The Airfare Network Problem

Keith Gladstone (keithag@princeton.edu)
Professor Mengdi Wang
ORF 360: Decision Modeling in Business Analytics

Department of Operations Research and Financial Engineering
Princeton University

May 6, 2016

Abstract

The objective of the airfare network problem is to maximize expected revenue when choosing to accept a request for a given itinerary within the airline network. It is an extension of the single-product capacity-control problem, which accounts for given prices and request arrival rates when making decisions regarding capacity. Price and frequency estimates will be based on real data, and several modeling assumptions will be made to allow the problem to be computed within reasonable time.

Contents

1 The Practical Problem

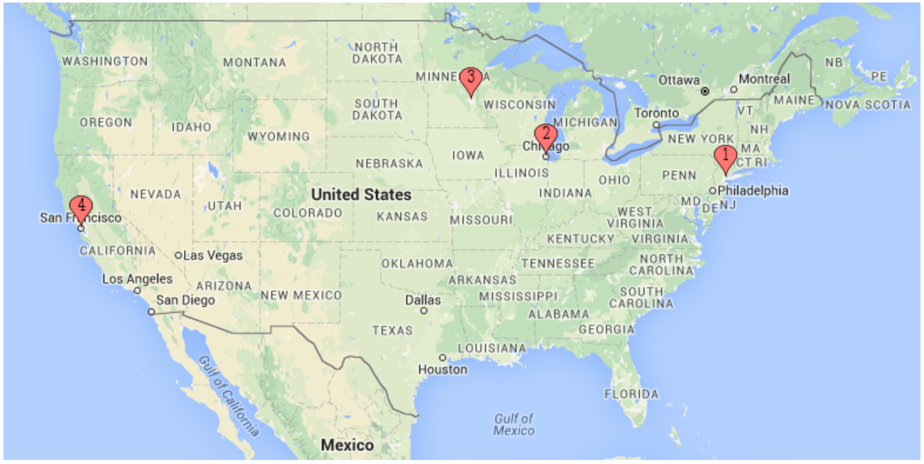
The goal of this project is to find an optimal decision-making strategy given capacity constraints, the consumer arrival process, flight routes, and ticket prices. Specifically, at a given time and capacity, which booking requests would we accept or reject? As opposed to a single-product capacity control problem, this problem takes into account the **network effects** of the flight network. Complete itineraries, either direct or with layovers, are referred to as **origin-destination (O-D) pairs**. Since not all flights are direct, we refer to a trip between any two stops within a flight-route as a **flight-leg**. In other words, if a passenger is traveling from City A to City C with a layover in City B, the flight-legs would be A-B and B-C, where the complete itinerary of the flight is A-B-C.

2 Mathematical Model

In this model, we refer to a flight-leg as a **resource** and an itinerary as a **product**. There are m resources that are used to create n products that have given sale prices. As an example, we could observe data from the airports in the following American cities, shown in **Figure 1**:

1. Newark, NJ (EWR)
2. Chicago, IL (ORD)
3. Minneapolis, MN (MSP)
4. San Francisco, CA (SFO)

Figure 1: Map of hub cities (Google)



These cities represent densely populated areas in different geographical regions of the United States, namely the Northeast, the Midwest, and the West Coast.

Using these airports as hubs, a variety of flight-legs can be constructed. We will limit the set of flight-legs to the following:

- EWR-ORD
- EWR-MSP
- ORD-MSP
- MSP-SFO

As there are 4 flight-legs in this example, we have $m = 4$. These flight-legs can be arranged into a specific set of products, limited to the following set of $n = 3$:

- EWR-ORD-MSP
- EWR-MSP-SFO
- ORD-MSP-SFO

2.1 Product Incidence Matrix

Call matrix $A = \{a_{ij}\}$ the **product incidence matrix**, size $m \times n$. In the case of our above example, this matrix would be size 4×3 (see **Table 1**). A_j is the j^{th} column of A, which states the resources of product j. For each product j there is a price p_j. The matrix A is populated as follows:

$$a_{ij} = \begin{cases} 1 & \text{product } j \text{ uses resource } i \\ 0 & \text{otherwise} \end{cases}$$

