Real-time seat allocation for minimizing boarding/alighting time and improving quality of service and safety for passengers

Danial Yazdani[a,*], Mohammad Nabi Omidvar[b,*], Igor Deplano[c], Charly Lersteau[c], Ahmed Makki[c], Jin Wang[c], Trung Thanh Nguyen[c,**]

[a]*Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China.*

[b]*Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.*

[c]*Liverpool Logistics, Offshore and Marine Research Institute, Department of Maritime and Mechanical Engineering, Liverpool John Moores University, Liverpool L3 3AF, United Kingdom.*

Abstract

Rail is considered as one of the most important ways of transferring passengers. High passenger loads has implications on train punctuality. One of the important parameters affecting punctuality is the average boarding/alighting time. Organizing boarding/alighting flows not only reduces the risk of extended dwell time, but also minimizes the risk of injuries and improves the overall service quality. In this paper, we investigate the possibility of minimizing the boarding/alighting time by maintaining a uniform load on carriages through systematic distribution of passengers with flexible tickets, such as season or anytime tickets where no seat information are provided at the time of reservation. To achieve this, the proposed algorithm takes other information such as passenger final destination, uniform load of luggage areas, as well as group travelers into account. Moreover, a discrete event simulation is designed for measuring the performance of the proposed method. The performance of the proposed method is compared with three algorithms on different test scenarios. The results show the superiority of the proposed method in terms of minimizing boarding/alighting time as well as increasing the success rate of assigning group of seats to group of passengers.

Keywords: Seat allocation, discrete event simulation, heuristics, optimization, rail system, transportation.

Notations and Abbreviations

Table 1 shows the list of notations and abbreviations which are used in this paper and provides a brief description for each of them.

*The first two authors contributed equally to this work.

**Corresponding author

Email addresses: danial.yazdani@yahoo.com (Danial Yazdani), m.omidvar@cs.bham.ac.uk (Mohammad Nabi Omidvar), igor.deplano@gmail.com (Igor Deplano), C.L.Lersteau@ljmu.ac.uk (Charly Lersteau), a.a.makki@2016.ljmu.ac.uk (Ahmed Makki), J.Wang@ljmu.ac.uk (Jin Wang), T.T.Nguyen@ljmu.ac.uk (Trung Thanh Nguyen)

Table 1: List of Notations and Abbreviations.

List	Description
$\mathcal{U}$	Set of all processing units (passenger or clique).
$\mathcal{Q}$	Set of all cliques ($\mathcal{Q} = \{u \in \mathcal{U} : \lvert u\rvert > 1\}$).
u_i	The ith member of $\mathcal{U}$, which itself is a set.
$\mathcal{P}$	Integer set of passenger indices.
p_i	The ith passenger.
$\mathcal{A}_s$	Set of alighting passengers in station s.
$\mathcal{B}_s$	Set of boarding passengers in station s.
$\mathcal{M}_s$	Set of empty seats in station s.
$\mathbf{X}_s$	A binary matrix with dimensions $\lvert\mathcal{B}_s\rvert \times \lvert\mathcal{M}_s\rvert$ that maps $\mathcal{B}_s$ to $\mathcal{M}_s$.
$\mathcal{S}$	Integer set of all station indices.
$\mathcal{H}$	Integer set of all half-carriage indices.
$\mathcal{C}$	Set of carriages with at least one free seat or proper seat-group.
θ	Environmental parameters.
$\mathcal{C}'$	Candidate carriages.
$\lvert\cdot\rvert$	Cardinality of a set.
n	Total Processing units ($\lvert\mathcal{U}\rvert$).
m	Total Passenger number ($\lvert\mathcal{P}\rvert$).
$D(x)$	Index of the destination station for passenger x.
$A(x)$	Index of the arrival station for passenger x.
$H(x)$	Index of the carriage to which passenger x is assigned.
$L(x)$	Returns 1 if passenger x has big luggage and 0 otherwise.
α	A constant for determining the trade-off between the impact of boarding and alighting distribution.
β	Threshold for luggage ratio to clique size.
$B(u;\beta)$	Checks the existence of big luggage for a processing unit based on a threshold β.
$\pi(u)$	Returns the preferences of a processing unit.
$\eta_1(j)$	Number of passengers with big luggage assigned to carriage j in the current station.
$\eta_2(j)$	Number of passengers without big luggage assigned to carriage j in the current station.
$\eta_3(j)$	Number of passengers with big luggage assigned to carriage j who have the same destination.
$\eta_4(j)$	Number of passengers without big luggage assigned to carriage j who have the same destination.
CFR	Clique failure ratio.
ABAT	Average boarding/alighting time.
CA	Cellular automata.
RSA	Real-time seat allocation.
RND-I	Random seat allocation algorithm without clique support.
RND-II	Random seat allocation algorithm with clique support.
LPS	Last passenger sitting time.
FPA	First passenger alighting time.

1. Introduction

Rail travel occupies a significant share of long and short distance travel services around the world. The number of passengers choosing rail for their journeys is constantly increasing in many countries (uic.org, 2016, 2015). As a result, many train networks are struggling to be punctual. Reducing delays in preplanned dwell time in train timetables has a significant impact on punctuality (Coxon et al., 2015). In the rail system, dwell time in short stops is the duration of time a train stands at a platform for the purpose of allowing passengers to board and alight (Douglas, 2012). Therefore, after finishing the boarding/alighting operation, the trains immediately continue their journey (Li et al., 2016, 2018). In this paper, we focus on the journeys with short stops. For brevity, we use the term dwell time to refer to dwell time in short stops. Reliable estimation of dwell time allows the operators to forecast service capacity and runtime, which can be used to improve service punctuality (DAriano and Pranzo, 2009).

Many factors affect the dwell time among which passenger boarding/alighting time is the most significant one (Szplett and Wirasinghe, 1984, Wirasinghe and Szplett, 1984, Parkinson and Fisher, 1996, Daamen et al., 2008). In fact, the estimated dwell times for short stops in the train timetables are short (Li et al., 2016) which potentially put them at risk of delay in peak-hours due to increased boarding/alighting number of passengers. The crucial role of boarding/alighting time in the dwell time is reflected in a wide range of models proposed in the literature whose functions are designed based on the number of boarding/alighting passengers (Douglas, 2012, Harris, 2006, Harris and Anderson, 2007, Puong, 2000, Lam et al., 1998, Lin and Wilson, 1992, Aashtiani and Iravani, 2002).

At off-peak-hours, dwell time depends in principle on the train control strategy rather than the boarding/alighting time, because at off-peak-hours trains stop a fixed time at each station. On the other hand, at peak-hour when there are high passenger demands, the dwell time becomes more dependent on boarding/alighting operation (KFH-Group, 2013). Qi et al. (2008) investigated the influences of different group sizes and characteristics such as personal activity, tendencies, individual desires, and pressure from passengers behind on boarding/alighting time. The effect of pedestrian traffic management in the boarding/alighting time is studied in (Seriani and Fernandez, 2015).

Wirasinghe and Szplett (1984) showed that the dwell time is affected by the maximum number of boarding/alighting passengers at a door and the interaction (or friction) between them. Szplett and Wirasinghe (1984) showed that nonuniform distribution of boarding and alighting passengers is commonplace. Congestion at critical doors – i.e., the ones with the maximum number of boarding and alighting passengers – increases the boarding/alighting time (Daamen et al., 2008) as well as the likelihood of injuries (Hulse, 2013). Indeed, research shows that injury due to falling while boarding or alighting a train is the most common platform/train interface incident (Hunter-Zaworski, 2017). This clearly shows that optimizing boarding/alighting flows not only decrease the likelihood of delayed dwell time, but also has implications on the entire service performance including safety and customer satisfaction.

Boarding/alighting time depends on several factors such as passenger flow, passenger behavior, carriage design, physical design of the station, weather conditions, and operational factors (Thoreau et al., 2016, Puong, 2000, Douglas, 2012, Kraft, 1975, San and Masirin, 2016, Fernandez et al., 2010, Heinz, 2003). In this paper, we focus on minimizing boarding/alighting time by organizing the passenger flows. One important factor that can significantly affect the passenger flow is the chaotic behavior of passengers without preassigned seat information on their tickets, including season tickets, day travel tickets, and anytime tickets which are common in many countries such as the UK railway system (Nguyen et al., 2017). Lack of control over passengers with such tickets can potentially increase the boarding/alighting time, the risk of delays in dwell time, and the risk of injuries by creating the following undesirable situations:

- Unbalanced distribution of boarding/alighting passengers across carriages resulting in increased boarding/alighting time.

- Congestion at critical doors resulting in increased boarding/alighting time and the risk of injuries due to behaviors such as rushing and jostling.

- Unbalanced distribution of passengers with big luggage causing overloaded storage areas, forced movements between carriages for luggage storage, and aisle blockage.

- Increased movement of passengers and cliques between carriages to find empty seats due to overloaded carriages or lack of consecutive empty seats for cliques.

In this paper, we propose a real-time seat allocation (RSA) algorithm to control the distribution of passengers with free-seat tickets aiming at minimizing the boarding/alighting time across an entire route. This not only improves the service quality, but also minimizes the risk of extended dwell time allowing the trains to stick to their scheduled timetable. RSA is designed to run at entrance gates or self-service kiosks where passengers receive seat information including the seat, carriage, and door numbers upon scanning their tickets and providing simple information such as the number and type of luggage they carry. The incentive of using RSA is improved customer experience by decreased stress of finding available seat, decreased likelihood of injuries, reduced customer interaction and standing time, increased likelihood of finding consecutive empty seats for group travelers, as well as increased likelihood of finding empty luggage area.

