A naive possibility would be to prohibit such situations altogether by enforcing an outside-in pattern: For any pair of passengers \((p_1, p_2) \in \mathcal{P}^2, p_1 \neq p_2\), seated in the same row \(r = r(p_1) = r(p_2)\), w.l.o.g. with \(s(p_1) < s(p_2) \leq k^1_r\), this can be ensured by adding the constraints

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

(17)

to IPA, i.e., to the MIP model (8)–(16); analogously for passenger pairs on the other side of the aisle. Note that even with such quite restrictive constraints, the MIP still allows for more general boarding sequences than the plain outside-in heuristic.

More generally, we can 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. \(s(p_1) < s(p_2) \leq k^1_r\)). To that end, suppose we are given data \(\eta_p \in \mathbb{Q}_{\geq 0} (p \in \mathcal{P})\) for the time passenger \(p\) needs to get up (to let someone pass) and settle in again. Let \(z_{i,p} \in \mathbb{Q}_{\geq 0}\) 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 their seat in row \(r\), conditioned on \(p\) being the \(i\)-th one to board. The constraints

\[
M_r(x_{p,i} - 1) + \sum_{p' \in \mathcal{P}(r)} \sum_{s(r') < s(p)} \eta_{p'} x_{p',i'} \leq z_{i,p} \quad \forall i \in [P], r \in \mathcal{R}, p \in \mathcal{P}(r) : s(p) < k^1_r
\]

(18)

ensure (for each \((i,r,p)\)) that \(z_{i,p}\) 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., \(M_r := \sum_{p \in \mathcal{P}(r)} \eta_p\)). Additionally, constraints (15) in IPA need to be modified to

\[
t^A_{ir} - t^A_{ir} \geq \sum_{p \in \mathcal{P}} (\tau_{p,r} + \mathbbm{1}_{r=r(p)} z_{i,p}) \quad \forall i \in [P], r \in \mathcal{R},
\]

(19)

where \(\mathbbm{1}_{r=r(p)} := 1\) if \(r = r(p)\), and 0 otherwise. Then, the overall time minimization brings each \(z_{i,p}\) down to the respective bound from (18) (if positive; to zero otherwise). Naturally, analogons of constraints (18) and (19) can straightforwardly be set up for the other side of the aisle as well.

**Inseparable passenger groups:** Typically, there are various groups of passengers who wish to board together (e.g., couples or families), which can be accounted for by adding 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 \(\{p_1, \ldots, p_{k}\} \subseteq \mathcal{P}\), we ensure that its members will be mapped as \(p_1 \mapsto i_1, \ldots, p_{\ell} \mapsto i_{\ell}\) with \(i_{j+1} = i_j + 1 (j = 1, \ldots, k - 1)\) by enforcing

\[
x_{p_{j+1},i+j+1} = x_{p_{j},i+j} \quad \forall i \in \{0, \ldots, P-k\}, j \in [k-1].
\]

(20)
In the more general case that the order within a group \( G \subseteq P \) is itself also to be optimized, let \( \mu^G_i \in \{0,1\} \), for \( i \in \{0,\ldots,P-|G|\} \), be decision variables indicating whether the group starts boarding at position \( i+1 \) (in the boarding sequence) or not. The following constraints then ensure that the group \( G \) boards together:

\[
\frac{P-|G|}{|G|} \sum_{i=0}^{P-|G|} \mu^G_i = 1, \tag{21}
\]

\[
\frac{1}{|G|} \sum_{p \in G} \sum_{j=1}^{|G|} x_{p,i+j} \geq \mu^G_i \quad \forall i \in \{0,\ldots,P-|G|\}. \tag{22}
\]

Indeed, note that exactly one \( \mu \)-variable must be 1 (21) and that \( \mu^G_i = 1 \) forces the \(|G|\) consecutive \( x \)-variables associated with the group \( G \) to 1 (22).

Finally, the big-M in (13) needs to be adapted as well; for simplicity, we may (and later do) use the objective value of the best-known (heuristic) solution here.

