Amazon Logistics
Last Mile Cost Optimization

Abbey Chaver and Benjamin LeRoy

May 15, 2015

Abbey Chaver is a second-year Operations Research and Management Science student. She is interested in data science, user experience research, and scalable business models. She also dabbles in design. To build this model, she drew on knowledge from Economics, Statistics, and Industrial Engineering courses.

Ben LeRoy is a senior Applied Math and Statistics double major, graduating in December of this year. He's interested in data science, statistical modeling, machine-learning, and all cool statistic stuff. His focuses outside Math and Stats including Economics and IEOR.

1 Challenge

Last-mile transportation is one of Amazon's biggest marginal costs. Optimizing this process is challenging due to daily variance in demand. Given a demand distribution Normal ($\mu = 12,000$, $\sigma = 2,400$) on weekdays and Normal ($\mu = 8,000$, $\sigma = 1,600$) on weekends, how much should Amazon pay:

1. Guaranteed drivers, who must be paid whether or not there are packages for them to deliver.
2. Option drivers, who are paid a reservation price to be available but need not be paid a full wage if there are not packages for them.
3. Spot drivers, who are only paid if they are needed that day.

2 Approach

One of the first questions we considered was whether or not Amazon should pay a premium wage, or efficiency wage, above the market rate. From looking at labor data, we found that truckers have a very low rate of voluntary absenteeism - most absenteeism is attributed to illness or injury. Therefore, offering a higher wage is unlikely to be cost effective. We recommend paying the market wage.

We made the following assumptions about the contracting process: Each type of driver is under a contract that once taken, will be filled. We assume that drivers will be willing to take a contract at the market wage for a guaranteed daily payment. We assume that each driver takes a uniform packages to deliver for each day. We also assume that option workers and spot workers, if not hired by Amazon, are able to enter the general spot market, where they may be able to find work at a wage approximately equivalent to the spot wage that Amazon offers. While there is reasonably high demand for delivery drivers, we assume that the probability a driver would find work on the open market is less than 1. Otherwise, there would be no risk in being a spot driver and all drivers would prefer to be spot drivers.

Essentially, Amazon must set wages for each type of driver such that a driver will be willing to take any of the three contracts, allowing Amazon to hire the optimal number of each.

We decided to model the wage as a base rate - the market wage - for each package, plus an additional premium to compensate drivers for the risk of not being hired by Amazon for the day. To solve the problem numerically, we set certain parameters at figures we considered reasonable. However, these figures can and should be adjusted to reflect available data.

3 Model

When building our model, we realized that optimizing total cost required solving for the number of each type of driver as well as the wage we paid them. We defined these terms for our calculations:

3.1 Terms

Y: Daily demand for packages to be delivered (distributed Normal(μ, σ)).

$\mathbf{x_1}$**:** Number of packages allotted to guaranteed drivers

$\mathbf{x_2}$**:** Number of packages allotted to option drivers

$\mathbf{x_3}$**:** Number of packages allotted to spot drivers

$\mathbf{c_1}$**:** Market rate for delivering a single package, paid to guaranteed drivers no matter what.

$\mathbf{c_2}$**:** Risk premium for delivering a single package, paid to option drivers no matter what. If they are hired that day, they are payed c_2 in addition to c_1.

$\mathbf{c_3}$**:** Risk premium for delivering a single package, paid to spot drivers only if they are hired that day, in addition to c_1.

s: Reflects the probability that a spot driver will find work on the open market ($\in [0, 1]$).