Overall, RSA works by taking advantage of passengers with free-seat tickets to reduce passenger imbalance across carriages along an entire route. The RSA heuristic assigns seat information including seat, carriage, and door numbers by taking the following four parameters into account: 1. Clique size (equals one for a single passenger); 2. Number of big luggage; 3. Arrival station; and 4. Departure station. RSA performs the seat allocation procedure in two phases: 1. Assigning a carriage door for boarding based on distributing passengers according to the above mentioned information, and 2. Finding a proper seat or a group of seats within a candidate carriage according to the passengers' destination station based on a sweeper approach.

To simulate the problem and evaluate the efficacy of the proposed seat allocation algorithm, we also propose a simulation system and a performance indicator to compare and evaluate RSA against other algorithms. The simulation software is a discrete event-based system using cellular automata (CA) (Chopard and Droz, 1998) and queues (Allen, 1990) to model passenger movement for measuring the performance of various seat allocation algorithms including the proposed RSA. In addition to RSA, we also propose a greedy algorithm that resembles the situation in which passengers do not have any seat information and try to find free seats by themselves. Moreover, we introduce a new performance indicator, Clique Failure Ratio (CFR), which is related to customer satisfaction and indicates the ratio between the number of cliques whose passengers cannot sit together to the total number of arrived cliques. RSA is tested alongside a greedy and two random versions of seat allocation algorithms on different scenarios on the simulation software for multi-carriage trains on multi-station journeys. The simulation results show that RSA performs seat allocation with significantly better performance in terms of minimizing CFR and average boarding/alighting time (ABAT) as compared to other methods.

Seat allocation problem is an important issue which is investigated in most transportation industries such as airlines (Nyquist and McFadden, 2008) and railways (Armstrong and Meissner, 2010). In the airline literature, there are studies on minimizing boarding time (Soolaki et al., 2012, Jaehn and Neumann, 2015) by considering passenger agility and the effect of carrying hand luggage (Tang et al., 2012, Milne and Salari, 2016, Qiang et al., 2014), clique (passengers traveling in a group) (Zeineddine, 2017, Yoon et al., 2010), clique and agility (Notomista et al., 2016), discount policies (Obeng and Sakano, 2012), and revenue management (Lan et al., 2015, Ma and Qiu, 2010, Yoon et al., 2012, Subramanian et al., 1999). However, the results of these studies cannot be readily applied to the railway industry due to intrinsic differences between airplanes and trains. For example, there are differences in the carriage structures such as the number of doors, number of carriages, seat configuration patterns, and luggage areas. Additionally, controlling the passenger ordering, which is possible in airlines, is not feasible for railways. Moreover, unlike air journeys, in rail journeys there are numerous intermediary stations along a route, which affect the overall boarding/alighting time.

In the train literature, although minimizing boarding/alighting time has been considered, revenue maximization has remained the prime concern in seat allocation (Hetrakul and Cirillo, 2014, Xie et al., 2013, Jiang et al., 2015, Wang et al., 2016, Sumalee et al., 2009, Li et al., 2010). (Ahn et al., 2016) analyzed passenger congestion as a function of boarding/alighting distribution, and proposed a real-time information system that showed the information about the expected loading factor of each carriage to the waiting passengers, aiming at distribution of passengers across the platform to attain a uniform load on the carriages. To the best of our knowledge, no study has been dedicated to minimizing boarding/alighting time for trains by means of a real-time seat allocation process aiming at passengers with free-seat tickets.

The remainder of this paper is structured as follows. In Section 2 the problem statement and the proposed simulation are described. The proposed RSA algorithm at its technical specifications are presented in Section 4. Comparison algorithms, performance indicators, and parameter settings are described in Section 5. Section 6 shows the experimental results. In the final section, the main findings and suggested directions for future work are summarized.

2. Problem Statement

The considered problem in this paper is seat allocation for passengers whose tickets do not have seat information. In its simplest form, the problem can be seen as an assignment problem where at station $s \in \mathcal{S}$, a set of boarding passengers $\mathcal{B}_s$ are to be assigned to a set of empty seats $\mathcal{M}_s$. The space of all possible assignments, or seat allocations, can be modeled using a binary matrix $\mathbf{X}_s$ with dimensions $|\mathcal{B}_s| \times |\mathcal{M}_s|$. The entry x_{ij} is set to one if the ith boarding passenger is assigned to the jth empty seat, and zero otherwise. Hence, at each station, one can search the space of all possible assignments $x_{ij} \in \{0,1\}, (i,j) \in \mathcal{B}_s \times \mathcal{M}_s$

with the aim of finding one that minimizes a given function $\tau(\cdot)$ measuring the overall dwell time. $\tau(\cdot)$ represents any model that maps the number of boarding/alighting information to the overall dwell time.

In order to find the optimal seat allocation across an entire route, the total dwell time should be minimized as a function of seat assignment:

$$\begin{aligned}
\underset{\mathbf{X}_1,\ldots,\mathbf{X}_{|\mathcal{S}|}}{\text{minimize}} \quad & \sum_{s\in\mathcal{S}} \tau(\mathbf{X}_s; \mathcal{A}_s, \mathcal{B}_s, \mathcal{M}_s, \theta), && (1) \\
\text{subject to} \quad & \sum_{i\in\mathcal{B}_s} x_{i,j} = 1, \forall j \in \{1,\ldots,|\mathcal{M}_s|\}, && (2) \\
& \sum_{j\in\mathcal{M}_s} x_{i,j} \leq 1, \forall i \in \{1,\ldots,|\mathcal{B}_s|\}, && (3)
\end{aligned}$$

where $\mathcal{S}$ is the set of all stations along a route, $\mathbf{X}_s$ is the assignment matrix, $\mathcal{A}_s$ and $\mathcal{B}_s$ are the set of alighting and boarding passengers p with their respective properties (such as clique membership, having big luggage, arrival station, and destination), $\mathcal{M}_s$ is the set of empty seats, and θ captures the model parameters. The purpose of the two constraints is to force a one-to-one mapping between the boarding passengers and the empty seats.

The above formulation assumes that the boarding and alighting passengers p are known across an entire route. However, in a real-time seat allocation system, the information of boarding and alighting passengers p beyond the current station is not known. Therefore, without a model to predict the boarding and alighting dynamics of passengers over time for all stations within a journey, it is not possible to solve the above seat allocation problem as a classic optimization problem. In other words, a particular seat allocation at the ith station ($\mathbf{X}_i$) has bearings on the seat allocation at the jth station ($\mathbf{X}_j$). This is why the minimization problem in (1) must be simultaneously solved for all $\mathbf{X}_1,\ldots,\mathbf{X}_{|\mathcal{S}|}$ in order to find the global optimal solution.

To cope with the above issues, we propose a seat allocation *heuristic* which improves the boarding/alighting time by reducing the imbalance among the load of carriages, taking into account the impact of passengers with big luggage.

The second objective considered by the algorithm is to minimize the clique failure which happens when a group of co-travellers cannot be seated together:

$$\text{minimize} \quad \sum_{q\in\mathcal{Q}} F(q), \tag{4}$$

where $\mathcal{Q}$ is the set of all cliques, and $F(\cdot)$ checks clique failure by returning one if the members of a clique q are not seated together and zero otherwise. It should be noted that this objective has conflict with the previous objective defined in (1), the details of which is further discussed in Section 6.

To summarize, the goal of this paper is to devise a real-time seat allocation heuristic which improves the two objectives defined in (1) and (4). A consequence of minimizing the above two objectives is to reduce the risk of delays in dwell time, improve customer experience by decreasing the stress in finding an empty seat, decrease the likelihood of injuries, and reduce customer blocking and standing time. To simulate the problem and evaluate the performance of the algorithm on minimizing boarding/alighting time, a discrete event-based simulation is designed, which is presented in the next section.

3. Simulation

3.1. Assumptions

The assumptions presented in this section are to avoid unnecessary complications in the simulation software while maintaining a good degree of resemblance to real-world situations. The assumptions can be classified into three groups: simulation related, passengers/cliques related, and carriage related.

1. Simulation

   (a) At each timestep of the simulation, a passenger may arrive with a constant probability at each station. This does not mean a uniform load across all stations.

   (b) The maximum number of passengers on board is bounded by the number of seats. In other words, we do not consider standing passengers during journeys.

   (c) The dwell time is longer than the boarding/alighting time. Therefore, the train does not depart the station before boarding/alighting completion.

   (d) The CA cells cannot be shared between passengers.

   (e) The carriage doors can be crossed by one person at a time.

   (f) For each door, the boarding will start after the alighting is completed.

   (g) Check-in is done at the entrance gate.

2. Passengers and cliques

   (a) Passengers with big luggage can be determined at the entrance gate. In real-world situations, this can be done using different ways, for example:

   - Declaration by passenger by means of self check-in kiosks.
   - Having separate gates for passengers with big luggage. For example, this is common practice in big stations in the UK.

   (b) Each passenger can carry maximum one piece of big and one piece of hand luggage. These can be changed in the simulation software.

   (c) Passengers walk at a constant speed, and there is neither a panic nor a rush situation.

   (d) Passengers have knowledge of approximate door locations before train arrival.

   (e) All clique members have the same destination.

