

For office use only

T1 \_\_\_\_\_

T2 \_\_\_\_\_

T3 \_\_\_\_\_

T4 \_\_\_\_\_

For office use only

F1 \_\_\_\_\_

F2 \_\_\_\_\_

F3 \_\_\_\_\_

F4 \_\_\_\_\_

---

**2016**

**19th Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet**  
(Please make this the first page of your electronic Solution Paper.)

**Team Control Number: 7204**

**Problem Chosen: B**

Please paste or type a summary of your results on this page. Please remember not to include the name of your school, advisor, or team members on this page.

In a world dancing to the rhythm of technology and innovation, increasingly rapid response-times have become the norm. With the advent of widespread internet usage, facets of life like research, communication, even socializing are reaching levels of unprecedented speed.

As commerce and retail begins to make the transition to the web, similar patterns are emerging. Dying are the days when a consumer must suffer the arduous trek to the nearest bookstore when Mathematics: Modelling Our World can be ordered, packaged, and delivered to the front door without leaving the comfort of bed. In reflection of the changing times, our company, SportsmanSHIP, has requested an analysis on the optimal placement of store warehouses across the continental United States in order to guarantee one-day shipping times through the United Parcel Service (UPS) for every customer in the 48 contiguous states.

To do so, a model of shipping times from various locations across the US was created by collecting all of the almost 700 delivery-time maps kept by UPS. By choosing to use exact data directly from the maps as opposed to constructing mathematical approximations, we were able to ensure our analysis was based on information of the utmost accuracy while simultaneously leveraging our computational method of analysis: Markov Chain Monte Carlo (MCMC). MCMC itself is a family of algorithms designed to probe a parameter space for points of interest. Our particular implementation of MCMC was built to optimize the number and placement of warehouses across ZIP Codes located within the continental US, considering factors like percentage of the area covered by one-day shipping, cost of maintaining warehouses, population of surrounding area, sales tax, and tax on clothing and other apparel.

Ultimately, the results of our analysis presented a curious conundrum. On one hand, when optimizing for percentage area covered by one-day shipping times, almost complete coverage of the US is possible with surprisingly few warehouses. On the other hand, when optimizing for profits alone, coverage is sacrificed for areas of sparser population. Thus, the boss of SportsmanSHIP must ultimately decide whether he or she would still like to pursue the one-day shipping guarantee when our analysis suggests greater profits may be obtained elsewhere. Regardless, coverage remains relatively complete in both cases, heralding the dawn of a new age of rapid, digital retail.



# Sportsman SHIP

Team 7204

Team 7204

November 14, 2016

Anthony Bausse  
President of SportsmanSHIP  
Department of Administration  
SportsmanSHIP Headquarters  
Manchester, New Hampshire 03104

Mr. Bausse —

As per your instructions, we have carried out preliminary analysis on the optimal placement of warehouses in the continental United States to guarantee every customer in the contiguous 48 states one-day shipping time. To do so, we constructed an algorithm designed to search for ideal locations based on the following factors:

- Percentage of the US covered by the one-day delivery times, as given by UPS
- Cost of maintaining the warehouse balanced against potential revenues
- Population of the surrounding area
- State-specific sales tax imposed on goods sold from a particular location
- Clothing and other apparel-related tax and tax exemptions

Our results indicated a slight inconsistency between coverage and profits. Considering the original goal of ensuring one-day shipping across the US, such a feat may be accomplished with as little as 57 warehouses, their precise locations chosen carefully by our algorithm and presented in the following report. However, in the course of performing follow-up analysis centered purely on optimizing profits, we discovered a more attractive strategy than the one-day guarantee. By sacrificing coverage of certain, sparsely populated areas in the interest of greater revenues elsewhere, we stand to increase company earnings significantly. Keep in mind, coverage remains excellent in both cases. The difference is merely a two-day delivery time for very few individuals living in highly rural areas of the United States in return for over double the profit increase. Thus, our final recommendation is to depart from the original plan of one-day shipping time for everyone in the US and adopt a modified strategy still focused on high coverage but with added emphasis on the ultimate goal of the operation: profits.

While perhaps a somewhat unexpected proposal, it is a recommendation nonetheless founded upon solid justification. The model we used was based upon exact data extracted directly from delivery-time maps provided by UPS, facilitating equal levels of accuracy in

our conclusions. Furthermore, the algorithm used to perform the analysis, called "Markov Chain Monte Carlo," is a widely-accepted, well-tested, robust method for performing the optimization.

In the end, while one-day shipping coverage is certainly feasible, for the ultimate sake of the company, we recommend a configuration of warehouses more focused on profit, as presented in greater detail in our report. While the blinding speed of response times certainly pressures us into the former strategy, sometimes, the tortoise really does win the race.

Best,

Team 7204

# Maximizing Customer Convenience with Hill Climbing

Team #7204

November 2016



---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Assumptions and Justifications</b>	<b>3</b>
<b>3</b>	<b>Methods of Data Gathering</b>	<b>4</b>
3.1	Scraping Data . . . . .	5
3.2	Extracting Information from UPS Maps . . . . .	5
<b>4</b>	<b>Model</b>	<b>7</b>
4.1	Markov Chain Monte Carlo . . . . .	7
4.2	Modifications . . . . .	8
<b>5</b>	<b>The Utility Function</b>	<b>9</b>
<b>6</b>	<b>Incorporating Sales Tax</b>	<b>10</b>
6.1	Analysis of Variables . . . . .	10
6.2	Revision of Utility Function . . . . .	11
<b>7</b>	<b>Tax on Clothing and Apparel</b>	<b>11</b>
7.1	Revision of Utility Function . . . . .	11
<b>8</b>	<b>Extending our Model</b>	<b>12</b>
8.1	Power to the People . . . . .	12
<b>9</b>	<b>Results</b>	<b>13</b>
9.1	Coverage . . . . .	13
9.2	Vetting the Perfect Solution . . . . .	14
9.3	Incorporating Sales Tax . . . . .	15
9.4	Incorporating Tax on Clothing and Apparel . . . . .	16
<b>10</b>	<b>Comprehensive Economic Model</b>	<b>17</b>
<b>11</b>	<b>Discussion</b>	<b>19</b>
11.1	Implications of Some Utility Function . . . . .	19
11.2	Considering Coverage . . . . .	19
11.3	To Profit or to Serve . . . . .	21
11.4	Analysis of Methods . . . . .	21
11.4.1	Strengths . . . . .	21
11.4.2	Weaknesses . . . . .	21
11.4.3	Looking Ahead . . . . .	21
<b>12</b>	<b>Conclusion</b>	<b>23</b>
<b>13</b>	<b>Appendix</b>	<b>25</b>

# 1 Introduction

As technology advances and the pace of life quickens, companies are expected to continually provide better services, be more accessible, and respond to requests faster. While such expectations have led to the rapid growth of retail stores and service locations throughout the United States, delivery services have arguably been the most influenced by these rising demands.

Amazon.com is an online storefront that has seen tremendous growth throughout its lifetime, and it only continues to grow. Founded in 1994 as an online bookstore, the digital retailer eventually expanded its services to include technology, home goods, and even eBooks. Amazon's response to its customer's rising demands of customers was to grow and evolve. In February of 2005, Amazon launched Amazon Prime, a service that provided express shipping for hundreds of thousands of products. Amazon believed that fast deliveries should not be an occasional indulgence, but an everyday experience [1]. While the company previously advertised a four-to-six day shipping time, Amazon Prime boasts a mere two day shipping time. Prime was so popular with customers that in 2014, the electronic retailer decided to evolve even further, adding Prime Now, a same day delivery option, to its list of services, granting members free two hour delivery.

Prime Now, PrimeNow.com, and Amazon's constant growth makes one wonder, "How can Amazon achieve such absurdly short delivery times?" A likely answer lies with efficiently distributed warehouses.

While it is possible for a delivery service company to operate solely from its headquarters, such centralization would result in inconsistent, sub-optimal delivery times, with times differing between locations nearer and farther away. The solution is to have warehouses scattered throughout the country, ensuring the presence of nearby locations to service any given customer. On the same note, it is important to consider the physical viability of such an option. A warehouse can be expensive to maintain, so the amount and placement of the warehouses must be optimized such that the service requirement can be met while also being cost efficient. Companies like Amazon who can afford to offer free delivery succeeded in balancing coverage area and company costs, and our company, SportsmanSHIP, intends to do the same.

