Stochastic Control and Optimization Project 2 – Dynamic Programming

Submitted by: Chaitra Setty cs63687, Garrett Sooter gas3254, Shreya Bhootda sb5863, Rohitashwa Chakraborty rc47878

Introduction

As an airline company, it is important to us to reduce the number of empty seats on any given flight in order to maximize profit. To achieve this goal, we have to sell more seats than are available on the flight based on assumptions of the number of people that will cancel or not show up to their flight. My team of four analysts has conducted analysis to identify the optimal number of tickets to offer for sale and maximize the discount profit. The manager does not want to overbook first-class tickets but is open to overbooking tickets in coach. When there are more people that show up than the number of coach tickets, then those passengers will be upgraded to first-class. This imposes an overbooking cost on the airlines because service in first-class is more costly.

Setting up the problem

Flight tickets are sold every day of the year and the flight capacity is a total of 120 seats including 100 seats for coach and 20 seats for first class. We assume that coach ticket holders show up for their flight 95% of the time and first-class ticket holders show up 97% of the time.

We first defined some assumptions and analyzed the expected revenue when only 5 seats are oversold. We assumed that if coach tickets are priced at $300 there is a 65% chance a ticket is sold and at $350 there is a 30% chance a ticket is sold. Similarly, if a first-class ticket is sold for $425 there is an 8% chance a ticket is sold and at $500 there is a 4% chance a ticket is sold. It costs $50 to upgrade a coach ticket to first-class and $425 to bump a coach passenger from the plane. We also took into account a 15% annual discount rate and calculated the daily discount rate.

Below is the setup of the problem based on the assumptions stated above.

First Class

```
]:  FC = {
        'seats': 20,
        'price_low': 425,
        'purchase_proba_low': 0.08,
        'price_high': 500,
        'purchase_proba_high': 0.04,
        'probability_showup': 0.97,
    }
```

Coach

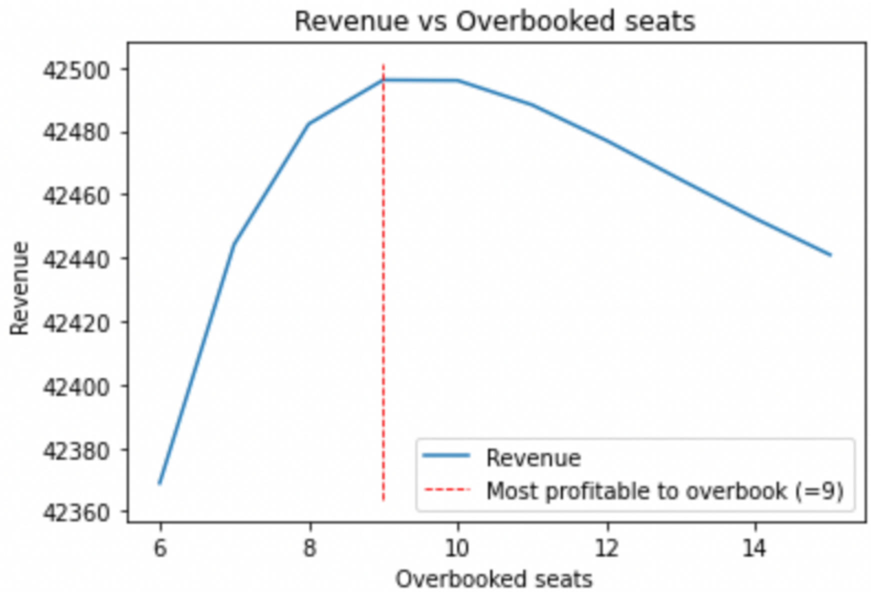
```
]:  COACH = {
        'seats': 100,
        'price_low': 300,
        'purchase_proba_low': 0.65,
        'price_high': 350,
        'purchase_proba_high': 0.30,
        'probability_showup': 0.95,
    }
```

Considering that there is no revenue on the day that flights take off, we initialize a value function and choice function that take in the coach ticket, first class ticket, and time. Then if tickets are overbooked, we calculate the probability of passengers showing up. Based on this, we can calculate the revenue of when the prices for coach and first-class are set high and high, low and low, high and low, and low and high respectively.