   (f) Clique members arrive at the gate together. Furthermore, according to the investigated circumstance in this paper where passengers' tickets do not consist any seat information, the possibility of having a clique with more than four passengers is very low (Nguyen et al., 2017).

   (g) Passengers do not have any special preferences about seat location.

   (h) In the middle stations, all passengers arrive at the platform before train arrival.

(i) Passengers use the same door for boarding and alighting.

(j) Passengers do not change seats during a journey.

3. Carriage and platform

(a) The entrance gate is next to the 1st carriage.

(b) The carriage plan is based on Figure 1 and it has 72 seats.

(c) All seats in carriages are forward facing and there are no table seat.

(d) There is no stair or considerable gap at the carriage doors.

(e) The luggage area has infinite space.

(f) The carriages are of economy class.

(g) Each carriage has two active doors for alighting and boarding.

3.2. Simulation of Boarding and Alighting in a Train System

In order to simulate the problem and create a test bed for the proposed seat allocation algorithm, we need to have an adequate simulation for multi-carriage trains in multi-station journeys. The simulation is divided into two separate parts: inside carriages and the platform. The inside part is CA-based and the platform works based on queues.

3.2.1. Simulation Parameters

In the simulation, the information of passengers is generated by several random number generators. The information of each passenger forms the first group of simulation parameters:

- Arrival Station: the station where the passenger starts his/her journey.
- Arrival Time: shows when the passenger joins the queue of the entrance gate.
- Destination Station: passengers' final destination station.
- Clique Size: number of individuals traveling in a group. Clique of size one represents a single passenger.
- Agility Parameter: the speed of a passenger in completing an actions such as walking, stowing, and collecting luggage. This parameter can be exploited to model slower passengers such as elderly or disabled people.
- Hand Luggage: small-sized luggage items that passengers stow in overhead lockers.
- Big Luggage: such as big suitcases, bicycles, and baby prams, which need to be stowed in designated areas.
- Seat Number, Carriage Number, and Door Number: these parameters are the output of the seat allocation algorithm.

The second group of parameters are related to times and durations. These parameters indicate the number of timesteps or epochs for performing a certain action by a passenger with average agility. Therefore, a function will set the value of these parameters for each individual based on his/her agility level. A list of these parameters is given below:

- Walking from row to row: it shows the average number of timesteps that walking between two adjacent rows takes.
- Stowing big luggage: the average number of timesteps for stowing a big luggage in a luggage area.
- Collecting big luggage: the average number of timesteps for collecting big luggage from a luggage area.
- Stowing hand luggage: the average number of timesteps for stowing hand luggage in hand baggage compartments above the seats.
- Collecting hand luggage: the average number of timesteps for collecting a hand luggage from hand baggage compartments above the seats.
- Moving between aisle seat and window seat: the average timesteps for moving from a seat to the next empty one (from window to aisle seat or vice versa).
- Sitting on an aisle seat: the average number of timesteps needed by a passenger to sit on an aisle seat from its neighboring position in the aisle.
- Sitting on a window seat with interference: this is for a passenger who wants to sit on a window seat but there is an interference due to a nonempty aisle seat.
- Standing up from a window seat with interference: this is for passengers on a window seat who wants to stand up and go to the aisle and there is an interference due to a non-empty aisle seat.
- Standing up from an aisle seat: this is the average timesteps needed by a passenger on an aisle seat who wants to stand up and move to the next position in the aisle.
- Passing the entrance gate: the number of timesteps that a passenger spends to pass the entrance gate in which the seat allocation procedure is done.
- Passing a carriage: the average timesteps it takes a passenger to walk through platform and pass a carriage.
- Open time: the number of timesteps that the entrance gate is opened before the train arrives at the station.

The third group of parameters are those related to the journey and the train:

- Number of stations.
- Number of carriages.

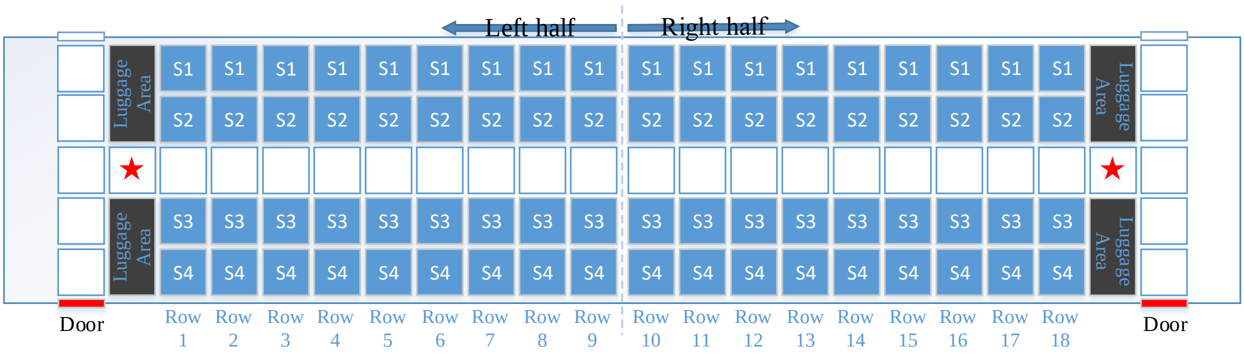

Figure 1: Carriage map in the simulation.

3.2.2. Carriage Plan

The map of a typical carriage that we adopted for our simulations is given in Figure 1. This carriage has 18 rows of seats, each having four seats, resulting in a total of 72 seats. The seats are shown with filled blue squares and the aisle blocks are shown with white squares with blue outlines. We do not consider forward and backward facing seats as well as table ones. The carriage has four doors (two on each side of the carriage), but the ones adjacent to the platform are active during boarding/alighting. The luggage areas that are utilized for storing big luggage are located next to the doors and passengers can access them from the cells marked with a red star.

3.2.3. Simulation Structure and Procedure

In the simulation, each carriage is represented by a CA (Chopard and Droz, 1998) with 110 cells (5×22) according to Figure 1. The neighborhood definition is as follows: in each row, seats 1 and 2 are each other's neighbors (similarly for seats 3 and 4). Additionally, there is an aisle block in the middle of each row, which is assumed to be an immediate neighbor of all four seats in the same row. Moreover, each block in the aisle neighbors the previous and the next aisle blocks. The simulation works based on a timestep counter and the status of all cells in CA are updated simultaneously.

The activities of a passenger start from the entrance gate when the event of `passenger arrival` occurs. At the entrance gate, there is a queue with a server (Allen, 1990) in which the seat allocation algorithm runs. The service time, including inputting information, the computation of the seat allocation algorithm, and printing seat information is constant for all passengers. The queue is processed in a first come first serve fashion. The generation of arriving time of passengers is made by a random number generator with Poisson distribution in a range that matches the gate's active period (Open Time). When a clique arrives, they are serialized and their arriving times are set accordingly. For generating the group size, the algorithm uses a uniform random number generator.

After receiving the seat information from the entrance gate, the passengers go to their assigned doors, or to their expected locations, prior to train arrival and form a queue. The order of passengers on these queues may be different form the order in which they arrived at the entrance gate. For each passenger, the movement time is calculated based on the distance between the entrance gate and the designated door, as well as its agility parameter. For a clique of passengers, the agility parameter is set according to its slowest member.

When the train arrives, the passengers start boarding one at a time by entering the carriages (stepping unto the first cell of CA). The passengers inside a carriage can have four different states: `walking`, `waiting`, `acting`, and `seated`. Passengers are in the `walking` state when they are in the aisle and their next aisle position is either empty or filled with another walking passenger. A passenger enters the `waiting` state when he/she cannot go to the next position due to another `waiting` or `acting` passenger. The countdown timer for passengers with the `waiting` state is stopped when their status changes. Passengers with an `acting` state are those who are stowing or collecting luggage, moving to their seat row from the aisle or vice versa. When a passenger is in the `acting` state, the other passenger whose next position is the current or next position of this passenger will enter to `waiting` state. Passengers in `seated` state are those whose current position is their assigned seat (cell).

In the simulation, no alighting occurs in the first station, and no boarding occurs in the last station. In the intermediary stations however, both boarding and alighting are performed. Boarding from a door starts only when the alighting of the same door is completed. For each passenger's current action, a countdown timer is set according to the time parameters. If the passenger is not in `waiting` status, then this count down timer works (it pauses for passengers in `waiting` status) and once it is equal to zero, the action is done or the passenger moves to the next cell.

4. Proposed Algorithm

In this section, we describe the details of the proposed real-time seat allocation (RSA) algorithm. The main purpose of this algorithm is to minimize the average boarding/alighting time (ABAT) to increase the service quality and safety as described in Section 1. Due to the nature of flexible tickets described before, RSA has no knowledge of the future arrivals. It therefore performs seat allocation according to its current information about boarded and arriving passengers, and the information provided by the user at the check-in such as big luggage information. RSA works based on half-carriages. This is because each carriage has two doors at its front and back, allowing it to be divided into two half-carriages. For brevity, we use the term carriage to refer to a half-carriage unless explicitly stated otherwise.

RSA has two major phases: 1. Choosing a door (a half-carriage) for a passenger/clique; and 2. Finding proper seat(s) in the chosen carriage from the first phase. The process of the first phase is done according to uniform distribution of passengers among doors (half-carriages) considering number of passengers with big luggage, the number of boarding passengers, and the number of alighting passengers in each station. In the second phase, seats are assigned to passengers according to the clique size and their destinations. Upon successful seat assignment, the system outputs *seat number*, *door number*, and *carriage number*.

