演讲稿

**Scripts for presentation**

—— stOOrz

**TBD: 衔接、标题选择**

**(P2)**

**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 aircrafts and on various occasions.

**This flowchart demonstrates the process of boarding. The main factor that causes the queue is passengers’ stowing their luggage.**

**Our model can be divided into three parts: Math Model, Optimisation in a mathematical account and Programming.**

**(P3)**

Assumptions should **make our model discrete** but also **plausible**. This is the main reason for making most of the **seemingly impulsive hypothesis** afterwards. We omitted the subsidiary assumptions, and you can refer to them in the essay.

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.**

**(P4)**

**Now we justify in detail that the velocity in a particular cell remains constant. This assumption enables us to simplify the calculation for the velocity (as it was only used when calculating distance) and wouldn’t dramatically affect the total time for the basic timestep (1/6 sec) is short.**

**(P5)**

Here are the meanings of different colours you might see in the notation part: Constant A, B and variables.

**(P6)**

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

**(P7)**

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

**(P8)**

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

**(P9)**

So far, we are only looking into the regular cases where passengers move freely without being blocked. There aren’t scenarios of contradictions such as equals infinity in this case.

**(P10)**

First, we’ll calculate the velocity according to the density. The density defined in the model is as shown on the slide. The visibility range is taken as because is proper and realistic. It also decides the time step (if taken as , the time step can be or sec), reducing the complexity of the simulation.

**(P11)**

Next, we use Greenshields speed-density linear model to develop the relation between and.

**(P12)**

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

**(P13)**

Additionally, the distribution has correspondent associations with the velocities by using partial summation. To be more specific, we label the cells and trace between which two adjacent cells is located.

**(P14)**

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

**(P15)**

As mentioned before, now we’ll come to the second scenario: when someone is causing a queue.

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).

**(P16)**

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

Here is the schematic diagram for this procedure.

**(P17)**

To somehow *unite* these two 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 do this with the interconversion formula recursively within every timestep.

**(P18)**

The formula is now given, which is further improved compared with our essay. Note that the multiplication of matrixes and vectors again ensures linearity.

**(P19)**

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

**(P20)**

Here we give the results. The weights can be calculated by accumulating all the and selecting the element. This can be easily done with matrix multiplications.

**(P21)**

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

**(P22)**

This part will focus on the modelling approach to minimize the total time. 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 the aisle cells that are occupied. 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)**

We’ll prove the intuitive idea proposed in 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)**

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.

**(P25)**

Disembarking is simply the reverse of boarding, only adding the prerequisite that passengers leave from aisle to window. **Therefore, the best strategy should be similar to boarding: to reach the highest parallelity.** The passengers have already been in an ideal queue, thus spending less time than boarding because of higher parallelity.

**(P26)**

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, respectively 1, 250 and 10. 1 is for standardization, and the others are based on real-life experiences and united magnitudes to consider totally.**

**(P27)**

**These are the results of our simulation.** **Steffen Sub-Perfect is the fastest and back to front is the most satisfactory plan.**

**(P28)**

**We use compliancy index to measure the sensitivity in figure. It shows the predictability of changes in the model, and the function to compare with is a relationship proved by our model.**

**(P29)**

**Case one is a longer stowing time. We use a random model – sigmoid model, as shown in the slides – to distribute the discompliance of passengers in a relatively realistic method, due to its speciality.**

**(P31)**

**We can conclude that in terms of longer stowing time random boarding is the most sensitive while front-to-back seems not sensitive. Queue jumping affects all methods, while random is the most sensitive to reduction of passengers.**

**(P32)**

These are the major conclusions drawn from our sensitivity analysis: **Random is far more sensitive than front to back, because 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.**

**(P33)**

Then we will apply our model to different aircrafts, and we will start with the development of coordinates, as demonstrated on the screen.

**(P34)**

And for the TETA aircraft, we define the grid as shown. **The seats with a negative -coordinate are the first class.**

**(P35)**

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

**(P36)**

**In the end, we come up with the conclusion: these two aircrafts can be divided into smaller individual parts that are similar to ordinary one-aisle aircrafts.**

**(P37)**

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

**(P38)**

And here is the result for the TETA aircraft.

**(P39)**

**To conclude, as for the strengths of our model, we considered various situations and used programs to simulate the process, so the model can be accurate. Multiple situations are considered using similar codes, so universality and efficiency are two benefits.**

**And for the weaknesses, we introduced many variables, some of which are a bit abstract, making our model complex. The strict boarding sequences make it difficult to operate.**

**(P40)**

**Last but not least, we write a letter to provide the airline executives with some suggestions**. First, we **point out two critical factors: hommization and efficiency**. **Secondly, we draw a simple chart to illustrate our plan** and **offer some simple tips**. **Airline executives need to prevent passengers from being stuck in aisles, provide them with enough space to place their luggage, and avoid queue-jumping.**