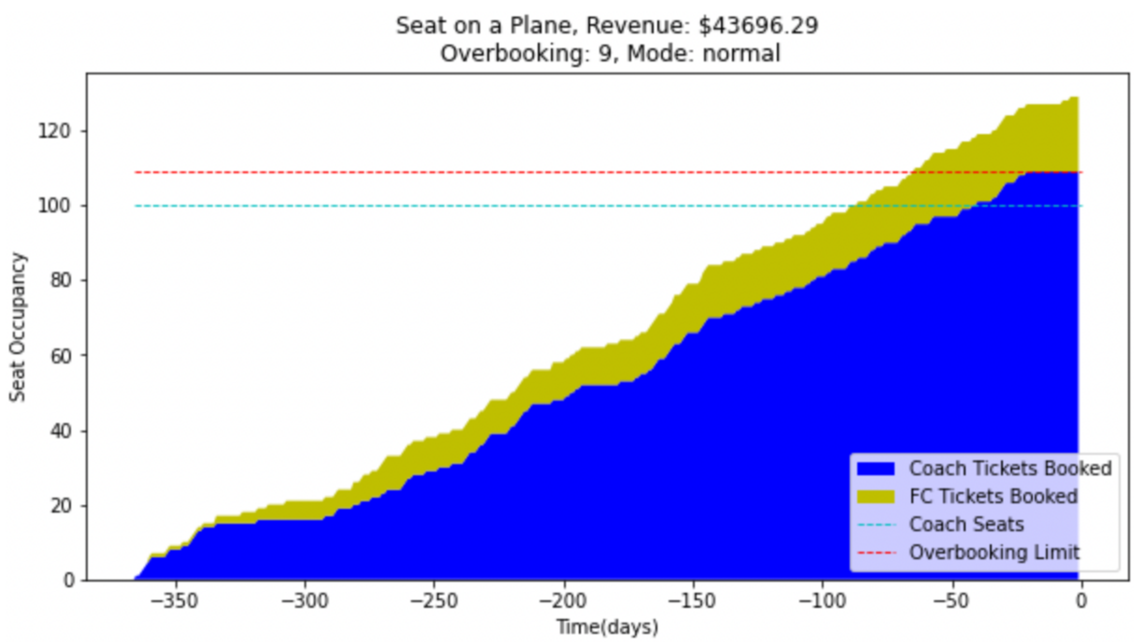
```
PRICING_OPTIONS = [
    (COACH['price_low'], COACH['purchase_proba_low'], FC['price_low'], FC['purchase_proba_low']),    #COACH_LOW_FC_LOW
    (COACH['price_low'], COACH['purchase_proba_low'], FC['price_high'], FC['purchase_proba_high']),  #COACH_LOW_FC_HIGH
    (COACH['price_high'], COACH['purchase_proba_high'], FC['price_low'], FC['purchase_proba_low']),  #COACH_HIGH_FC_LOW
    (COACH['price_high'], COACH['purchase_proba_high'], FC['price_high'], FC['purchase_proba_high']), #COACH_HIGH_FC_HIGH
    # No Sale Strategy pricing options
    (0, 0, FC['price_low'], FC['purchase_proba_low']), #COACH_NOSALE_FC_LOW
    (0, 0, FC['price_high'], FC['purchase_proba_high']) #COACH_NOSALE_FC_HIGH
]


decision_lookup = {
    0: 'COACH_LOW_FC_LOW',
    1: 'COACH_LOW_FC_HIGH',
    2: 'COACH_HIGH_FC_LOW',
    3: 'COACH_HIGH_FC_HIGH',
    4: 'COACH_NOSALE_FC_LOW',
    5: 'COACH_NOSALE_FC_HIGH'
}
```

As a result of these calculations, if 5 tickets are oversold, the expected revenue would be $42,242.86.