Regarding the heuristics, note that both deterministic and randomized outside-in strategies avoid seat interferences by construction, as does the Steffen method. This is not the case for the max-settle-row strategy, and forcing it would render the boarding scheme equivalent to outside-in. One might consider a modification that weighs estimated seat interference waiting times against settle-in times to decide in each round for each row whether to prioritize avoiding seat interferences or selecting a passenger with maximal settling-in time. However, doing so somewhat destroys the simplicity of the max-settle-row strategy and, assuming seat interferences are less time-intensive as settling-in actions, it seems quite unlikely to actually change the constructed sequence. Finally, note that if seat interferences were to be avoided during random boarding, one would simply end up with randomized outside-in boarding.

Inseparable passenger groups can be taken into account by all boarding schemes in several ways. Given the greedy-like nature of most boarding heuristics, we felt the most natural option was this: While constructing a boarding sequence according to some strategy, as soon as a member of a group is encountered, schedule the whole group (in an order that, if not fixed a priori, best matches the strategy’s intent, e.g., outside-in) rather than just the one member. Note that when inseparable groups are considered, seat interferences may occur even for strategies that otherwise avoid them.

5.2 Computational Experiments

For the sake of brevity, we restrict our evaluation of the modified boarding strategies to random and outside-in boarding as well as the exact Gurobi/IPA method. We conducted experiments with either just groups of inseparable passengers (the arguably most important refinement of the ABP in terms of realism) or additionally penalizing seat interferences, both in the basic and the disruption setup (cf. Section 4). To that end, we extended the IPA model with passenger group constraints (20) (a priori fixing the order within each group in outside-in fashion, which seemed reasonable) and seat interference constraints (18) and (19). Since the earlier experiments indicated that the differences between working with precise or loosely estimated time data are ultimately not severe, we do not also incorporate time data perturbations in the experiments here and instead focus on assessing the impact of late and displaced passengers (or passenger groups – indeed, note that rather than single passengers, whole groups of passengers are treated as late or displaced if any one of their members is selected as being so affected during instance generation).

The test instances are based on the same mp-sp data we used in the previous experiments, with the following additions: Seat interference waiting times \( \tau_p \) were set to 10 seconds for all passengers. Thus, the time a row is blocked by a passenger \( p \) settling in effectively increases by 10 s for each passenger that has to
Table 7. Experimental results on testset mp-sp with inseparable passenger groups. Numbers are average values over 40 instances; 10 each for 4 airplane sizes.

<table>
<thead>
<tr>
<th>groups</th>
<th>method</th>
<th>objective</th>
<th>w/ 2-opt</th>
<th>% impr.</th>
<th>% gap</th>
<th># opt</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>random</td>
<td>1951.4</td>
<td>1186.4</td>
<td>39.20</td>
<td>—</td>
<td>—</td>
<td>14.8</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>1369.3</td>
<td>947.8</td>
<td>30.78</td>
<td>—</td>
<td>—</td>
<td>11.6</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>929.6</td>
<td>—</td>
<td>1.88</td>
<td>35.63</td>
<td>11</td>
<td>2954.8</td>
</tr>
<tr>
<td>50%</td>
<td>random</td>
<td>2028.2</td>
<td>1272.4</td>
<td>37.27</td>
<td>—</td>
<td>—</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>1941.3</td>
<td>1129.3</td>
<td>41.83</td>
<td>—</td>
<td>—</td>
<td>11.5</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>1089.9</td>
<td>—</td>
<td>3.11</td>
<td>43.14</td>
<td>10</td>
<td>1985.2</td>
</tr>
<tr>
<td>75%</td>
<td>random</td>
<td>2058.5</td>
<td>1319.4</td>
<td>35.91</td>
<td>—</td>
<td>—</td>
<td>9.3</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>2438.4</td>
<td>1239.8</td>
<td>49.15</td>
<td>—</td>
<td>—</td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>1187.2</td>
<td>—</td>
<td>2.85</td>
<td>47.34</td>
<td>10</td>
<td>1996.5</td>
</tr>
</tbody>
</table>

Table 8. Experimental results on testset mp-sp with inseparable passenger groups and seat interference consideration. Numbers are average values over 40 instances; 10 each for 4 airplane sizes.