Table 1: Product Incidence Matrix

	EWR-ORD-MSP	**EWR-MSP-SFO**	**ORD-MSP-SFO**
EWR-ORD	1	0	0
EWR-MSP	0	1	0
ORD-MSP	1	0	1
MSP-SFO	0	1	1

2.2 States

In this problem, resources are limited since the number of seats on a flight-leg are limited to a certain capacity. The state of the network is denoted with $\mathbf{x}$= $(x_1, ..., x_m)$, which represents the remaining capacity for each resource $i \in \{1, ..., m\}$.

2.3 Transitions

Given the state vector **x**, if product j is sold, then the following transition occurs on the state vector: $\mathbf{x} \to \mathbf{x} - A_j$. This simplifies to:

$$x_i = \begin{cases} x_i - 1 & \text{product } j \text{ uses resource } i \\ x_i & \text{otherwise} \end{cases}$$

The selling horizon is divided into T small periods, and in each period, there is a probability q_j that a request for product j is made. The selling periods are small enough such that no more than one request is made during a given period. Hence, we require the following condition for **q**:

$$\sum_{j=1}^{n} q_j \leq 1$$

2.4 Decisions

We have now established the model of the problem such that we can address the important question: **at a given time and state, which product requests would we accept upon their arrival**?

Below is the decision vector:

$$\mathbf{u}(t, \mathbf{x}) = \{u_j(t, \mathbf{x})\}_{j=1}^{n}$$

where

$$\mathbf{u}(t, \mathbf{x}) = \{0, 1\}^n$$

The decision vector has the following feasibility constraint:

$$A_j u_j \leq \mathbf{x}, \ \forall j$$

2.5 Dynamic Programming Problem Formation

Given the product incidence matrix, the states, the transitions, and the decision vectors, we can now form a **Dynamic Programming Problem (DP)** that will solve for the maximum expected revenue at a given state **x**. Let $V_t(\mathbf{x})$ be the maximum expected revenue function. The intuition behind the following Bellman equation is that the maximum expected return to be achieved in the remainder of the time horizon, given the current state and time, is the maximum expected return to be achieved given a certain product request arrival multiplied by the probability that the given product request actually arrives. Its structure is similar to that of an expectation formula. It appears as follows:

$$V_t(\mathbf{x}) = \max_{\mathrm{u(t,\ \mathbf{x})\in U(\mathbf{x})}} \left[\sum_{j=1}^{n} q_j (p_j u_j(t, \mathbf{x}) + V_{t+1}(\mathbf{x} - A_j u_j(t, \mathbf{x}))) + (1 - \sum_{j=1}^{n} q_j) V_{t+1}(\mathbf{x}) \right]$$

Boundary condition:

$$V_{T+1}(\mathbf{x}) = 0$$

Feasible region:

$$U(\mathbf{x}) = \{\mathbf{u} \in \{0,1\}^n : A_j u_j \leq \mathbf{x}, \; \forall j\}$$

The Bellman equation can be rearranged to simplify computation:

$$V_t(\mathbf{x}) = V_{t+1}(\mathbf{x}) + \max_{\text{u(t, } \mathbf{x}\text{)} \in \text{U(}\mathbf{x}\text{)}} \left[\sum_{j=1}^{n} q_j u_j(t,\mathbf{x})(p_j + V_{t+1}(\mathbf{x} - A_j) - V_{t+1}(\mathbf{x})) \right]$$

2.6 Optimal Pricing Strategy

The **opportunity cost** of a decision to use resources A_j is:

$$V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - A_j)$$

As long as there is enough capacity on the flight-legs to be able to accept the request for the trip, we will choose to book the trip with flight-legs A_j if the selling price for trip j, p_j is greater than the opportunity cost. Formally, if a request for product j is made, then the optimal decision is:

$$u_j(t,\mathbf{x}) = \begin{cases} 1 & \text{if } p_j \geq V_{t+1}(\mathbf{x}) - V_{t+1}(\mathbf{x} - A_j) \text{ and } A_j u_j \leq \mathbf{x} \\ 0 & \text{otherwise} \end{cases}$$

3 Justification of the Model

Airfare networks are complex problems, and as such, some models are better than others in capturing their mathematical attributes. Some models deal with simple consumer arrival and others take advantage of behavioral economics such as the reference-price effect. Optimal strategies consist of dynamic pricing, fare-class control, and price skimming.