4.1. Input Parameters

The system accepts passengers as input to initiate the seat allocation process. The processing unit of the algorithm is either a single passenger or a clique. The entire processing units are shown by the set $\mathcal{U} = \{u_1, \ldots, u_n\}$, where n is the total number processing units and u_i is a set representing a single passenger or a clique. If the cardinality of u_i is one (i.e., $|u_1| = 1$), it represents a single passenger and a clique otherwise. The set $\mathcal{U}$ can be flattened out to represent the entire set of passengers as follows:

$$\mathcal{P} = \{p_1, \ldots, p_m\} = \bigcup_{i=1}^{n} u_i, \tag{5}$$

where m is the total number of passengers and $m \geq n$. In the absence of cliques $m = n$. As an example, the following set represents a hypothetical set of processing units: $\mathcal{U} = \{\{p_1\}, \{p_2, p_3, p_4\}, \{p_5\}, \{p_6, p_7\}\}$.

In addition to processing units and passengers, the algorithm also requires the index of stations and carriages. Consequently, the set $\mathcal{S}$ contains the indices of all stations, and the set $\mathcal{H}$ contains the indices of all half-carriages. This means that if a train has k full-carriages, the cardinality of $\mathcal{H}$ will be $2k$. Passenger and clique related information such as the arrival station, destination station, existence of a big luggage, and the carriage they are assigned to are accessible with relevant functions that operate on a passenger:

$$D(x) : \text{index of the destination station of passenger } x, \tag{6}$$

$$A(x) : \text{index of the arrival station of passenger } x, \tag{7}$$

$$H(x) : \text{index of the carriage to which passenger } x \text{ is assigned}, \tag{8}$$

$$L(x) : \text{returns 1 if passenger } x \text{ has big luggage and 0 otherwise.} \tag{9}$$

4.2. The Seat Allocation Process

In this part, the procedure of RSA is described. RSA performs seat allocation in two main phases. The first phase determines a carriage in three steps. The first step distinguishes carriages in which there is at least an empty seat (for single passengers) or a proper seat-group (for cliques). In the second step, some candidate carriages are specified according to the balance maintenance between carriages. Finally, one carriage is chosen from the candidate carriages in the third step. After choosing a carriage in the first phase, a seat/seat-group is assigned to passenger(s) in the carriage using a sweeper based process.

4.2.1. Phase 1 (choosing a carriage)

RSA chooses a carriage for the processing unit u with the aim of uniformly distributing passengers among carriages based on their arrival station, destination stations, and whether they carry big luggage. Whether a processing unit is assumed to be carrying a big luggage is determined by the following equation:

$$B(u; \beta) = \begin{cases} 1 & \text{if } \frac{1}{|u|} \sum_{p \in u} L(p) \geq \beta \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $u \in \mathcal{U}$ is a set of size one or higher, and β is a threshold for the ratio between the number of big luggage items and the clique size.

When a processing unit applies for a seat, the algorithm utilizes four different values to determine a potential carriage from which a seat is allocated. These include:

- η_1: number of passengers with big luggage assigned to the jth carriage in the current station (i).

$$\eta_1(j) = |\{p \in \mathcal{P} : A(p) = i \wedge L(p) = 1 \wedge H(p) = j\}|, \tag{11}$$

- η_2: number of passengers without big luggage assigned to the jth carriage in the current station (i).

$$\eta_2(j) = |\{p \in \mathcal{P} : A(p) = i \wedge L(p) = 0 \wedge H(p) = j\}|, \tag{12}$$

- η_3: number of passengers with big luggage in the jth carriage having the same destination k.

$$\eta_3(j) = |\{p \in \mathcal{P} : D(p) = k \wedge L(p) = 1 \wedge H(p) = j\}|, \tag{13}$$

- η_4: number of passengers without big luggage in the jth carriage having the same destination k.

$$\eta_4(j) = |\{p \in \mathcal{P} : D(p) = k \wedge L(p) = 0 \wedge H(p) = j\}|, \tag{14}$$

RSA uses η_1 and η_2 in order to uniformly distribute the total boarding population (with and without big luggage) in each station across all carriages and doors to reduce the boarding time. Furthermore, RSA utilizes η_3 and η_4 for establishing equilibrium between the number of alighting passengers (with and without big luggage) among all carriages for each station. Next, RSA performs three steps to determine a carriage for a clique.

Step 1. In the first step of RSA, the procedure of finding a seat or a seat-group for the processing unit $u \in \mathcal{U}$ is done on all carriages ($\mathcal{H}$) and the ones with at least one proper seat or seat-group are chosen to form the set $\mathcal{C}$, which will be sent to the next step. For the purposes of this step, the term *proper* refers to the following two criteria: 1. Whether a carriage has a sufficient number of seats to accommodate u; 2. Whether the potential carriages contain seats that satisfy a preference criteria $\pi(u)$. For a single passenger ($|u| = 1$), all empty seats satisfy $\pi(u)$ by definition. For a clique however, $\pi(u)$ attempts to find the preferred seat-group to keep the members of a clique close to each other. Figure 2 shows different patterns of proper groups of seats that are used in this paper for cliques of size two, three, and four. For simplicity, the preferred patterns are restricted to those with a maximum size of four. Nevertheless, investigating what constitutes the best preferred pattern for cliques of different sizes is not the focus of this study. It should be noted that the algorithm is flexible enough so that $\pi(u)$ can be replaced with any other preference policy. For larger cliques, $\pi(u)$ recursively divides the clique in half until all resultant units are of size four or less. If a preferred group for a clique of size four is not found, the recursion continues to form cliques of smaller sizes and the empty seats in carriages are matched against the patterns in Figure 2 until a match is found.

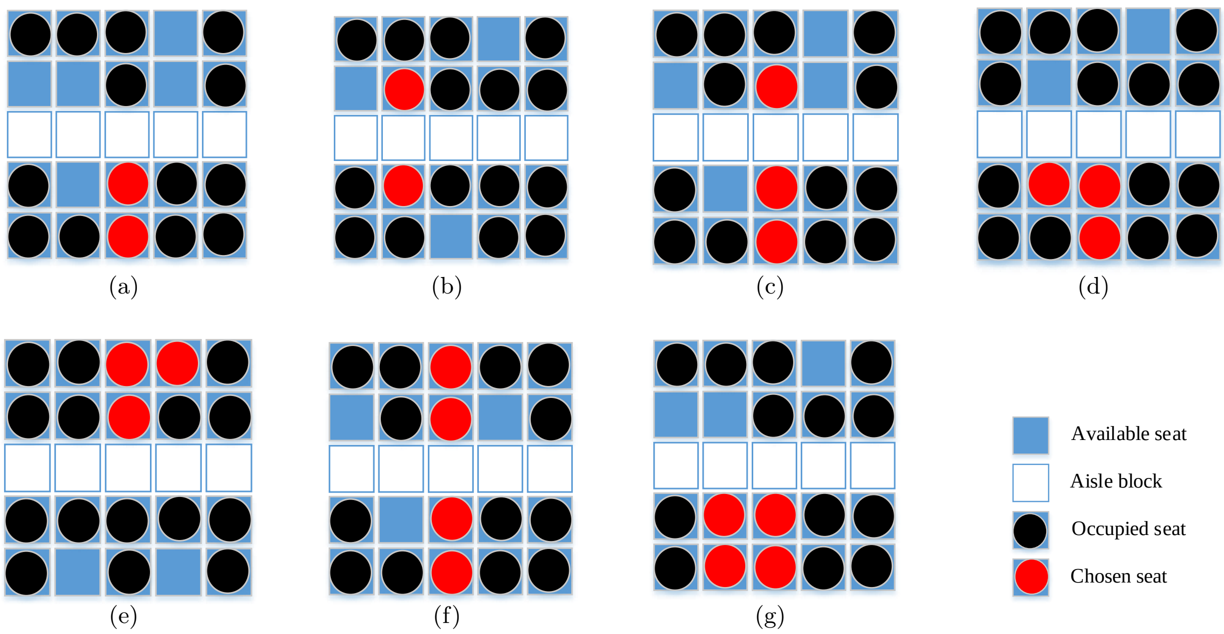

Figure 2: Proper group of seats for cliques with size of two, three and four passengers.

Step 2. In the second step, RSA is concerned with the distribution of passengers with the aim of maintaining a uniform balance among carriage loads. To do so, RSA processes $\mathcal{C}$ from the previous step to form a new set of candidate carriages $\mathcal{C}'$. The selection criteria to populate $\mathcal{C}'$ is as follows:

$$\mathcal{C}' = \begin{cases} \{c \in \mathcal{C} : \eta_1(c) \leq \ \min\{\eta_1(i) : i \in \mathcal{C}\} + \alpha\} & \text{if } B(u) = 1 \\ \{c \in \mathcal{C} : \eta_2(c) \leq \ \min\{\eta_2(i) : i \in \mathcal{C}\} + \alpha\} & \text{otherwise,} \end{cases} \tag{15}$$

where u is the unit being processed, and α is a constant determining the trade-off between influences of boarding and alighting passenger distributions in the process of choosing a carriage. The parameter α affects the number of candidate carriages in Eq. (15). A larger α increases the possibility of having a larger number of candidate carriages and vice versa.

