INSTITUT FÜR INFORMATIK
Algorithmische Bioinformatik

Universitätsstr. 1 D–40225 Düsseldorf

Evaluating Different Methods for Aircraft Boarding

Hao Tian

Bachelorarbeit

Beginn der Arbeit:	04. November 2019
Abgabe der Arbeit:	04. Februar 2020
Gutachter:	Prof. Dr. Gunnar Klau
	Prof. Dr. Egon Wanke

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 04. Februar 2020

Hao Tian

Abstract

Every once in a while air travel is the chosen method to reach a destination. However, passengers often feel unpleasant as a result of being blocked in the aisle of the aircraft. Additionally, airlines have to pay fees because of long boarding time.

Algorithms and models, such as Steffen boarding method, Random boarding method, WILMA boarding method... exist to optimize the boarding time. More advanced algorithms, such as Back-to-front boarding method and Rotating-zone boarding method, also take team boarding(group behaviour) into account.

The Steffen boarding method is very effective. However, under certain circumstances, this method can be improved.

In a real-life, team boarding needs to be taken into account. Therefore, this thesis evaluates different existing boarding methods to attempt to optimize the Steffen method. Furthermore, it also put forward a new boarding method for team boarding, which bases on Steffen boarding method.

The newly created model in this thesis is a result of pinpointing and improving the bottlenecks of existing methods. The model used in this thesis is a result of combining the thoughts from various literature and authors. This thesis introduces a simulator using Java and queuing theory to simulate the aircraft boarding process, and finally obtained the output(total boarding time) by simulating different boarding methods to evaluate the results.

Contents

1 Introduction

Since the 21st century, researchers have continuously proposed boarding methods and created simulators to reduce boarding time. Aircraft boarding process refers to the process from the beginning of the first passenger entering the cabin to the end of all passengers have been entered their seats. The total time of the boarding process called boarding time. For passenger's boarding behaviour, it divides into individual boarding behaviour and team boarding behaviour. The definition of the theoretical boarding method refers to considering only individual boarding behaviour, which means it can assign any passenger to any seat according to the algorithm. However, considering the actual situation, a feasible boarding method must consider team boarding behaviour, which means that passengers who belong to the same boarding crew need to be allocated in the same row in the cabin as much as possible. Tang et al., 2019 contains a summary of most of the boarding strategies to date. Some of them are theoretical algorithms such as Steffen, 2008 boarding method. Some of them also consider team behaviour and the feasibility of modern technology such as Notomista et al., 2016.

However, the number of airports is too large. Novel technologies, such as machine learning are not necessarily feasible. Therefore, this thesis will build an essential boarding simulator, and try to evaluate and improve the theoretical boarding method, then consider team behaviour to create a feasible team boarding method by improving algorithm only.

It is crucial to evaluate boarding methods based on different factors(Delcea et al., 2018b), such as long boarding time, which is also the factor that the thesis focuses on. The existing theoretical algorithms to reduce boarding time are based on reducing passengers' congestion time in the aisle to arrange the boarding route accordingly. For example, the Steffen boarding method(Steffen, 2008) considers the time delay caused by passengers loading luggage, which leads to subsequent passengers' congestion. Steffen optimizes the boarding route with idealized theoretical algorithms, which is defined as "set buffer" in this thesis. That means if one passenger is carrying luggage, leave some aisle space for the next passenger so that two passengers can enter their seats almost at the same time. It can also be summarized as rational usage of aisle space (Steffen and Hotchkiss, 2012).

Therefore, as an algorithm for reducing the boarding time, it is indispensable to adjust the boarding order according to various conditions, such as Bazargan, 2007. Of course, one has to keep in mind that the theoretical setup is not entirely in line with the actual situation. One can improve boarding methods by looking for more situational factors. The Steffen boarding method is only based on individual boarding behaviour, but a realistic boarding situation, such as team boarding behaviour, is not applicable in the Steffen boarding method.

For example, Back-to-front and Rotating-zone boarding methods(Delcea et al., 2018a) consider team boarding behaviour by setting different areas in the cabin. However, these two methods also have their disadvantages because of setting unreasonable buffer area. The question is "How do you design a reasonable buffer?".

Thereby, this thesis will create a new theoretical boarding method according to the reasonable buffer setting by using the greedy algorithm, which is based on the thesis(Milne and Kelly, 2014) that has been improved Steffen boarding method. After this, came up with the question about considering team boarding behaviour. "If combining the team boarding behaviour and Steffen's idea, will it be better than other boarding methods which also consider team boarding behaviour?". That is, if there is a situation in which is the team boarding behaviour, the buffer area is expanded according to the number of boarding crew's members and the number of boarding crew's luggage. If it is an individual boarding behaviour, the idea of the Steffen boarding method is retained.

2 Simulator Introduction

Several theses had created some simulators, such as Schultz et al., 2008 and Mas et al., 2013. Also, Jafer and Mi, 2017 is a very detailed introduction to the design of the simulator.

However, due to different research directions, the performance and standards of each simulator are different. Therefore, a simulator has to been created by using java to meet the objectives of the thesis. The advantages of the simulator are the universality of the java language and detachability of the simulator. Which means the simulator can be divided into multiple functions, such as creating the required standard data, calculating different boarding orders[2.3] according to different boarding algorithms, and simulating the boarding processes according to different data and boarding orders. Thereby, the advantages are very convenient for the subsequent development of the simulator.

This chapter will introduce the models in the simulator and simulate the boarding processes of three basic boarding methods: WILMA, Random and Steffen.

At the same time, the benefits of Steffen's idea are evaluated by collecting data through the results of the simulated boarding processes, and it is also proved that "set buffer" is essential to reduce boarding time under the model.

2.1 Aircraft Model

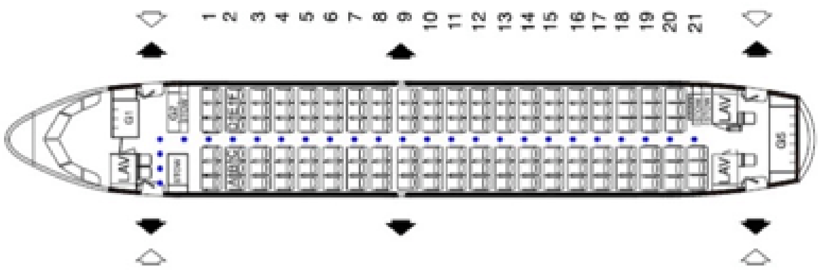

Figure 1: A319 Aircraft Seat Map
[1](The subsequent use of this figure is derived from Skytrax(Skytrax, n.d.) the URL for the figure is from (FINNAIR, n.d.))
[2](This figure is for research reports only and not for commercial. Fair Use Copyright: Law, n.d.)

Throughout the whole thesis, use A319 model aircraft to run the simulations (Figure 1). For ease of simplicity, omitted first class. The blue dots in the figure represent the aisles' positions. Only consider 20 rows (120 seats) for the seats. The aircraft has only one entrance on the front of the cabin.

2.2 Passenger Model

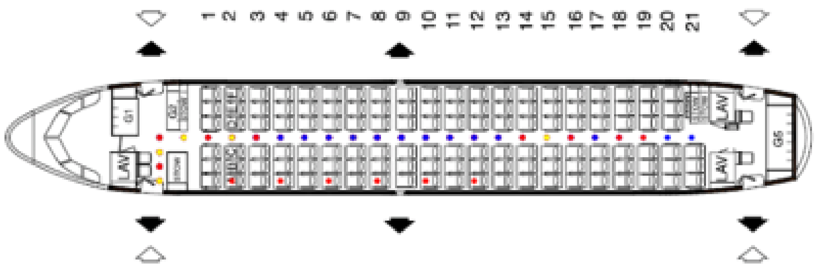

Figure 2: Passengers are boarding

For the passenger model, each passenger can carry one or none luggage on board, and each passenger has to choose whether to board as part of a team (for the evaluation of the follow-up team boarding methods). Only consider full aircraft (with 120 passengers). Figure 2 shows the process of passengers boarding. Every red dot represents a passenger, and if they carry luggage, add a yellow dot behind the passenger. Each passenger must wait until the aisle position in front of him is available. Each red dot and a yellow dot will occupy an aisle position and disappear together after the passenger has boarded. This model does not consider the time required to pass another passenger, who is the aisle seat. For related important parameters, please refer to Table 1.

2.3 Parameters Table

Name	Type	From	Value
xPosition	double	Aircraft seat Model	
yPosition	double	Aircraft seat Model	
seatNumber	int	Aircraft seat Model	0-119
waitArea	String	Aircraft seat Model	waitArea
aislePosition	String	Aircraft seat Model	Aisle1-23
havePerson	boolean	Aircraft seat Model	true/false
passengerName	String	Passenger Model	Passenger1-Passenger120
luggage	int	Passenger Model	0,1
groupID	int	Passenger Model	0-n
boardingOrder	int[]	Boarding Methods	Randomly new int[120]
startTime	long	Simulation process	System start time
endTime	long	Simulation process	System end time
waitTime	long	Simulation process	Depends on congestion
walkTime	int	Simulation process	Each step 200ms
loadingTime	int	Simulation process	200ms/700ms

Table 1: Parameters from Program

The above variables are essential parameters for building the simulator. First, we need to draw the aircraft seat map to a UI window by using java. Then according to the "xPosition" and "yPosition" in UI window, we can determine the "seatNumber" which are used to assign to passengers, the "aislePostion" which passengers move on it, and the "waitArea" which passengers wait on it to enter the cabin.