<table>
<thead>
<tr>
<th>groups</th>
<th>method</th>
<th>objective</th>
<th>w/ 2-opt</th>
<th>% impr.</th>
<th>% gap</th>
<th># opt</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>random</td>
<td>2018.4</td>
<td>1200.0</td>
<td>40.55</td>
<td>—</td>
<td>—</td>
<td>21.0</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>1369.3</td>
<td>953.6</td>
<td>30.36</td>
<td>—</td>
<td>—</td>
<td>15.8</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>937.1</td>
<td>—</td>
<td>1.72</td>
<td>36.31</td>
<td>10</td>
<td>2913.3</td>
</tr>
<tr>
<td>50%</td>
<td>random</td>
<td>2086.5</td>
<td>1262.8</td>
<td>39.48</td>
<td>—</td>
<td>—</td>
<td>17.3</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>1942.0</td>
<td>1134.6</td>
<td>41.58</td>
<td>—</td>
<td>—</td>
<td>16.2</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>1088.3</td>
<td>—</td>
<td>3.39</td>
<td>43.00</td>
<td>10</td>
<td>1900.3</td>
</tr>
<tr>
<td>75%</td>
<td>random</td>
<td>2113.7</td>
<td>1342.0</td>
<td>36.51</td>
<td>—</td>
<td>—</td>
<td>13.4</td>
</tr>
<tr>
<td></td>
<td>outside-in</td>
<td>2440.2</td>
<td>1247.4</td>
<td>48.88</td>
<td>—</td>
<td>—</td>
<td>12.7</td>
</tr>
<tr>
<td></td>
<td>IPA</td>
<td>1196.1</td>
<td>—</td>
<td>3.22</td>
<td>47.88</td>
<td>10</td>
<td>1923.6</td>
</tr>
</tbody>
</table>

get up to let \( p \) pass. For the groups, one has to consider both the total number and the sizes of groups as well as the associated seat assignments. To that end, we generated three test sets with 25%, 50%, and 75% of passengers belonging to groups, respectively, and the group sizes were distributed as 70% couples, 20% groups of three, and 10% groups of four people. More precisely, for an instance with \( P \) passengers, total percentage \( p_t \in \{0.25, 0.5, 0.75\} \) of groups and percentage \( p_g \in \{0.7, 0.2, 0.1\} \) of group size, the number of groups for a group size \( g \in \{2, 3, 4\} \) is given by \( P \cdot p^\prime \cdot p^g / g \), rounded to the nearest integer. For example, instances for the 20 × 2 cabin layout with 50% passengers in groups have 7 couples, one group of three, and one group of four passengers each, and instances for the 30 × 6 cabin with 75% passengers in groups all exhibit 47 couples, 9 groups of three, and 3 groups of four passengers. The seat assignments for the groups are generated as follows: Start with a free seat drawn uniformly at random, and while the number of chosen seats is smaller than \( g \), repeatedly select another seat with minimum total taxicab distance (w.r.t. row and seat index differences) to all seats already chosen for the current group. If there is more than one seat with the same distance, preference is given to one directly beside one of the chosen seats; if those are unavailable, the seat is selected uniformly at random otherwise.

In Tables 7 and 8, we summarize the experiments for instances with passenger groups and additionally taking seat interferences into account, respectively, and Table 9 provides an overview of the results for robustness experiments (considering both passenger groups and seat interferences). The columns are labeled analogously to the earlier tables in Section 4.

A first observation that is immediate from comparing Tables 7 and 8 is that seat interferences apparently do not have a large influence on boarding times overall. Indeed, the differences are essentially negligible across
Table 9. Results of robustness experiments on testset mp-sp with inseparable passenger groups and seat interferences. Considered were disruptions caused by late passengers, passenger displacements, and both combined. Numbers are average values over 120 instances; 30 each for 4 airplane sizes with 25%, 50%, and 75% of passengers in groups.

<table>
<thead>
<tr>
<th>method</th>
<th>obj.</th>
<th>late</th>
<th>displ.</th>
<th>comb.</th>
<th>% comb. loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>random + 2-opt</td>
<td>1268.2</td>
<td>1569.3</td>
<td>1432.2</td>
<td>1701.4</td>
<td>[36.9, 63.9]</td>
</tr>
<tr>
<td>outside-in + 2-opt</td>
<td>1111.9</td>
<td>1418.2</td>
<td>1299.4</td>
<td>1565.2</td>
<td>[31.4, 60.8]</td>
</tr>
<tr>
<td>IPA</td>
<td>1073.8</td>
<td>1387.7</td>
<td>1267.1</td>
<td>1537.0</td>
<td>[30.1, 60.1]</td>
</tr>
</tbody>
</table>