Step 3. Finally in the third step, the final carriage is chosen from the set of all candidate carriages $\mathcal{C}'$ based on the following formula:

$$\text{Selected Carriage} = \begin{cases} \arg\min\limits_{c \in \mathcal{C}'}(\eta_3(c)) & \text{if } B(u) = 1, \\ \arg\min\limits_{c \in \mathcal{C}'}(\eta_4(c)) & \text{otherwise} \end{cases} \tag{16}$$

As described in the second step, α influences the contributions of considering distribution of boarding passengers and distribution of alighting passengers in determining the chosen carriage. If α is set to zero, the algorithm chooses the carriage with minimum η_1 (or η_2) from $\mathcal{C}$, which can be potentially more than one. On the other hand, the maximum feasible value for α is the total number of seats in a carriage (36 in this paper), which leads to selecting all carriages in $\mathcal{C}$ in the second step of the algorithm based on Eq. (15). As a result, the carriage with the lowest η_3 or η_4 is chosen among all $c \in \mathcal{C}$. Note that there can be more than one such carriages. If this the case, the carriage with the lowest η_1 or η_2 is chosen. Therefore, a lower value of α increases the effect of boarding distributions based on the smallest η_1 or η_2 of $\mathcal{C}$ on choosing a carriage for the arrival clique. Furthermore, a higher value of α increases the effect of the alighting distribution based on the smallest η_3 or η_4 of $\mathcal{C}$ in this process.

4.2.2. Phase 2 (choosing a seat or group of seats)

After determining a carriage in the first phase, RSA searches for the best potential seat within the selected carriage to minimize the alighting time based on the final destination of the unit being processed. To find a seat or seat-group, RSA uses a sweeper method that scans the rows to find the first available seat. The initial sweeper position (row) and the sweeping direction are determined by a policy based on how far the final destination of the processing unit is from the current station. The purpose of the sweeper algorithm and its movement policy is to assign the passengers with farther final destinations closer to the end of the carriage. Similarly, passengers with shorter journeys will be assigned to the seats closer to doors to minimize the total boarding and alighting time for all stations. In other words, the end of each carriage is for passengers with longer journeys. Similarly, the seats closer to doors belong to passengers with shorter journeys. Therefore, most of the boarding and alighting are done from the seats closer to the doors, resulting in reduction of the total alighting and boarding time. As a measure of journey length for passengers, the stations from the current one to the last one in a route is divided into four equal groups. The first group contains the stations closest to the current one and the last group (fourth) contains the stations farthest away from the current one.

To allocate a seat based on relative journey length, the sweeper can start from three different positions inside a carriage: the first row, the last row, and the middle row. If the destination of a processing unit (u) is the first group (closest), then the sweeper starts from the first row (closest to the door) and sweeps toward the end of carriage and allocates a seat or group of seats for the unit as soon as it finds one while it scans. It is clear that if $|u| = 1$ (single passenger), the first empty seat is chosen. For cliques ($|u| \geq 2$), the sweeper uses the sub-algorithm $\pi(u)$ from Section 4.2.1 (first step) for determining a proper seat-group while scanning the rows. If the destination of the unit lies in the second group, the sweeper starts from the middle row and scans toward the first row (carriage door). If there was no empty seat(s) in this part, it starts again from the middle row but this time toward the end of the carriage. For the third destination group, the sweeper starts from the middle row and moves toward the end of the carriage and if no seats are found, it restarts from the middle row and scans toward the first row. Finally, for the fourth group (farthest away), the sweeper starts from the last row and moves toward the first one.

If the number of remaining stations is not divisible by four, then the RSA tries to round them and assign each station to the nearest group. Additionally, if the number of the remaining stations is less than four, the algorithm changes the sweeper plan as follows:

- If there are three remaining stations, RSA maps them to the first, second and fourth groups, so that we do not use the third group procedure.
- If there are two remaining stations, RSA maps these two stations to the first and third groups.

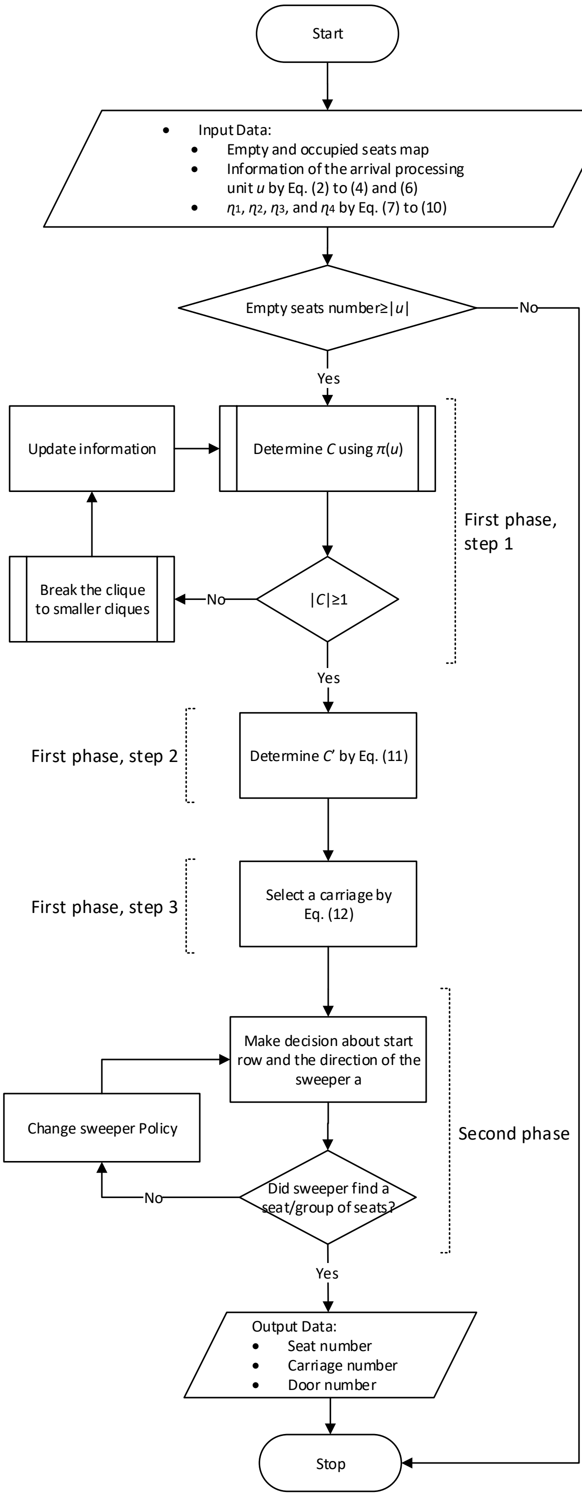

Figure 3: The flowchart of RSA.

- Finally, if there is only one remaining station, then RSA maps it to the first group procedure.

The flowchart of the proposed algorithm is illustrated in Figure 3.

5. Experimental Settings

In this part, two performance indicators for measuring the efficiency of RSA and comparing the performance of different algorithms are described in Section 5.1. Additionally, three comparison algorithms including a greedy, and two random based methods are explained in Section 5.2. Finally, the parameter settings of algorithms and the simulation system are presented in Section 5.3.

5.1. Performance Indicators

In this paper, two different performance indicators are used for measuring the performance of seat allocation algorithms. The first one is the averation boarding/alighting time (ABAT) across all stations, and the second one is the clique fail ratio (CFR). To calculate ABAT, we assume that alighting starts when passengers move from their seats toward carriage doors and boarding finishes when the last passenger sits on his/her seat:

$$\text{ABAT} = \frac{1}{|\mathcal{S}| - 2} \sum_{i=2}^{|\mathcal{S}|-1} (\text{LPS}_i - \text{FPA}_i), \tag{17}$$

where $|\mathcal{S}|$ is the number of stations, LPS_i represents the last passenger sitting time at the ith station, FPA_i is the alighting starting time of the first passenger at ith station. As can be seen, the first and the last stations are excluded from ABAT calculation because the boarding time in the first station and the alighting time in the last station are not very restrictive since the train usually waits longer.

The second performance indicator is CFR that shows the ratio between the number of failed cliques, i.e. those that the algorithm failed to seat together, and the total number of cliques with two or more passengers:

$$\text{CFR} = \frac{\text{\# of failed cliques}}{|\mathcal{Q}|}. \tag{18}$$

Note that if the train has no empty seat, the gate will reject the clique and this rejection will not be counted. However, if there are enough seats, the group will be partitioned in a dichotomy fashion until all the members have a seat, counting of course each division as a failure. As example, when a clique of four passengers arrives and there is no available empty group of four seats in the train, the algorithm will divide the clique into two sub-cliques of size two (one failure). If the new sub-cliques cannot be processed, the algorithm further divides them into two single passengers (another two failures). As a result, for the initial clique of size four, the algorithm failed three times before allocating seat for all its members.

5.2. Comparison Algorithms

5.2.1. Greedy Method

In this section, a real-time greedy heuristic algorithm is proposed to simulate the situations where seat information are not provided on tickets. This simulates a common situation in a large part of the UK's railway system (Nguyen et al., 2017), which we use as a baseline to assess the performance of seat allocation algorithms. Since this algorithm does not have a seat allocation mechanism, it resembles the situations where passengers try to find an empty seat by themselves (a greedy behavior).