3.2 Relating Terms

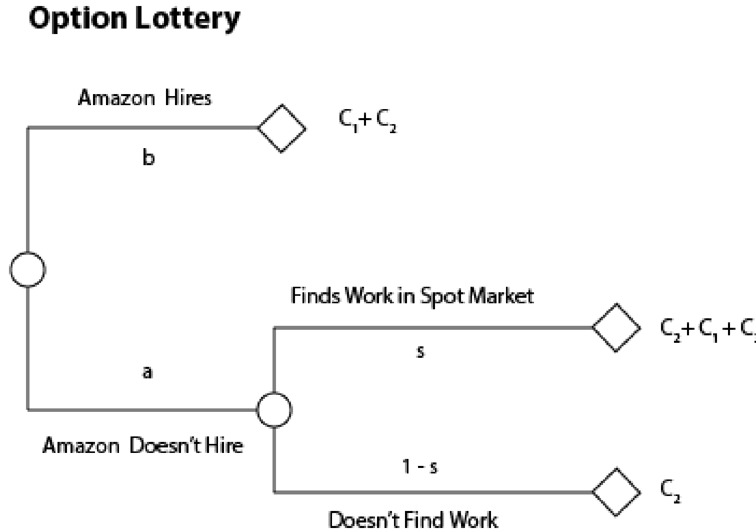
We consider the wages c_2 and c_3 risk premiums, so we need to quantify how much risk is perceived by a driver in each situation. We also consider c_1 to be a market rate that we cannot set independently, so essentially a constant. We also assume that if an option or spot driver is not hired by Amazon, they can seek work with other companies. We assume that the market will perform the same calculation as us, resulting in a spot market wage of (c_1 + c_3).

We used a risk-averse utility function to capture a driver's desire for a guaranteed wage. By defining c_2 and c_3 as functions of x_1, x_2, and x_3, we can minimize a total cost function with respect to fewer variables. We use the relative risk aversion function:

$$u(c) = (1 - e^{-\gamma c})$$

3.3 Option Drivers

First, we want to choose c_2 such that a driver is indifferent between c_1 and the option lottery displayed below:



The above diagram leads us to equalize the following equation:

$$u(c_1) = [a \cdot [s \cdot u(c_2 + c_1 + c_3) + (1 - s) \cdot u(c_1)] + b \cdot u(c_1 + c_2)]$$

Later we will want to express c_2 in terms of x_1,x_2, and c_1; the math is worked out in Math Appendix .

The probability that any certain option driver is hired by Amazon is dependent on either of these two scenarios occur: $Y \geq (x_1 + x_2)$, or $x_1 \leq Y \leq (x_1 + x_2)$. A particular option driver is selected out of all the x_2 option drivers (we assume each option driver is equally likely to be chosen) if $x_1 \leq Y \leq (x_1 + x_2)$.

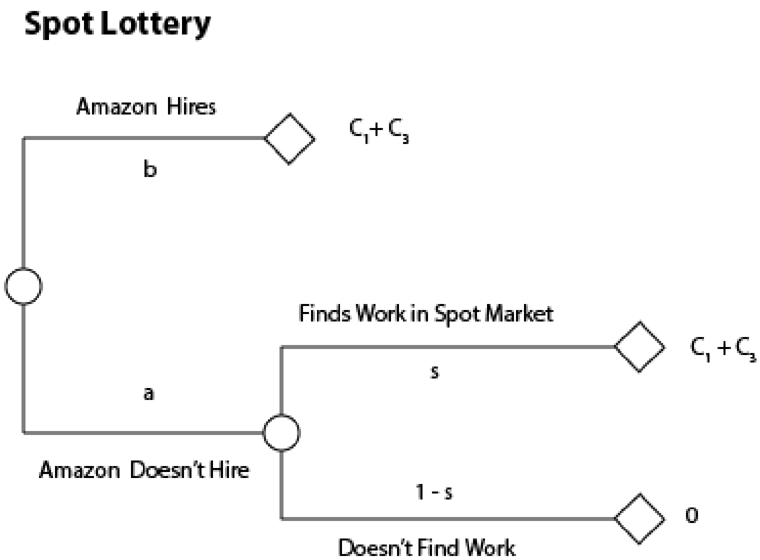
So the probability of getting hired, b, is:

$$b = \int_{x_1}^{x_1+x_2} \frac{y - x_1}{x_2} \mathbb{P}(y) dy + \mathbb{P}(Y > x_1 + x_2)$$

$$a = 1 - b$$

3.4 Spot Drivers

Similarly, we want drivers to be indifferent between c_1 and the spot lottery:



The above diagram leads us to equalize the following eqation:

$$u(c_1) = [a \cdot [s \cdot u(c_1 + c_3) + (1 - s) \cdot u(0)] + b \cdot u(c_1 + c_3)]$$

You can see how we express c_3 as a function of x_1,x_2 and c_1 in Math Appendix.