The natural question that arises next is what would the optimal overbooking amount be, for this strategy, to maximize revenue. After conducting a grid search over a range of values, we observe that there exists a clear maxima/upper-bound to the maximum revenue that can be earned with this strategy. **In our case, the maxima is attained at overbooking 9 seats and revenue is at $42,496.11**. After this point, the costs associated with overbooking increase and revenues diminish.



Below is a visualization of what a possible booking pattern would look like over time:

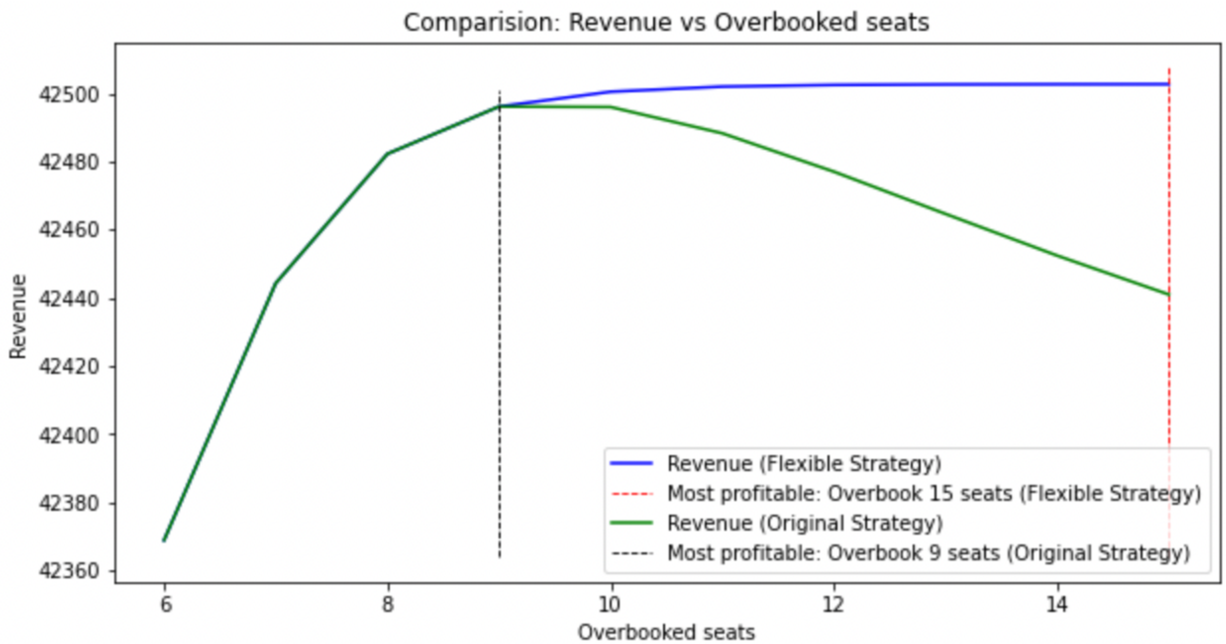


A new way ahead

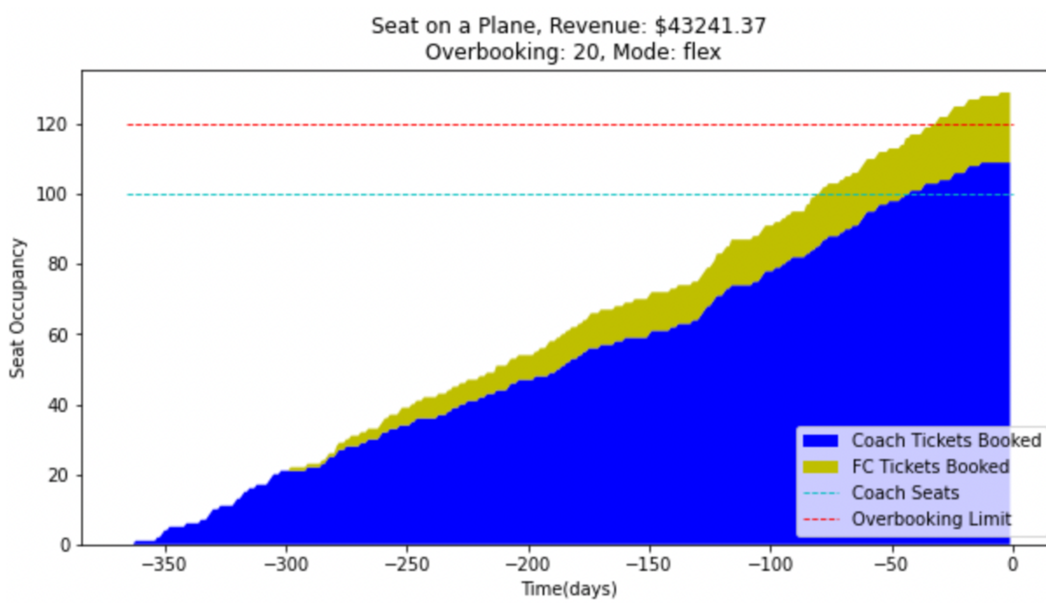
Furthermore, we explored a flexible strategy (the former being the normal strategy) which is when the airline has the option to not sell any coach tickets on a given day. Mathematically speaking, the flexible strategy is identical to the normal strategy with the addition of a ticket level which has a purchase probability of 0. This strategy grants the airlines greater control on its sales and therefore, should earn greater profits.

```
def NET_PRESENT_VALUE(c, fc, t, coach_price, coach_purchase_proba, fc_price, fc_purchase_proba, V):
    decision = None
    if c < COACH['seats'] + OVERBOOKING_COUNT and fc < FC['seats']: # seats available
        value = (
            (0 + DELTA*V[c, fc, t+1] ) * (1 - coach_purchase_proba ) * (1 - fc_purchase_proba)  # Neither tickets sold
            + (coach_price + DELTA*V[c+1, fc, t+1] ) * (coach_purchase_proba ) * (1 - fc_purchase_proba) # Only Coach tickets sold
            + (fc_price + DELTA*V[c, fc+1, t+1] ) * (1 - coach_purchase_proba ) * (fc_purchase_proba) # Only FC tickets sold