For passengers' information, the program will read a txt file as input, which include 120 passengers' information. Furthermore, each row in the file includes a "passengerName", a "luggage" and a "groupID(boarding crew number)".

After that, the program assigns each passenger a "seatNumber" based on different boarding methods. Then group all "seatNumber" into an array to obtain the "boardingOrder".

Finally, input the passenger information one by one, and each passenger corresponds to a "seatNumber" in "boardingOrder". Then the simulator begins to simulate the boarding process. For the boarding process, the "startTime" is the System-Time when the first passenger enters the cabin, and the "endTime" is the System-Time when the last passenger enters his seats. The time that each passenger moves on "aislePosition" is called "walktime". The time that each passenger to load his luggage and enter his seat is called "loadingTime". If there is a passenger on an "aislePosition", "havePerson" of the "aislePosition" is true. If subsequent passengers are congested due to the current "havePerson" being true, the time counted in "waitTime" of the passenger who causes the congestion.

2.4 Define the Problem of Evaluation

With the description in 2.3, we can define the problem as shown below.

Input: A Passengers Set P = $\{p_i\}$ i $\in$ {1,2,3,...,120} , **A Boarding Order Set Q** = $\{q_i\}$ i $\in$ {1,2,3,...,120}

Define the luggage Set **L**, the total walking time **Twalking**, the total waiting time **Twaiting**, the total loading time **Tloading** , and the boarding crew number Set **C**.

Definition 2.1. The sequence of collecting each passenger's name in the order of the data is called "Passenger Set".

Definition 2.2. The sequence of collecting each passenger's seat number in the order of "Passenger Set" is called "Boarding Order Set."

Definition 2.3. The sequence of collecting each passenger's luggage in the order of "Passenger Set" is called "Luggage Set.

Definition 2.4. The sequence of collecting each passenger's boarding crew number in the order of "Passenger Set" is called "Boarding crew number Set."

$$p_i = \{q_i, l_i, twalking_i, twaiting_i, tloading_i, c_i\}, i \in \{1, 2, ..., 120\} \tag{1}$$

$$L = \{l_1, l_2, \dots, l_{120}\}, l_i \in \{0, 1\}, i \in \{1, 2, ..., 120\} \tag{2}$$

$$Twalking = \sum_{i=1}^{120} twalking_i \tag{3}$$

$$Tloading = \sum_{i=1}^{120} tloading_i \tag{4}$$

$$Twaiting = \sum_{i=1}^{120} twaiting_i \tag{5}$$

Output: :

$$Tsum = Twalking + Twaiting + Tloading \tag{6}$$

Problem: As one uses the same model, simulating different boarding methods under different data, and obtaining the minimised Tsum, which is the total boarding time for the whole boarding process. The simulator calculates T_sum = T_end - T_start.

2.5 Simulation Steps

The simulator takes into account of the parameters mentioned above and is created based on queuing theory(Kendall, 1951) and multithreading of Java. According to the definition of the above problem, it needs to input different data as program arguments to run the program for different situations, so through the following simulation steps, it can completely simulate the boarding process in different situations and collect the boarding time to evaluate the boarding methods.

2.5.1 Create suitable Data

Each data file(txt) has three columns for the passengers' name (or number), the number of luggage that each passenger carries, and the boarding crew number for each passenger.

A data set basically has 29 data for each evaluation. Each data has 120 passengers, and the passengers randomly carry luggage according to a different percentage of luggage conditions(0%-100%). For example, the percentage is 0%, that means none passengers carry luggage. Three different data for each percentage case and each data has luggage set in a different sequence. Nevertheless, the 0% data and 100% data will not change the luggage set in a different sequence, so there is only one data for each of these two cases.

If there are other situations for creating data, it will put forward in the content of the subsequent thesis.

Basic Data Form:

A Passengers Set P$^{2.1}$, **A Luggage Set L**$^{2.3}$, **A Boarding Crew Number Set C**$^{2.4}$, = {c_i},i ∈ {1,2,3,. . . ,120} ,

$$c_i = \begin{cases} 0 & \text{individual boarding behaviour} \\ n & \text{n>0,with boarding crew n} \end{cases}$$

[3](For all theoretical boarding methods, the c_i will always be 0. Moreover, for team behaviour boarding methods, the rules of setting C will show in 4.2.1.)

A Data Condition D$^{2.5}$ = d_i, d_i ∈ {0%,10%,20%,30%,40%,50%,60%,70%,80%,90%,100%} , i ∈ {0,1,2,3,. . . ,10}.

Definition 2.5. The percentage of passengers with luggage in the total number of passengers in each data. The set collecting all percentage case is called "Data Conditions Set."

Total number of passengers with luggage :

$$\#Passengers\_with\_luggage = 120 * d_i \tag{7}$$

2.5.2 Calculate Boarding Order

After inputting the data set created in 2.5.1, the program will assign the seat number to each passenger by different boarding algorithms. Each algorithm assigns the seat number based on the order of each passenger, the luggage they carry and whether the passenger is boarding for the team, and then return an array to store the seat numbers in order of passenger set, which is the boarding order.

There are two ways to get the boarding order. For some methods with stable boarding order[1], such as Steffen boarding method and WILMA boarding method, it can directly input the boarding order according the method's rules. For other methods that have an unstable boarding order[2], the boarding order needs to be calculated through specific algorithms.

2.5.3 Collect the Results

After inputting the data into the program, the simulator will simulate the boarding process according to the specified boarding order. After each simulation, the total boarding time for the simulation will be output. For each boarding method, it will create a table to store the performance (total boarding time) of the method in each case (0% -100%).

As mentioned in 2.5.1, there are three data for each case. For each method, calculate the average of the three performances (total boarding time) in each case, then collect the average of different methods into a new table to compare with, then draw the graph of the new table, and finally conclude.

[1]No matter how the information of passengers changes, passengers get a seat number in a fixed order.

[2]Passengers get a seat number according to different algorithms conditions. Which mean if the passenger's information changes, the seat number assigned to him may also change.

2.6 Show Simulation-Process by Three Basic Methods

These three methods have been simulated under the other models refer to Steffen and Hotchkiss, 2012 and Cimler et al., 2012. However, the results are different here. Although the Steffen method still performs best, differences of performances for the other two methods.

Evaluate these three methods according to 2.5. Pick the simulations of any one of the data and use representative pictures to help quickly understand the methods. For a detailed description of the methods, refer to Steffen and Hotchkiss, 2012.

Each method performed 29 simulations, and each data has a fixed order of 120 passengers. However, according to the data condition, which is the percentage of luggage, passengers randomly carry luggage with a fixed number. In other words, each percentage case has three data, and the luggage set for the same percentage data in three different sequence. Furthermore, the boarding crew number of all passengers for all data in this chapter is 0, because these three methods are individual boarding behaviour.

2.6.1 Random Method

For a randomly arranged boarding queue[3], each passenger corresponds to a random seat, and boarding is performing according to the boarding queue and boarding order. For the boarding process, refer to Figure 3.

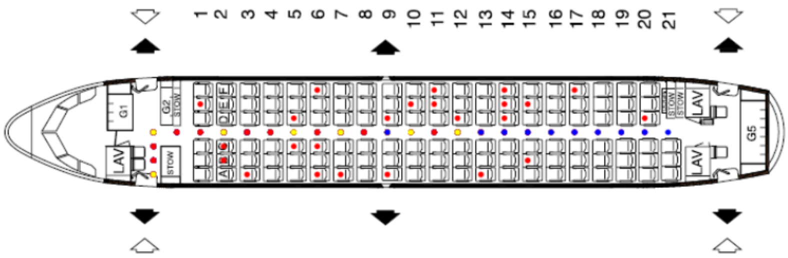

Figure 3: Random boarding Process

2.6.2 WILMA Method

For a randomly arranged boarding queue, each passenger corresponds to a fixed seat, the seat is arranged in accordance with the rules from back to front, from the window to the aisle, and boarding is performed according to the boarding queue. For the boarding process, refer to Figure 4.

[3]"Boarding queue" is a set of passengers, which can also be understood as a queue waiting for boarding before the gate. Passengers board one by one following the queue order.

