Learning Carrier Choice Models for Load Pricing in Digital Freight Platforms

by

Alexandra Li

S.B. in Computer Science and Engineering Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by:	Alexandra Li Department of Electrical Engineering and Computer Science Jan 15, 2024
Certified by:	Saurabh Amin Professor of Civil and Environmental Engineering, Thesis Supervisor
Accepted by:	Katrina LaCurts Chair, Master of Engineering Thesis Committee

Learning Carrier Choice Models for Load Pricing in Digital Freight Platforms

by

Alexandra Li

Submitted to the Department of Electrical Engineering and Computer Science
on Jan 15, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

With the expansion of digital commerce and growth of the economy, the freight transportation scene has adapted to reflect such changes. Digital freight platforms, acting an intermediary between shippers and carriers, have gained traction to modernize the process and leverage technology to improve efficiency and increase the ease-of-use for all parties involved. Through their role in setting prices and presenting loads, these platforms can reduce the negative environmental impact of freight while simultaneously increasing the efficiency of carriers and satisfying the needs of shippers. The key challenge that these digital freight platforms face is understanding how carriers strategically select an action on the platform, which is difficult to capture despite having large amounts of data because naive estimation methods on historical data produce unrealistic results for different pricing methods.

This thesis addresses this challenge by developing a simulation to evaluate the practicality of these estimates and iteratively revise the parameters based on constraints until they produce desirable results. In our research, we model the behavior through which carriers select a load to accept or reject with a 2-way latent class multinomial logit model. We tune the parameters of this model through a feedback loop where we perform a maximum likelihood estimate on the data to obtain model parameters, evaluate these parameters in the simulation, and use the results to perform a re-estimation to eventually obtain parameters that are both representative of the data and produce the expected results.

We use this system to evaluate optimized pricing and load presentation methods. We experiment with bundling, or grouping a sequence of loads together to reduce the overhead time carriers spend finding suitable loads and to produce routes with less CO2 emissions. We solve for a mixed-integer linear program that maximizes the total utility of bundles proposed by the platform to generate few and non-overlapping bundles. We develop a dynamic programming based pricing method to generate carrier and time specific prices for bundles. We evaluate these methods in our model and analyze the effects of such methods on carrier interactions and behavior. Although these methods do not yet show a substantial decrease in freight carbon emissions, we have laid the groundwork for modeling this complex system and hope that future work can be done to reduce the negative environmental that the freight transportation sector leaves on this planet.

Thesis supervisor: Saurabh Amin
Title: Professor of Civil and Environmental Engineering

Acknowledgments

First, I would like to thank Saurabh for all his support and mentorship over the last two years. His experience and expertise in the field and inquisitive perspective towards approaching problems have been invaluable in this project. Under his guidance, I've grown as a researcher, communicator, and person, and I thank him for always pointing us in the right direction. His dedication to helping students reach their full potential shines through, and it has motivated me to make the most of the M.Eng experience.

I would like to thank my labmates, Maxime and Aron, for their welcoming and encouraging attitudes. I am truly grateful for all they've taught me and they're openness in discussing ideas and trading opinions. I'm lucky to have had the chance to collaborate with them both.

I would also like to thank Mazan Danaf, a senior economist at Uber Freight, for the time and energy he has put into assisting this project. His wealth of industry knowledge and willingness to share data insights and utilizations have significantly enhanced the depth and quality of our work.

These past years and this last semester were made possible by my roommates and friends, especially those who stayed in Boston with me. I will be forever grateful for your company and the joy you bring.

Finally, I would like to express my gratitude to my family for their unwavering support throughout this journey.

This accomplishment would not have been possible without any them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2021, freight movement accounted for 23% of transportation greenhouse gas emissions in the US, with the transportation sector taking up the largest portion of all emissions [1]. As the supply chain continues to expand with the online-nature of the economy, it becomes apparent that the current methods of operations will have serious negative effects on the environment and propel climate change forward unless large steps are taken to make the process more efficient and sustainable.

The freight industry has historically been relatively old-fashioned despite being such an integral part of the economy. Typically, a company that wishes to ship products, also known as a shipper, shares the information on the delivery it hopes to get fulfilled and can either list it on a load board, where many other companies can also list deliveries, or communicate the delivery with a freight broker. The freight broker is essentially a middleman between the shippers and the carriers, or the truck drivers who fulfill the deliveries [11]. To take the burden of finding loads to deliver off of drivers, carriers will typically work with brokers who either find deliveries on load boards, already have a small set of deliveries from the companies they work with, or get deliveries through cold calling. In this system, there are many tedious steps, including the paperwork involved in fulfilling a delivery on both the shipper and carrier side, the phone-calling done by the broker, and the decentralization of information.