SportsmanSHIP is a recreational equipment company based in New Hampshire. Currently, we have a physical location that customers can buy from, but most of our business is conducted through online sales. Due to an increased demand for our products, we have decided to expand our company by opening warehouses across America. Amazon currently has great coverage across the United States, guaranteeing two day shipping for all areas, and two hour shipping for select locations, but we believe that we can do better. We are a team of SportsmanSHIP employees that also share a passion for mathematical modeling, and we have been tasked with optimizing SportsmanSHIP's warehouse locations. Our warehouse distribution will allow us to provide one day shipping to nearly all of the United States, and we will do so with minimal warehouses.

In this paper, we analyze the crux of large delivery service companies: efficient warehouse distribution. We identify important factors of efficiency such as amount of warehouses, population density, and national coverage. Using these factors, along with data given in regional maps, we constructed a comprehensive model for efficient warehouse placement. Our ultimate goal was to lower shipping time below one day for customers across the United States while minimizing the number of warehouses. In addition, we were asked to consider and discuss the implications of state sales tax on the model, as well as to account for taxes on clothing and apparel, which SportsmanSHIP plans on adding to our inventory in the near future.

To tackle the task at hand, we implemented a model that used a Markov Chain Monte Carlo (MCMC) algorithm to determine the utility of a sample distribution. For the first task, maximizing

national coverage and minimizing number of warehouses, this utility function considered in the percentage of the United States covered by a one-day warehouse shipping facility as well as the number of warehouses used. For the next task, we extended this model to include a sales tax calculation factor, negatively impacting the utility of a model with a large amounts of sales tax. For the third task, we modified the sales tax factor to include a weighted tax factor that took into account both clothing and sales taxes. Finally, we developed a separate economic model that compares utility based off of profit only, and does not weight the percent of the United States covered nearly as strongly. This was implemented as another way to look at the problem of optimizing warehouses, since it has been found that only 4% of customers truly care about same-day or one-day shipping [10].

Using our models, SportsmanSHIP's expansion is guaranteed to be a success, and one day, we may even surpass Amazon. All it takes is a bit of warehouse knowledge, and a lot of math.

## 2 Assumptions and Justifications

**Assumption:** The data presented to us by the United Parcel Service is correct.

*Justification:* UPS is a reputable company that has been operating for years. It has also commercialized this data to the user; therefore, we will assume it is reputable.

**Assumption:** The curvature of the Earth is negligible enough to be approximated by a polar transform and a linear scaling factor.

*Justification:* In the process of translating pixels on a map into usable ZIP Codes, we considered ways to project the flat map onto a spherical Earth. Since we worked on a relatively small scale, it is not necessary to accurately model the curvature of the Earth. However, since the scale is big enough, we decided to use a linear correction factor to ensure proper transformation.

**Assumption:** Distances on the UPS maps translate consistently to distances on Earth by the same factor.

*Justification:* Similar to the previous justification, because the scale of the map is relatively small when compared to the entirety of the Earth, the factor with which distances on the map must be multiplied to obtain the actual distance would remain effectively constant across the entire map.

**Assumption:** The first three digits of the ZIP Code are precise enough to delimit a region in the United States.

*Justification:* In the course of determining warehouse placement, for the purpose of easing computational expense, locations were determined roughly with a granularity corresponding to the first three digits of a ZIP Code. These three digits do specify small regions of states and are used widely to determine mailing areas around the United States.

**Assumption:** Population by ZIP Code has not changed significantly since 2010.

*Justification:* The population in the United States is determined by the census taken every ten years. Since, 2010 was the last census taken, we can assume it will stay accurate until the next one is taken in 2020.

**Assumption:** We, and the customers, wish to minimize tax liability.

*Justification:* Even if we are not directly impacted by sales tax, with no sales tax, we are able to advertise a lower price, making it more attractive to customers. These customers will likely buy from us in the future.

**Assumption:** The States having "None or limited tax on clothing and shoes" in the problem notes can be represented with either no sales tax or a reduced sales tax of 1%, depending on whether or not they have clothing sales tax.

*Justification:* According to TaxFoundation.org, many states offer some sort of exemption for sales tax on clothing. Some are completely exempt and some are exempt to a cap. These caps vary widely, from \$50 to \$250, and we cannot predict what our customers will purchase, but instead of making all of these states have 0% sales tax, we want to make our model more comprehensive. One factor in our model measures the utility of a distribution of warehouses using sales tax and population affected. When incorporating clothing, our model cares more about whether or not a state has sales tax than the exact sales tax. 1% is a quantity that is very small, but not 0%, so we decided that 1% is a reasonable tax rate to assign to states with caps or limitations that on clothing sales tax.

**Assumption:** National sales data accurately models a given store's regular departmental sales.

*Justification:* Ultimately, departmental stores are simply samples of national data since they cater to a general, non-specific population.

**Assumption:** All warehouses are stocked with appropriate merchandise required to satisfy local customers. That is, no customer needs to wait for merchandise from a warehouse two days away because the warehouse one day away is out of stock.

*Justification:* The focus of the problem was on the placement of warehouses, so the process of stocking said warehouses is beyond the scope of our analysis. We leave it to the SportsmanSHIP Department of Logistics to figure out how to keep warehouses supplied with relevant merchandise.

### 3 Methods of Data Gathering

Shipping logistics, by necessity, is an extremely data-driven field. This has fortunately presented us with a plethora of quantitative information with which we may rigorously calculate an optimal solution. This data ranges from UPS Time-in-Transit maps and US Census records to sales tax tables and corporate financial statements.

But like many data-informed projects nowadays, this abundance of data comes with the burden of *over-information*, where the datasets we query are vast enough to make retrieval occupy



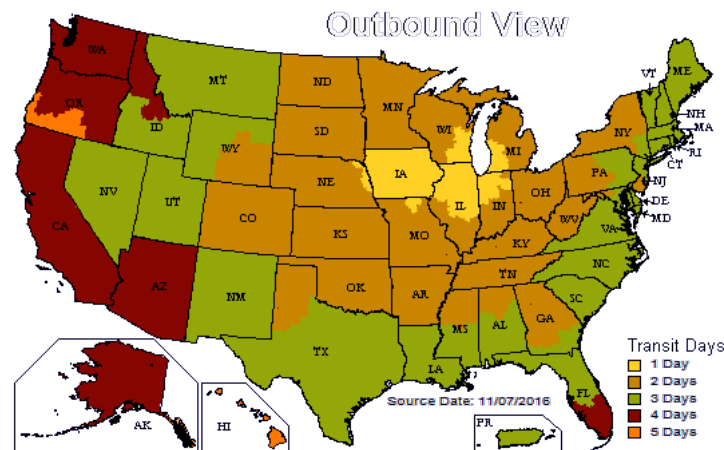
significant time and make computations on them intractable. Thus, we have devised methods to simplify the data without loss of accuracy. These methods are described in this section.

Qualitative understanding of the mechanics of logistics has been indispensable as well. In learning how retailers collect sales tax and how customers decide on a vendor, we have been able to make sense of the vast sea of data available to us.

### 3.1 Scraping Data

Although databases and CSV's are nice, the most useful of data often needs to be mined and refined before it can be applied to solve a problem. We accomplished this by creating HTTP-enabled scripts written in Python3 capable of crawling websites and CSVs to harvest data.

The first and most important data set which we scraped in this fashion was the UPS Time-in-Transit Map database. This website takes a shipment origin ZIP Code and returns a map indicating the estimates number of days required for the package to arrive. Fortunately, these maps were placed in a publicly accessible directory and sequentially labeled from 0001.gif to 0691.gif, making it easy to quickly download them all using a script.



There's a catch though: UPS provides 691 Time-in-Transit maps, while there over 43,000 ZIP Codes in the United States. With query latency on the order of *seconds*, it would take ages for a script to check UPS's website with all of these ZIP Codes. To save time, we only retrieve what we need: ZIP Codes are 5-digit designations that can precisely identify a *neighborhood* in the country. Since we only need to map ZIP Codes to region maps, we can just use the first three digits of the ZIP Code, which has precision down to the region of the state. This reduces the query space from 43,000+ ZIP Codes to about only 900, drastically saving time. Additionally, as validation for this simplification, we confirm that several of these 3-digit ZIP Code blocks fit into each of UPS's Time-in-Transit maps, indicating that we have more granularity than the data that we will be working with.