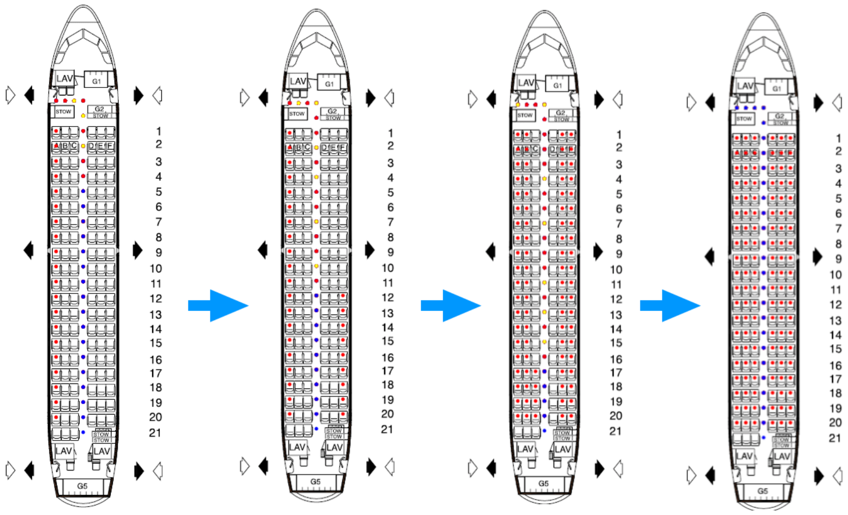

Figure 4: WILMA boarding Process

2.6.3 Steffen Method

For a randomly arranged boarding queue, each passenger corresponds to a fixed seat, and the seat is arranged according to a line interval, which means each passenger will set 1-line "buffer" for the next passenger, from back to front, from the window to the aisle, and boarding is performed according to the boarding queue and boarding order. For the boarding process, refer to Figure 5.

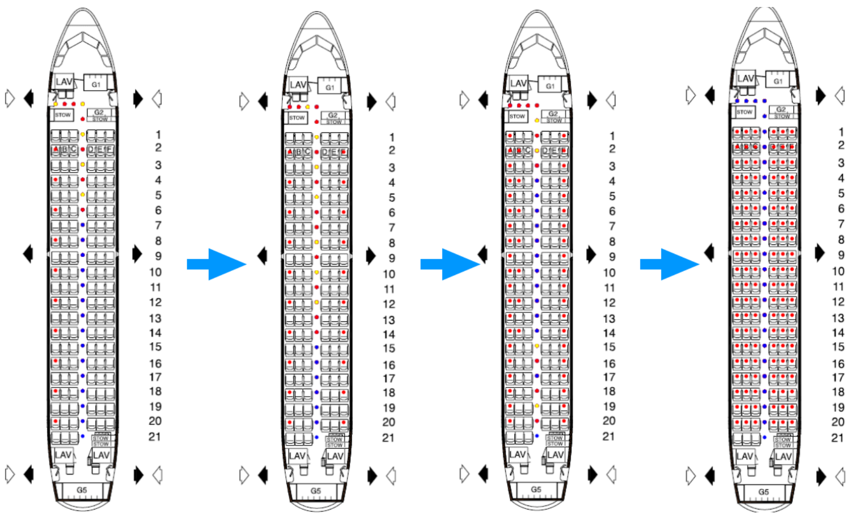

Figure 5: Steffen boarding Process

2.6.4 Evaluating the Result

Each simulation of the boarding process of each data produces a boarding time. Afterwards, calculate the averages of each percentage data group and added into the table(Table 2) [4]. In the end, create a graph(Figure 6) to compare the results.

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
WILMA	29	40	50.3	61	72	82	93.7	104	115	125	136
Random	34	42.3	47	52	57	62.3	68.7	71.3	77	84	92
Steffen	30	34	37	39.7	43.3	47.3	48.7	53.3	55.3	58.3	61

Table 2: Boarding Time(WILMA vs Random vs Steffen)
[3](The unit of all tables that are not specifically described in the following is second.)

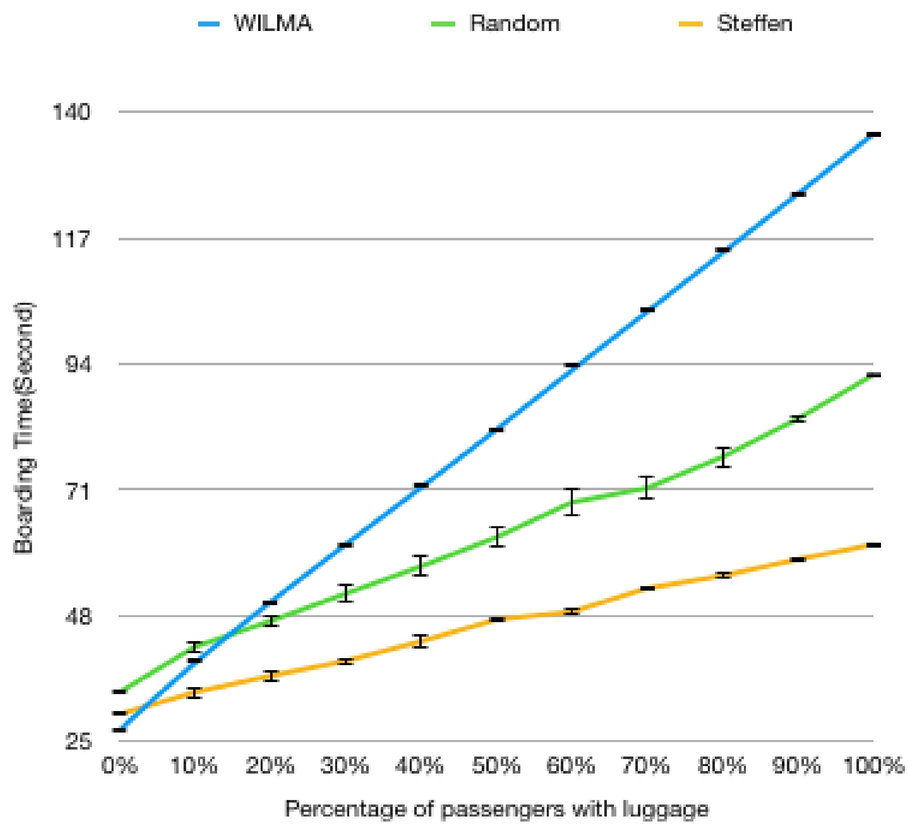

Figure 6: Result Graph

[4]The first column in the table is the methods name, and the first row is the different percentage of passengers carrying luggage. The value of each grid in the table is the average of boarding time by simulating three different data in the same percentage of passengers carrying luggage data groups.

As one can see from the results, WILMA is performing the best when passengers do not carry any luggage. With the increase of passengers carrying luggage, the Steffen method requires significantly less boarding time than the other two methods. The more passengers are carrying luggage, the more significant the impact.

Because in the model, luggage will also occupy one aisle position, as the number of passengers carrying luggage increases, when using the WILMA method, more and more passengers need to wait for previous passengers with luggage to board before they arrive at their seat lines.

The Random method solves part of this problem, which means some passengers will set the "buffer" for the next passenger, and no matter how large this buffer is, they will board at almost the same time, so there will be fewer congestions than that of WILMA method. However, it will also not perform optimally because of randomness, which means a passenger might set a buffer for the next passenger too large or too small.

As Steffen method has set up a buffer zone for each passenger, passengers can still arrive at their seat lines, whether or not the previous passenger is carrying luggage. So as the number of passengers boarding with luggage increases, the superiority of the Steffen method becomes apparent.

Therefore, setting buffers is very useful and necessary in the case of passengers carrying luggage. It also proves that, in most cases, the Steffen method is currently the most useful theoretical boarding method.

3 Improve Steffen Boarding Method

3.1 A new Method Introduction

This chapter will introduce a new individual boarding method called Steffen-Greedy boarding method(after this referred to as SG method) based on a thesis(Milne and Kelly, 2014), which optimised the Steffen method (after this referred to as Steffen-New method).

Steffen-New's method is to allocate boarding seats according to the amount of luggage carried by passengers. Under this model, as one can understand that passengers with luggage board first, and passengers without luggage board later.

However, as it has analysed in the previous chapter, the WILMA boarding method is better than the Steffen boarding method when none of the passengers carries luggage. Hence, a greedy algorithm used to optimise the Steffen-New boarding method.

Algorithm 1 Steffen-Greedy For Each Colunm

Input:
  Colunm number 1-6;
  Passenger i p_i,, Passenger i+1 $p_i + 1$;
  Luggage of Passenger i, l_i;
  ArrayList to collect free seats of this colunm, $seats$
Output:
  buffer for Passenger i+1, $q_i + 1$;
  if $l_i = 0$ **then**
    $buffer = 0$
  else
    $buffer = 1$
    $seats.add(getSeatNumberByPosition(currentSeatsPosition + 1))$
  end if

The proposed algorithm will capture whether the previous person who boarded, carrying luggage. If the previous passenger carried one, the current passenger would enter his seat in an interlaced way(Steffen method, set one line "buffer" for the next passenger). If this were not the case, the passenger would enter his seat immediately(WILMA method).

It is worth mentioning that the greedy algorithm applied by the SG method allocate seat numbers to the boarding process of each column. Which means, for each column, if the previous passenger does not carry luggage, all passengers after him who do not carry luggage will enter their seats immediately using WILMA method. After the processing of this column of passengers by the greedy algorithm, when the next batch of the column of passengers comes, there will be a passenger sequence(ArrayList in the algorithm), which equals to the number of empty seats in the current column to fill the empty seats. Specific examples will introduce in the description of the SG method.