From this, a few startups have become invested in building digital freight platforms in hopes of moving this entire system online, increasing its efficiency, and improving the job quality for all parties involved. The interactions between of shippers and carriers with a digital freight platform are illustrated in Figure 1.1. The benefits of a digital freight network are very quickly realized: by incorporating technology into this relatively archaic system, the overall shipping costs are lowered while maintaining the same level of compensation for the carriers because the need for a broker and the associated brokerage fees are removed. In addition, carriers earn more and drive more efficient routes since deliveries are listed in a centralized manner, as waiting times are reduced and carriers can better align their deliveries with more options. By bringing payments online, carriers can get paid instantaneously. Overall, the potential to improve the freight industry is huge, as having an online platform allows providers to easily track supply chain data and gain insights into where resources are being wasted and where unnecessary harm is done through emissions.

To optimize routing and driving assignment with emissions in mind, one consideration

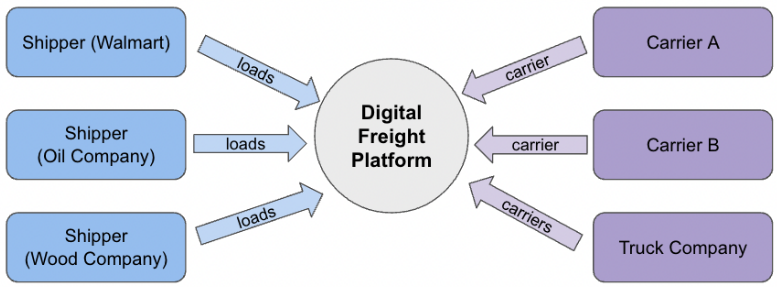


Figure 1.1: Illustration of shippers and carriers interacting with the digital freight platform

is to reduce the empty miles driven, which is when a truck head is pulling a trailer without a load. This can happen if a carrier makes a delivery from city A to city B, but then, due to non-ideal planning and coordination, has to then make a delivery from city C to city D. In this case, the carrier has to drive the *empty miles* from city B to city C without getting paid for this distance and resulting in CO2 emissions. Our research will work towards incorporating empty miles into the evaluation of the total costs of the platform to analyze the system from the perspective of emissions.

In general, digital freight platforms have the ability to affect the choices of carriers and deliveries through the presentation of loads and the prices they set. We will explore how both of these concepts can be used to create a more efficient system.

The main problem we address with our research is how to capture the strategy through which carriers select an action based on the load characteristics, which is crucial for these platforms to understand and model because the pricing and load presentation methods must be backed by an understanding of how the carriers would interact with these changes. This problem is challenging, despite the large amounts of historical data that these platforms have, because creating models from estimation methods on the data does not give realistic results over different pricing methods.

In our research, we create a simulation for the interaction of shippers, carriers, and load bookings on the digital freight platform to provide these large-scale platforms with ways to improve their efficiency and reduce emissions while further reduce costs to the platform through bundling and more refined pricing methods.

A challenge in creating a realistic simulation is the absence of real-world data, as this is all sensitive data that cannot be revealed between competitors. We partner with a large digital freight platform to get rough data distributions and slightly modified data to build a realistic model. It is worth noting that for privacy reasons, the data is also incomplete, as we do not have information regarding the price at which loads were accepted or the location and status of the carriers that accepted the loads. We also use the data tracked by the platforms to perform analysis on how supply matches demand and where the biggest improvements can be made.

We set up the simulation by using the rough carrier and load start locations and times provided by our collaborator. We experimented with different methods of generating bundles to present to carriers, ranging from generating all feasible bundles to a more optimized method. We also compared different pricing methods, including static pricing where bundles from one city to another were always the same price, dynamic pricing where the prices of bundles change over time as the pickup time of the delivery approaches, and heterogeneous pricing where prices are different per carrier. We modeled the process of a carrier selecting a bundle with a latent class choice model and proceeded until the time frame of the simulation was completed.

We will make use of the concept of *bundles*, introduced by multiple freight platforms, which is the idea of combining multiple loads into a bundle and presenting them to the carriers as one option. This process makes it easier for carriers to deliver more loads quickly by reducing wait time and increasing efficiency by calculating more optimal routes for the drivers [2].

We hope that this research can leave a positive impact on freight platforms, carriers, and shippers, as the improvements we analyze and propose result in wins for all parties. The route optimization is of interest to platforms because it results in more efficient drivers that will lower both costs and time spent on deliveries, and our work in bundling will hopefully better plan routes for carriers as well as give them more confidence for booking loads in advance.

1.1 Related Work

Much work has been done in regard to the fleet routing optimization problem. We are interested in creating a simulation to model the interactions of the freight network and basing the model on real-world data. We will do a comprehensive analysis of the literature regarding each of these areas.

In addition, we are modeling a decentralized fleet, as digital freight platforms tend to work with many independent carriers and shippers. This differs from previous work because most of the route-planning research assumes that the platform or entity is able to assign carriers to loads, so it does not handle or capture the complex process in which the carriers select loads with their preferences.

1.1.1 Simulating the Routing Problem

Schroeder et al. describe a multi-agent freight transport model as a micro-simulation by separating the decision-making of shippers, transport service providers (TSP), and carriers [10]. The transport service providers in their case are the ones that connect the transport chain. The goals of the shipper agents are to minimize total logistic costs, TSP agents setup transport chains and commission carriers to loads, and the carriers have different capacities and respond to routing and pricing.

