**Scripts for presentation**

**—— stOOrz**

(P1) Eason

We are Team #22768821 from Shanghai Experimental School, and today we will introduce The Optimal Boarding & Disembarking Method.

(P2) Eason

This is the structure for our presentation.

(P3) Lyra

To begin with, we’ll look at the structure of our model.

(P4) Lyra

Time and efficiency play a vital role in air transportation. Therefore, it's necessary to build a model that provides the best strategy for different types of aircraft and on various occasions.

One of the most innovative aspects of our model is that we included quite a few theoretical mathematical analyses. They can justify our model's notions, concepts, and results and simplify the calculation.

Our model can be divided into three parts: Math Model, optimization in a mathematical account and programming.

(P5) Lyra

This flowchart demonstrates the process of boarding. Luggage-stowing is the major contributor to queuing.

(P6) Raymond

Assumptions should make our model discrete but plausible, which is the main reason for making most of the seemingly impulsive hypothesis afterwards.

We assume that the time wasted while passengers try to stuff extra luggage into their seats is qualitatively equivalent to that spent while stowing extra luggage.

Additionally, we hypothesize that passengers always walk at the maximum possible speed.

We also assume that the velocity in a particular cell remains constant, since it can simplify calculations without causing much inaccuracy.

(P7) Raymond

In the model, the definition of time and velocity differs from SI. We make these changes to make the calculations simpler.

(P8) Eason

In the following part, we’ll mainly calculate the total time systematically.

(P9) Eason

The loop here indicates that we calculate the total time based on recursion. According to the discreteness of our model, this can be reduced to finding a recursion formula for a distinctive passenger-based variable. We chose (velocity) as that variable out of simplicity and authenticity concerns.

(P10) Allan

Before turning the spotlight on the analysis, we’ll first construct the space of cells and coordinates as shown on the slide.

(P11) Allan

First, we label the cells and trace between which two adjacent cells is located to get the current cell of .

(P12) Allan

Next, we use dot products of vectors to calculate distribution according to density, as the slide shows.

(P13) Allan

After that, we use Greenshields speed-density linear model to develop the relation between and. We’d stress that the visibility range is taken as because is proper and realistic. It also decides the time step to be one sixth or one twelfth a second rather than being too large or small, therefore reducing overall complexity.

(P14) Allan

**(Play Animation)** This animation here shows how we calculate time and velocity. We can see how we do this recursively. Actually, the constants in the recursion formula here can be specified using only the previous states. This can significantly simplify calculations since we’re only conducting it with .

(P15) Allan

Finally, we get the recursion formula, showing that speeds are linearly interconnected, as the reciprocal of can be understood as the real-life speed. And we’ll also use two methods to justify our deductions: first in this section and then in the Sensitivity Analysis part.

(P16) Allan

Now we’ll tackle the congestion state. We divided the task into two parts: stowing luggage and offering seats. The first is trivial (but we’ll later add *discompliance* factors to this in the SA).

Offering seats can be calculated mathematically using permutation and the preservation of order.

Here is the schematic diagram for this procedure.

(P17) Allan

To somehow unite these seemingly separate states, we use a *state* parameter to indicate a passenger’s current movement status: moving, stowing luggage, or waiting in a queue. These states are interconvertible, and we’ll recursively do this with the interconversion formula within every time step.

(P18) Allan

The transformation formula is given as shown. Note that the multiplication of matrixes and vectors again ensures linearity.

Therefore, the only remaining problem is when a passenger takes his seat and disappears. According to programmatic views, we use deletion in the queue to tackle this, which is relatively trivial.

(P19) Allan

Here we give the results. The weights can be easily calculated with matrix multiplications.

(P20) Allan

After obtaining all these indicators, we will calculate the total time.

(P21) Eason

This part will focus on the modelling approach to minimise the total time.

(P22) Eason

Our work can be divided into inspiration from our previous calculations and strict mathematical proof.

Empirically, we’ll raise *parallelity* to describe the proportion of occupied aisle cells. The higher the parallelity, the more efficient the system is and the faster the strategy is. The formulae of parallelity are shown in the slide.

(P23) Eason

We’ll prove the intuitive idea proposed on the previous slide.

Based on the linear model, we can do these analyses according to the fact that linear optimums occur on the verges. The flow chart gives detailed analyses.

(P24) Eason

Secondly and mathematically, we can justify this using a critical claim. You can refer to this in the essay, and we’ll not attach importance to this as we’ve already explained the ideas behind it.

(P25) Raymond

Besides the total time, passengers’ satisfaction is also an essential factor to consider. In real-life experiences, dissatisfaction mainly comes from queuing and offering seats. Additionally, fellow passengers may be split, causing dissatisfaction. The total dissatisfaction index is the weighted sum of the three factors, the weights respectively 1, 250 and 10.

(P26) Eason

These are the results after the application of our model.

(P27) Eason

We can see that Steffen Sub-Perfect is the fastest and back to front is the most satisfactory plan.

(P28) Raymond

The starting point and destination of the passengers are exchanged, only adding the prerequisite that passengers leave from aisle to window. Therefore, the best strategy should be similar to boarding. The passengers have already been in an ideal queue, thus spending less time than boarding because of higher parallelity.

**Animation**

(P29) Eason

We conducted a Sensitivity Analysis on different passengers.

(P30) Eason

Random boarding is the most sensitive for longer stowing time and reduced passengers, while queue-jumping affects all strategies.

(P31) Eason

These are the major conclusions drawn from our sensitivity analysis: Random is far more sensitive than front to back, for randomised sequences can result in immeasurable effects. Back-to-front is the best overall because it is the least sensitive and has better time and satisfaction.

(P32) Raymond

Now we will extend our model to different aircrafts.

(P33) Raymond

We start with the development of coordinates, as demonstrated on the screen.

(P34) Raymond

And for the TETA aircraft, we define the grid as shown.

(P35) Raymond

This is Claim Two, which helps us find the best strategy for the two kinds of aircraft. We prioritised the cells from main to block and introduced efficiency to assess how good a scheme is qualitatively.

(P36) Raymond

In the end, we conclude that these two aircraft can be divided into smaller individual parts that are similar to ordinary one-aisle aircraft.

(P37) Raymond

To optimise the whole plan, we need to apply it to in-group and between-group sequences. We arranged for inner group passengers to fill empty blocks to ensure that every cell is used. The graph shows our best strategy for Flying Wing aircraft.

(P38) Raymond

And here is the result for the TETA aircraft.

(P39) Lyra

To sum up the strengths and weaknesses,

(P40) Lyra

our model features mathematical and programmatic accuracy. Multiple situations are considered using similar codes, so universality and efficiency are other benefits.

And for the weaknesses, we introduced quite a few abstract variables, making our model complex. The boarding schemes we devised are also hard to operate in reality.

(P41) Lyra

Thank you!