This algorithm combines the advantages of the Steffen-New and WILMA boarding methods and optimises the theoretical boarding method under this model.

3.2 Evaluating Steps

The evaluation mainly follows the steps as described in 2.5. However, for the Steffen-New and SG methods need some adjustments.

3.2.1 Create data

Create a new data set with a varying amount (0% to 100%) of passengers carrying luggage. The data is sorted by the following condition: passengers with luggage board before passengers without luggage.

Only 10 data are needed here because in each case, the passengers' order is sorted according to the specified condition, which means that the boarding queue[3] will not change. Additionally, all passengers also have boarding crew number 0.

Data Form:

A Passengers Set $P^{2.1}$, **A Luggage Set $L^{2.3}$**, **A Boarding Crew Number Set $C^{2.4}$** = $\{c_i\}$, $c_i = 0$, $i \in \{1,2,3,\ldots,120\}$, **A Data Condition $D^{2.5}$**.

Thorem : For the Luggage Set L, each l_i follows:

$$l_i = \begin{cases} 1 & i \leq 120 * d_i \\ 0 & others \end{cases}$$

3.2.2 Simulate Steffen-New Method and Steffen-Greedy Method

At each step, input the data set(created in 3.2.1) to the simulator to execute the Steffen-New method and the SG method. The result is then collected and added to a table. The simulation steps are consistent with the steps of 2.5.2 and 2.5.3.

3.2.3 Comparing the Result with original Steffen Method

This chapter needs to evaluate whether the two new methods are more effective than the original Steffen method. Therefore, it needs to compare the boarding time of these two new methods with the boarding time collected by the original Steffen method in previous simulations(Refer to Table 2). After collecting all the boarding time of these three methods, a graph is drawn based on the boarding time of the three methods for the next step of evaluating.

3.2.4 Conclusion and analyse Results

After creating the graph, the performances of the three methods under the different percentage of passengers carrying luggage are evaluated and analysed. After that, as one can look at the amount and reason for the improvement of the best performing method.

3.3 Steffen-New Boarding Process Introduction

The only difference between the Steffen-New method and the Steffen method is that all passengers with luggage will board first.

The example with 50% of passengers carrying luggage is taken as an example to introduce the Steffen-New method. Figure 7 shows the boarding process of this method. The first 60 passengers are boarding with luggage, and the remaining 60 passengers are boarding without luggage.

In the figure, the first to third pictures show that the first 60 passengers are boarding with luggage, and they boarded in the order following the Steffen boarding method. In the third picture, the last 60 passengers without luggage begin to appear at the end of the queue. The fourth picture shows that the passengers without luggage also boarded with the rules of the Steffen boarding method.

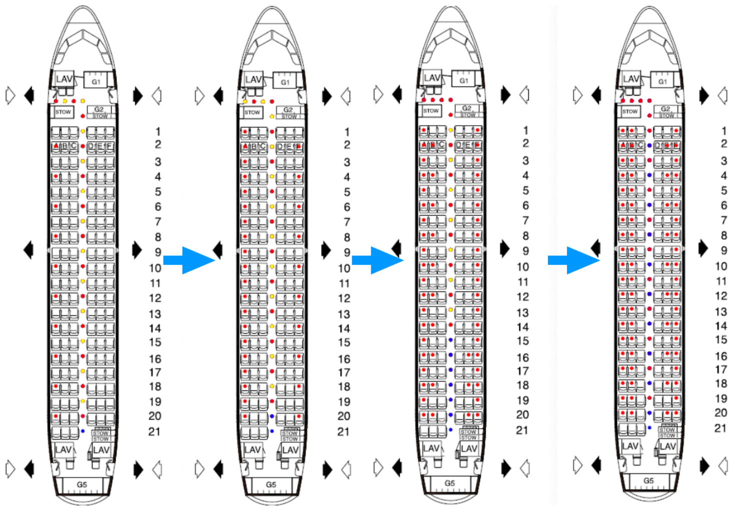

Figure 7: Steffen-New Boarding Process

3.4 Steffen-Greedy Method Boarding Process Introduction

The same example with 50% of the passengers carrying luggage used to explain SG boarding method (Figure 8). First, execute the Steffen-Greedy algorithm for the first column(the first picture to the second picture), and ten passengers enter the first column of seats according to the Steffen boarding rules. At this time, ten seats in the first column are empty, so the next ten passengers will enter these ten empty seats. Use the same algorithm for the 6th and second columns (the third picture). For the remaining 60 passengers without luggage, they will enter the 5th, 3rd, and 4th columns in order according to Wilma's method (the fourth picture). However, this is the ideal situation for the Steffen-Greedy algorithm.

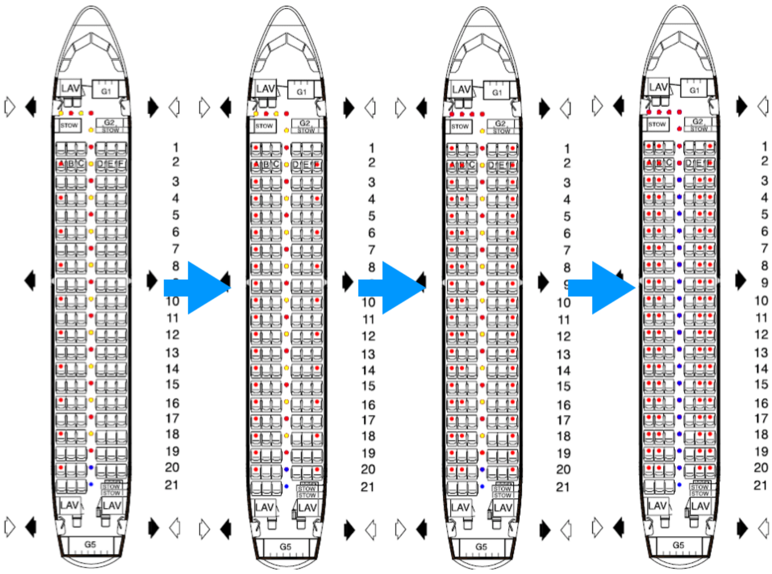

Figure 8: SG Boarding Process with 50% Passengers with Luggage

Figure 9 shows the problem mentioned above more apparent by a simple example, caused by applying the greedy algorithm for each column.

Assume that the red passenger is the last passenger to board who carry luggage, and it will set a buffer which is 1 line for the first yellow passenger. Then the rest yellow passengers will board immediately by using WILMA boarding method; however, as one can see that the first column has one empty seat(green dot). So when the next batch of the column of passengers(black dots) comes, there will be a passenger to fill the green one first. After then, all passengers without luggage will board by using WILMA boarding method.

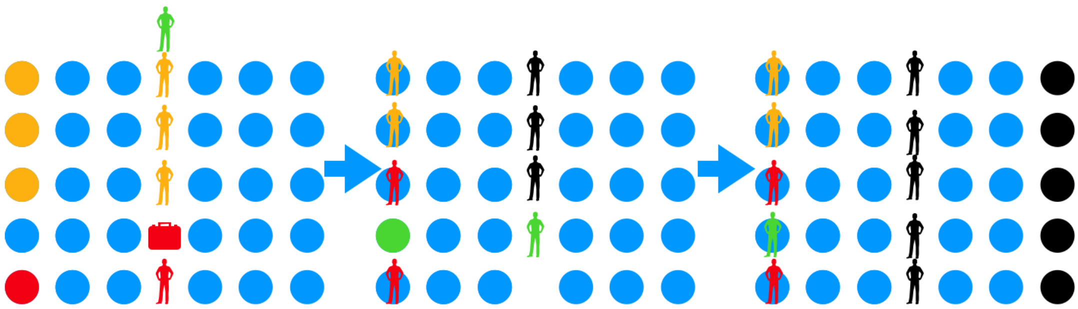

Figure 9: Fill empty Seats caused by Greedy Algorithms

3.5 Result

After completing the steps in 3.2, one can get the following Table 3[5] and Figure 10. For the boarding times of Steffen-Orginal are from Table 2.

