**(P1)**

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

**(P2)**

**To begin with, we will introduce the overall boarding process, which is shown in this chart on the slide. While boarding a plane, passengers will first go to their assigned seat, put their luggage on the rack and then get seated. While a passenger is stowing their bags, other travellers who are stuck behind should wait until the passenger finishes the process, which will cause a queue.**

**(P5)**

**Our model can be divided into three parts: Math Model, Optimization in a mathematical account and Program.** The program part can verify the other two parts, and the optimisation part can verify the models. The Math model part mainly discusses the factors that decide whether a plan is good: total boarding time and passenger satisfaction, including some optimisation. The optimisation part is about ways to make the strategy better. And the program part is to simulate the boarding process to verify our conclusion given by the model part.

**(P6)**

The inherent structure of the problem indicates that **our model will be discrete.** Therefore, apart from the intuitive ones, other assumptions, such as the third 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.

**(P7)**

**These two assumptions are about the moving state of passengers, and we’ve given correspondent justifications.**

**(P8)**

**These assumptions are respectively proposed to simulate reality and to simplify our calculations.**

**(P9)**

**Here are the assumptions of Model A.** In Model A, we would **consider the single-aisle case (although in our presentation, we tend to combine it with other types of aircraft due to their similar deduction methods)**. 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 qualitatively equivalent to that spent while stowing those extra luggage.**

**For the second assumption on this slide, we disambiguated the expression in our essay by using this interpretation: passengers always maintain maximum theoretical speed. A corollary to this is that queuing has the same effect on a *block* of adjacent passengers.**

**(P10)**

**Here are the last two assumptions. We’ll explain the second assumption soon afterwards.**

**(P11)**

**As for the velocity, we assume that the velocity in a certain cell remains constant.** This means that could be thought as points on the velocity function. Though it might not be so realistic, **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** (meaning that it wouldn't cause much change to 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. Thus according to the formula of calculating velocity, **the velocity in a certain timestep wouldn't change much** so that it can be seen as a constant.

**(P12)**

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

**(P14-P16)**

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. **Constant B** may vary in the entire scope but won’t change for a particular set of passengers and plane types. **Variables** will be for different initial sequences of passengers. (the slide changes through the speech)

**(P17)**

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 valuable.

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

**(P18)**

In addition, so far, we are only looking into the regular cases where passengers move freely without being blocked. In this case, there don’t exist scenarios of contradictions such as .

**(P19)**

First, we’ll calculate the velocity according to the density. The density defined in the model is as shown in the slide. The visibility range is taken as because is not too big or too small and is realistic, and it also decides the time step (if taken as , the time step can be or sec), reducing the complexity of the simulation.

**(P20)**

Next, e use Greenshields speed-density linear model (Ref. 3 in the essay) to develop the relation between and .