In the greedy algorithm, the first step is to find a carriage that seems suitable for the passenger or the clique. The first involved parameter in this decision making process is the number of passengers inside the carriage and passengers usually try to look inside the carriages through the windows to find relatively less crowded carriages. This happens usually at the first station in which passengers arrive when the train is already there and doors are open. However, there are usually several middle stations in which passengers arrive before the train arrival. Therefore, passengers anticipate the location of carriages doors and naturally build a queue at each anticipated door location. Since usually passengers prefer walking shorter distances, the queues closer to the entrance gate have more population. However, the difference between these queue sizes should not be too high. In the proposed greedy algorithm, the first door queue can be larger $\omega \geq 1$ times more than the last door one. The queue size from the first door decreases uniformly door by door to the last one. If $\omega = 1$ then the size of all queues behind each door is almost equal. Passengers are assigned to door queues in the way that the queue size relationships could be maintained.

In this study, the greedy algorithm cannot use the information about empty and occupied seats in carriages because passengers do not have such information. As a result, there is a possibility that there are not enough empty/available seats according to the queue size at each carriage, so that some passengers need to search for another carriage after the train arrival. Therefore, after the train arrival, when the chosen carriage by passengers does not have enough empty seats, the algorithm chooses another carriage that has empty seats according to empty and occupied seat information. The reason that the algorithm uses this information for the alternative carriage is that we assume that after the train arrival, passengers can detect empty seats through windows.

Another issue that happens in this situation is increasing clique failure, because passengers do not have information about available group of seats in each carriage. Therefore, it is possible that passengers in cliques cannot find a proper group of seats and they need to sit separately.

Inside the carriage, passengers try to find seats based on their preferences. Most single passengers prefer to find the closest empty window seat that its next aisle seat is empty and they choose aisle seat when there is no other choice. However, there are passengers that prefer to find the closest seat without considering any further empty window seats and also if there is another passenger beside them. Passengers in cliques usually choose the first group of proper seats like Figure 2 but if they cannot find such a proper group of seats then the clique is divided based on the available empty seats.

5.2.2. Random Method (RND-I)

In the random method, the algorithm chooses a random seat number among all carriages for each passenger. This method does not support clique of passengers, so almost all members of cliques will fail to seat together. The algorithm has access to the empty and occupied seat information.

5.2.3. Random Method with Clique Support (RND-II)

This method resembles the random method in its carriage selection process; however, it also uses a sweeper mechanism similar to RSA to support cliques. RND-II chooses a carriage randomly among all carriage that have enough empty seats. Then, it uses a sweeper method similar to the RSA method except that the starting position of the sweeper and its direction are chosen randomly. The algorithm sweeps along a carriage row by row and allocates a seat or a group of seats as soon as it finds one. For cliques the criteria described in Figure 2 is used to find a proper seat-group while sweeping. If a row contains more than one empty seats or seat-groups then the algorithm chooses one at random.

5.3. Parameter Settings

For the simulation, the duration of each timestep is set to one second (Nguyen et al., 2017). For producing different benchmark scenarios, a journey is created for a passenger by randomly selecting its arrival station, destination station, and the arrival time. The arrival station (i) is chosen when a processing unit (passenger or clique) arrives during the station's open time. The destination station (j) is also chosen uniformly from the set $\{j \in \mathcal{S} : i < j \leq |S|\}$. The choice of a uniform is arbitrary; however, these distributions can be changed to any desired types in the simulation.

The arrival time of passengers is generated by a Poisson distribution. We assume that the train dwell time for the first station is 900 timesteps and passengers arrive during this period. For the middle stations, open time is 900 timesteps and arrival time of passengers are generated during the open time, so after that no passenger arrives at the station. For clique size, we assume that 60% of passengers travel alone, 25% of them are in cliques with a size of two, 10% form a clique of three passengers and 5% travel in a clique size of four (Nguyen et al., 2017). As described before, since having cliques with more than four members in the investigated situation in this paper is rare (Nguyen et al., 2017), we assume that the maximum size of cliques is four. The agility of passengers is a number that affects time parameters in the simulation. In this paper, passengers' agility is generated by a normal distribution: $\mathcal{N}(1, 0.15)$. For cliques, the agility is generated for all members but the value will be set to the lowest among them. For luggage, we assume that around 25% of passengers carry different types of big luggage and 25% carry hand luggage that needs to be stowed and cannot be kept by passengers during a journey (Nguyen et al., 2017). Note that considering hand luggage and agility is for making the simulation more realistic.

Table 2 shows the values of time parameters (Nguyen et al., 2017). It is worth mentioning that these values are estimated time duration (in seconds) for an average passenger in terms of its agility parameter a. For calculating these values for ith passenger, his/her agility (a_i) parameter is multiplied by the factors shown in Table 2 to obtain a custom value for each passenger according to his/her agility. Note that after multiplying a to time parameters, the obtained values are rounded to the nearest integer number to make them compatible with the granularity of timesteps.

Experiments are done on six different scenarios described in Table 3. For each scenario, the simulation is run 50 times for each algorithm with different random seed numbers for all random number generators.

Table 2: time parameters: Average time duration for passengers with average agility for performing listed actions.

Time Parameter	Values (Second)
Walking from row to row	1.5
Stowing big luggage	10
Collecting big luggage	15
Stowing hand luggage	5
Collecting hand luggage	5
Move between aisle and window seats	2
Sit to aisle seat	2
Sit to window seat with interference	10
Stand up from window seat with interference	12
Stand up from aisle seat	3
Passing entrance gate	3
Passing a carriage	21

Table 3: the parameter settings of six scenarios.

Scenario #	1	2	3	4	5	6
No. of Carriages	3	6	3	6	3	6
No. of Stations	7	7	10	10	15	15

All simulations in this paper are performed on a PC with the following specifications: CPU is the Intel (R) core (TM) i7-4770K 3.50 GHz, 16GB of RAM and the OS is Windows 10 64bit. In each table, the best results based on Wilcoxon signed-rank test with Holm-Bonferroni p-value correction at 5% significance level are shown in bold.

For RSA, the value of β is 0.5, and α is set to 5 according to the sensitivity analysis provided in Table 4. The results show that RSA performs best when the value of α is set to 5 and 10. According to the multi-comparison statistical test, the performance of RSA is not significantly different in these two cases. However, decreasing the value of α to lower values such as 0 decreases the efficiency of the algorithm because the distribution of passengers during alighting becomes chaotic, which leads to increased alighting time and consequently ABAT. On the other hand, increasing α results in not having uniform distribution of passengers in boarding, which results in increased boarding time. For the greedy heuristic algorithm, ω=2, and 20% of passengers choose the closest empty seat and 80% of them prefer to pursue other preferences such as finding the first empty window seat.

Table 4: Average ABAT (and its standard deviation inside parenthesis) obtained by RSA with different values of α in Eq. (15).

Scenario #	ABAT				
	0	5	10	20	36
1	172.48(11.44)	**166.27(10.66)**	**165.06(11.04)**	171.93(11.20)	174.19(10.95)
2	182.12(6.25)	**171.34(6.47)**	**172.16(5.71)**	178.81(6.67)	186.64(23.93)
3	168.20(7.44)	**157.96(7.78)**	**156.81(7.73)**	164.14(6.71)	167.85(7.26)
4	185.93(6.08)	**169.45(5.79)**	**171.42(6.27)**	176.73(6.18)	190.36(7.09)
5	150.95(4.39)	**140.58(4.21)**	**141.94(5.02)**	147.14(5.73)	152.22(6.11)
6	163.17(5.10)	**147.09(4.46)**	**149.45(4.19)**	156.02(3.92)	163.41(4.71)

6. Experimental Result

In this section, the experimental results of four seat allocation algorithms, i.e., RSA, Greedy, Random without clique support (RND-I), and random with clique support (RND-II) are presented. Table 5 shows the mean and standard deviation of ABAT over 50 independent runs. The values represent the number of timesteps used by each algorithm for the seat allocation process. The average number of transported passengers ($\overline{m}$) for each scenario is also shown in Table 5. In different scenarios, passenger loads are from medium to high depending on the station number. In this paper, we do not consider low passenger loads because it defies the very purpose a real-time seat allocation system which is to manage high loads.

According to the reported results in Table 5, RSA obtains the best results by obtaining the lowest ABAT values among the algorithms. RSA decreases the alighting time by controlling the passenger loads with common destinations and distributing them uniformly across carriages. It also reduces the boarding time by equalizing carriage loads through systematic distribution of the arrived passengers at each station across all carriages. RSA also attempts to maintain a uniform load on big luggage areas across all carriages to prevent congestion of passengers with big luggage in certain carriages which causes a slow passenger flow. This also minimizes the likelihood of encountering a full luggage area for passengers, which eliminates the undesired movements between carriages. Additionally, RSA assigns passengers with longer journeys to the end or middle of each carriage, and those with shorter journeys to the seats closer to the doors. This minimizes ABAT because most alighting and boardings are done from the seats closer to the carriage doors.

The obtained ABAT values by the greedy algorithm, which simulates the cases where passengers find seats by themselves according to their preferences, are around 20% longer in comparison with RSA. In this situation, passengers risk their safety by rushing and jostling in order to find a free seat. Additionally, passengers natural behavior does not result in uniform load on carriages, which increases congestions at doors. This not only increases ABAT but also affects passenger satisfaction (Nguyen et al., 2017). As described in Section 5.2.1, the greedy algorithm does not access to information about free and occupied seats before train arrival. Therefore, it is likely for passenger to board a carriage with insufficient empty

Table 5: Average ABAT (and its standard deviation inside parenthesis) obtained by algorithms based on the number of timesteps in the six scenarios alongside the average of total passenger numbers ($\overline{m}$) for each scenario.