The optimal value of x_3 is essentially infinite: we assume that we will be able to find enough spot workers to cover all of our demand if we offer a fair wage. However, having no upper limit to the number of spot drivers available makes it difficult to calculate the probability that any one spot driver will be hired by Amazon, as we were able to do for option drivers. Therefore, we assumed that it's reasonable to solve the problem within three standard deviations of the distribution: we would almost never expect to hire beyond that. So we define:

$$x_3 = (\mu + 3\sigma) - (x_1 + x_2)$$

Then we can define a spot worker getting hired as the scenario where

$$Y \geq (x_1 + x_2)$$

and that this driver is selected out of the x_3 option drivers. The probability b' that this occurs is:

$$b' = \int_{x_1+x_2}^{x_1+x_2+x_3} \frac{y-(x_1+x_2)}{x_3} \mathbb{P}(y)dy$$

3.5 Objective Function

We now have every term defined as a function of x_1, x_2, and c_1, where c_1 is a constant. We can therefore set up a Total Expected Cost function (TEC) to minimize in only two variables.

Note: Because we considered absenteeism to be a constant proportion for each type of driver, we simply removed it for clarity's sake.

$$\min_{x_1,x_2} \text{TEC}_Y =$$

(Fixed costs for guarantee and option workers:)

$$c_1 \cdot x_1 + c_2(c_1, x_1, x_2) \cdot x_2$$

(Variable costs for option workers:)

$$+c_1 \cdot \Big(\big(\int_{x_1}^{x_1+x_2} (y - x_1)\mathbb{P}(Y = y)dy \big) + (x_2) \cdot (1 - \Phi_y(x_1 + x_2)) \Big)$$

(Variable costs for splot workers:)

$$+[c_1 + c_3(c_1, x_1, x_2)] \cdot \int_{x_1+x_2}^{\infty} (y - x_1 - x_2)\mathbb{P}(Y = y)dy$$

Where Φ_y is the CDF of the Normal distribution, and y determines the mean and standard deviation.

4 Example Solution

Setting Parameters To test whether our model returned reasonable figures, we plugged in some values.

c_1 **= 1**
For c_1, we needed to determine a base market rate for per-package delivery. Assuming that a driver can deliver on average 200 packages in a 10 hour day and expects \$20/hour, we set the wage for each package at a convenient \$1.

γ **= 0.5**
We found that $\gamma = 0.5$ returned reasonable figures for our units, so we used it for simplicity's sake. Determining γ empirically will be necessary for an accurate model:

Determining a person's risk aversion is typically done by surveying them about how much they would pay to buy a certain lottery. Typically, a person will pay less than the expected value of the lottery, because they are averse to risk. Measuring to what to degree can be done with a series of questions that can then be fit to a concave utility function, such as the one we used. Other utility functions can provide very different results. Our function cannot return utility above 1, whereas, concave functions like $\sqrt{x}$ and $\log(x)$ are unbounded.

s = 0.5
For the probability that a driver can find work in the spot market, we estimated a probability of 0.5 to reflect risk.

We also decided not to change the market rate c_1 for the weekend, because truckers typically work overtime at the same rate.

Results: In order to solve the model, we built it in R (See Code Appendix for R code). While technically the number of packages only come in integers, the numbers are large enough that it is appropriate to solve this problem continuously and round to the nearest discrete value.