The model described in the previous section of this paper is useful in capturing the fact that flight-legs can be viewed as building blocks that form complete itineraries. Indeed, this is the network extension of the single-product capacity control problem. Not all customers perform due diligence when booking trips, simply taking the first deal they see instead of perhaps the best one. By processing historical ticket data, we need to come up with values for the following *fixed* parameters:

- p_j price of product j for all $j \in \{1, ..., n\}$
- q_j probability that a request for product j is made for all $j \in \{1, ..., n\}$
- x_i the initial number of seats on flight-leg $i \in \{1, ..., m\}$
- T the number of selling periods

We will also have to formulate the product-incidence matrix, A.

4 Computation of Optimal Pricing Strategy

In order to actually compute the optimal pricing strategy, we need to use the initial data to populate the V matrix using "value iteration", which will then be used to make the decisions. This code was completed in R and can be viewed at this link: `https://github.com/kgladstone/airfare-network`.

Note that for ease of computation, we are assuming that all planes in the network (flight-legs) have the same initial capacity k. As I will justify in the next section, these capacities will be quite small so that the code takes only a few minutes to run. In other words,

$$x_{i,t=1} = k, \forall i \in \{1, ..., m\}$$

5 Parameter Variation

In this section we will see if varying parameters results in significant changes in the optimal pricing strategy.

5.1 Curse of Dimensionality

Coined by Richard E. Bellman himself, the **curse of dimensionality** is the phenomenon that describes how the computation of certain algorithms can take a very large amount of time if the dimensions of the variables exceed a certain threshold. The presence of this phenomenon is precisely why I chose to limit the number of airports to 4, the number of flight-legs to 4 and the number of itineraries to 3. In reality, airfare revenue managers have to deal with dozens if not hundreds of airports, and thousands of itineraries, to the point where executing the value iteration of the Bellman equation becomes prohibitive. However, tweaking certain modeling variables such as the number of periods, initial capacities of flight-legs, as well as adjusting the real-data estimates of price and request frequency, can yield different V matrices for the Bellman equation and change the way the airfare revenue manager would make decisions.

5.2 Estimating with Real Prices

Priceline.com lists information for hundreds of upcoming flights. As such, it is an appropriate website from which to pull itinerary price data. In the example above, we illustrated a network that involves four cities:

- Newark, NJ (EWR)
- Chicago, IL (ORD)
- Minneapolis, MN (MSP)
- San Francisco, CA (SFO)

Table 2: Ticket prices for various itineraries (priceline.com)

Trip	Ticket Price ($)	Airline
EWR-ORD-MSP	725.60	United
EWR-MSP-SFO	404.60	Delta
ORD-MSP-SFO	147.60	Delta

From these four cities we can construct three trips that pass through the network, sharing some flight-legs. See **Table 2** for Priceline data on these trips[1]. This gives us the following price vector:

$$\mathbf{p} = \begin{bmatrix} 725.60 & 404.60 & 147.60 \end{bmatrix}$$

5.3 Hidden City Phenomenon

We will attempt to discover the **hidden-city phenomenon** in our data. The phenomenon is illustrated as follows. Let the hub network consist of City A, City B, and City C. Let the passenger desire to fly from City A → City C. Suppose there are two itineraries:

- Trip I: City A → City C, $p_I = \$h$
- Trip II: City A → City C → City B, $p_{II} = \$g < \$h$

If the passenger booked Trip II and traveled from City A, making the layover in City C, but terminated his trip at the layover, then the passenger essentially completed an itinerary City A → City C, despite using a ticket for City A → City C → City B. And in doing so, he saved $\$(h - g)$. For instance, if City A is Newark (EWR), City B is Minneapolis (MSP) and City C is Chicago (ORD), then we may see that a flight from EWR to MSP with a layover in ORD may be cheaper than a flight from EWR to ORD direct. This would be the hidden-city phenomenon in practice.