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Steffen-Greedy	28.9	32.4	35.5	38.3	41.6	45.3	48.2	51.5	54.6	57.7	61
Steffen-Original	30	34	37	39.7	43.3	47.3	48.7	53.3	55.3	58.3	61
Steffen-New	30	33.3	36.1	39.2	42.2	46	48.6	52.8	55.2	57.7	61

Table 3: Boarding Time(Steffen-Greedy vs Steffen-Original vs Steffen-New)

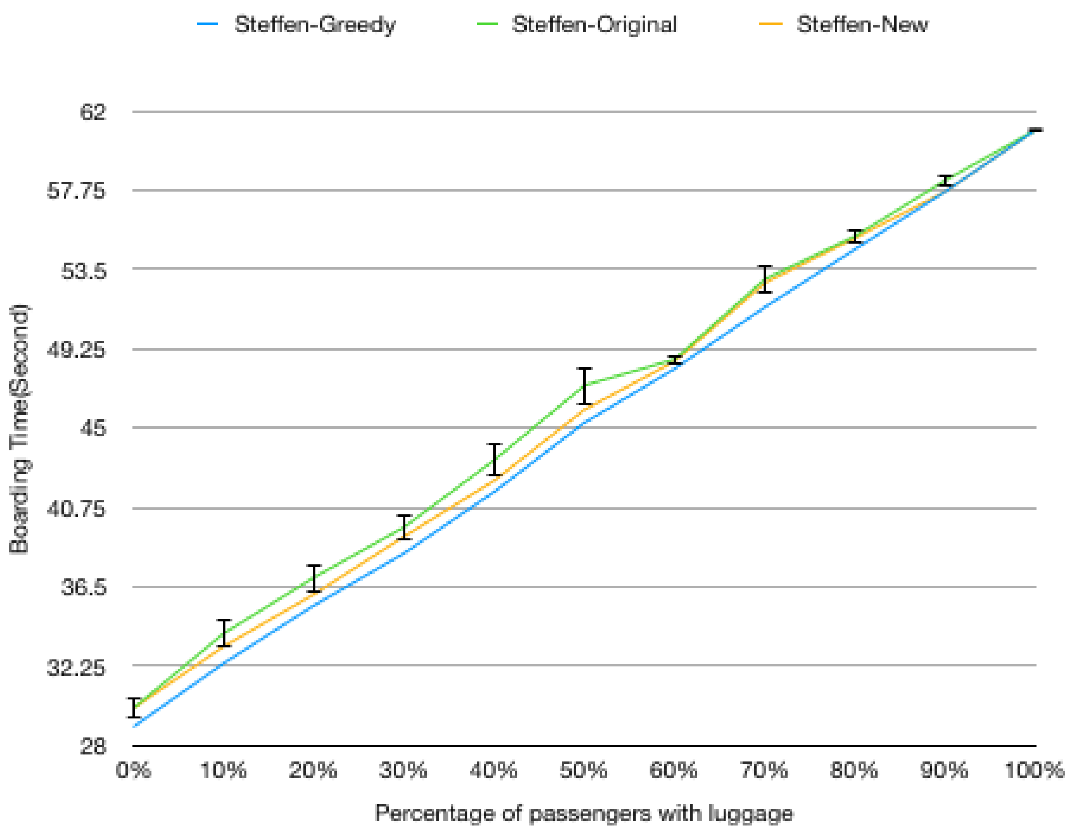

Figure 10: Result Graph

[5]The first column in the table is the methods name, and the first row is the data of the different percentage of passengers carrying luggage. The value of each grid for Steffen-Greedy and Steffen-New is the boarding time of each percentage of luggage data. Not the average! Only one data for each case, because the passenger with luggage always boards first. Therefore, only Steffen-Original has standard deviation.

As shown in the figure, compared to the other two methods, the SG method had a better performance for test cases where 90% or less of the passengers are carrying luggage. The next formula is used to calculate the percentage of optimisation:

$$\frac{TimeOfOldMethod - TimeOfNewMethod}{TimeOfOldMethod} * 100\% \tag{8}$$

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
SN vs SO	0	2.06	2.43	1.26	2.54	2.75	0.21	0.94	0.18	1.03	0
SG vs SO	3.33	4.71	4.05	3.53	3.93	4.23	1.03	3.25	1.27	1.03	0
SG vs SN	3.33	2.7	1.66	2.3	1.42	1.52	0.82	2.46	1.09	0	0

Table 4: Optimaisation percentage

Table 4[6] collects the percentage improvement of the Steffen-Greedy method compared to the other two methods, and the percentage improvement of the Steffen-New method compared to the Steffen-Orginal method.

Because both methods(SG method and Steffen-New) are optimized only for the congestion caused by a small number of boarding passengers, they perform similarly. However, due to the need to quickly measure the data, the simulator uses small values of the time variables, such as only 500ms for loading luggage. When these time variables' value expanded, the SG method will perform better.

In general, the Steffen-New method and the SG method have some optimisations over the Steffen-Orginal method. The SG method is more optimised than Steffen-New method. This optimisation is particularly noticeable when 80% or less of the passengers are boarding with luggage.

The reason for this effect is that for Steffen-New, compared to Steffen-Original, the boarding time of passengers seated in the first row of several rounds is reduced. Since both methods require 12 boarding rounds[7] of boarding, because Steffen-Original has a random boarding queue, the worst case may be that each round of boarding will place the passenger seat with luggage in the first row, which will cause more waiting time for the next round. The boarding passenger must wait for the last passenger in the last round to complete the boarding before proceeding.

Steffen-New boarding method let the passengers with the luggage board first, when the passenger with luggage in the previous round boarded, the waiting time for next round is the same as Steffen-Original. However, for passengers who do not carry luggage in the next few rounds, the waiting time for next round is shortened. Because passengers who seat on the first row in more rounds have no luggage, thus the boarding is faster.

[6] The first column in the table is the methods name, and the first row is the data of the different percentage of passengers carrying luggage."SO" means "Steffen-Original", "SN" means "Steffen-New", "SG" means "Steffen-Greedy". All data units in the table are percentages.

[7] A passenger causes congestion on the first or second line of the aircraft, and it is said to have performed a "boarding round." For example, Steffen boarding method uses interlaced boarding, so it takes two boarding rounds to fill a column of seats, and 12 boarding rounds to complete the entire boarding.

For the SG method, it has the same advantages as Steffen-New. However, it also optimises the number of boarding rounds, because when the passengers with luggage are boarded, after then for an empty seat column, the passengers without luggage will use WILMA boarding method. That is, for a column with all empty seats, Steffen-New requires two boarding rounds, and the last boarding passenger of the two boarding rounds will cause two congestions in the aisle. However, the SG method requires only one boarding round, so it will only cause congestion once in the aisle. That is why the SG method is better than Steffen-Orginal boarding method and Steffen-New boarding method.

As one can see the optimisation becomes significantly smaller for test cases where 80% or more of the passengers carry luggage. The reason for this is that the more passengers carry luggage, the fewer rounds there are to show the advantages of the two new methods.

For example, if no passengers carry luggage, there will be 12 boarding rounds by using Steffen-New method, but it will only have six rounds by using the SG method. As the number of passengers carrying luggage increases, for another example, 70% passengers carry luggage, the rounds of boarding will be 12 by using Steffen-New method and 11 by using the SG method. However, when more than 108(90%) passengers carry luggage, the rounds of boarding will be 12 by using these two methods, which is also why the SG method is 0% optimised for Steffen-New when more than 80% of passengers are carrying luggage (refer to Table 4).

In conclusion, Steffen-New method has the advantage of optimizing the congestion time caused by the last passenger in each round who sits the first two row of the cabin. The SG method has this advantage, and it also can minimize the boarding rounds as much as possible.

4 Evaluate a new boarding method with team boarding

4.1 Introduction

In the previous chapter, it has proven that the Steffen boarding method is very effective. In particular, the most optimised Steffen(SG-method) boarding method also sets the buffer area by Greedy-Algorithm according to the amount of luggage carried by each passenger. This chapter will create a new algorithm called Steffen-Greedy-Team boarding(after this referred to as SGTB method) which also takes account of team boarding behaviour and setting up the buffer. After this, the newly created algorithm will put on test with the most commonly used methods by airlines, and finally to conclude.

For SGTB method, the passengers will randomly divide into different boarding crews, which means each passenger have a boarding crew number, and passengers board as a team boarding behaviour have the same boarding crew number except 0. The passenger has boarding crew number 0, that means he will board as individual boarding behaviour. Then set the aircraft seat map to 20 containers from back to front. Each container consists of 6 seats. Next, each boarding crew will set a buffer area for the next boarding crew according to the number of luggage and the number of members in this boarding crew. Finally, SGTB will take the ideas of bin packing and job scheduling to determine the best seat row of the next boarding crew.

According to some thesis(Müller, n.d. and Jafer and Mi, 2017), the most common boarding methods currently used by airlines are Back-To-front boarding method and Random method.

So in the following paragraph, first to use a random set of boarding crew numbers to replace the boarding crew set of the data set, which in 2.5.1 has created. Afterwards, simulated by using the Random method and SGTB method to collect output to compare the two methods. For other "Set-Areas" methods commonly used today, which also consider team boarding behaviour. In order to pursue the standard and higher performance of these methods, must create some suitable data for them(Explanation in 4.4.1), then compare them with SGTB and Random methods, and conclude.

4.2 Evaluating Steps

4.2.1 Create Data

In this section will create two data sets, the first one is for the SGTB method and the Random method. The second is for SGTB, Random, Back-To-Front(BTF), Back-To-Front_mix(BTFm) and Rotating-Zone(RZ) methods. For these two data sets, the passenger set and luggage set are the same as the data created in 2.5.1, except that the boarding crew number set has changed. The boarding crew number set must apply the following rules described below for each data set.

For the first data set, each data has 120 passengers, and 60 of them have their boarding crew number, other passengers board as individual boarding behaviour, which means their boarding crew number is 0. Moreover, each boarding crew has a random number of members.