	Week	Weekend
$\mathbb{E}(Y)$:	12,000	8,000
$\sigma(Y)$:	2,400	1,600
Expected Costs:	\$13,127	\$8,759
c_1:	1	1
c_2:	\$0.0942	\$0.0942
c_3:	\$0.7339	\$0.7339
x_1:	6,232	4,155
x_2:	6,204	4,136
s:	.5	.5

5 Summary

This challenge demands a fairly complicated model of labor supply for the delivery market, a quickly changing industry. While most of our parameters were estimates that should be validated by actual data in order to return accurate solutions, our model describes the cost structure of Amazon's last-mile transportation defensibly, based on strong logic and reasonable assumptions.

6 Math Appendix: Mathematics of c_2 and c_3 worked out

Part 1: Math for equalizing option driver's options (solve for c_2)

$$u(c_1) = [a \cdot (1-s) \cdot u(c_2) + a \cdot (s) \cdot u(c_2 + c_1 + c_3) + b \cdot u(c_1 + c_2)]$$

$$1 - e^{-c_1} = [a \cdot (1-s) \cdot (1 - e^{-c_2}) + a \cdot (s) \cdot (1 - e^{-c_2 - c_1 - c_3}) + b \cdot (1 - e^{-c_1 - c_2})]$$

$$1 - e^{-c_1} = a \cdot (1-s) - a \cdot (1-s) e^{-c_2} + a \cdot (s) - a \cdot (s) \cdot e^{-c_2 - c_1 - c_3} + b - b \cdot e^{-c_1 - c_2}$$

$$1 - e^{-c_1} = \Big[a \cdot (1-s) + a \cdot (s) + b\Big] - a \cdot (1-s) e^{-c_2} - a \cdot (s) \cdot e^{-c_2 - c_1 - c_3} - b \cdot e^{-c_1 - c_2}$$

RHS: the first 3 terms sum to 1, so we can reduce both sides by 1:

$$-e^{-c_1} = -a \cdot (1-s) e^{-c_2} - a \cdot (s) \cdot e^{-c_2 - c_1 - c_3} - b \cdot e^{-c_1 - c_2}$$

$$e^{-c_1} = a \cdot (1-s) e^{-c_2} + a \cdot (s) \cdot e^{-c_2 - c_1 - c_3} + b \cdot e^{-c_1 - c_2}$$

$$e^{-c_1} = \Big[a \cdot (1-s) + a \cdot (s) \cdot e^{-c_1 - c_3} + b \cdot e^{-c_1}\Big] \cdot e^{-c_2}$$

$$\frac{e^{-c_1}}{a \cdot (1-s) + a \cdot (s) \cdot e^{-c_1 - c_3} + b \cdot e^{-c_1}} = e^{-c_2}$$

$$\Rightarrow c_2 = \ln\Big(\frac{a \cdot (1-s) + a \cdot (s) \cdot e^{-c_1 - c_3} + b \cdot e^{-c_1}}{e^{-c_1}}\Big)$$

Part 2: Math for equalizing spot driver's options (Solve for c_3)

$$u(c_1) = [a' \cdot s \cdot u(c_1 + c_3) + b' \cdot u(c_1 + c_3)]$$

$$1 - e^{-c_1} = [a' \cdot s \cdot (1 - e^{-c_1 - c_3}) + b' \cdot (1 - e^{-c_1 - c_3})]$$

$$1 - e^{-c_1} = (a' \cdot s + b') \cdot (1 - e^{-c_1} e^{-c_3})$$

$$1 - e^{-c_1} = (a' \cdot s + b') - (a' \cdot s + b') \cdot e^{-c_1} e^{-c_3})$$

$$1 - e^{-c_1} - a' \cdot s - b' = -(a' \cdot s + b') \cdot e^{-c_1} e^{-c_3})$$

$$\Rightarrow c_3 = ln\Big(\frac{(a' \cdot s + b') e^{-c_1}}{e^{-c_1} + a' \cdot s + b' - 1}\Big)$$

Code Appendix

May 4, 2015

Code with notes

We first had to set our gamma value (need for the rest of the items below). This relates to the risk aversion function we used.

```
gamma=1/2
```

For the functions below we had to include parameters to change the functions depending upon the time of the week it was run (since the number of packages varied between weekend and week).