            + (coach_price + fc_price + DELTA*V[c+1, fc+1, t+1] ) * (coach_purchase_proba ) * (fc_purchase_proba) # Both tickets sold
        )

    elif c < COACH['seats'] + OVERBOOKING_COUNT and fc >= FC['seats']: # coach seats available, fc seats not
        value = (
            (0 + DELTA*V[c, fc, t+1] ) * (1 - coach_purchase_proba )    # Neither tickets sold
            + (coach_price + DELTA*V[c+1, fc, t+1] ) * (coach_purchase_proba )  # Only Coach tickets sold
        )

    elif c >= COACH['seats'] + OVERBOOKING_COUNT and fc < FC['seats']: # coach seats not available, fc seats available
        value = (
            (0 + DELTA*V[c, fc, t+1] ) * (1 - fc_purchase_proba)    # Neither tickets sold
            + (fc_price + DELTA*V[c, fc+1, t+1] ) * (fc_purchase_proba) # Only FC tickets sold
        )
    else: # c >= COACH['seats'] + OVERBOOKING_COUNT and fc >= FC['seats'] # No more seats available
        value = DELTA*V[c, fc, t+1] # No tickets sold
        decision = -1 # No more seats available

    return value, decision
```

A comparative analysis of the two strategies reveals that both strategies perform equally well upto 9 seats of overbooking,
However, as the margin for overbooking increases, the strategies diverge with the second strategy outperforming the first. One must note that the second strategy does not have a visible upper bound in the trend visualized below, however, the **strategy plateaus beyond a limit of 15 seats for overbooking.**



This can be explained because the airlines has greater control over its seats and therefore may choose not to overbook to the limit. This can be seen in the simulation below:

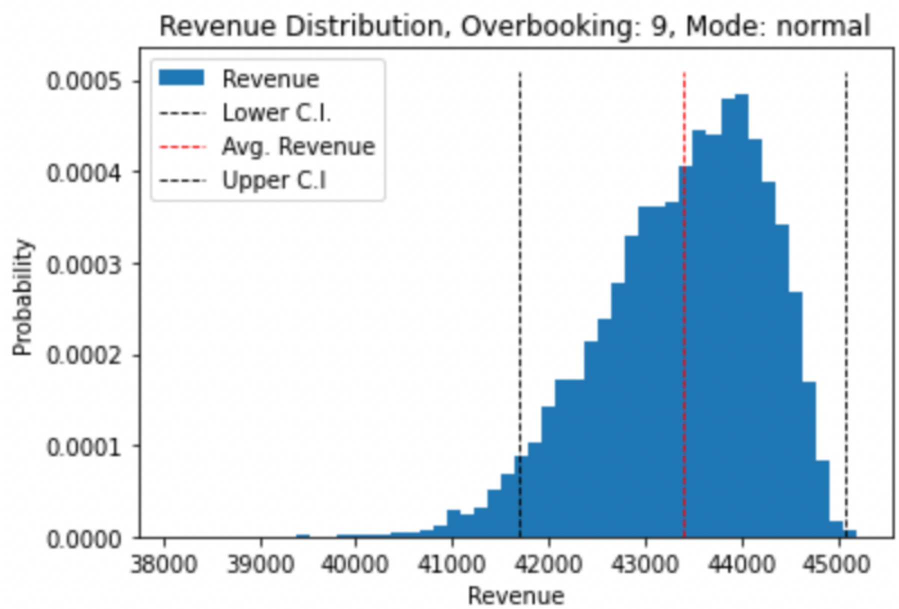


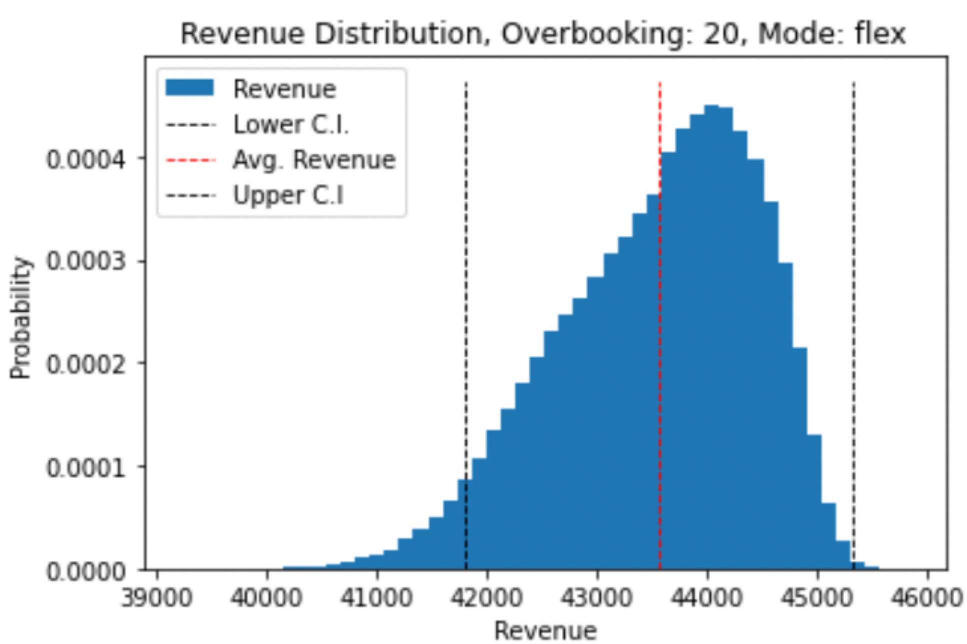
Therefore, if the airlines mandates upto 20 seats of overbooking, the second strategy will earn maximum revenue at an overbooking limit of 20 seats. The following are the statistics of the revenues earned by the two strategies:

- Strategy - Original (Overbook: 9): Optimal Revenue: $42496.11
- Strategy - Flex (Overbook: 20): Optimal Revenue: $42502.67

Simulation Forward

So far, we have been solving the problem backwards. Now, we have used the optimal booking policy to simulate 50,000 bookings and use some key data points and plots to make policy decisions.