all group percentages and sizes and cabin layouts. Thus, these results can be seen to confirm earlier claims from the literature that seat interferences are less important than aisle interferences (see, e.g., van den Briel et al. (2005), Nyquist and McFadden (2008), Kierzkowski and Kisiel (2017)). To be fair, this impression might change if the seat interference waiting times are set to (much) larger values than the 10 s we assumed here, but probably not when keeping them at a realistic level (relative to all the other time data in an ABP instance). Note also that the average MIP solver runtimes are slightly lower with additional seat interference constraints than without.

The overall boarding completion times grow significantly compared to the setting where inseparable passenger groups (and seat interferences) are disregarded (cf. Table 2). Nevertheless, the main conclusions still apply in the extended context here: Our 2-opt local search post-processing significantly improves the sequences produced by random and outside-in boarding. The amount of improvement decreases for random boarding with increasing number of passengers assigned to inseparable groups, but increases for outside-in; in fact, when 75% of the passengers belong to groups, outside-in boarding itself yields worse boarding completion times than random boarding, but applying the 2-opt improvement heuristic remedies and reverses this situation. Overall, in all cases, outside-in boarding (+ 2-opt) still significantly improves over random boarding. Interestingly, Gurobi/IPA now achieves larger reductions (about 2-3%) in the boarding completion time compared to the best heuristic solution than we could observe in the experiments that disregard groups and seat interferences (only ca. 0.2%). On the downside, the IPA MIPs do not appear to become easier to solve (for Gurobi) with the additional constraints; indeed, the average gaps are clearly larger than in the earlier experiments, ranging from about 36% to roughly 48% compared to the 12.37% from Table 2, though the number of instances that were solved to optimality within the two-hour time limit is essentially the same. Judging from the overall average runtimes, the MIPs seem to be easiest when the percentage of passenger belonging to inseparable groups is about 50%. Nevertheless, the objective lower bounds provided by Gurobi/IPA show that empirically, the combination of (adapted) outside-in and 2-opt refinement has an empirical average approximation ratio of about 1.55 for the extended ABP problem.

Turning to Table 9, we can observe that for the extended ABP, the relative differences in final boarding completion times between heuristics and the exact IPA method (with time limit) are comparable to those in the earlier robustness experiments (cf. Table 5). Due to considering seat interferences and inseparable passenger groups, the boarding times are generally much larger overall, though outside-in still outperforms random boarding (after improving the respective boarding sequences with 2-opt local search), and Gurobi/IPA yields some further improvements. Interestingly, in the extended ABP, late-comers clearly have a more pronounced negative impact on the boarding times (from any method) than passengers that board in different positions than their assigned place in the respective computed boarding sequences. The combination of all such disruptions naturally yields the worst overall boarding completion times, which are between 30% and over 60% larger than the ideal boarding times (for undisrupted sequences); the sequences computed with Gurobi/IPA are now the most stable, incurring the smallest average losses compared to the ideal boarding times. The loss percentage intervals are wider than in the previous experiments (without passenger groups and seat interference considerations), with notably smaller lower bounds and comparable, slightly larger upper bounds. The spread is likely explained by the MIPs being harder to solve (yielding larger optimality gap estimates when terminating prematurely due to running into the time limit), but the smaller lower ends of the loss intervals are actually good news: If, as may be expected (since it is quite typical MIP solver behavior), the large gaps are mostly linked to the MIP solver having trouble increasing the
MIP objective lower bounds but the best-known solutions are already (close to) optimal, then the smaller
lower loss bounds compared to the basic ABP (see Table 5) indicate that the extended ABP is actually
more robust to the disruptions caused by late or displaced passengers. Although larger total boarding times
are unavoidable in the more realistic setting with (in particular) inseparable passenger groups, the further
loss due to disruptive passenger behavior would be only about 30% here, in contrast to almost 50% for the
basic ABP.