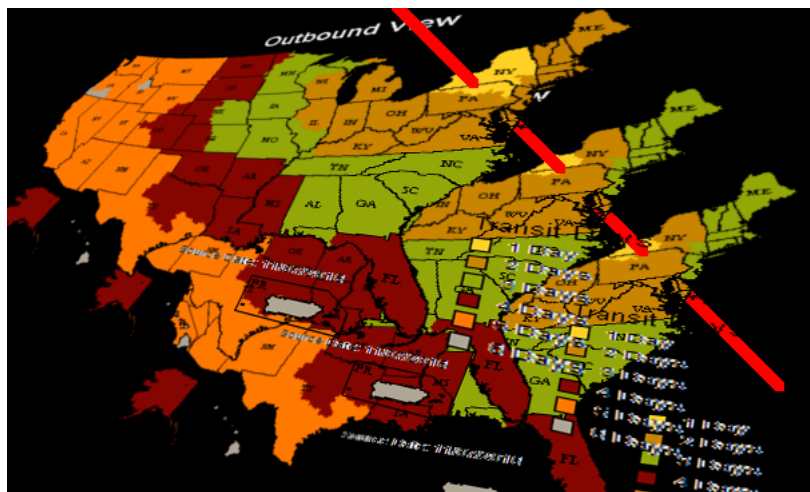
Outside of this, scraping data from static datasets, like CSV's and text files, was trivial. A series of Python scripts allowed us to parse ZIP-Code-to-city/state/coordinate data, as well as census and legal data connecting ZIP Codes to population counts and sales taxes.

### 3.2 Extracting Information from UPS Maps

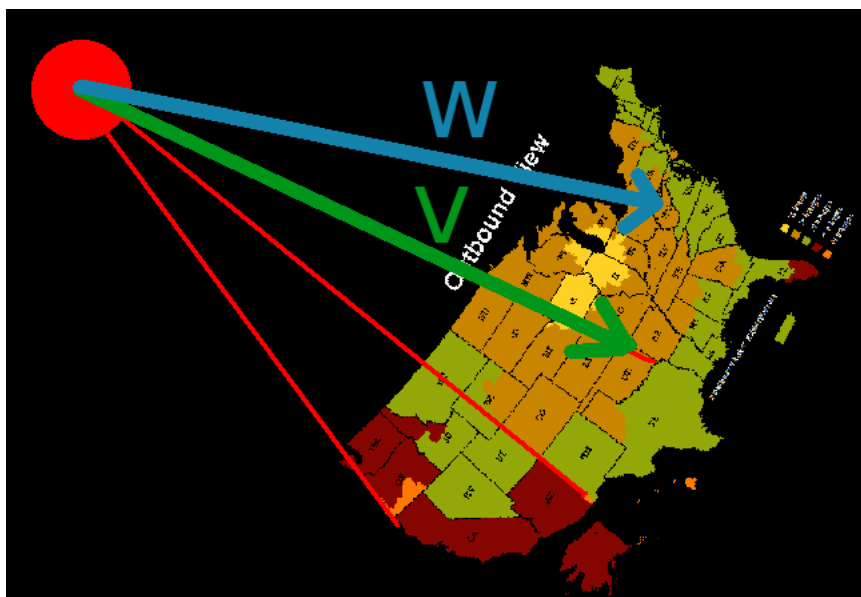
Although they compile real data into a condensed visual format, UPS's Time-in-Transit maps are designed as a tool for a humans shipping managers to estimate shipping times. UPS could

hardly expect anyone to try to use a computer program to scrape data from them! However, these maps contain just as much information as any UPS shipping API available for a thousandth of the time cost: Since our optimization program will need to make tens of thousands of Time-in-Transit queries as it searches the parameter space, it would be impossible for it to make all of these queries over the internet under heavy latency. Therefore, it is only viable to try to extract as much information as possible from these maps as possible.

This, then, is a matter of image processing, at task in which Python3 excels. Thankfully, all of the images are standardized with the same color and pixel locations. Therefore, to determine whether a particular place in the country resides in a 1-day Ground Delivery zone, one can sample a slice of *all* the maps and check if the particular yellow corresponding to a 1-Day Ground Delivery appears. This is highly optimized through matrix slicing operations in NumPy.



In order for these pixel samples to be meaningful, however, it is necessary to tie each pixel to a real location (Latitude, Longitude) on the globe. This becomes tricky, since it is unknown what projection this particular map uses to account for the Earth's curvature (note how the latitudinal state borders curve towards the pole). However, since the scale of the map is small relative to the curvature of the Earth, one can make a fairly accurate **polar approximation of Earth's curvature**.



This is accomplished as shown in the figure above with a large amount of trigonometry and some vector transformations. With this method and its inverse, it is possible to map any pixel on the image to a real coordinate, and vice versa, with minimal deviation at the far East and West edges of the image.

This accomplishment is the core of our ability to draw conclusions about warehouse placement: Since we can map ZIP Codes to coordinates to pixels on Time-in-Transit maps, we can determine the UPS shipping time between any two locations in the United States. This map transform algorithm is embedded deeply into our program in *mapper.py*, a module which unites all of UPS's shipping data with the geographic, economic, and population data gathered during our research. With all this data at hand, we are ready to begin work on our model.

## 4 Model

First, we wish to figure out how to heuristically determine the minimum number of warehouses that can still have maximum coverage. Therefore, we utilized a variant from the family of algorithms called Markov Chain Monte Carlo (MCMC) to systematically find the best distribution.

The model from which our implementation of MCMC samples from consists of a series of discrete maps depicting regions of the United States within a set delivery time given the location of a delivery source — in this case, our warehouses. These maps were obtained from the United Parcel Service's website application used for estimating package-delivery times.

### 4.1 Markov Chain Monte Carlo

To determine the optimum number and placement of warehouses, a Markov Chain Monte Carlo algorithm was designed. The initial problem facing an analysis of warehouse placement is the potentially enormous number of warehouses required to cover the continental United States, with each warehouse along with the total number of warehouses translating to their own individual parameters. The resulting parameter space may easily be in sixty, seventy, even eighty or more dimensions, presenting an immense computational obstacle. Thus, to address this issue, MCMC was chosen for its simplicity and efficacy in determining optimal values for a high number of parameters [8].

The basic form of the algorithm proceeds as follows. Let  $\mathbf{x}$  be a collection of ZIP Codes representing the locations of potential warehouses. Let  $u(\mathbf{x})$  be a utility function determining the "utility," that is to say effectiveness, of a particular configuration of warehouses. The specifics of the utility function are at a later time. Below are the steps of the algorithm:

1. Randomly select an initial  $\mathbf{x}$ . Let this vector be  $\mathbf{x}_n$
2. Randomly select a second  $\mathbf{x}$  close to the first. Let this vector be  $\mathbf{x}_{n+1}$
3. Compare  $\mathbf{x}_n$  and  $\mathbf{x}_{n+1}$ .
  - If:**  $u(\mathbf{x}_{n+1}) > u(\mathbf{x}_n)$ ,  
keep  $\mathbf{x}_{n+1}$
  - Else if:**  $\frac{u(\mathbf{x}_{n+1})}{u(\mathbf{x}_n)} > U \in [0, 1)$ , where  $U \in [0, 1)$  is a random draw from a uniform distribution between 0 (inclusive) and 1 (exclusive),  
keep  $\mathbf{x}_{n+1}$
  - Else:**  
keep  $\mathbf{x}_n$

- 
4. Let the vector kept from step 3 be the new  $\mathbf{x}_n$ . Repeat from step 2.

For the purpose of determining the optimum number of warehouses to use, a second loop was added to the traditional MCMC routine described above.

If  $n$  of  $\mathbf{x}_n$  is a multiple of 100,

1. Let  $s_m$  be the size of  $\mathbf{x}_n$ . That is,  $s$  is the number of elements in  $\mathbf{x}_n$
2. Let  $s_{m+1}$  be a value close to  $s_m$ . Expand or shrink  $\mathbf{x}_{n+1}$  through the addition or removal of random values until it matches the size prescribed by  $s_{m+1}$
3. Let the main loop of the algorithm run through another 100 iterations.
4. Compare  $s_m$  and  $s_{m+1}$ .
  - If:**  $u(s_{m+1}) > u(s_m)$ , where  $u(s)$  is the accumulated utility throughout the batch of 100 iterations associated with the particular  $s$ ,
    - keep  $s_{m+1}$
  - Else if:**  $\frac{u(s_{m+1})}{u(s_m)} > U \in [0, 1)$ ,
    - keep  $s_{m+1}$
  - Else:**
    - keep  $s_m$
5. Let the value kept from step 3 be the new  $s_m$ . Repeat from step 2.

A value of 100 was chosen to allow the algorithm to settle the warehouse-locations into a configuration before potentially disturbing it with the addition or removal of further warehouses. If no such delay was incorporated, the constant flow of random warehouses into each  $\mathbf{x}$  would disturb the optimization attempted by the main loop of the MCMC tremendously.

Note also that in both cases, the rather vague phrasing "choose a value close to..." was used. To chose this nearby value, they were drawn from a Gaussian distribution centered on the present value, giving greater probability to choosing values closer to the current one, but not making it impossible to select a value further away. Selecting appropriate standard deviations for the Gaussian is more art than science, and were picked through a combination of common sense and experimentation.