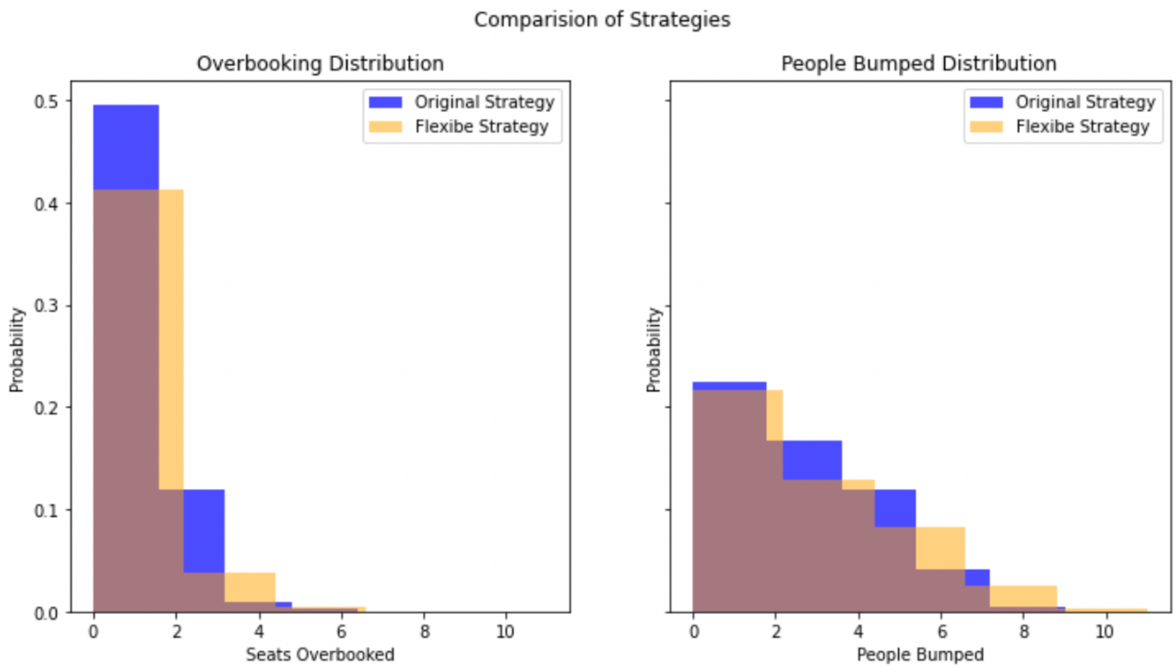




From the above analysis we could see that both strategies performed similar to each other, the differences that could be seen are:

KPI	Policy1 - Normal Mode	Policy2 - Flex mode
Coach overbooked	99.98%	100.00%
Passengers Kicked	72.68%	71.67%
Avg Overbooking Cost	$1,040.47	$1,237.79
Avg. Revenue	$43,396.96	$43,555.17
Std. Revenue	$855.25	$899.06
Avg Ticket Sold - Coach	108.44	109.07
Avg Ticket Sold - FC	19.55	19.49

The above graph we could deduce that policy 2 has a larger mean i.e profit however has a larger standard deviation which implies more risk



The above probability distribution we could conclude that using policy 1 would lead to a lower probability of kicking people off the plane and also lead to similar or less probability of overbooking a flight. In spite of this, Policy 2 leads to a larger revenue because Policy 2 ends up booking more seats and therefore covers the losses incurred by overbooking and asking people to leave.

Conclusion

Case 1: Overbooking limit > 9

From the above data points we could conclude that we would end up making more profit from Policy 2 - Flex mode. However, this model is more prone to risk as it has a higher standard deviation when compared to Policy 1.

Case 2: Overbooking limit < 9

For this use case we found Policy 1 to be more suitable, as it leads to greater profit. Additionally, it had a lower “Passengers Kicked” percentage which would also lead to a lesser impact on the companies reputation