By referencing Table 1, observe that United Airlines flight from Newark to Minneapolis (EWR-ORD-MSP) is more expensive than the Delta Air Lines flight from Newark to San Francisco (EWR-MSP-SFO). Since the Delta flight makes a layover in Minneapolis, an astute customer who wants to travel from Newark to Minneapolis could book the Delta flight for $404.60 and simply remain in Minneapolis instead of boarding the second leg of the trip to San Francisco. In booking the Delta flight instead of the United flight whose marked destination is Minneapolis at $725.60, the customer saved over $300 and still made it to his desired city, never having to set foot in San Francisco at all.

5.4 Methodology for Request Frequencies

We now need to estimate values for $\mathbf{q}$, the probabilities that in a given period a request for product j is made for all products $j \in \{1, ..., n\}$. Given the condition

[1] Flight data accessed on May 2, 2016 for trips occurring on May 9, 2016 (priceline.com)

Table 3: U.S. Metropolitan Statistical Area Populations (2010 Census)

Airport	Metro Area	Population
EWR	New York-Newark-Jersey City, NY-NJ-PA	19,567,410
ORD	Chicago-Naperville-Elgin, IL-IN-WI	9,461,105
MSP	Minneapolis-St. Paul-Bloomington, MN-WI	3,348,859
SFO	San Francisco-Oakland-Hayward, CA	4,335,391

$\sum_{j=1}^{n} q_j \leq 1$, we can make the following assumption about $\mathbf{q}$: the probability that a request for a certain flight be made is correlated with the population of the cities involved. As such, we will compute the following metric to be referred to as the **raw O-D population score** for an origin city with population w_j and a destination city with population $w_{j'}$:

$$\text{score}(j, j') = w_j^2 + w_{j'}^2$$

We have the following data on United States Metropolitan Statistical Area (MSA) populations[2], shown in **Table 3**. With these data we can compute the raw O-D population score for the itineraries:

- EWR-ORD-MSP score = $3.94098390709981 \times 10^{14}$
- EWR-MSP-SFO score = $4.01679149230981 \times 10^{14}$
- ORD-MSP-SFO score = $1.08308122943906 \times 10^{14}$

The sum of these scores is $9.04085662884868 \times 10^{14}$, hence when divided by the sum, the **normalized O-D scores** become:

- EWR-ORD-MSP normalized score = 0.4359
- EWR-MSP-SFO normalized score = 0.4443
- ORD-MSP-SFO normalized score = 0.1198

Suppose that in a given period the probability that no request is made is equal to *five percent*. We can scale each of the normalized O-D scores by 0.95 and then assign the resulting values to the $\mathbf{q}$ vector:

$$\mathbf{q} = \begin{bmatrix} 0.4141 & 0.4221 & 0.1138 \end{bmatrix}$$

5.5 Results

We will start by holding all of the following parameters constant.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

[2] Metropolitan Staistical Area population data from 2010 U.S. Census (census.gov)

$$\mathbf{p} = \begin{bmatrix} 725.60 & 404.60 & 147.60 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} 0.4141 & 0.4221 & 0.1138 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 7 & 7 & 7 & 7 \end{bmatrix}$$

$$T = 30$$

The result of running the R-code yields a value of

$$V_{t=1}(\mathbf{x} = (7, 7, 7, 7)) = \$7,894.24$$

6 Analysis

Now, let's run a sensitivity analysis by adjusting some parameters, holding all else equal to their values in the original run above. See **Table 4** for side-by-side sensitivity analysis results. Indeed, in the table, the results for **q** and **p** are those that minimize and maximize the value function from a large sample of testing data.