After every iteration, the algorithm saves the data and locations to a file. Therefore, once the algorithm runs through its set iterations or achieves its desired value, it is able to demonstrate the warehouse distribution that held the maximum utility. This is what the program will output as the best distribution.

## 4.2 Modifications

While this method allows us to systematically find the optimum number, the method can be very slow and extremely computationally expensive. Therefore, some slight modifications were made.

- Variable Step Size

This method was implemented in order to help a walker agent, an instance of the MCMC algorithm, find a maximized area of coverage. The idea is that when the coverage becomes high, an optimal solution is nearby, so it is very important for the agents to slow down. This way, they are able to find an optimal solution. This helped the algorithm go faster and find more optimal solutions. Therefore, the step-size of the agent (i.e the range at which it could

choose its random values from) decreased as a function of how much the map was covered. The function is outlined below:

$$10 + 200\sqrt{1 - p}$$

Where  $p$  represents coverage. This variable will be more rigorously defined in the utility function section.

- Initial State MCMC

This method was implemented in the third task in order to hasten the model's run time by choosing, beforehand, a favorable configuration for the problem. This configuration was determined by an MCMC run of the first task of the problem, but nevertheless, still had a favorable utility.

This method is now sufficiently optimized to produce a quality result quickly. Nevertheless, this method is only as good as its utility function, since that is the basis of the walker's measurements. Therefore, we will go on to define the utility function.

## 5 The Utility Function

Now that we have a model and a method to find the best possible solution, it is absolutely necessary to determine a measure of a good distribution.

A utility function is defined as a mathematical relationship that determines the quality of a particular system. In our analysis, the function accepted the configuration of warehouses and returned a particular value corresponding to the utility of said configuration, with higher values implying greater utility. There were two factors that went into constructing the function: the number of warehouses and the percentage of the continental United States covered by one-day UPS delivery times. To account for further factors like sales and clothing taxes, a revised utility function is discussed in a later section.

The first component considered in creating our function was the number of warehouses. Maintaining a warehouse requires significant costs and resources, often costing anywhere between \$15 and 25 million annually, so the goal was to minimize their numbers, with a large number of warehouses corresponding to a smaller utility. To achieve such an affect, we considered the following function:

$$U = \frac{1}{s}$$

where  $u(s)$  is our utility function and  $s$ , a somewhat poorly-named variable standing for "size," is the number of warehouses. The simple inverse proportion enables our function to generate higher utilities for lower numbers of warehouses.

Then for area covered, because the ultimate goal of the warehouses is to ensure one-day delivery for nearly everyone living in the continental US, the area covered must be maximized. Ideally, area covered should be considered in such a way as to reward nearly full coverage and penalize anything less, yet such a scheme presents a unique problem for MCMC. Because our algorithm relies on random-walks through parameter space, the potentially steep distribution generated by such a utility function would increase the difficulty for the algorithm to discover the optimal configuration of warehouses. Thus, we turned to a variety of gentler alternatives, experimenting with various concavities and scaling values.

Ultimately, through empirical observations, we determined the following utility function to be the most effective (the component describing number of warehouses is included):

$$U = \frac{p^2}{s}$$

where  $p$  is a value in the range  $[0,1)$  representing the percentage of the continental US covered by one-day delivery times or “coverage”. The process of calculating this percentage from the configuration of warehouses is described in the following section.

A brief aside, the simplicity of the utility function proved to be a somewhat surprising find, outperforming our highly-adjusted logistical-growth monster and other more convoluted functions. Simplicity, as it happens, really is the key.

## 6 Incorporating Sales Tax

As a business, we are required to follow certain legal regulations in order to ensure that we are paying our fair share to the state and country we reside in. One of these legal obligations is Sales Tax. While sales tax laws are generally complicated and messy, we will utilize the simplistic guidelines given to us, which state that shipments made between a warehouse and customer in the same state will have to pay the state’s sales tax, while shipments made between a warehouse and a customer in different states will not pay any sales tax. We will utilize this data in order to construct an improved utility function.

In order to account for Sales Tax, it is necessary to determine how sales tax will impact the overall utility. Therefore, we will implement the sales tax factor, making the new utility equation look as follows:

$$u(\mathbf{x}) = \frac{p^2 t}{s}$$

Where  $t$  is the sales tax,  $p$  is percentage covered, and  $s$  is the number of warehouses. In perfect utility, each warehouse would have a sales tax factor of 1, since sales tax would not influence utility at all. However, with sales tax, there needs to be a penalty given to the sales tax factor.

### 6.1 Analysis of Variables

- **The Percentage Shipping Population Within The State**

In order to ensure this is as accurate as possible, we must take into account how the population is spread throughout its surrounding region. For instance, consider two warehouses that can ship both within its state and outside of its state in one day. Let us say that warehouse one has 25% of its shipping population in its own state, while warehouse two has 75% of its shipping population in its own state. The first warehouse should have greater utility since there is a lesser proportion of people within the warehouse’s own state. This means that the majority of warehouse one’s customers do not have to pay sales tax, which consequently should result in greater utility.

Additionally, this is also more useful than taking into account area, because area is not necessarily coordinated with population. A distribution in which there is a warehouse in Utah with greater sales tax area but less sales tax population should have less utility than a region in New York with a lower sales tax area but a greater sales tax population.

In order to calculate this for each warehouse, we will look at all the ZIP Codes that the area serves, and from there determine the proportion of ZIP Codes for each state. Then using

population information about each ZIP Code, we can find the percentage of the shipping populations within the state. We will denote this value as  $k_i$  where  $i$  is the warehouse number

- **Weight of Sales Tax**

The weight of the sales tax is an important factor in building our utility function. Clearly an item bought in a state with 5% sales tax should receive a lower utility value than an item purchased in a state with only a 1% sales tax. Additionally, it is important to ensure that a 0% sales tax will not affect the utility at all. In order to ensure that this value is between 0 and 1, we will divide by the maximum sales tax that is given in the problem. This value is 7.5%. We will calculate this for each warehouse by looking at its ZIP Code and the state the ZIP Code is in. We will denote this parameter as  $w_i$  where  $i$  is the warehouse number.

## 6.2 Revision of Utility Function

Now that we have analyzed what variables to consider, let us fully construct the utility function, specifically the parameter  $t$ . Normally, under perfect utility, this value will be one, however we need to ensure that this is corrected for by accounting for the sales tax and the percentage of the shipping population within the state. Since they are both normalized, we may simply multiply them to obtain the correction factor. Subtracting by one, we obtain a new sales tax utility for each warehouse. Since sales tax is based on the warehouse locations themselves, we must compute this factor and average over all of the regions. In terms of parameter  $t$  this can be summarized as:

$$t = \frac{\sum_{i=1}^s 1 - k_i w_i}{s}$$

Where  $t_s$  is the tax of warehouse,  $k_i$  is the percentage population of the shipping reason with respect to warehouse  $i$ ,  $w_i$  is the weight of the sales tax with respect to warehouse  $i$ , and  $s$  is the number of warehouses, consistent with the previous equation. Plugging this in for the parameter  $t$ , we obtain the function below:

$$u(\mathbf{x}) = \frac{p^2 \sum_{i=1}^s 1 - k_i w_i}{s^2}$$

This is our desired utility function.

## 7 Tax on Clothing and Apparel

Another idea our business would like to implement is a clothing line. Clothing lines are unique by themselves since they have many of their own tax laws. Therefore, we need to consider how we will incorporate the taxes of clothing into our model. In order to do this, we will simply modify the sales tax factor  $w_i$ .

### 7.1 Revision of Utility Function

This factor denoted as  $w_i b_n + b_a a_i$ , where  $b_a$  is the apparel tax weight,  $b_n$  is the non-apparel tax weight,  $a_i$  is the apparel tax, and  $w_i$  is the sales tax calculated exactly how it was before. This is essentially a linear congruence that splits the goods that are apparel and non-apparel into the proportion denoted by the ratio  $b_a : b_n$ .  $a_i$  is calculated exactly like  $w_i$  is except with apparel tax rates rather than sales tax rates. Additionally, to ensure this term is still normalized between 0

and 1, we will make sure the weights  $b_a$  and  $b_n$  add up to 1. Plugging in our equation for the previous value of  $w_i$  is therefore:

$$u(\mathbf{x}) = \frac{p^2 \sum_{i=1}^s 1 - k_i(w_i b_n + b_a a_i)}{s^2}$$

Since we are modeling the growth of a clothing department, we will model both a store that has a 0.9 : 0.1 non-apparel to apparel ratio of its store as well as a store that has a 0.7 : 0.3 non-apparel to apparel ratio to its store. The first situation signifies a smaller department which would likely be the case of a starting department, while the second situation models a regular sports apparel store. The ratio is based on the national ratio of wholesale non-apparel sports sales to the wholesale apparel sports sales taken from statistica.com [12, 13].