As for the second data set, each data has 120 passengers, and also 60 of them have their boarding crew number. However, each boarding crew has most 3 passengers, and the passengers of the same boarding crew can not board on different boarding areas when using "Set-Areas" methods.

4.2.2 Test if Steffen-Greedy-Team Boarding method more effctive

For testing, if the SGTB method is more effective, as one can compare the SGTB method with the most commonly-used boarding method, which is the Random boarding method. First, use the first dataset(refer to 4.2.1) as input arguments, then use 2.5.2 to calculate the boarding order for each method for each data. At last use 2.5.3 to collect the results of both methods, and draw a graph of the result.

For the second test, it uses the second dataset(refer to 4.2.1) to compare SGNB method with all commonly-used boarding methods, Back-To-Front, Back-To-Front_mix, Rotating-Zone and Random boarding methods. Once again, uses 2.5.2 and 2.5.3 to simulate by using these methods to get the result.

4.2.3 Conclusion and analyse Result

After completing the above steps, one can conclude the result. Then find explanations for why one method outperforms another one. At last, evaluate and analyse the feasibility of the best team boarding method.

4.3 Steffen-Greedy-Team Boarding Method vs Random Method

4.3.1 Steffen-Greedy-Team Boarding Method introduction

In general, the SGTB method based on setting a buffer according to previous boarding crew's information, which are numbers of crew's members and numbers of luggage, and using bin packing theory and job scheduling theory to find the best row for each boarding crew. Step by step, the first step is to determine a "best row" based on the buffer set by the previous boarding crew. For a simple example, refer to Figure 11.

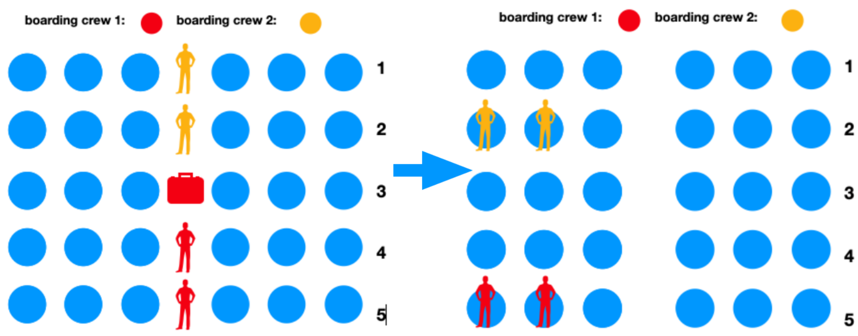

Figure 11: SGTB find "best row"

As figure 11 shows, boarding crew 1 does not have a previous boarding crew, so the "best row" for boarding crew 1 is line 5. Then it has two passengers, and one of them has luggage, then it will set a buffer which is two lines for boarding crew 2, so the "best row" for boarding crew 2 is line 2. Occasionally the "best row" for the boarding crew is already filled up completely and in this case "best row" will go up one line until the free seats of this line is enough for the boarding crew. Refer to figure 12.

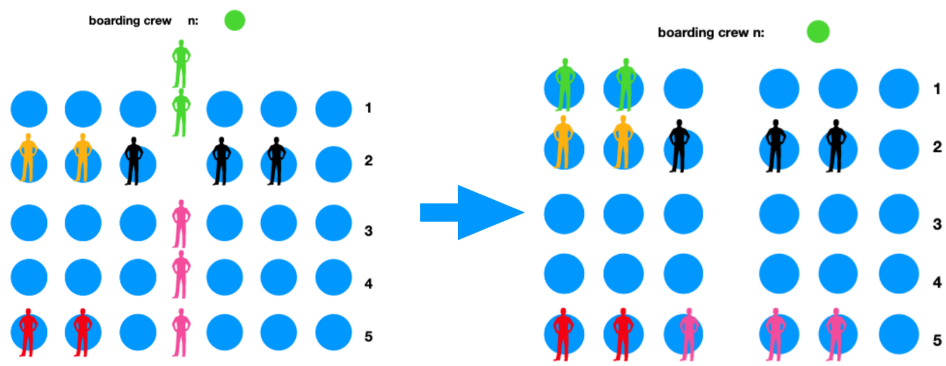

Figure 12: “best row” go up 1 line

As figure 12 shows, the green boarding crew has two passengers, and the previous boarding crew is the pink one. The pink boarding crew has three passengers without luggage, and the “best row” for pink crew is 5, so it will set a two lines buffer for the green boarding crew, the “best row” for green crew is supposed to be 2, but there are not enough free seats for the green boarding crew. Therefore the “best row” will go up 1 line, which is line 1. Nevertheless, what if all lines do not have enough free seats for a boarding crew? In this case, split the boarding crew from the “best row”. It will always go up 1 line to find free seats to allocate until all passengers of the boarding crew are assigned seats. Refer to Figure 13.

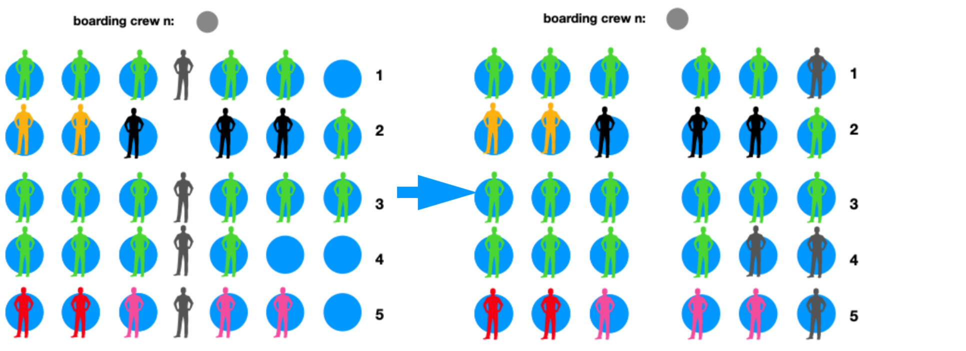

Figure 13: Split Boarding Crew

As figure 13 shows, suppose the grey boarding crew is the last, and suppose its "best row" is line 5, then the "best row" needs to go up 1 line. However, each of the lines does not have enough empty seats for this boarding crew, so split the boarding crew and assign seats to each passenger from "best row" which is line 5. In the figure, one can see that passenger 1 will sit on line 5, passenger 2 and passenger 3 will sit on line 4, and passenger 4 will sit on line 1. If there will be more boarding crews, we need to consider how large to set the buffer area. At this time we need to find the last group in the split boarding crew. In this case, the last group has only one passenger without luggage, so it will set a buffer which is 1 line for the next boarding crew.

The SGTB algorithm pseudocode refer to the appendix 2.

4.3.2 Random Method(Team) introduction

For the random boarding method, each boarding crew will randomly pick a "best row". Whenever that "best row" does not have enough free seats or the boarding crew needs to split, then the same rules as SGTB are applied. A simple example shows the difference between the two methods, refer to Figure 14.

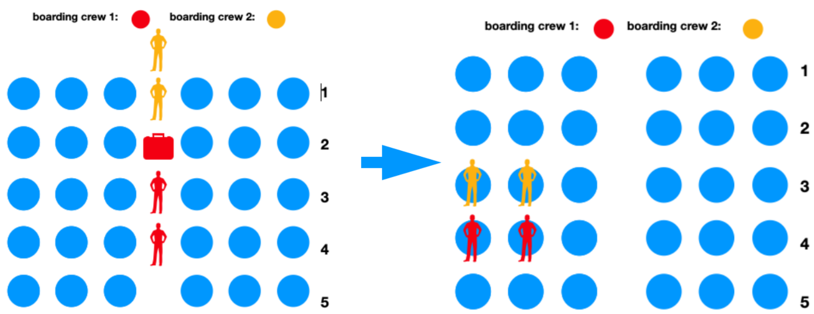

Figure 14: Random boarding method(Team)

As figure 14 shows, the boarding crew 1 picks a random line(here pick the line 4), and for the boarding crew 2, it will also randomly pick a line form line 1 to line 5(here pick the line 3).

4.3.3 Result

Input the first data set created in 4.2.1 as program arguments. Each method has simulated 29 times, and collect the average of the boarding time of each percentage case, as mentioned in the first paragraph of 4.2.2.

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Random	48.6	55.7	63.7	71.1	74.7	82	85.7	92.9	101.4	105	109.4
SGTB	43.9	49.1	53	57.7	63.1	66.1	70.2	75.2	82.4	88.5	91.9

Table 5: Random vs SGTB

Based on Table 5[8] , that can draw a graph (refer to figure 15). It presents a data comparison of the two methods in each case in the form of a graph.