6 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. We proved strong NP-hardness for four out of six considered models of passenger
moving and settle-in times, 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 (encom-
passing all commonly used ones, including outside-in) can, in general, yield boarding times that are far
from optimal. Moreover, we proved 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 then developed the first algorithmic approach that directly tackles
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 a new MIP model with several heuristics; we were able to solve almost half of our
test instances to optimality within a two-hour time limit, 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. We also
assessed the robustness of the considered boarding methods with respect to perturbations of the passenger
time data and passengers disrupting the planned boarding sequence. Finally, we described how to adapt
the exact MIP model and boarding heuristics to an extended variant of the airplane boarding problem
that also takes inseparable passenger groups and delays due to seat interferences into account, along with
a computational evaluation.

The robustness experiments, in particular, revealed that there is still a large potential for improvements
that may be gained by incorporating robustness aspects explicitly. Thus, we believe it to be an interesting
topic for future research to tackle the (extended) boarding time minimization problem from the perspective
of robust optimization (see, e.g., (Ben-Tal, El Ghaoui, and Nemirovski 2009)). Moreover, it may be worth
spending some effort to devise a dedicated branch-and-cut solver for the ABP that incorporates bounds like
those from Lemma 2.2 locally, heuristics, and further MIP techniques (branching rules, cutting planes, etc.)
still to be investigated for the present problem. (Prior research on MIPs and heuristics for related permuta-
tion flow shop problems, e.g., (Wilson 1989, Jessin, Madankumar, and Rajendran 2020) and associated
references, might be a good starting point for such endeavors.) A more efficient MIP algorithm would open
the door to more extensive simulation studies, involving further heuristics and larger test instance sets.
Furthermore, it will be interesting to see how further aspects such as allowing passengers to store carry-on
luggage at arbitrary locations could be incorporated into the newly proposed MIP framework.

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
(near-)constant factor approximation algorithms, 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). Finally,
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
To show correctness, it suffices to prove the assertion that $C(i, r)$ is indeed the time row $r \in \mathcal{R}$ becomes accessible again after being occupied by a passenger $p \in \{\pi^{-1}(1), \ldots, \pi^{-1}(1)\}$. To that end, we proceed by induction over $i \in [P]$.

Since the very first passenger $\pi^{-1}(1)$ is never blocked, this clearly holds true for $C(1, 1), \ldots, C(1, R)$. For the induction step, consider passenger $p := \pi^{-1}(i)$ (and presume the $C(i, r)$-values are computed in increasing row-index order $r = 1, 2, \ldots, R$, i.e., following the progression of $p$ through the plane). For $r \leq r(p)$, $p$ can access row only after no predecessor $(\pi^{-1}(i - 1), \ldots, \pi^{-1}(1))$ 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 $t_{P,r}$ time moving on to the next row (if $r < r(p)$) or $t^*_{p}$ time settling in (if $r = r(p)$). In the former case, if row $r + 1$ is still occupied by any predecessor $(\pi^{-1}(i - 1), \ldots, \pi^{-1}(1))$ 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 $O(PR)$ to obtain $C_{\text{max}} = \max_{r \in \mathcal{R}} C(P, r)$, note that by iterating over passengers (in sequence $\pi$) and, for each passenger, over the rows (front to back), every required value is readily available (computed earlier or as given data) and computation of any $C(i, r)$ then reduces to a constant number of arithmetic operations. This completes the proof. \hfill \square

## B Proof of Theorem 2.1

The following result on a certain flow shop problem will be used in the proof of Theorem 2.1 below and may be of interest in its own right:

**Lemma 6.1.** The decision version of the three-machine permutation flow shop scheduling problem with blockages, $F_3|\text{perm}, \text{blocking}|C_{\text{max}}$, 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.** Proof. To show hardness, we begin with an instance of $F_2|b = 1|C_{\text{max}}$, a flow shop scheduling problem that was shown to be strongly NP-hard in (Papadimitriou and Kanellakis 1980) (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 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).

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, a so-called permutation flow shop (PFS) additionally requires an identical job order on all machines. It is known (see, e.g., (Brucker 2007, 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., \( F_2\{\text{perm}, b = 1\} | C_{\text{max}} \) is necessarily identical to its PFS variant, \( F_2\{\text{perm}, b = 1\} | C_{\text{max}} \).