## 8 Extending our Model

We have seen that our model is an excellent approximation of the general demographic, geographic, and economic features of the United States in that it is able to identify the general utility of warehouse placement in some particular region of the country. However, let's go further.

### 8.1 Power to the People

In the real world, no matter how much one may deliberate on the placement of a warehouse based on demographics or logistics, the ultimate trial by fire is this: does the volume of shipment *and therefore revenue* justify the cost of maintaining a warehouse at that particular location? In other words, *is our arrangement of warehouses profitable?*

Hence, we consider the following model as the ultimate "real world" metric of whether our arrangement is desirable: Suppose that we select a large sample of customers (in our simulation, roughly 450,000) around the nation based on population distribution and note the location of each of these customers. They are all interested in purchasing sports equipment/apparel from SportsmanSHIP, and each has \$54.56 dollars to spend ( $\approx$  Average purchase per customer per year for sports equipment = Revenue from sales for Dick's Sporting Goods/Dick's Sporting Goods' Customer Base)[9, 6]. So when they make their purchase, selecting from the warehouse that most appeals to them in terms of shipping time and taxation, their money contributes to that warehouse's profitability and therefore overall utility. We're giving that deciding power to the people; it's simple competitive economics.

With that, the utility of a single warehouse  $k$  is then computed to be

$$W_k(\mathbf{x}) = \$54.56 \times f_k - C_k$$

Where  $f$  is the number of customers which order from the warehouse each year and where  $C$  is the annual cost of maintaining the warehouse. (Surveying the current warehouse market, it is evident that \$100,000/year is a reasonable average price for a sizable warehouse. We can then set this price as our standard, treating it like our average budget per warehouse.) In summary, the utility of a warehouse is exactly the same as its individual profitability.

Keeping in mind that some warehouses will be more profitable than others based on their locations (due to factors such as surrounding population and shipping distances), we can take the utility of the entire arrangement to be the sum of all warehouses' utilities. We augment this with a small penalty from customers who are not within the desired one-day shipping range of *any* warehouse, in order to simulate factors such as bad publicity, and to penalize us for falling short of our universal coverage goal. Therefore, the net utility function becomes:



$$u(\mathbf{x}) = \sum_W W_k(\mathbf{x} - \$54.56d)$$

Where  $W$  is the set of warehouses in our arrangement and  $d$  is the number of dissenting customers who cannot enjoy one-day shipping. This final utility function then expresses the overall expected profit for our company; maximizing this via warehouse placement maximizes the productivity of our enterprise.

**This economic interpretation is extremely powerful.** Not only does it model try to optimize *profit, rather than coverage*, but it also *implicitly* considers all of the previous factors, including coverage and reduction of sales tax liability, through its profitability metric. This is due to the how we've set up this model: warehouses that reduce a customer's sales tax liability are selected over warehouses that don't; and warehouses that are placed in more populous regions where they can serve more customers profit more than those which aren't. This model accounts for the common-sense idea that placing a distribution center in the middle of an uninhabited desert is ill-advised; the model also accounts for the fact that fewer, more strategically placed distribution centers can target more people and improve profits. It follows then that it's no surprise that our economic interpretation strongly resembles the actual placement of Amazon fulfillment centers [7]. A comparison of this model's similarity to this is considered in the discussion.

The most promising part is that this economic interpretation provides a means to easily integrate more fine-grained factors into our warehouse-placement question, factors like property tax, zoning regulations, actual lease rates, climactic costs, and more, simply by converting each of these into a monetary value, a well-established practice in the world of business.

## 9 Results

### 9.1 Coverage

For the coverage simulation, the MCMC found a model with around 95.760% coverage. This model had a utility of 0.016, which was fairly useful, given there were about 57 warehouse locations. The maps of the warehouse locations as well as heat maps and region coverage maps can be seen below.

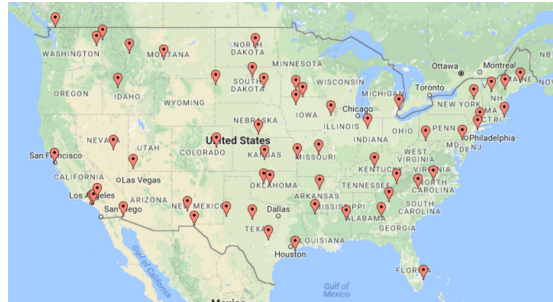


Figure 1: Location of ZIP Codes

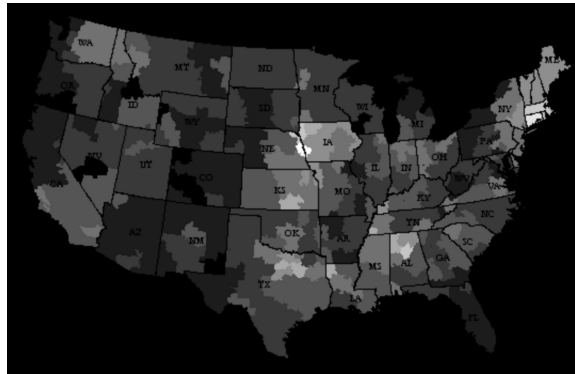


Figure 2: Heat Map (Brighter areas have more warehouse coverage)

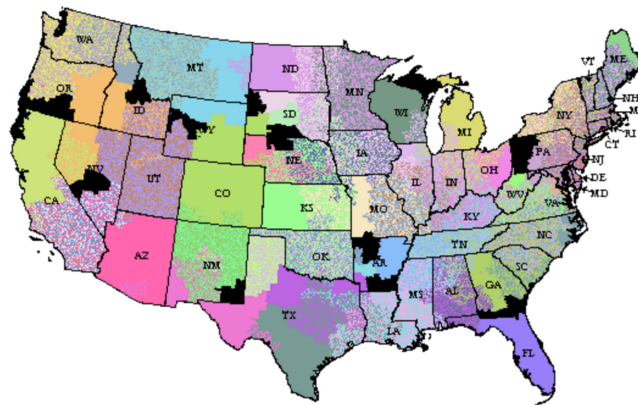


Figure 3: Regional Map Coverage (Areas with multiple colors signify overlapping regions)

This set of maps was created using a utility function that considered number of warehouses and percent coverage of the U.S. only. Overlap of warehouse range was not considered a detriment, so manual vetting was necessary in order to find a more optimal solution.

## 9.2 Vetting the Perfect Solution

As can be seen, this Markov Chain Monte Carlo algorithm is only able to produce a solution that covers a region 95% to 97% optimal. Therefore, in order to produce a perfect solution, the resulting map was manually vetted with a couple more regions. The resulting map is shown below.

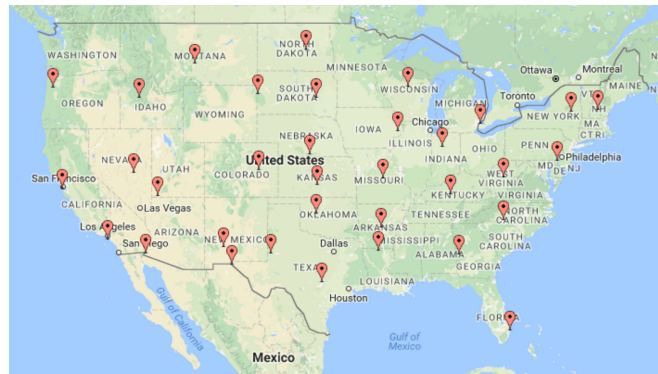


Figure 4: Location of ZIP Codes

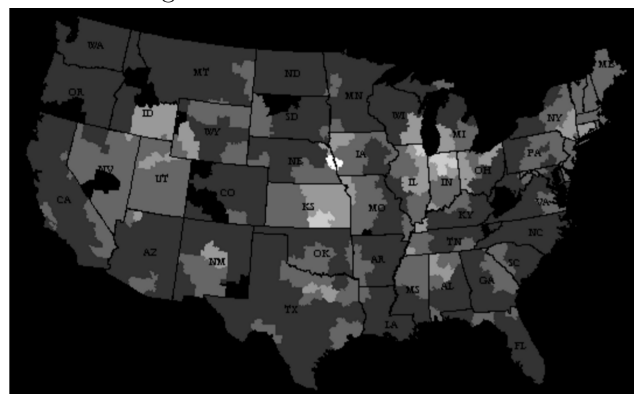


Figure 5: Heat Map (Brighter areas have more warehouse coverage)

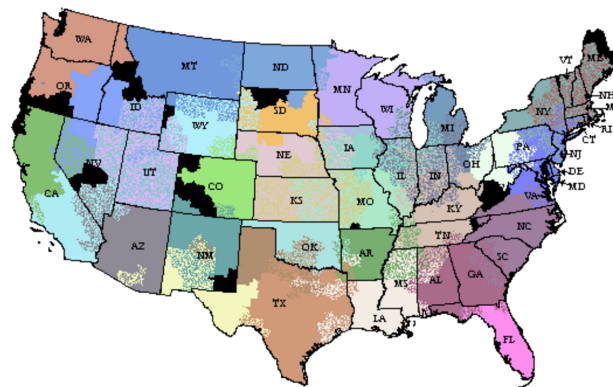


Figure 6: Regional Map Coverage (Areas with multiple colors signify overlapping regions)

In this solution, the vetting has taken out 23 of the warehouse that the MCMC found. While the percent covered is still the same, most of those areas are by definition non-serviceable areas such as national forests, parks, or deserts.