Creating the c_2, c_1 values as functions of x_1, x_2, and c_1

```
#################
# c_2 as a function of c_1,x_1 and x_2
Cost2<-function(x_1,x_2,c_1,dist_mean,dist_sd,s){
  k=c_1+Cost3(x_1,x_2,c_1,dist_mean,dist_sd,dist_mean+3*dist_sd,s)
  #using pnorm for P(Y<y)
  b=1-(pnorm(x_1+x_2,,mean = dist_mean,sd = dist_sd))  +
    integrate(f = function(y){return( (y-x_1)/(x_2) * dnorm(y,mean=dist_mean,sd = dist_sd))},
              lower = x_1,upper = x_1+x_2 )$value
  a=1-b
  output<-log(  (a*(1-s)+a*exp(-gamma*k)+b*exp(-gamma*c_1)) / (exp(-gamma*c_1)) )
  return(output)
}
#####
Cost3<-function(x_1,x_2,c_1,dist_mean,dist_sd,total_available=dist_mean+3*dist_sd,s){
  bprime= integrate(f = function(y){return( (y-(x_1+x_2))/(total_available -(x_1+x_2)) *
                                              dnorm(y,mean=dist_mean,sd = dist_sd))},
                    lower = x_1+x_2,upper = total_available )$value
  aprime=1-bprime

  inside<-( (exp(-gamma*c_1)*(aprime*s+bprime))/(exp(-gamma* c_1)+aprime*s+bprime-1) )
  if(inside<1){inside=1}
  output<-log(inside)
  return(output)
}
```

Creating the Total Expected Cost function (TEC).

```
#####
#TEC
Total_Expected_Cost<-function(x_1,x_2,c_1,dist_mean,dist_sd,s){
  #creating total available given distribution
  #3 sds above the mean
  total_available<-dist_mean+3*dist_sd
  fixed<-c_1*x_1+Cost2(x_1,x_2,c_1,dist_mean,dist_sd,s)*x_2
  option<-(c_1)*integrate(f=function(y){return((y-x_1)*dnorm(y,mean = dist_mean,sd = dist_sd))},
                                                                  lower=x_1,upper=x_1+x_2)$value
```

```
  option2<-(c_1)*(x_2)*(1-pnorm(x_1+x_2,mean=dist_mean,sd=dist_sd))
  spot<-(c_1+Cost3(x_1,x_2,c_1,dist_mean,dist_sd,total_available,s))*
    integrate(f=function(y){return((y-x_1-x_2)*
                                     dnorm(y,mean = dist_mean,sd = dist_sd) )},
              lower = x_1+x_2 ,upper = total_available)$value

  output<-(fixed+option+option2+spot)
  return(output)
}
```

Before we go any further, we needed to set our s value (probability to find work on general spot market), and the c_1 value.

```
s_value=.5
c_value=1
```

Now, in preparation for optimizing the TEC, we made a second TEC function that just a function of 2 variables, which was fairly simple to optimize.

```
Total_Expected_Cost_Optimize<-function(xvalues){
  return(Total_Expected_Cost(x_1=xvalues[1],x_2=xvalues[2],c_1=c_value,
                             dist_mean=12000,dist_sd = 2400,s=s_value))
}
```

Since the optimization function looks for local minima, the answer depends on the starting point. Below is one such optimization sample starting point.

```
#Since the optimization function looks for local minima, the answer
# depends on the starting point. Here is one such optimization sample starting point.
data<-optim(c(2500,2500),fn=Total_Expected_Cost_Optimize,lower=c(0,0),method="L-BFGS-B")
```

and here is the report from the optimization function:

```
data
```

```
## $par
## [1] 6218.777 6217.339
##
## $value
## [1] 13127.65
##
## $counts
## function gradient
##       17       17
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

To make sure we didn't miss a lower local minimum (the global minimum), we decided to create a small grid of potential plausible starting values.

