**(P2)**

**Time and efficiency play a vital role in air transportation. For normal passenger flights, periods that require a significant amount of time include the boarding and disembarking of passengers. Therefore, it's necessary to build a model that provides the best strategy for different types of aircrafts and on various occasions.**

**To begin with, we will discuss the boarding and disembarking process, which is shown in the flowing chart. From the chart, we can find that the main factor that causes the queue is passengers’ stowing the luggage.**

**We need to design a model to calculate the total time and to apply to real-life planes to find a best strategy.**

**(P3)**

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

**(P4)**

The inherent structure of the problem indicates that **our model will be discrete.** Therefore, apart from the intuitive ones, other assumptions, such as the second one in this slide, will hinge on this property of discreteness. On the other hand, **it’s also essential to make our assumptions plausible.** This is the main reason for making most of the **seemingly impulsive hypothesis** afterwards.

**(P5)**

**Here are the assumptions of Model A.** In Model A, we would **consider the single-aisle case and combine them with other aircrafts based on their similarity.** It’s important to note that since we’ll vary the stowing time of passengers in the later slides, the first assumption is relatively reasonable. **The time wasted while passengers try to stuff extra luggage into their seats is in quantity equivalent to that spent while stowing extra luggage.**

**We also assume that passengers always walk at the maximum possible speed because they want to get seated as quickly as possible (though not properly stated in the essay).**

**(P6)**

**As for the velocity, we assume that the velocity in a certain cell remains constant.** This means that could be considered as points on the velocity function. **This assumption enables us to simplify the calculation for the velocity (for it was only used when calculating distance) and wouldn't cause a lot of inaccuracy** which will not dramatically affect the total time, as shown in the graph. This is partly because **the basic timestep is only , a concise time** that wouldn't influence the velocity and distribution of passengers much. Therefore **the velocity in a certain timestep wouldn't change much** so that it can be seen as a constant.

**(P7)**

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

**(P8)**

In the first model, we divide the variables into three types. **Constant A,** marked as A in this table, refers to the constants that will not change in the whole scope, in other words, the properties of the model. **Constant B** may vary in the entire scope but won’t change for a particular set of passengers and plane types, or the properties of an aircraft or a strategy. **Variables, describing properties of passengers,** will be for different initial sequences of passengers. (The slide changes through the speech)

**(P9)**

In this model, we will calculate the total time of boarding. According to the discreteness of our model, this task can be reduced to finding a recursion formula for any passenger-based variable. Out of simplicity and authenticity concerns, we chose (velocity) as that variable.

**(P10)**

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

**(P11)**

In addition, 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.

**(P12)**

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.

**(P13)**

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

**(P14)**

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

**(P15)**

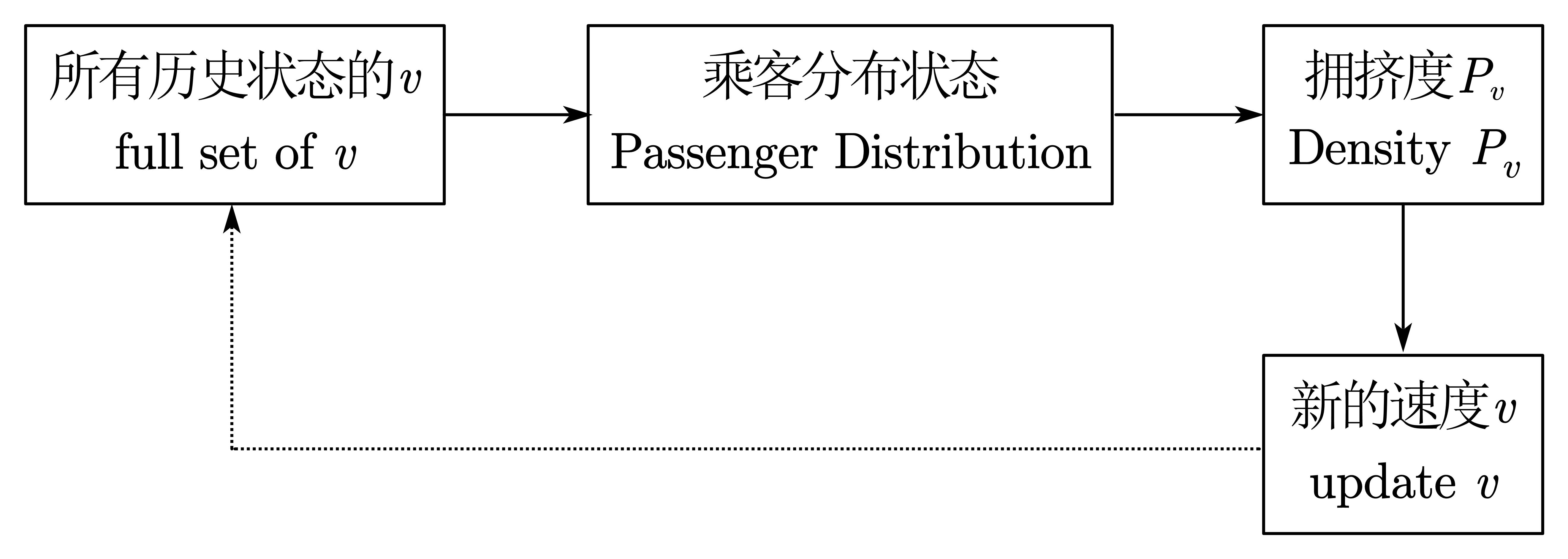
Additionally, the distribution has correspondent associations with the velocities by using partial summation.