### 9.3 Incorporating Sales Tax

For the sales tax model, we were able to find an optimal solution with 87 warehouses and a coverage of around 90%. The maps are shown below.

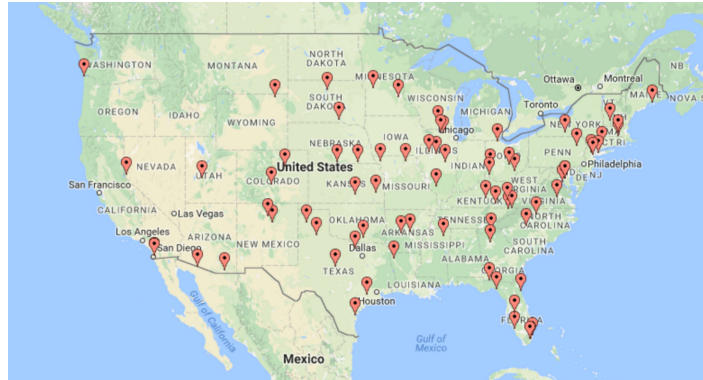


Figure 7: Location of ZIP Codes

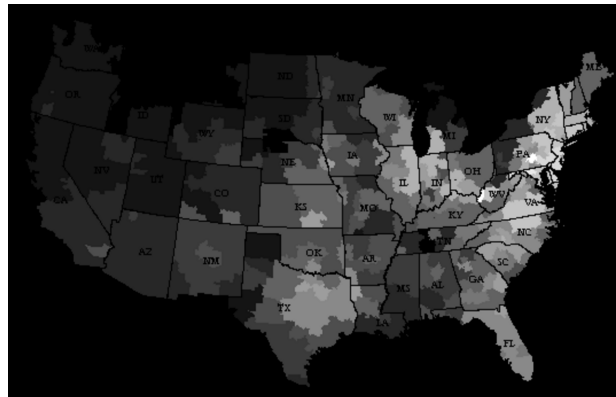


Figure 8: Heat Map (Brighter areas have more warehouse coverage)

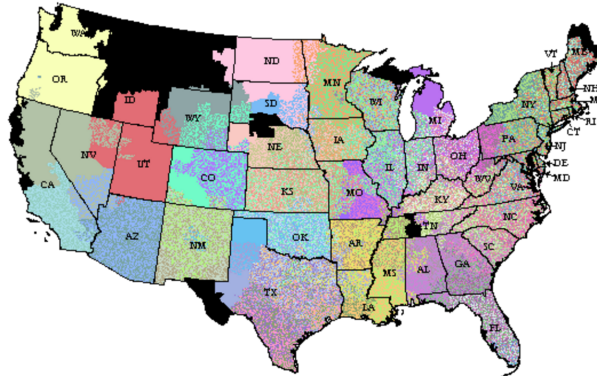


Figure 9: Regional Map Coverage (Areas with multiple colors signify overlapping regions)

This set of maps was created using a utility function that considered population. Population varies throughout states, so warehouse distribution was less even across the country. Warehouse density is much higher near the east coast and Midwest than it is near the west coast. There is still significant overlap of warehouse ranges, but manual vetting was not performed on this distribution. The effect that sales tax has on utility depends on the population affected by the sales tax. This means manual vetting could possibly reduce the optimality of this solution by a significant amount, so no manual vetting was performed.

#### 9.4 Incorporating Tax on Clothing and Apparel

For the clothing and sales tax model, we were able to find an optimal solution with once again, 57 warehouses and 95% coverage. This strongly mimicked our first model from before.

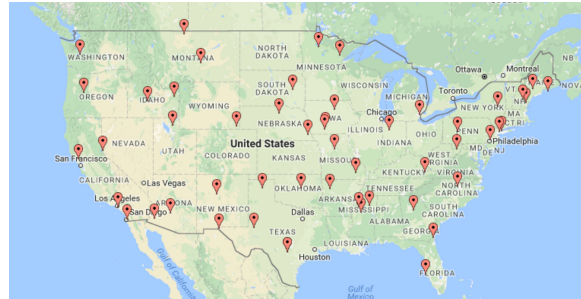


Figure 10: Location of Zip-Codes

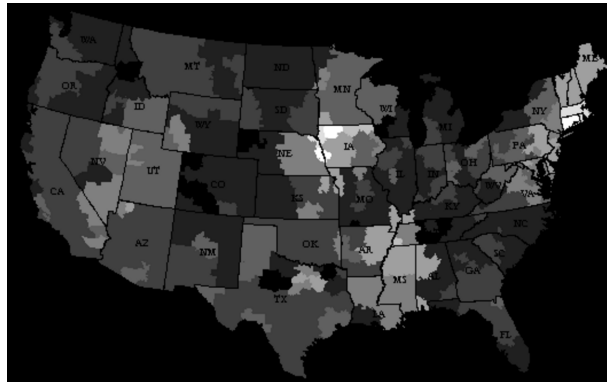


Figure 11: Heat Map (Brighter areas have more warehouse coverage)

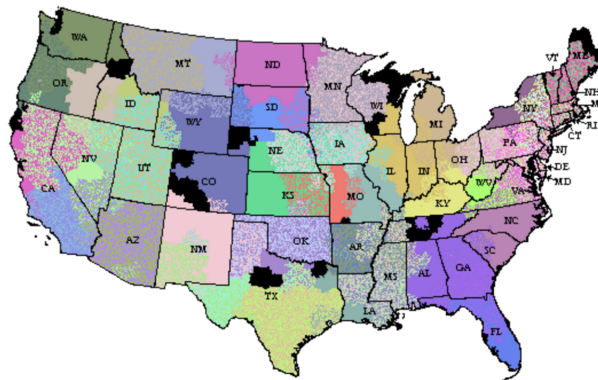


Figure 12: Regional Map Coverage (Areas with multiple colors signify overlapping regions)

The utility function for this set of maps was very similar to the function used to generate the previous set of maps, however it had added details regarding the special tax regulations on clothing and apparel. These special regulations changed the relative value of warehouses in certain states, and this is reflected in the new, more even distribution of warehouses. Percent coverage is very comparable to the first model's, however the areas left uncovered are slightly different. While not as extreme as in the second model, warehouse density is still greater in the east coast and Midwest than in the west coast.

## 10 Comprehensive Economic Model

For the comprehensive model, we were able to obtain an 83% coverage rate, with 67 warehouses and a \$14200765.594149007 profit margin for the company. This means that if the company were to distribute as shown below, it would ultimately earn approximately 14 million dollars in additional revenue.