To now move to three machines, we note that Hall and Srisvandarajah (1996, Lemma 1) observed that \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \) (and thus, \( F_3\{\text{perm}, b = 1\} | C_{\text{max}} \)) can be expressed equivalently as an instance of \( F_3\{\text{blocking} \} | C_{\text{max}} \) 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 \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \) 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 \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \) (asking whether the makespan does not exceed a given value) is trivial, which completes the proof.

\[\text{Proof.}\] Proof of Theorem 2.1. For rational data (moving and settle-in times, and \( T \)), containment in NP is easy to see: A “yes”-certificate is given by a permutation \( \pi \) of \( \mathcal{P} \), for which the boarding completion time \( C_{\text{max}}(\pi) \) can be computed in polynomial time by Lemma 1.1; thus, \( C_{\text{max}}(\pi) \leq T \) can be verified in polynomial time.

To show hardness, we reduce from \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \), the permutation flow shop problem with blockages, which is strongly NP-hard by the above Lemma 6.1. Recall that in this problem, we are given a set \( J \) of \( n \) jobs and three machines \( m_1, m_2, m_3 \), and 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 \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \) asking for a feasible permutation schedule with \( C_{\text{max}} \leq C \). By Lemma 6.1, 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 \( T := C \), \( \mathcal{P} := [n + 3] \), \( \mathcal{R} := [n + 3] \), \( S^1_r := \{(r, 1)\} \) and \( S^2_r := \emptyset \) for each \( r \in \mathcal{R} \), and \( t^*_{p} := 0 \) 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

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

We claim that this instance admits a boarding sequence \( \pi \) with boarding completion time at most \( T \) if and only if the input instance of \( F_3\{\text{perm}, \text{blocking} \} | C_{\text{max}} \) has a feasible solution \( \rho \) with makespan at most \( C \). Indeed, \( \pi \) without the three “dummy” passengers \( n + 1, n + 2, n + 3 \) is in one-to-one correspondence with \( \rho \): Suppose we have a permutation flow shop solution (i.e., job sequence identical for all machines) \( \rho = (\rho_1, \ldots, \rho_n) \) with \( C_{\text{max}} \leq C \). 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; it allows to cover the case of full plane occupancy). Conversely, a boarding sequence \( \pi \) with completion time at most \( T \) can be mapped directly to a desired flow shop schedule \( \rho \) by simply removing the dummy jobs. \( \square \)
C Proof of Theorem 2.2

To show hardness, we reduce from the 3-PARTITION problem, which is well-known to be strongly NP-complete, cf. (Garey and Johnson 1979). In this problem, we are given a finite set \( A := \{1, 2, \ldots, 3m\} = [3m] \) \((m \in \mathbb{N})\), an integer \( B \in \mathbb{N}_{>3}\), 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 \( A_1, A_2, \ldots, A_m\) 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,

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

\((k_r^2 = 0 \text{ and } S_r^2 = 0 \text{ for all } r \in \mathcal{R})\), 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

\[
    t_s^r := \begin{cases} 
        a_p, & 1 \leq p \leq 3m, \\
        B, & 3m + 1 \leq p \leq 4m,
    \end{cases}
\]

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 \( A_1, A_2, \ldots, A_m\) be such that \( A_1 \cup A_2 \cup \ldots \cup A_m = A\), and \( |A_j| = 3\) as well as \( \sum_{i \in A_j} a_i = B\) for all \( j \in [m]\). Furthermore, let \( A_j = \{i^j_1, i^j_2, i^j_3\}\) (arbitrarily ordered) for every \( j \in [m]\). We set

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

For any \( j \in [m]\), the group of four passengers \( \pi^{-1}(4j - 3), \ldots, \pi^{-1}(4j)\) 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 \( \pi^{-1}(1), \ldots, \pi^{-1}(4)\) are indeed not blocked and thus finish settling in at time \( B\). Inductively, since every fourth passenger in the sequence \( \pi\) sits in the first row and thus blocks all subsequent passengers (from getting on board) until having settled in, each passenger group \( \pi^{-1}(4j - 3), \ldots, \pi^{-1}(4j)\) with \( 2 \leq j \leq m\) has settled in at time \( jB\), respectively. Consequently, the total boarding completion time is exactly \( mB\).