Scenario #	ABAT				$\overline{m}$
	RSA	Greedy	Random	Rnd-Sw	
1	**166.27(10.66)**	194.21(12.04)	197.76(13.85)	199.79(15.61)	378.14(20.53)
2	**171.34(6.47)**	209.30(11.18)	204.55(10.48)	205.46(11.75)	738.36(23.93)
3	**157.96(7.78)**	184.41(9.14)	188.64(7.73)	191.02(9.70)	498.36(17.14)
4	**169.45(5.79)**	209.82(9.18)	203.57(9.19)	204.42(10.22)	1018.92(26.69)
5	**140.58(4.21)**	166.16(7.88)	171.98(6.45)	168.67(8.54)	613.84(20.55)
6	**147.09(4.46)**	188.40(5.84)	180.42(6.28)	183.39(7.49)	1224.47(30.01)

seats. This creates passenger movement between carriages which is time consuming and results in increased ABAT. Another issue with the greedy algorithm is its ignorance of passenger destination, which may result in some carriages having more passengers to alight. Since this algorithm resembles passengers' greedy behavior, other undesired situations which increase ABAT are: higher load on carriages closer to the entrance gates, overloading of big luggage areas, and poor distribution of passengers within a carriage.

Table 5 clearly shows that RSA outperforms both RND-I and RND-II. This shows that systematic selection of carriages, maintaining uniform load on big luggage areas, and considering passengers' final destinations in the seat allocation process can substantially decrease ABAT. Compared to the greedy method, it is interesting to note that random methods consistently perform better on scenarios with more carriages (scenarios 2, 4, and 6). This is because unlike random methods, the greedy method does not have access to empty seat information of the carriages, which increases the likelihood that a passenger encounters a full carriage. Therefore, for longer trains (more carriages) it takes a passenger longer to move along the platform searching for a carriage with empty seats. With respect to the two random methods (RND-I,-II), the results are almost identical, with RND-I having a marginal advantage over RND-II. The clique processing feature of RND-II is an aisle blocking action, which lengthens the seat allocation process.

Table 6 compares the obtained ABAT by the standard RSA and RSA without clique supporting ($\widehat{\text{RSA}}$) on the six scenarios. The results clearly demonstrate that $\widehat{\text{RSA}}$ significantly outperforms RSA in terms of ABAT. The reason is that $\widehat{\text{RSA}}$ does not support cliques and it only focuses on the boarding/alighting time minimization by uniformly distributing boarding and alighting passengers among carriages. On the other hand, standard RSA is a multi-objective method that considers the minimizing clique failure (Eq. (4)) alongside with the boarding/alighting time minimization (Eq. (1)) where there is a conflict between these two objectives. The processing units of $\widehat{\text{RSA}}$ are single passengers, therefore, it distributes passengers in cliques among different carriages to have an almost perfect uniform distribution of boarding and alighting passengers. In fact, after passing several stations when the map of empty seats becomes cluttered, $\widehat{\text{RSA}}$ can easily find single empty seats in the most proper carriage. Moreover, the sweeping procedure can find empty

Table 6: Average ABAT (and its standard deviation inside parenthesis) obtained by Standard RSA and a RSA without clique supporting ($\widehat{\text{RSA}}$) based on the number of timesteps in the six scenarios.

Scenario #	ABAT	
	RSA	$\widehat{\text{RSA}}$
1	166.27(10.66)	145.50(6.72)
2	171.34(6.47)	154.83(7.51)
3	157.96(7.78)	127.79(7.06)
4	169.45(5.79)	137.25(7.25)
5	140.58(4.21)	116.15(6.00)
6	147.09(4.46)	105.51(5.28)

single seats closer to the desirable points (start points of the sweeper). Therefore, passengers are distributed based on their journey length inside carriages. On the other hand, standard RSA supports cliques which leads to a deteriorated ABAT. In fact, supporting cliques for increasing quality of service for passengers does not allow RSA to always assign cliques to the best carriage in order to maintain uniform distribution of boarding and alighting passengers. The reason is that there is a possibility that the procedure of $\pi(u)$ cannot find a proper group of $|u|$ seats in the carriage. The same situation can happen for the sweeper where it cannot always assign the most suitable seats to passengers based on the length of their journey because it needs to support the cliques.

Table 7 contains the clique failure information for the RSA, RND-II, and greedy algorithms. RND-I is excluded from the table due to its lack of clique support. The results clearly show the superiority of RSA with respect to minimize CFR. Unlike RND-II which processes cliques at the expense of increased ABAT, RSA minimizes CFR without having a detrimental effect on ABAT. This can be attributed to RSA's sweeper policy which takes passengers' destination information into account in the seat allocation process. RND-II however, uses a random sweeper policy which assigns seats to passengers randomly irrespective of their final destinations. This clutters up the empty seats in a carriage which increases the likelihood of clique failure. Finally, the greedy algorithm exhibits the worst performance due to its lack of access to empty seat information of carriages. This increases clutter in the distribution of free seats, which in turn increases the likelihood of clique failure due to lack of sufficient consecutive empty seats.

Table 8 shows the average and maximum CPU time for allocating a seat by RSA for each passenger in all scenarios. The reported CPU times in Table 8 show that RSA can be used as a real-time algorithm. RSA is a heuristic algorithm with time complexity of $\mathcal{O}(n)$ for allocating a seat to a passenger in a train with n seats.

Table 7: Average clique failure number (f) and clique failure ratio (CFR) of algorithms (standard deviations inside parenthesis) in the six scenarios.

Scenario #	Clique failure number			CFR		
	RSA	Greedy	Rnd-Sw	RSA	Greedy	Rnd-Sw
1	**0.20(0.72)**	9.80(3.68)	3.34(3.31)	**0.003(0.01)**	0.20(0.06)	0.06(0.06)
2	**0.08(0.34)**	20.62(5.62)	6.22(3.21)	**7.68e-4(0.003)**	0.22(0.06)	0.06(0.03)
3	**5.76(2.89)**	23.48(4.71)	17.50(5.35)	**0.09(0.05)**	0.39(0.08)	0.27(0.07)
4	**13.96(5.14)**	51.38(8.12)	35.12(6.54)	**0.10(0.04)**	0.40(0.06)	0.25(0.04)
5	**18.68(5.12)**	38.54(7.19)	33.60(6.72)	**0.24(0.06)**	0.51(0.08)	0.39(0.06)
6	**39.10(6.72)**	76.42(10.81)	63.66(9.20)	**0.25(0.04)**	0.50(0.06)	0.38(0.05)

Table 8: Mean, standard deviation (in parenthesis), and maximum CPU time to allocate a seat for each passenger by RSA.

Scenario #	1	2	3	4	5	6
Mean (std.)	7.92e-4(2.79e-4)	9.09e-4(3.41e-4)	8.14e-4(3.26e-4)	7.86e-4(3.32e-4)	5.99e-4(3.30e-4)	5.74e-4(3.36e-4)
Max	0.003	0.004	0.003	0.005	0.003	0.021

7. Conclusion

In this paper, we studied the effect of organizing passenger flows on minimizing the boarding/alighting time by means of a real-time seat allocation (RSA) algorithm aiming at passengers with flexible tickets. The objective of the algorithm is to find seats for the passengers with open tickets to minimize the load discrepancy among carriages. The system is designed to operate on self check-in kiosks and entrance gates where passengers scan their tickets and provide some basic information to receive their seat number, the carriage where it resides, and the closest door from which it can be accessed. This not only minimizes the boarding/alighting time and the risk of delays in dwell time, but also minimizes passenger interaction, reduces the risk of injuries, and increases customer satisfaction. RSA uses a heuristic that takes passenger destination, existence of big luggage items, and group travelers (cliques) into account. This incentivizes the passengers to use the system in order to increase the likelihood of finding an empty seat and luggage area, as well as proper seat-groups for group travelers.

To test the performance of RSA and simulate the problem, a simulation software was design based on cellular automata and queues to model passenger movement. The simulator has a parametric design which allows it to model various environmental aspects such as passenger agility, big/hand luggage, passenger arrival distribution, and passenger loads at each stations. In addition to measuring the boarding/alighting time, the simulator also incorporates a performance indicator to measure clique failure ratio which shows the degree of maintaining clique integrity. The experimental results based on several scenarios with different train sizes, passenger loads, and journey lengths show that RSA significantly improves both the boarding/alighting time and clique failure ratio as compared to cases where passengers find a seat using a greedy approach or at random.

Although this research showed the advantage of unifying passenger loads across carriages by means of systematic distribution of passengers with flexible tickets, several real-world situations remain for future investigations. These include: 1. investigating the effect of standing passengers and the possibility of minimizing the standing time, 2. investigating the effect of noise caused by undesired passenger behavior such as using doors other than the assigned one and the effect of bidirectional passenger flows between carriages, 3. incorporating more passenger preferences such as the seat type (window, aisle, and table seats), availability of electricity and USB outlets, and closeness to luggage area, 4. simulating disabled passengers and their effect on the boarding/alighting time, and 5. finding better mechanisms for clique processing and sub-division.

Acknowledgements

This work was supported by a Dean's Scholarship by the Faculty of Engineering and Technology, LJMU, a Newton Institutional Links grant no. 172734213, funded by the UK BEIS and delivered by the British Council, a NRCP grant no. NRCP1617-6-125 delivered by the Royal Academy of Engineering, and a DFT Transport Technology Reasearch Innovation Grant (T-TRIG 2016).

References

Aashtiani, H. Z. and Iravani, H. Application of dwell time functions in transit assignment model. In *Transportation Research Record*, pages 88–92, 2002.