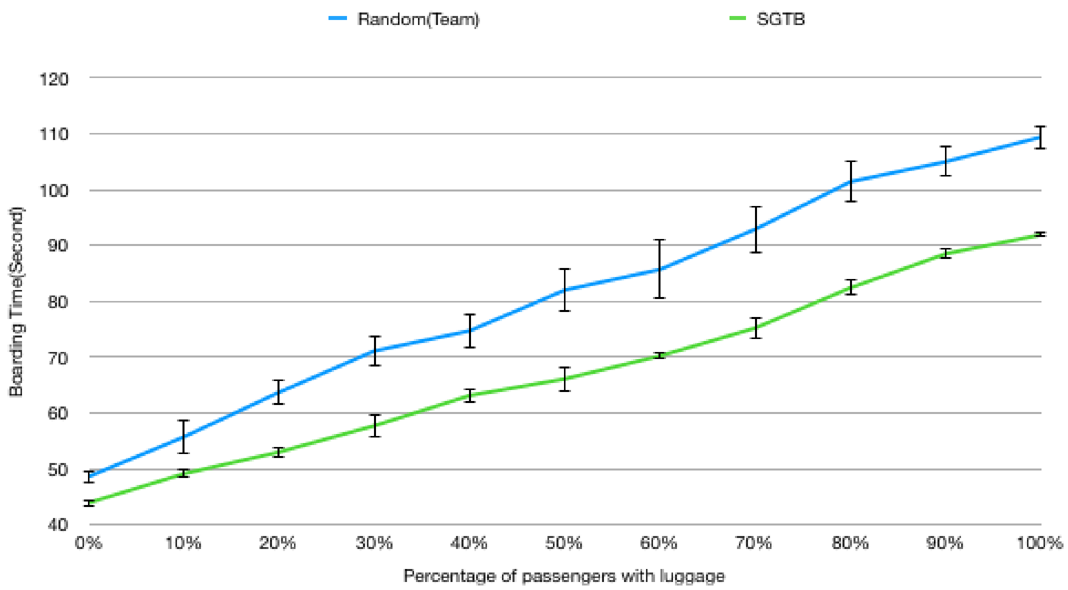

Figure 15: Result Graph

As shown in the figure, in all cases, the SGTB method is significantly better than the Random method. With the increase in the number of passengers carrying luggage, it can be observed that the interval between the two chart lines is larger. Which means that the more passengers carry luggage, the better the SGTB method performs over the Random method.

[8]The first column in the table is the methods name, and the first row is the data of the different percentage of passengers carrying luggage.

For the SGTB method, it effectively uses the boarding buffer. To the extent that it is guaranteed that passengers boarding the crew can sit in the same row, and the buffer area from the previous boarding crew is minimal, which means that passengers on multiple boarding crews can board at the same time. Compared with Random boarding, these points may affect the boarding time.

Because, for Random boarding method, choosing a row of the seat is random, the buffer area of the two boarding crews may be too large to maximise the simultaneous boarding of multiple boarding crews. It may also cause the buffer area of the two boarding crews to be too small to cause unnecessary congestion, which means that the next boarding crew must wait for several passengers or all passengers of the previous boarding crew to complete the boarding.

From this, it can conclude that, under the model, by optimising the buffer between boarding crews, compared with the commonly used Random method, the boarding time has been greatly optimised.

4.4 Steffen-Greedy-Team Boarding method vs Other Methods

In addition to the Random method, many airlines choose to use the Back-To-Front boarding method now, and a small number of airlines also choose the Rotating-Zone boarding method. So this subsection will choose the Back-To-Front, Rotating Zone, Back-To- Front mix methods for simulation.

Because all three methods divide the aircraft into several boarding areas, in order to meet the requirements of team boarding and to make these three methods perform best, one must recreate the appropriate simulation data set. See above(4.2.1 second data set) for details. In the following three methods' introduction will also explain why the data set needs to be reconstructed.

Finally, use the created data to simulate (following 2.5 simulation steps) the three methods mentioned above, as well as the Random and SGTB methods, and then compare them to conclude.

4.4.1 Other Three Methods Introduction

Under the model, each boarding area has to be four rows, which means that each boarding area allows 24 passengers to board. The only difference between these three methods is that the boarding orders of the boarding areas are different.

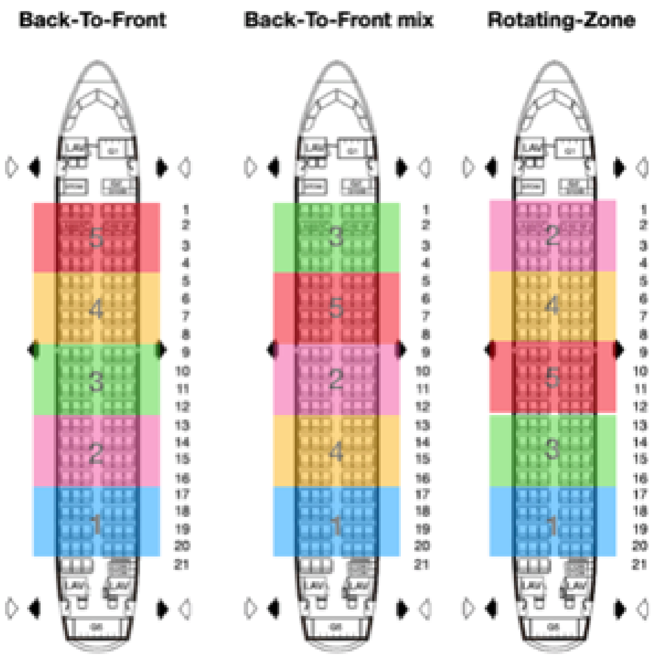

Figure 16: Boarding Areas for 3 different Methods

As shown in Figure 16, different methods will board in different orders. For example, Back-To-Front boarding method, there will be 24 passengers boarding in boarding area 1, and after completing, it will continue to fill in boarding areas 2, 3, 4, 5.

Each boarding area must have the best boarding strategy, which is the rules introduced in the SGTB method. It will reduce the boarding time of each boarding area as much as possible. Similar to SGTB, as one can treat each boarding area as a complete boarding process, and then perform the boarding process of each boarding area in turn according to the order of the boarding area in different methods. In other words, there are 24 passengers in each boarding process, and four containers (4 rows of seat) for each boarding process are set following the boarding area order. For each boarding process, the buffer is also set for each boarding crew according to the number of team members and the number of luggage.

At the same time, it can also explain why we need to reset the data set. Each boarding team will not have more than three passengers, because each boarding area has only four rows, so if there are more than three passengers, there is no way to set the buffer, which will not reflect the advantages of the method. For the case of fewer than three people, different situations can be mixed (buffer is set or not), which is more effective and closer to the real situation and let these three methods have higher performance.

Moreover, for another rule of the dataset, let us take Rotating-Zone method for an example. Each boarding area allows 24 passengers on board, so what if the 24th passenger and 25th passenger are the same boarding crew? The 24th passenger will board the boarding area 1, and the 25th passenger will board the boarding area 2. Even if they are from the same boarding crew, their seats are so far away. A team boarding method must avoid this situation.

4.4.2 Result and realistic Conditions

Input the second data set created in 4.2.1 as program arguments. Each method has simulated 29 times, and collect the average of the boarding time of each percentage case, as mentioned in the second paragraph of 4.2.2.

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Rotating-Zone	48.3	58	66	74.1	78	86.1	94.9	103	107	115.3	121.6
Back-To-Front	47.7	58.8	65.4	75.1	80.2	87.3	96	103.5	108.1	116.8	123.6
BTF_mix	47.1	57.1	64.4	72.3	77.4	86.7	93.8	101.4	106.6	114.1	121.3
Random	47.9	52.8	61.7	68.6	71.6	79.5	82.9	89.7	94.4	100.7	109.5
SGTB	44.3	47	51.3	58.4	60.3	64.9	69.9	71.8	76	81.1	85.4

Table 6: Boarding Time of 5 Methods

Based on all the data in Table 6[9] , create a graph of the result. Figure 17 visually shows the performance of five different team boarding methods in different situations.

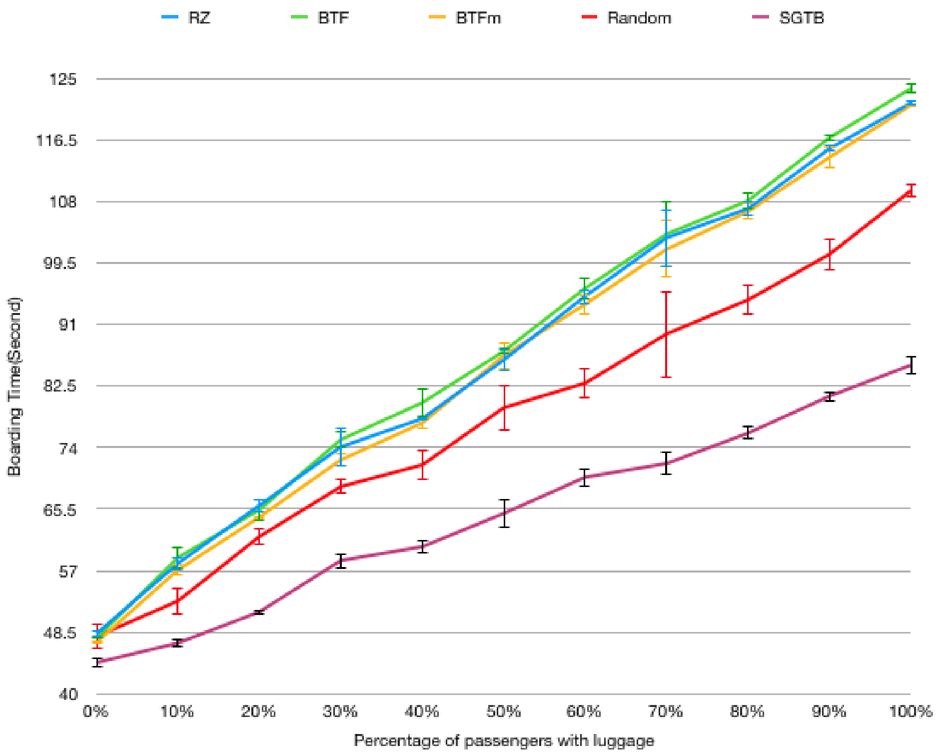

Figure 17: Result Graph

[9]The first column in the table is the methods name, and the first row is the data of the different percentage of passengers carrying luggage.