\(\Leftarrow\): Let \( \pi\) be a boarding sequence for \( \mathcal{I}\) with completion time \( C_{\text{max}}(\pi) = mB\). We may and do assume w.l.o.g. that \( \pi(3m + 1) < \pi(3m + 2) < \cdots < \pi(4m)\), 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 \( \pi\), 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 $|P'| \geq 4$ or $|P'_j| \geq 4$ 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

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

if $\ell = 1$, or

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

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 $3m + 2, \ldots, 4m$ 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 $|P'| \leq 3$ and $|P'_j| \leq 3$ 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

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

Next, assume (again by contraposition) that either

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

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

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

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 $\pi$, 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 $P', P'_1, \ldots, P'_{m-1}$ 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

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

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. \qed

D Proof of Corollary 2.1

Since the seat assignments $\sigma$ 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. \qed
E Proof of Proposition 2.1

This can easily be seen by considering the boarding sequence \( \pi \) and propagating the changes in \( t^{m_1} \)- 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 \( \pi \) is a feasible (but not necessarily optimal) solution of the modified ABP instance, which therefore has optimal completion time \( C_{\text{max}} \leq C_{\text{max}}(\pi) \).

\( \square \)

F Proof of Lemmas 2.1 and 2.2 and Theorem 2.3

\textit{Proof.} Proof of Lemma 2.1. The first passenger \( p_1 := \pi^{-1}(1) \) in the boarding sequence needs time \( \sum_{r=1}^{R-1} t^{m_{p_1,r}} + t^{s}_{p_1} \) to finish boarding. The second passenger \( p_2 := \pi^{-1}(2) \) can pass a row \( r \leq R - 2 \) either directly in time \( t^{m}_{p_2,r} \) or in time at most \( t^{m}_{p_1,r+1} \) 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:

\[
\sum_{r=1}^{\pi^{-1}(i) - 1} \max \left\{ t^{m}_{\pi^{-1}(i),r}, t^{m}_{\pi^{-1}(i-1),r+1}, \ldots, t^{m}_{\pi^{-1}(1),R-1} \right\} + t^{s}_{\pi^{-1}(i)}.
\]

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 \( k \)) need for boarding. Clearly, the next group of \( R \) passengers can begin boarding \textit{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 \( C_{\text{max}}(\pi) \).

\( \square \)

\textit{Proof.} Proof of Lemma 2.2. Each passenger \( p \) has to pass rows \( 1, \ldots, r(p) - 1 \) before eventually settling in at row \( r(p) \). If a passenger is not blocked along the way, he/she needs time exactly \( \sum_{r=1}^{r(p) - 1} t^{m}_{p,r} + t^{s}_{p} \) 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 \( \sum_{p:r(p) \geq r+1} t^{m}_{p,r} + \sum_{p \in P_r} t^{s}_{p} \). Additional time is required for the first passenger who will block row \( r \) to pass rows \( 1, \ldots, r - 1 \) (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 is obviously bounded from above by \( \min \{ R, P \} \), (6) indeed provides a lower bound on the minimum achievable boarding time.

\( \square \)

\textit{Proof.} Proof of Theorem 2.3. With \( t^{m}_{p,r} = c^{m} \) and \( t^{s}_{p} = c^{s} \) 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 because the outside-in strategy clearly satisfies the requirement (2) from Lemma 2.1) reduces to

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

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 \( i = 0, 1, \ldots, k - 1 \), the maximum is attained for \( j \) such that \( r(\pi^{-1}(j + iR)) = R \). Since \( P = |P| = |S| = kR \), this upper bound coincides with the lower bound (5), which immediately shows optimality of the outside-in boarding sequence. For the running time bound \( O(P) \), note that we can compute the position of any passenger \( p \in P \) in the outside-in sequence directly using (1). Therefore, iterating over all passengers once is indeed sufficient to obtain the full boarding sequence.

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

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

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

\lproof

H Proof of Theorem 2.4
First consider the case $t^m_{p,r} = c^m_p \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in \mathcal{P} \times \mathcal{R}$, $t^s_p = c^s \in \mathbb{Q}_{\geq 0}$ 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

$$kc^s + \sum_{i=0}^{k-1} \max_{p \in \mathcal{P}, \sum_{s=1}^\infty (r(p)-1)c^m_p},$$

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

Now suppose $t^m_{p,r} = c^m_p \in \mathbb{Q}_{\geq 0}$ for all $(p,r) \in \mathcal{P} \times \mathcal{R}$, $t^s_p \in \mathbb{Q}_{\geq 0}$ (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 $k\max_{p \in \mathcal{P}}\{(r(p)-1)c^m + t^s_p\}$, which is again equal to $k$ times the lower bound (4).