```
fulldata<-matrix(0,nrow=3,ncol=3)
i=1

#######
#checking over multiple starting points
#basically checking starting points via grid approach
for(index1 in c(2500,5000,7500)){
  j=1
  for(index2 in c(2500,5000,7500)){

    data<-optim(c(index1,index2),fn=Total_Expected_Cost_Optimize,lower=c(0,0),method="L-BFGS-B")
    fulldata[i,j]<-data$value
    j=j+1
  }
  i=i+1
}
```

It should be noted that some of the entries are zeros because the code gets errors. This is due to the fact that we assumed that total available drivers was 19,200 for week days, which is less than the starting point $x_1 = 10,000$ and $x_2 = 10,000$, it seems as if we found a good result anyway, even without a complete grid.

```
fulldata
```

```
##          [,1]     [,2]     [,3]
## [1,] 13127.65 13127.65 13127.65
## [2,] 13127.65 13127.65 13127.65
## [3,] 13127.65 13127.65 13127.65
```

From the grid above we selected the optimal starting point and ran the model

```
#selecting minimum value from fulldata grid
data<-optim(c(5000,2500),fn=Total_Expected_Cost_Optimize,lower=c(0,0),method="L-BFGS-B")
```

And below is all the data from our predicted best model

```
#print out values
data
```

```
## $par
## [1] 6232.181 6203.933
##
## $value
## [1] 13127.65
##
## $counts
## function gradient
##       20       20
##
## $convergence
```

```
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```
#checking results
Total_Expected_Cost_Optimize(data$par)
```

```
## [1] 13127.65
```

```
#getting C_2 value
Cost2(x_1=data$par[1],x_2=data$par[2],c_1=c_value,dist_mean=12000,dist_sd = 2400,s=s_value)
```

```
## [1] 0.0941948
```

```
#getting C_3 value
Cost3(x_1 =data$par[1],x_2=data$par[2],c_1=c_value,dist_mean=12000,dist_sd = 2400,s=s_value)
```

```
## [1] 0.7339184
```

```
#re-examing s and c_1 values
s_value
```

```
## [1] 0.5
```

```
#c_1 value
c_value
```

```
## [1] 1
```

Weekend

The above optimum was just for the week, so below is the code for the weekend (no additional commentary)- we just changed the distibution function.

```
 #Weekends
########
Total_Expected_Cost_Optimize<-function(xvalues){
  return(Total_Expected_Cost(x_1=xvalues[1],x_2=xvalues[2],
                             c_1=c_value,dist_mean=8000,dist_sd = 1600,s=s_value))
}
#Since the optimization function looks for local minima, the answer
# depends on the starting point. Here is one such optimization sample starting point.
data<-optim(c(5000,2500),fn=Total_Expected_Cost_Optimize,lower=c(0,0),method="L-BFGS-B")
data
```

```
## $par
## [1] 4154.749 4135.982
##
```

```
## $value
## [1] 8751.769
##
## $counts
## function gradient
##       14       14
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

```
#We could grid to look at possiblities, but this value looks pretty good


Total_Expected_Cost_Optimize(data$par)
```

```
## [1] 8751.769
```

```
#getting C_2 value
Cost2(x_1=data$par[1],x_2=data$par[2],c_1=c_value,dist_mean=8000,dist_sd = 1600,s=s_value)
```

```
## [1] 0.09419361
```

```
#getting C_3 value
Cost3(x_1 =data$par[1],x_2=data$par[2],c_1=c_value,dist_mean=8000,dist_sd = 1600,s=s_value)
```

```
## [1] 0.7339166
```

```
#re-examing s and c_1 values
s_value
```

```
## [1] 0.5
```

```
#c_1 value
c_value
```

```
## [1] 1
```