Figure 13: Location of Zip-Codes

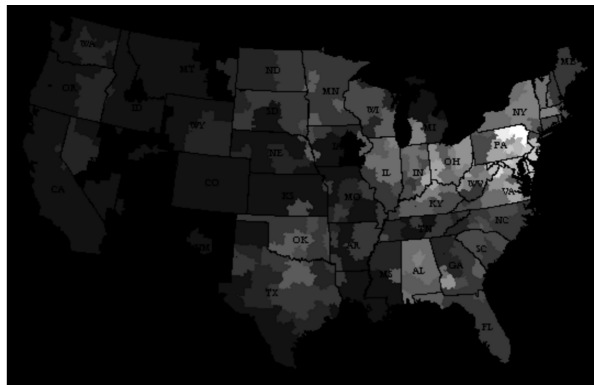


Figure 14: Heat Map (Brighter areas have more warehouse coverage)

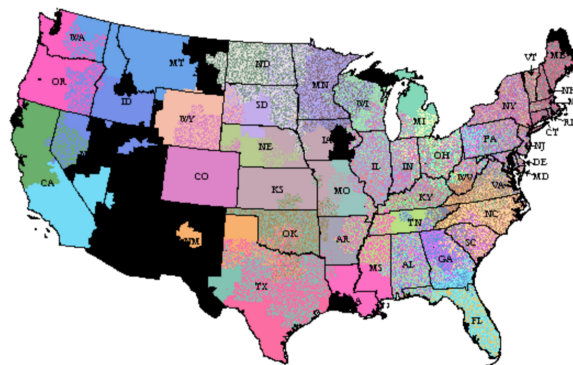


Figure 15: Regional Map Coverage (Areas with multiple colors signify overlapping regions)

The utility function for the comprehensive model differed greatly from the previous models. The focus of this function was company profit as opposed to coverage. The distribution of warehouses is extremely uneven throughout the country. The density gradient is even more extreme than in t

## 11 Discussion

### 11.1 Implications of Some Utility Function

Curiously, when comparing warehouse configurations across different utility functions, there was minimal difference in the resulting values. For example, when taking the configuration produced from the MCMC run geared towards optimizing coverage and feeding it into the utility function incorporating clothing and sales tax, the change was less than one percent. Thus, though the warehouse configurations ultimately generated from runs geared towards optimizing coverage, sales tax, and clothing tax differ, a particular configuration of warehouses seems to have relatively universal utility once its been optimized in one scheme.

### 11.2 Considering Coverage

The end product that our model produces is a distribution of warehouses across the United States, with each warehouse identified using the ZIP Code where it will be located. One of our primary goals for the distribution was to be able to provide one-day delivery to as much of the United States as possible, but to do so with as few warehouses as possible. Using our utility function of  $\frac{p^2}{s}$  where  $p$  is coverage and  $s$  is the number of warehouses, the theoretically most optimal result would be one warehouse covering 100% of the U.S., however this is not feasible. The limitation lies in the UPS maps our model draws data from, which show the locations to which a warehouse can deliver products in one day. As seen in Figure (1), our model found that the most optimal balance between percent coverage and number of warehouses was to cover roughly 95.76% of the United States using only 57 warehouses. While it is possible to cover more of the U.S. by adding one or two more warehouses, achieving nearly 100% coverage, this is not feasible in real life. Even with manual vetting, 100% was unable to be achieved, although we were able to significantly reduce the number of warehouse. Warehouses, factories, and locations of any sort are expensive to maintain, so even if they help the company reach more customers, they will still be losing money. Amazon's Prime Now currently advertises two hour shipping, but only to select locations in the U.S. If profit was not a concern, Amazon would easily open up more locations, providing two hour shipping to 100% of the U.S. and gaining more customers, but overall they would lose money because some locations would not be utilized frequently enough to break even. This is why our model covers most, but not all of the United States; because coverage is not the only important factor.

When we used our model to see the effects that state sales tax would have on our warehouse distribution, new factors were added into the utility function. Factors such as sales tax and the population affected by the sales tax had to be incorporated into our utility function, and were done for every warehouse. This is similar to a company considering the value of location. While it is important to be available to as many customers as possible, what if there are not many potential customers in an area? What if the sales tax is very high? These factors must be considered when a company chooses their locations. For example, Dick's Sporting Goods does not have any locations in Montana. This could be explained by the fact that Montana's population is relatively low compared to other U.S. states, and that it has no sales tax. Because of its low population, Montana has fewer potential customers and is not an optimal location for businesses. In addition, having no sales tax means that it would cost the same for a buyer in Montana to receive their order from an in state or out of state warehouse. Simply put, placing a warehouse in Montana and having complete coverage of it would be more detrimental than beneficial to a company. This

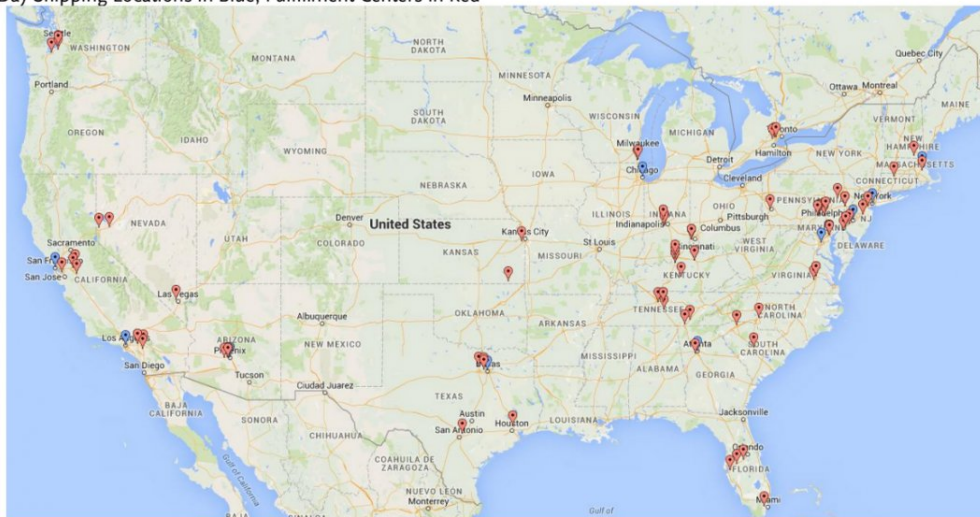
agrees with our model's prediction when sales tax was incorporated. According to Figure (9), our revised model's most optimized distribution of warehouses has roughly 90% coverage using 80 warehouses, but no warehouses in Montana, and very low coverage of Montana.

The comprehensive model resulted in a very different distribution of warehouses than the other three models, but it also valued the criteria of the utility function very differently. It used a much more economically focused criteria and aimed to generate the most profit for the company. This is why warehouses densities are much higher near metropolitan areas and larger cities. Not only are the populations higher there, but large companies and business districts are located there as well. This allows large companies to interact with other large companies and make deals, etc. for further economic benefit. Areas that have lower population or are more rural also have significantly less warehouses. They are not only not profitable, but also in unsuitable areas for large companies such as Amazon. Amazon's fulfillment centers, which are the equivalent of our warehouses, are located around the United States in a distribution very similar to our comprehensive model, as seen below.



Figure 16: Location of ZIP Codes for Comprehensive Model

**MAP OF FULFILLMENT CENTERS AND EXISTING SAME-DAY SHIPPING REGIONS**  
Same-Day Shipping Locations in Blue, Fulfillment Centers in Red



Source: Piper Jaffray, geobatch, MWVPL Logistics

Figure 17: Distribution of Amazon Fulfillment Centers in United States

Both maps show a preference for the northeast, near large cities and other large companies. The northwest and the Great Plains are largely devoid of warehouses. These locations are inconvenient from an economic standpoint, and utility standpoint. Large cities are sparse and natural obstructions such as the Rocky Mountains make this area extremely inconvenient to build a warehouse.



Overall, we see that coverage is not the most defining factor in choosing a warehouse distribution. The consideration of other factors can greatly affect the distribution. Furthermore, various criteria adjust the result in varying ways. The best, and most realistic distributions can only be modeled with a comprehensive model, with criteria and values that are more reflective of real companies.

### **11.3 To Profit or to Serve**

Interestingly, when considering the results from the MCMC runs focused primarily on optimizing coverage as compared to the runs focused on optimizing profits, discrepancies arise. Near-perfect coverage may be obtained in the former case, yet as the results indicate, perfect coverage may neither be cost-effective nor practical. Considering the resulting coverage provided by runs aimed at maximizing revenue, the areas missed tend to be fairly rural—places like national forests, Native American reservations, mountainous regions, areas with little population that would likely not mind receiving their recreational merchandise after two days instead of one, a reality balanced against the approximately 150 percent increase in profits calculated to occur with this configuration of warehouses as opposed to the configuration promoting perfect coverage.

### **11.4 Analysis of Methods**

#### **11.4.1 Strengths**

The strength of the model itself rests in its direct utilization of UPS's delivery-time maps. While approximation of coverage with mathematical functions may yield a model more readily available for traditional paper-and-pencil analysis, using the exact coverages as given by the maps ensures complete accuracy of delivery-times and leverages more computational methods of analysis like MCMC. Furthermore, Markov Chain Monte Carlo algorithms are a robust, proven method for sampling a parameter-space. In conjunction with the maps, the algorithm has provided us with results founded on precise data and accurate considerations with minimal approximations, facilitating equally accurate predictions and results.

#### **11.4.2 Weaknesses**

A weakness in our analysis to consider is the time it takes for an MCMC routine to successfully probe all important maxima in parameter space. Being an ergodic process, given enough time, the algorithm would eventually find every point of interest. But considering the unique time-restraints imposed upon the present analysis, the MCMC runs may not have discovered every possible optimal combination of warehouse locations. The computation time is not lessened by the parallelization of the algorithm. Because MCMC is recursive in nature, it is impossible to properly parallelize. The best that can be done is to run multiple chains of MCMC routines simultaneously and select the best results produced across all them, which was done in this analysis. Thus, while we recovered optimal configurations of warehouse locations, there is the possibility they are not the most optimal configuration possible.