\lproof

I Proof of Proposition 2.3
First suppose $P = |\mathcal{S}|$. Let $k \in \mathbb{N}_{\geq 2}$ and $0 < \varepsilon \leq k(k-1)/(3k-2)$, let $c^m, c^s \in \mathbb{Q}_{\geq 0}$ 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 $(r(p),s(p)) \in \mathcal{S}^1_{(p)}$ for all $p \in \mathcal{P}$. Only the passengers seated in the last row have positive moving times, namely $t^m_{p,r} = c^m_p$ 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 $t^s_p = c^s$ 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 their 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

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

(23)

times the optimal completion time. Let $R := [k(k-1)/\varepsilon] - k + 2$, $c^m := 1$, and $c^s := (R - k - 1)$. Then, $(R-1)c^m + (k-1)c^m = c^s$ and term (23) collapses to

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

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.
let settle in simultaneously. Again, an optimal boarding strategy clearly yields boarding time \( T \). were passenger \( p \in P(R) \) 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 their 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 \( t_{m}^s \) and \( t^s \)-values are zero. Then, the strategy yields boarding time \( 2T \), since row \( R-1 \) is blocked by another passenger \( p \in P(R) \) while passenger \( \pi^{-1}(1) \) is settling in. Consequently, we have

\[
U \leq k(R-1)c^m + H_k \max_{r \in R \subseteq P_{(r)}} \sum_{p \in P_{(r)}} t_p^s.
\]

Furthermore, the denominator of (24) can be bounded from below by

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

Consequently, we have

\[
\frac{U}{L} \leq \frac{k(R-1)c^m + H_k \max_{r \in R \subseteq P_{(r)}} \sum_{p \in P_{(r)}} t_p^s \left( + H_k(R-1)c^m - H_k(R-1)c^m \right)}{\max \left\{ k(R-1)c^m + \min_{r \in R \subseteq P_{(r)}} \sum_{p \in P_{(r)}} t_p^s, (R-1)c^m + \max_{r \in R \subseteq P_{(r)}} \sum_{p \in P_{(r)}} t_p^s \right\}}
\]
\[
H_k(R - 1)c^m + H_k \max_{r \in R} \sum_{p \in P(r)} t_p^s \\
\leq \frac{(R - 1)c^m + \max_{r \in R} \sum_{p \in P(r)} t_p^s}{k(R - 1)c^m + \min_{r \in R} \sum_{p \in P(r)} t_p^s} + \frac{(k - H_k)(R - 1)c^m}{k(R - 1)c^m + \min_{r \in R} \sum_{p \in P(r)} t_p^s}
\]
\[
\leq H_k + \frac{(k - H_k)(R - 1)c^m}{k(R - 1)c^m} = H_k + 1 - \frac{H_k}{k},
\]
which establishes the claimed approximation ratio \(1 + \frac{k-1}{k} H_k\).

\[ \square \]

### L Proof of Proposition 2.4

From the proof of Theorem 2.4, we know that

\[
k \max_{p \in P} \left\{ (r(p) - 1)c^m + t_p^s \right\} \leq k(R - 1)c^m + k \max_{p \in P} t_p^s \leq k(R - 1)c^m + kT
\]

is an upper bound on the outside-in boarding time. Since Lemma 2.1 also applies to max-settle-row, the bound (25) is actually valid for both boarding strategies. The same can be said about the lower bound from the proof of Theorem 2.6,

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

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

\[
\frac{k(R - 1)c^m + kT}{k(R - 1)c^m} = 1 + \frac{T}{(R - 1)c^m} \xrightarrow{R \to \infty} 1,
\]
which completes the proof.

\[ \square \]
Bibliography


Coudert D, 2016 A note on Integer Linear Programming formulations for linear ordering problems on graphs. Research report hal-01271838, Inria; I3S; Université Nice Sophia Antipolis; CNRS.


Jessin TA, Madankumar S, Rajendran C, 2020 Permutation flowshop scheduling to obtain the optimal solution/a lower bound with the makespan objective. Sādhanā 45(1):Art. No. 228.