Table 4: Sensitivity Analysis of $V_{t=1}(\mathbf{x})$

Parameter	**Parameter (1)**	$V_{t=1}(\mathbf{x})$	**Parameter (2)**	$V_{t=1}(\mathbf{x}))$
T	20	7,514.44	50	7,911.39
x	(5, 5, 5, 5)	5,649.84	(8, 8, 8, 8)	8,991.58
q	(0.914, 0.005, 0.081)	5,138.15	(0.504, 0.444, 0.052)	7,908.17
p	(132.00, 185.09, 960.70)	4,220.08	(683.55, 593.60, 0.66)	8,921.08

Plots of the sensitivity analysis appear in the **Appendix**. General trends are summarized below:

- **Period Count Variation:** $V_{t=1}(\mathbf{x})$ increases in T, at a decreasing rate. When holding all other parameters equal to the original run, for $T > 40$, the marginal increase of V is small. Hence, it becomes less necessary to increase the number of periods beyond 40, for a problem whose other parameters are around the same order of magnitude as the ones in this problem. The growth of the value function is depicted in **Figure 2** of the **Appendix**.

- **Capacity Variation**: $V_{t=1}(\mathbf{x})$ increases in $\mathbf{x}$. When a regression was run on five linear data points, the result indicated that when each flight-leg gains an additional seat of capacity, the expected revenue to be gained from the optimal decision strategy increases by about $1,100. This is due to the fact that more tickets are allowed to be sold, but also shows that the solution to the problem would grow in a predictably linear manner if we scaled the capacity of each flight-leg. See **Table 5** for the output of the regression model, which summarizes the linear relationship. The growth of the value function is depicted in **Figure 3** of the **Appendix**.

- **Request Rate Variation:** As the vector $\mathbf{q}$ is multidimensional, it was necessary to collect a large sample of randomized inputs and the corresponding value function outputs, in order to see the effect of request rate variation. Holding all other parameters equal to the original run, in a sample of size 389, $V_{t=1}(\mathbf{x})$ is maximized when

  $$\mathbf{q} = \begin{bmatrix} 0.5043514 & 0.4436166 & 0.05203192 \end{bmatrix}$$

  The fact that $q_1^{max} > q_2^{max} > q_3^{max}$, is noteworthy since $p_1 > p_2 > p_3$. Indeed, $V_{t=1}(\mathbf{x})$ increases as $\mathbf{q}$ reflects proportions of $\mathbf{p}$. A three-dimensional scatterplot that depicts the value function with varied $\mathbf{q}$ inputs (which always sum to 0.95) can be found in **Figure 4** of the **Appendix**.

- **Itinerary Price Variation:** Like $\mathbf{q}$, the vector $\mathbf{p}$ is also multidimensional, and as such, we could conduct a similar analysis. Holding all other parameters equal to the original run, in a sample of size 388, $V_{t=1}(\mathbf{x})$ is maximized when

  $$\mathbf{p} = \begin{bmatrix} 683.5471316 & 593.5957835 & 0.657084869 \end{bmatrix}$$

  The value function increases as p_1 and p_2 increase and as p_3 decreases. A three-dimensional scatterplot that depicts the value function with varied $\mathbf{q}$ inputs (which always sum[3] to \$1,277.8) can be found in **Figure 5** of the **Appendix**.

It is important to note that this analysis is limited in the fact that the capacities are unreasonably low for filling seats of full-sized airplanes, and that network effects are limited by the problem's reduced size compared to a larger, connected hub network. However, it is useful to develop a model like this to understand the problem, and when perhaps given enough computing power, be able to solve for expected revenue maximization over a larger network with larger seating capacities.

Table 5: Linear relationship between $V_{t=1}(\mathbf{x})$ and initial seat capacity $\mathbf{x}$

	Dependent variable:
	$V_{t=1}(\mathbf{x})$
Initial Capacity $\mathbf{x}$	1,100.867***
Constant	165.619*
Note:	*p<0.1; **p<0.05; ***p<0.01

[3] This is the sum of the elements in the original vector $\mathbf{p}$: 725.60, 404.60, and 147.60

7 Conclusion

This paper explored the Airfare Network Problem, addressing the question of **what is the maximum expected revenue to be attained given a network of flight-legs, itineraries, ticket prices, consumer arrival process, and number of selling periods?** The formation of the problem makes use of the Bellman equation, that when solved illuminates the optimal decisions to be made when processing a booking request: **whether or not to accept that request**.