**(P16)**

Therefore, we get the result.

Notice that previous calculations have shown that real-time speeds are associated with a linear bound. And we’ll also use two methods to justify our deductions: first on the next slide and then in the Sensitivity Analysis part.

This is the recursion formula. It clearly displays the linearity. ( can be understood as the real-life speed.)

****

**(P17)**

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

We divide 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). The latter can be calculated as shown mathematically – using permutation and the preservation of order.

**(P18)**

Here are the calculations.

**(P19)**

Here is the schematic diagram for this procedure.

**(P20)**

Now we’ll show the formula for the interconversion of states. The formulas here are further improved compared to our essay.

**(P21)**

Here are the ideal formulas. It preserves the linearity.

**(P22)**

Deletion is relatively trivial according to programmatic views.

**(P23)**

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



**(P24)**

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

**(P25)**

This part will focus on the modelling approach to optimising (or minimising) the total time. Our work can be divided into two parts: inspiration from our previous calculations and strict mathematical proof.

We’ll raise *parallelity* to describe how many of the aisle cells 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.

**(P26)**

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

First, based on the model, we can do these analyses as shown.

**(P27)**

The linearity of our model preserves these properties.

**(P28)**

Secondly and mathematically, we’ll also prove this with two significant claims. Claim One, shown on the slide, is about the optimality of all cells being occupied. Claim Two will be helpful when dealing with more complicated aircraft.

**(P29)**

**Disembarking can be seen as the reverse of boarding**. **Therefore, the best strategy should be similar to boarding: to reach the highest parallelity.** However, as there are no offering cell procedures, the passengers have already been in an ideal queue, thus spending less time than boarding because of higher parallelity.

**(P30)**

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. And according to the strict sequence, some fellow passengers may be split, causing dissatisfaction (though mistakenly not written in the essay). **The total dissatisfaction index is the weighted sum of the three factors.**

**(P31)**

**The weights of the factors are respectively 1, 250 and 10. The reason for 1 is for standardisation, and the others are based on real-life experiences, and also to unite magnitudes to make the ultimate dissatisfaction index combine the three factors.**

**(P32)**

**These are the results of our simulation.**

This is a comparison between different methods. **We can see that Steffen Sub-Perfect performs the best overall with Steffen Perfect,** back-to-front, window-to-aisle following; it is evident that the random method even outperforms the front-to-back method.

**(P33)**

We need to conduct a sensitivity analysis on our model, but how? **We use compliancy index to measure in figure. It shows the predictability of changes in the model, and the function to compare with is a relationship proved by facts.**

**(P34)**

**Case one of our Sensitivity Analysis 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.** We choose the sigmoid function due to its speciality in its functions and value, as shown in the figure to the right.

**(P35)**

**These are how we determine whether the model is stable or unstable based on how the graph looks.**

**(P36-P37)**

**We can conclude that random boarding is the most sensitive while front-to-back seems not sensitive.**

**(P38)**

**Next, we analysed the queue-jumping situation and concluded that both methods are sensitive, meaning queue-jumping significantly impacts total results.**

**(P39)**

**Last but not least, we researched on the reduction of passengers and found out that random boarding is the most sensitive (see the distribution of points) while back-to-front is not so sensitive.**

**(P40)**

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.**

**(P41)**

Then we will apply our model to different aircrafts and we will start with the development of coordinates. For the Flying Wing aircraft, as we've already divided it into four blocks, we define the intersection point of the main aisle and the ith block aisle as , and the ith block aisle as its x-grid. And for the TETA aircraft, we define the entrance cell in the left as , and the direction of the two aisles as the x-grid. **The seats with a negative -coordinate are the first class. And for the rest, passengers with seats -coordinated 1 and 9 would board first, while those with 4 and 6 board last.**

**(P42)**

**Here are the coordinates for the Two Entrance Two Aisle Aircraft.**

**(P43)**

**This is Claim Two, which helps us find the best strategy for the two kinds of aircrafts.**

**(P44)**

**Here are the main ideas when we apply the model to different aircrafts.** Two Entrance Two Aisle and the Flying Wing are two kinds of multi-aisle aircrafts, and we found that **they can be divided into smaller individual parts similar to ordinary one-aisle aircrafts.**

**(P45)**

To optimise the whole plan, we obviously need to **optimise the boarding sequence inside groups,** and then we need to **optimise the between-group sequences. The details are included in the pseudocode of the essay.**

**(P46)**

To ensure that every cell is used, we arranged **a few inner group passengers to fill up empty blocks.**

**(P47)**

**Our best strategy of Two Entrance Two Aisle aircraft is shown on the graph.**

**(P48)**

**Then we would introduce our strengths and weaknesses. As for strengths, we considered various situations and used programs to simulate the process. As the result, the model can be accurate because of the various situations considered, and also universal and efficient due to the usage of programs.**

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

**(P49)**

**We write a letter based on our conclusions to provide the airline executives with some suggestions**. First, we **point out two critical factors** in our boarding and disembarking process - **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.**