#### **11.4.3 Looking Ahead**

In moving our analysis forward, the first logical step would be to let our MCMC algorithms run for longer periods of time. By allowing the routines to explore more fully the parameter space without the strict time-restraints limiting their progress, perhaps more optimal placements of warehouses

may be discovered. Another potential avenue for improvement would be using exact data from the United Parcel Service as opposed to their maps. At the moment, access to their data is restricted to queries sent directly to their website, severely restricting the speed of the MCMC algorithms as time passes between a request and response. As a result, we chose to extract data directly from downloaded maps, but to ensure the highest quality of accuracy possible when conducting the analysis, having direct access to the data itself would be ideal.

## 12 Conclusion

Overall, we used maps of UPS delivery-times to compile a model giving the number of days required for a shipment to reach a customer. We then designed an implementation of Markov Chain Monte Carlo algorithms aimed at optimizing placements of warehouses within this model, seeking to provide a maximum amount of coverage of one-day shipping times across all of the continental United States. Afterwards, the algorithm and associated utility functions were modified to take into account sales tax, tax on clothing, and other factors, producing slightly different results. Perhaps most interestingly, a final run of the MCMC seeking to maximize profits, though not required, led to the surprising conclusion that sacrificing coverage over negligible areas may result in significant profit increases, opening an alternative strategy for the company.

In the end, regardless if the configuration optimized for coverage or the one for profit is chosen, both seek to deliver products to customers at unprecedented rates. In an era of globalization, connectivity, and rapid response, such efforts to ensure one-day delivery times stand as milestones marking our collective knack for innovation, progress, and good SportsmanSHIP.

## References

- [1] About Amazon - Our Innovations - Amazon Prime. (2016). Amazon.com. Retrieved 13 November 2016, from <https://www.amazon.com/p/feature/zh395rdnqt6b8ea>
- [2] Amazon.com: Amazon Prime Now: Appstore for Android. (2016). Amazon.com. Retrieved 13 November 2016, from <https://www.amazon.com/Amazon-com-Amazon-Prime-Now/dp/B00R20MPW2>
- [3] Bittner, V. (2016). The 2010 US Census Population By Zip Code (Totally Free). The Splitwise Blog. Retrieved 13 November 2016, from <https://blog.splitwise.com/2013/09/18/the-2010-us-census-population-by-zip-code-totally-free/>
- [4] Complete List of United States Zip Codes | AggData. (2016). Aggdata.com. Retrieved 13 November 2016, from <https://www.aggdata.com/node/86>
- [5] DICK'S Sporting Goods' Hours, Locations & Stores. (2016). Dickssportinggoods.com. Retrieved 13 November 2016, from <http://www.dickssportinggoods.com/storeLocator/index.jsp>
- [6] Dick's Sporting Goods Inc.. (2016). Marketwatch.com. Retrieved 14 November 2016, from <http://www.marketwatch.com/investing/stock/dks/financials>
- [7] D'Onfro, J. (2016). Here are all of Amazon's warehouses in the US. Business Insider. Retrieved 14 November 2016, from <http://www.businessinsider.com/how-many-fulfillment-centers-does-amazon-have-in-the-us-2015-3>
- [8] Metropolis, N., & Ulam, S. (1949). "The Monte Carlo Method." *Journal of the American Statistical Association*, 44, 335-341. doi: 10.2307/2280232.
- [9] Physical Activity Council's annual study tracking sports, fitness, and recreation participation in the US. (2016) (1st ed.). Retrieved from <http://www.physicalactivitycouncil.com/pdfs/current.pdf>
- [10] Silver, H. (2016). How important is same-day delivery to shoppers?. *Internetretailer.com*. Retrieved 13 November 2016, from <https://www.internetretailer.com/commentary/2015/11/06/how-important-same-day-delivery-shoppers>
- [11] Software, P. (2016). Sporting Goods Retail Store Business Plan Sample - Market Analysis | Bplans. *Bplans.com*. Retrieved 13 November 2016.
- [12] Sports products US wholesale sales 2008-2015 | Statistic. (2016). Statista. Retrieved 13 November 2016, from <https://www.statista.com/statistics/240946/sports-products-industry-wholesale-sales-in-the-us/>
- [13] Sports apparel wholesale sales US 2008-2015 | Statistic. (2016). Statista. Retrieved 13 November 2016, from <https://www.statista.com/statistics/240985/athletic-apparel-wholesale-sales-in-the-us/>

## 13 Appendix

Below are the specific ZIP Codes of each optimized warehouse configuration specified in the paper.

### Most optimal coverage configuration:

'88543', '92329', '04040', '84780', '73078', '52751', '79608', '68947', '99022', '38755', '50546', '65105', '87935', '25283', '83804', '71277', '05083', '27722', '59802', '48236', '72136', '04496', '57783', '64752', '37753', '78732', '50465', '67568', '94132', '33429', '11208', '77551', '90703', '30518', '83637', '79360', '89314', '10560', '58486', '28075', '90847', '98264', '57312', '42722', '46942', '19884', '85366', '12134', '56120', '73772', '57455', '80830', '59036', '15451', '39323', '31804', '02576'

### Vetted coverage configuration:

'88543', '84780', '52751', '68947', '65105', '87935', '71277', '05083', '48236', '57783', '78732', '67568', '94132', '33429', '83637', '79360', '89314', '58486', '28075', '90847', '98264', '57312', '42722', '46942', '19884', '85366', '12134', '73772', '80830', '59036', '15451', '31804', '54401', '72202'

### Most optimal sales tax configuration:

'41537', '68971', '23205', '56226', '32080', '45227', '78416', '61105', '76246', '20540', '33199', '62268', '20317', '38869', '88429', '66748', '61436', '18644', '61008', '89712', '37621', '07204', '71909', '64455', '33812', '71171', '92182', '01653', '28240', '79257', '87567', '80136', '11020', '67277', '55337', '45406', '81223', '32348', '40935', '68352', '39829', '43076', '84644', '43604', '22195', '87538', '07927', '05141', '01570', '37664', '28222', '61839', '06804', '45711', '82731', '33990', '33155', '30366', '33020', '63501', '27427', '79010', '97121', '72117', '61613', '85341', '76834', '53575', '04629', '57571', '40442', '07932', '92119', '14812', '57420', '30560', '23120', '85670', '77434', '73460'

### Most optimal clothing tax-exemption configuration (with starting value taken from most optimal coverage configuration):

'88556', '92149', '04228', '84310', '72852', '50830', '79007', '68714', '98497', '38958', '50149', '65349', '85347', '24942', '83441', '71412', '04612', '26852', '59525', '48180', '72557', '03810', '56667', '63952', '38850', '78291', '50633', '68329', '94928', '34229', '10156', '78291', '90714', '30070', '82053', '79761', '87104', '11054', '57353', '27560', '89705', '97413', '57483', '41517', '47925', '19136', '85048', '12010', '56250', '74834', '55717', '83327', '59072', '15712', '38923', '31569', '03878'

### Most optimal clothing tax-exemption configuration (from random starting configuration):

'25422', '26268', '43502', '13471', '41063', '49076', '13489', '38701', '25820', '94005', '62240', '33853', '47834', '30119', '39112', '59468', '15747', '28738', '65463', '70819', '21405', '75963', '73071', '46962', '32777', '15254', '77302', '40078', '34674', '90638', '33946', '48725', '29082', '84743', '35021', '80442', '47116', '02917', '44699', '67480', '54101', '88028', '61864', '89121', '27910', '50661', '45699', '37361', '70607', '75630', '62709', '16825', '75852', '20839', '97050', '93604', '33305', '58077', '32541', '47590', '20825', '71662', '25669', '11204', '68773'

### Configuration optimized for profit:

'13132', '23307', '22202', '57022', '55750', '61820', '13210', '63010', '27259', '32710', '41237', '79248', '98610', '49780', '32417', '14549', '56064', '78144', '23603', '02632', '43317', '35587', '59832', '32578', '47970', '74563', '33601', '72747', '29836', '41544', '33309', '53537', '57448', '12040', '95988', '43117', '92805', '13209', '17404', '74036', '38762', '36429', '61803', '80423', '25162', '79414', '15936', '23307',

'17703', '66413', '45760', '29542', '57770', '16822', '57476', '83645', '53535', '25839', '11941', '69147',  
'17836', '76437', '30018', '04992'