Another phenomenon that arose during the exploration was the **hidden-city phenomenon**. To exploit this, consumers can travel to desired cities at prices that are not conspicuously advertised by finding cheaper flights with layovers at their personally desired destinations.

By developing a script in R and using real data from Priceline.com, we were able to find the maximum expected revenue of our model given the parameters, and observed that in higher dimensions, the computation time grows rapidly. As such, we acknowledged that the model can be extended to real-life airfare revenue management with larger data, but such computation is beyond the scope of this paper.

Instead we kept the modeling parameters (O-D pairs, capacity and periods) of our problem small enough to be able to run large samples ($n \approx 400$) of sensitivity analysis and observe the effect of adjusting certain multidimensional parameters (price and request rate). Understanding the relationships between parameters and the value function is particularly useful when applying this method to larger flight networks and conducting further analysis.

8 Appendix

Figure 2: Value function increases in number of periods, at a decreasing rate

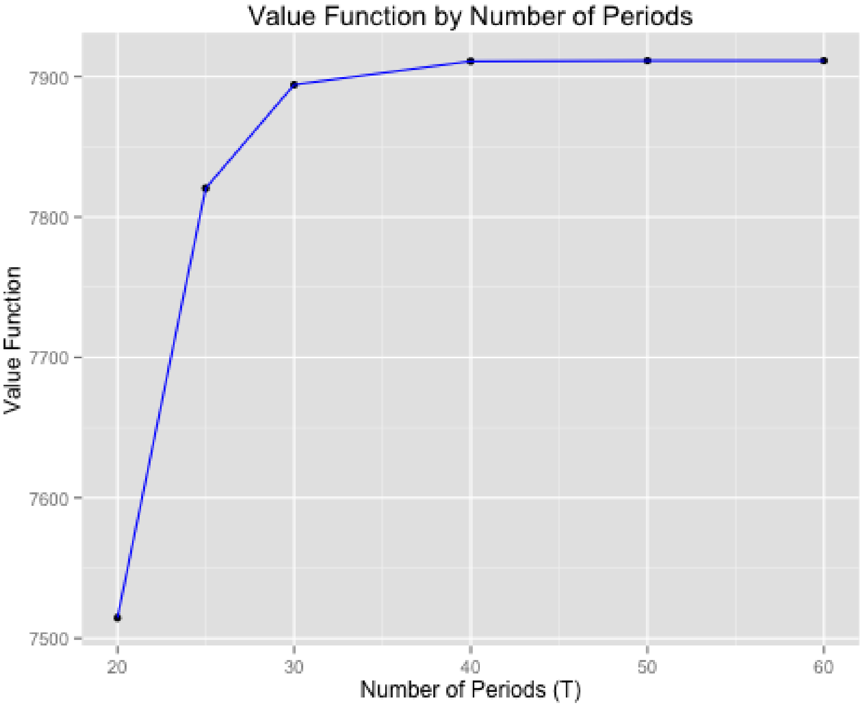


Figure 3: Value function increases in capacity, at a constant rate

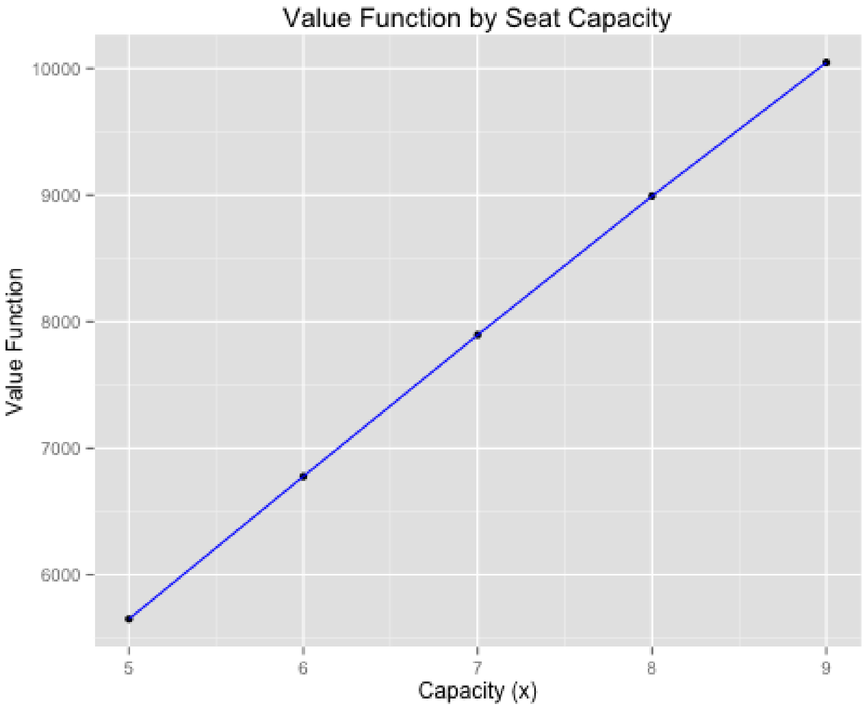


Figure 4: Value function increases as $\mathbf{q}$ reflects proportions of $\mathbf{p}$. The value function is maximized when $\mathbf{q}$ =[0.504 0.444 0.052]

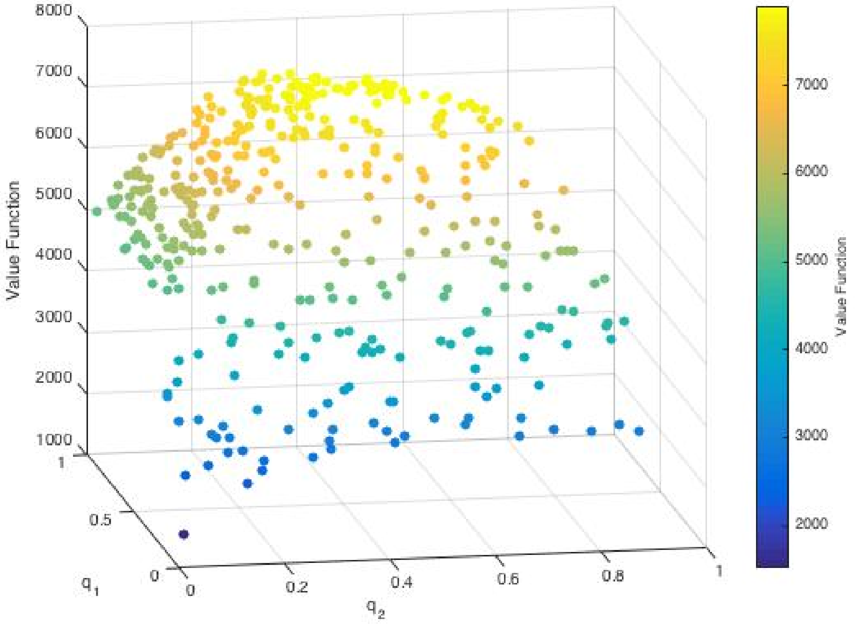


Figure 5: Value function increases as p_1 and p_2 increase and as p_3 decreases. In the sample, the Value function is maximized when $\mathbf{q}$ =[683.55 593.60 0.66]