**(P21)**

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

**(P22)**

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

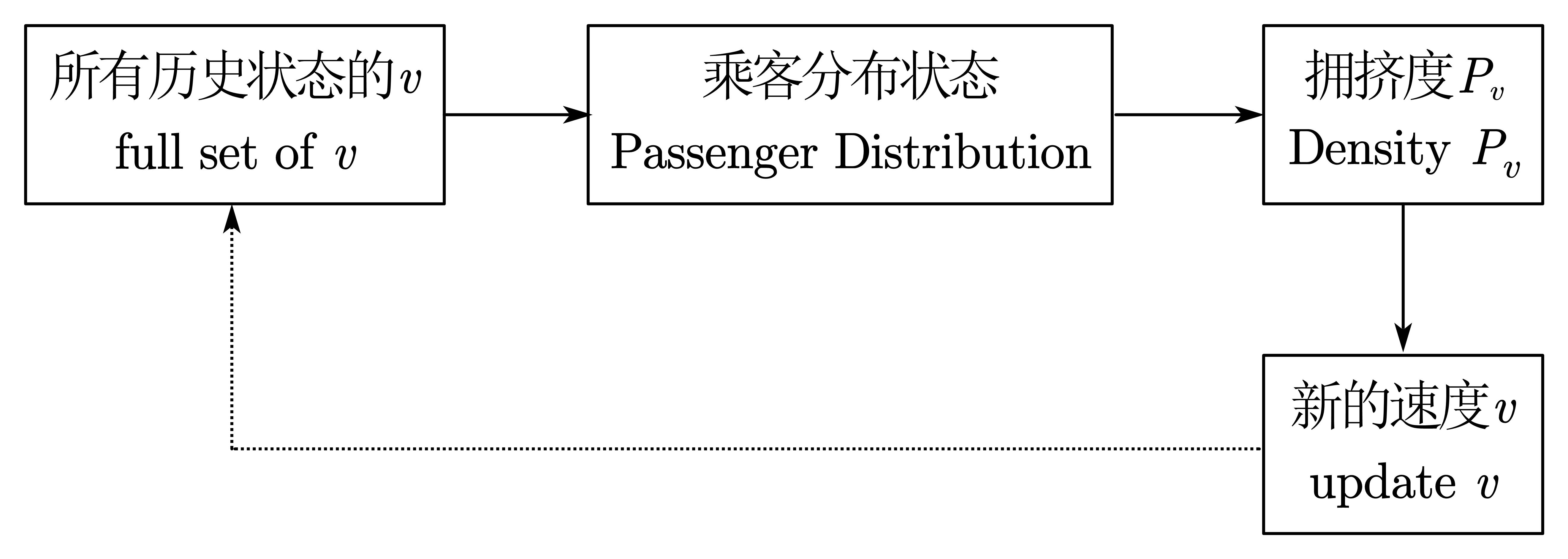
**(P23)**

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.

**(P24)**

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

****

**(P25)**

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.

**(P26)**

Here are the calculations.

**(P27)**

Here is the schematic diagram for this procedure.

**(P28)**

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

**(P29)**

Here are the ideal formulas. It preserves the linearity.

**(P30)**

Deletion is relatively trivial according to programmatic views.

**(P31)**

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



**(P32)**

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

**(P33)**

This part will focus on the modelling approach to optimising (or minimising) 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. Intuitively, 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.

**(P34)**

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

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

**(P35)**

The linearity of our model preserves these properties.

**(P36)**

Secondly and mathematically, we’ll also prove this with two significant claims. The first is about the optimality of all cells being occupied. The second will be helpful when dealing with more complicated aircraft.

**(P37)**

Here is Claim one. You can also refer to this in the essay. As the thesis has been shown before, we’ll not elaborate on this.

**(P38)**

**Disembarking is almost just the reverse of boarding,** for the motion is just the reverse of boarding. **So the best strategy should be similar to boarding.** However, for there are no offering cells, the passengers have already been in an ideal queue, thus spending less time than boarding because of higher parallelity.

**(P39)**

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.**

**(P40)**

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

**(P41)**

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

**(P42-P43)**

N/A.

**(P44)**

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

**(P45)**

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.**

**(P46)**

**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 dicompliance of passengers in a relatively realistic method.** We choose the sigmoid model due to its speciality in its functions and value, as shown in the figure to the right.

**(P47)**

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

**(P48)**

N/A.

**(P49)**

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

**(P50)**

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

**(P51)**

**Last but not least, we researched 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.**

**(P52)**

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.**

**(P53)**

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.**

**(P54)**

**Here are the coordinates for the TETA Aircraft.**

**(P56)**

**Here are the main ideas when we apply the model to different aircrafts.** TETA 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.**

**(P57)**

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.**

**(P58)**

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

**(P59)**

We have concluded some of the strengths of our model.

**First, accuracy. In our model, we consider several special situations. Also, we use several programs to facilitate our calculation**. This makes our result reasonable and precise.

**The second strength is universality**. In our model, **we succeeded in achieving visualizationtuations of the plane and successfully simulated the whole process of different boarding methods shown in the video clips just now.** This means that our model can be **applied to a variety of problems.**

**Finally, our model bears efficiency**. As shown in the second model, **we use a program to facilitate our calculations in finding the best strategy.** Therefore**, a lot of time is saved,** proving that our model has efficiency.

**(P60)**

Also, we have found some weaknesses that need to be improved.

**The first problem is complexity**. We introduce many variables and a variety of explanations in our model. **Some of them are a little bit abstract,** and some of our calculations conducted by programs aren't shown in this essay. This will **make our model more complex and less easy to understand**.

**The second weakness is that our model is challenging to operate** in reality. As seen in the descriptions above, **our model provides a plan with some details that must be strictly obeyed**. **This will increase the difficulty for the crews to exert this plan. However, we have thought of a method that can ease this difficulty.** **When there’s a passenger ahead waiting, we can first let him get to his seat.** According to our sensitivity analysis, **this will not significantly impact our boarding time**. Therefore, this kind of strategy is somehow reasonable and **flexible.**

**(P61)**

**We write a letter based o 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**. **Finally, we offer some simple tips** that could be applied to all kinds of planes. **Airline executives need to prevent passengers from being stuck in general aisles, provide them with enough space to place their luggage, and avoid queue-jumping.**