Ahn, S., Kim, J., Bekti, A., Cheng, L.-C., Clark, E., Robertson, M., and Salita, R. Real-time information system for spreading rail passengers across train carriages: Agent-based simulation study. In *Australasian Transport Research Forum (ATRF), 38th, 2016, Melbourne, Victoria, Australia*, 2016.

Allen, A. O. *Probability, Statistics, and Queueing Theory*. Elsevier, Academic press, 1990.

Armstrong, A. and Meissner, J. Railway revenue management: overview and models. 2010.

Chopard, B. and Droz, M. *Cellular automata*. Springer, 1998.

Coxon, S., Chandler, T., and Wilson, E. Testing the efficacy of platform and train passenger boarding, alighting and dispersal through innovative 3d agent-based modelling techniques. *Urban Rail Transit*, 1(2):87–94, 2015.

Daamen, W., Lee, Y.-c., and Wiggenraad, P. Boarding and alighting experiments: overview of setup and performance and some preliminary results. *Transportation Research Record: Journal of the Transportation Research Board*, (2042):71–81, 2008.

Douglas, N. Modelling train and passenger capacity. Technical report, DOUGLAS Economics, Transport for NSW, 2012.

DAriano, A. and Pranzo, M. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9(1):63–84, 2009.

Fernandez, R., Zegers, P., Weber, G., and Tyler, N. Influence of platform height, door width, and fare collection on bus dwell time: Laboratory evidence for santiago de chile. *Transportation Research Record: Journal of the Transportation Research Board*, 2143(1):59–66, 2010.

Harris, N. G. Train boarding and alighting rates at high passenger loads. *Journal of advanced transportation*, 40(3):249–263, 2006.

Harris, N. G. and Anderson, R. J. An international comparison of urban rail boarding and alighting rates. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 221(4):521–526, 2007.

Heinz, W. *Passenger service times on trains-theory, measurements and models.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2003.

Hetrakul, P. and Cirillo, C. A latent class choice based model system for railway optimal pricing and seat allocation. *Transportation Research Part E: Logistics and Transportation Review*, 61:68–83, 2014.

Hulse, J. Intrusion detection and prevention technologies available to rail transit. In *Proceedings of the 2013 APTA Rail Conference, Philadelphia, PA, USA*, pages 2–5, 2013.

Hunter-Zaworski, K. Analysis of passenger incident data from five rail transit systems. *Safety*, 3(3):21, 2017.

Jaehn, F. and Neumann, S. Airplane boarding. *European Journal of Operational Research*, 244(2):339–359, 2015.

Jiang, X., Chen, X., Zhang, L., and Zhang, R. Dynamic demand forecasting and ticket assignment for high-speed rail revenue management in china. *Transportation Research Record: Journal of the Transportation Research Board*, (2475):37–45, 2015.

KFH-Group. *TRB's Transit Cooperative Research Program (TCRP) Report 165: Transit Capacity and Quality of Service Manual, Third Edition*. The National Academies Press, 2013.

Kraft, W. H. *An analysis of the passenger vehicle interface of street transit systems with applications to design optimization.* PhD thesis, New Jersey Institute of Technology, Newark, USA, 1975.

Lam, W. H., Cheung, C., and Poon, Y. A study of train dwelling time at the hong kong mass transit railway system. *Journal of advanced Transportation*, 32(3):285–295, 1998.

Lan, Y., Ball, M. O., Karaesmen, I. Z., Zhang, J. X., and Liu, G. X. Analysis of seat allocation and overbooking decisions with hybrid information. *European Journal of Operational Research*, 240(2):493–504, 2015.

Li, C., Song, X., and Li, X. Seats allotment optimization for china railway operation management. In *Management and Service Science (MASS), 2010 International Conference on*, pages 1–4. IEEE, 2010.

Li, D., Daamen, W., and Goverde, R. M. P. Estimation of train dwell time at short stops based on track occupation event data: A study at a dutch railway station. *Journal of Advanced Transportation*, 50(5):877–896, 2016.

Li, D., Yin, Y., and He, H. Testing the generality of a passenger disregarded train dwell time estimation model at short stops: Both comparison and theoretical approaches. *Journal of Advanced Transportation*, 2018:16, 2018.

Lin, T.-m. and Wilson, N. H. Dwell time relationships for light rail systems. *Transportation Research Record*, (1361), 1992.

Ma, W. and Qiu, B. A robust model for airline seat allocation with multiple seat bookings. In *Management and Service Science (MASS), 2010 International Conference on*, pages 1–4. IEEE, 2010.

Milne, R. J. and Salari, M. Optimization of assigning passengers to seats on airplanes based on their carry-on luggage. *Journal of Air Transport Management*, 54:104–110, 2016.

Nguyen, T., Makki, A., Yazdani, D., Lersteau, C., Deplano, I., Ramada, J. P., Wang, R., and Ancele, Y. Intelligent real-time seat allocation for better rail services. Technical report, DfT Transport Technology Research Innovation Grant (T-Trig 2016), 2017.

Notomista, G., Selvaggio, M., Sbrizzi, F., Di Maio, G., Grazioso, S., and Botsch, M. A fast airplane boarding strategy using online seat assignment based on passenger classification. *Journal of Air Transport Management*, 53:140–149, 2016.

Nyquist, D. C. and McFadden, K. L. A study of the airline boarding problem. *Journal of Air Transport Management*, 14(4): 197–204, 2008.

Obeng, K. and Sakano, R. Airline fare and seat management strategies with demand dependency. *Journal of Air Transport Management*, 24:42–48, 2012.

Parkinson, T. and Fisher, I. *TCRP Report 13: Rail Transit Capacity.* Transportation Research Board, National Research Academy, National Academy Press, 1996.

Puong, A. Dwell time model and analysis for the mbta red line. Technical report, Massachusetts Institute of Technology Research Memo, 2000.

Qi, Z., Baoming, H., and Dewei, L. Modeling and simulation of passenger alighting and boarding movement in beijing metro stations. *Transportation Research Part C: Emerging Technologies*, 16(5):635–649, 2008.

Qiang, S.-J., Jia, B., Xie, D.-F., and Gao, Z.-Y. Reducing airplane boarding time by accounting for passengers' individual properties: A simulation based on cellular automaton. *Journal of Air Transport Management*, 40:42–47, 2014.

San, H. P. and Masirin, M. I. M. Train dwell time models for rail passenger service. In *The 3rd International Conference on Civil and Environmental Engineering for Sustainability*, volume 47, 2016.

Seriani, S. and Fernandez, R. Pedestrian traffic management of boarding and alighting in metro stations. *Transportation Research Part C: Emerging Technologies*, 53:76–92, 2015.

Soolaki, M., Mahdavi, I., Mahdavi-Amiri, N., Hassanzadeh, R., and Aghajani, A. A new linear programming approach and genetic algorithm for solving airline boarding problem. *Applied Mathematical Modelling*, 36(9):4060–4072, 2012.

Subramanian, J., Stidham Jr, S., and Lautenbacher, C. J. Airline yield management with overbooking, cancellations, and no-shows. *Transportation science*, 33(2):147–167, 1999.

Sumalee, A., Tan, Z., and Lam, W. H. Dynamic stochastic transit assignment with explicit seat allocation model. *Transportation Research Part B: Methodological*, 43(8):895–912, 2009.

Szplett, D. and Wirasinghe, S. An investigation of passenger interchange and train standing time at lrt stations:(i) alighting, boarding and platform distribution of passengers. *Journal of advanced transportation*, 18(1):1–12, 1984.

Tang, T.-Q., Wu, Y.-H., Huang, H.-J., and Caccetta, L. An aircraft boarding model accounting for passengers' individual properties. *Transportation Research Part C: Emerging Technologies*, 22:1–16, 2012.

Thoreau, R., Holloway, C., Bansal, G., Gharatya, K., Roan, T.-R., and Tyler, N. Train design features affecting boarding and alighting of passengers. *Journal of Advanced Transportation*, 50(8):2077–2088, 2016.

uic.org. synopsis 2015, 2015. https://uic.org/IMG/pdf/synopsis_2015_print_5_.pdf Accessed 20 September 2017.

uic.org. synopsis 2016, 2016. https://uic.org/IMG/pdf/synopsis_2016.pdf Accessed 20 September 2017.

Wang, X., Wang, H., and Zhang, X. Stochastic seat allocation models for passenger rail transportation under customer choice. *Transportation Research Part E: Logistics and Transportation Review*, 96:95–112, 2016.

Wirasinghe, S. and Szplett, D. An investigation of passenger interchange and train standing time at lrt stations:(ii) estimation of standing time. *Journal of advanced transportation*, 18(1):13–24, 1984.

Xie, Z., Zhu, W., and Zheng, L. A dynamic railway seat allocation problem. In *IIE Annual Conference. Proceedings*, page 3480. Institute of Industrial and Systems Engineers (IISE), 2013.

Yoon, M. G., Song, Y. S., and Lee, H. Y. An integrated seat allocation model for individual and group demand in airlines. In *Computers and Industrial Engineering (CIE), 2010 40th International Conference on*, pages 1–6. IEEE, 2010.

Yoon, M. G., Lee, H. Y., and Song, Y. S. Linear approximation approach for a stochastic seat allocation problem with cancellation & refund policy in airlines. *Journal of Air Transport Management*, 23:41–46, 2012.

Zeineddine, H. A dynamically optimized aircraft boarding strategy. *Journal of Air Transport Management*, 58:144–151, 2017.