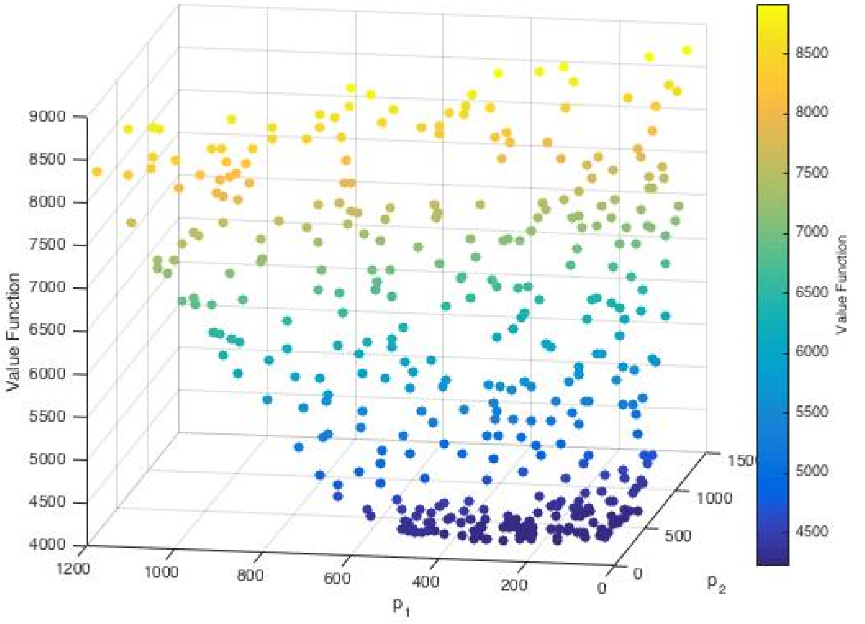
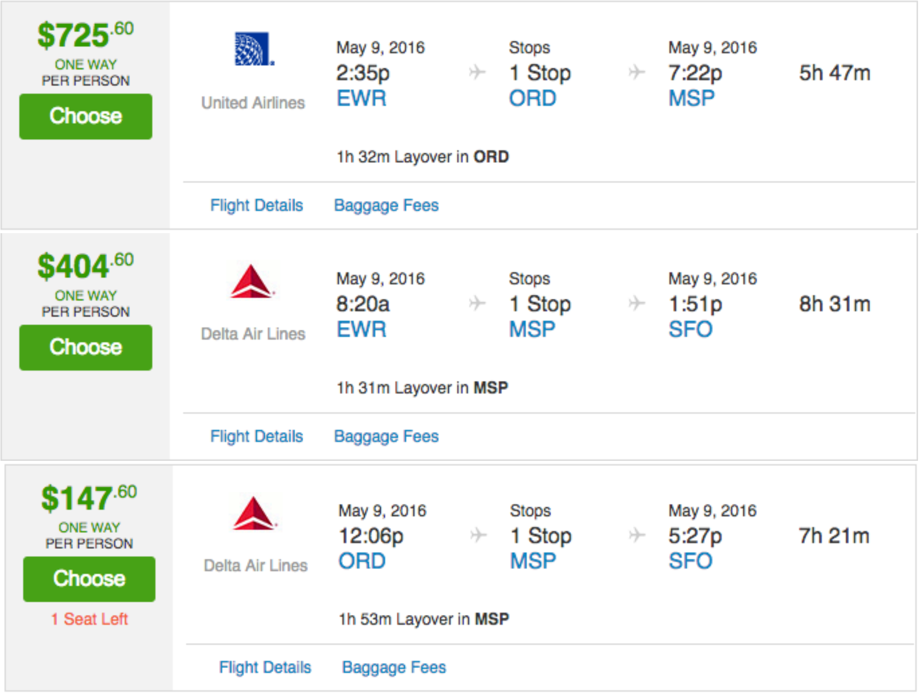


Figure 6: Priceline.com ticket prices, accessed May 2, 2016



$725.60
ONE WAY
PER PERSON

United Airlines

May 9, 2016
2:35p
EWR

Stops
1 Stop
ORD

May 9, 2016
7:22p
MSP

5h 47m

1h 32m Layover in **ORD**

$404.60
ONE WAY
PER PERSON

Delta Air Lines

May 9, 2016
8:20a
EWR

Stops
1 Stop
MSP

May 9, 2016
1:51p
SFO

8h 31m

1h 31m Layover in **MSP**

$147.60
ONE WAY
PER PERSON

1 Seat Left

Delta Air Lines

May 9, 2016
12:06p
ORD

Stops
1 Stop
MSP

May 9, 2016
5:27p
SFO

7h 21m

1h 53m Layover in **MSP**