As one can see in the figure, the Random boarding method is significantly better than the three partition boarding methods, and the SGTB method is significantly better than the Random method, and SGTB method is also the most stable method.

For the three boarding methods, the Back-To-Front boarding method performs the worst. Because, in the sequence of adjacent boarding areas, there is no buffer, which results in that few passengers in two adjacent boarding areas can board almost at the same time. The second is the Rotating-Zone method. Although set the buffer zone for adjacent boarding areas, but the buffer zone settings are not reasonable and stable. For example, boarding area 1 and 2 are too far apart. When boarding in boarding area 2, a large part of the aisle is idle, and all passengers will cause congestion at the head of the aircraft. The best performing BTF-mix of the three methods solves the disadvantages of the first two methods, but the time saved may be only the boarding time of single-digit passengers, so from the results, all three methods are not effective enough.

For the Random boarding method, due to the randomness and no boarding area, the usage rate of the aisle is higher than the three methods mentioned above. That is, more passengers can board almost at the same time; that's why this method can reduce total boarding time. Therefore, in the most commonly used method at present, most airlines still use Random boarding method.

For the SGTB boarding method, the algorithm can maximise the usage of the aisle, so the advantages of the method are distinct. However, from the actual situation, SGTB boarding method is also the most difficult to implement. For the methods of dividing the boarding areas, a specified number of passengers in a random queue can enter the designated boarding area in order. For the Random boarding method, as everyone is familiar with, airlines can give passengers a random seat when they are checking in. Then the passengers enter the aircraft to find their seats in turn according to the random queue.

However, for the SGTB method, the boarding order needs to be generated in real-time at the gate, which also means that passengers need to provide additional information (number of luggage, boarding crew) when they are checking in. Their seats will be assigned in real-time according to their boarding information and the position of the boarding queue when they pass the boarding gate (when scanning a QR-code). So airlines need recyclable electronic boarding cards for passengers, or passengers can use the E-Tickets on their phones.

Nowadays, more and more people use e-tickets on mobile phones for boarding, and it is also environmentally friendly for airlines to prepare recyclable electronic boarding passes, so I think the SGTB method is feasible.

5 Conclusion and Development

5.1 Results achieved

This thesis has obtained research results in three directions: The theoretical boarding method, the feasible boarding method and boarding simulator.

5.1.1 Theoretical Boarding Method

The theoretical boarding method means that regardless of the actual situation, the boarding process can theoretically be carried out as quickly as possible. Therefore, in this thesis, SG boarding method based on the Steffen-New boarding method is used to optimise the boarding time by sorting orders according to the passengers carrying luggage or not and using the Steffen boarding method and the WILMA boarding method in a reasonable combination.

5.1.2 Feasible Boarding Method

The feasible boarding method means taking reality into account. The most crucial factor is to include the conditions of team boarding. In this thesis design a new algorithm SGTB by combining the idea of the SG boarding method with the team boarding conditions. The simulation results are significantly better than the most commonly used boarding methods. At the same time, suggestions are also made on the actual conditions that need to be implemented for this boarding method.

5.1.3 Boarding Simulator

The simulator in this thesis consists of 2 significant functionalities. On the one hand, the simulator is capable of generating different boarding order by different algorithms. At the other hand, the boarding process can be simulated based on the boarding order. It is convenient for evaluating other boarding methods that are not shown in this thesis or for the new algorithms when someone will create in the further.

5.2 Development

For the algorithm and simulator, there is still much space for development. First of all, a more optimised structure can be made for the code. Also, when adding more actual parameters, questions like "how can the algorithm be improved to perform better?" or "How can improve the algorithm and simulator when dealing with different aircraft models?" remains open.

A Source Code

The following link was checked on January 27, 2020.

The source code of this bachelor thesis: https://github.com/th136006386/AircraftSimulator

B Steffen-Greedy-Team Boarding Algorithm Pseudocode

Algorithm 2 SGTB-Algorithm

Input:
 Boarding Crew c_i;
 bestRow from Boarding Crew $c_i - 1$;
 buffer from $c_i - 1$;
 count = 0;
Output:
 bestRow for Boarding Crew c_i;
 buffer for Boarding Crew $c_i + 1$;

```
bestRow = bestRow + buffer
if bestRow > 19 then
    bestRow = 0;
end if
while !currentRowHasEnoughSeats(bestRow) and count < 20 do
    bestRow = bestRow + 1; //"best row go up one line"
    if bestRow > 19 then
        bestRow = 0;
    end if
    count + +;
end while
if count < 20 then
    assignedSeat(c_i, bestRow);
    buffer = countMembers(c_i) + countLuggage(c_i);
else
    splitBoardingCrew(c_i);
    assignedSeatForEachFromBestRow(c_i, bestRow);
    g = findLastSplitGroup(c_i);
    bestRow = findBestRow(g);
    buffer = countMembers(g) + countLuggage(g);
end if
```

References

Kento Aida (2000). “Effect of job size characteristics on job scheduling performance”. In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, pp. 1–17.

Massoud Bazargan (2007). “A linear programming approach for aircraft boarding strategy”. In: *European Journal of Operational Research* 183.1, pp. 394–411.

Richard Cimler, Eva Kautzká, Kamila Olševičová, and Martin Gavalec (2012). “Agent-based model for comparison of aircraft boarding methods”. In: *Proceedings of 30th International Conference Mathematical Methods in Economics Agent-Based*, pp. 73–78.

Camelia Delcea, Liviu-Adrian Cotfas, and Ramona Paun (2018a). “Agent-based evaluation of the airplane boarding strategies’ efficiency and sustainability”. In: *Sustainability* 10.6, p. 1879.

Camelia Delcea, Liviu-Adrian Cotfas, Mostafa Salari, and R John Milne (2018b). “Investigating the random seat boarding method without seat assignments with common boarding practices using an agent-based modeling”. In: *Sustainability* 10.12, p. 4623.

FINNAIR (n.d.). *A319 Seat Map*. http://www.finnair.fi/filecontent/common/a319.gif.

Shafagh Jafer and Wei Mi (2017). “Comparative study of aircraft boarding strategies using cellular discrete event simulation”. In: *Aerospace* 4.4, p. 57.

David G Kendall (1951). “Some problems in the theory of queues”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 13.2, pp. 151–173.

Richard E Korf (2002). “A new algorithm for optimal bin packing”. In: *Aaai/Iaai*, pp. 731–736.

Digital Media Law (n.d.). *Fair Use*. http://www.dmlp.org/legal-guide/fair-use.

Silvia Mas, Angel A Juan, Pol Arias, and Pau Fonseca (2013). “A simulation study regarding different aircraft boarding strategies”. In: *International Conference on Modeling and Simulation in Engineering, Economics and Management*. Springer, pp. 145–152.

R John Milne and Alexander R Kelly (2014). “A new method for boarding passengers onto an airplane”. In: *Journal of Air Transport Management* 34, pp. 93–100.

Justin Müller (n.d.). *Optimal Boarding Methods for Airline Passengers*. https://www.fzt.haw-hamburg.de/pers/Scholz/arbeiten/TextMuellerJ.pdf.

Gennaro Notomista, Mario Selvaggio, Fiorentina Sbrizzi, Gabriella Di Maio, Stanislao Grazioso, and Michael Botsch (2016). “A fast airplane boarding strategy using online seat assignment based on passenger classification”. In: *Journal of Air Transport Management* 53, pp. 140–149.

Michael Schultz, Christian Schulz, and Hartmut Fricke (2008). “Efficiency of aircraft boarding procedures”. In: *Proceedings of 3rd ICRAT*, pp. 371–377.

Skytrax (n.d.). *World Airline Seat Map guide*. https://www.airlinequality.com/info/airline-seat-plans/.

Jason H Steffen (2008). “Optimal boarding method for airline passengers”. In: *Journal of Air Transport Management* 14.3, pp. 146–150.

Jason H Steffen and Jon Hotchkiss (2012). “Experimental test of airplane boarding methods”. In: *Journal of Air Transport Management* 18.1, pp. 64–67.

Tie-Qiao Tang, Shao-Peng Yang, and Liang Chen (2019). "An Extended Boarding Strategy Accounting for the Luggage Quantity and Group Behavior". In: *Journal of Advanced Transportation* 2019.

List of Figures

List of Tables