Their goals with this experiment were to explore how the simulation is affected by carriers with different load capacities, the introduction of tolls, and certain regulations on cities' rules. The traffic agents use a custom utility function, which we also implement, to evaluate the economic successes of their options, and the cost to carriers is determined by the distance and time costs incurred through delivering loads and some additional fixed costs. In their situation, the TSPs calculate their cost as the fees they pay to carriers minus the opportunity costs lost by not delivering a load within its given time window. This is similar to the idea of a digital freight platform that acts as a connector between shippers and carriers, as the goal of the platform is to maximize the difference between the cost a shipper will pay the platform and the price at which a load will accept a load while also fulfilling as many loads as possible. However, in their simulation model, the TSP agents are more limited, as they cannot influence the price of loads, whereas a large part of the decision making and influence behind a digital freight platform is the pricing of loads.

In addition, beyond the limitations on the TSPs, their research falls short in both the small scale of the simulation and their lack of analysis on how the pricing and routing factors affect the interactions between loads and carriers. As they explain in their conclusion, the focus of their research was on defining and implementing different agent types rather than on their behavior and response to different factors of the simulation. We plan to build on this by integrating digital freight platforms with much more fine-grained control over pricing and load proposals, and doing an in-depth analysis of how the carriers and shippers react to different methods.

The previous work done by Abed et al. also investigates simulating freight flows through an agent-based method, and it covers road, rail, and inland waterways [3]. They aim to capture the chains of actions and decisions made by various actors in the supply chain, from producer to carrier to consumer as well as hitting the time constraints, thus resulting in the agent-based modeling. This paper describes a simulation methodology that they plan to implement, so the details are rather vague but they estimate the cost of production firms, which are the entities that produce the goods to be transported. Our research is rooted in similar motivations of modeling how the different agents interact with the platform, but more in terms of analyzing what power the digital freight platform has to improve the overall supply chain experience.

1.1.2 Industry-based Data

For the mixed fleet problem presented by Schneider, Stenger, and Goek, the data used to evaluate their models are all generated instances, containing fleets that all originate from a single depot, which is unrealistic in our case of decentralized fleets, and synthetic data [9]

In addition, Schroeder et al. only work with micro-instances of 4 carriers working to serve the loads of 4 shippers, so in this case the data is very small scale [10].

For Abed et al., they are able to base their experiments on a significant amount of real-world data from firms based in Belgium, zoning dividers, network data containing travel times along roads, and vehicle information, but they are not able to base their model on the specific data for carriers accepting loads in terms of prices and time scales at which these agents perform these actions [3].

This lack of real-world data is not surprising among work done in this field, as it is quite difficult to gather a set of large, representative data of shipper and carrier constraints and locations. One explanation is that data only becomes somewhat centralized with the introduction of digital freight companies such as Uber. These companies have made progress in modernizing the trucking industry by building up a platform for shippers and carriers.

So far, these companies, and more specifically Uber Freight, have focused on improving efficiency through the idea of bundling, where multiple deliveries are combined together into one bundle that is presented to the carrier in an effort to minimize the distance between one delivery's drop off and the next's pickup to reduce empty miles [5]. The idea of forming bundles in routing relates closely to the optimal fleet routing problem we are solving because they follow the same objective functions and measures of cost, and bundling is a method of presenting delivery options to drivers. In the classic routing problem knowing deliveries and time constraints, given vehicle locations, we can deduce bundles from the path taken by each vehicle in the optimal solution. Alternatively, we could calculate the cost of bundling loads and solve the routing problem from the perspective of bundles.

1.1.3 Simulation-based Optimization

The iterative method we propose where we tune parameters through both mathematical optimization methods and simulation results is inspired by the field of simulation-based optimization, which describes the methodology in which a system is mathematically modeled and then evaluated in a computer-based simulation to gather more information about its behavior. The inputs are varied in different iterations and the effects on the simulation results are observed, and each iteration moves closer to the optimal solution [6].

These techniques are used in the context of urban transportation by Osorio and Bierlaire where they attempt to get a comprehensive understanding of traffic dynamics in urban networks [7]. A challenge in their research is accounting for all factors including vehicle performance, traveler decision-making, and supply network details. While performance and supply network information can be concluded mostly from historical data, the traveler decision-making part of the problem is much more nuanced and similar to the problem we face with modeling carrier choices in the digit freight network. The paper is novel because it proposes a metamodel that combines information from the traffic simulation tool with the analytical network model, which we take inspiration from in using both a mathematical estimate of the data and simulation results to improve our parameters. They show in their research that they are able to generate near optimal results in a manner that is very computationally efficient [7].

Overall, while there has been work done with modeling the freight network, there is still space to improve in both data and solution-finding. The usage of real-world data comes with the challenge of scale, as previous models would have to run for an unrealistic amount of time before coming up with a near-optimal solution. These are both issues we hope to address in our work.

1.2 Key Contributions

In our research, we provide a relatively robust way to model the interactions of shippers and carriers on a digital freight platform. We partner with a large digital freight platform to get a more comprehensive look at the movement of freight in the Texas Triangle, specifically for the loads, carriers, and bookings. The main contributions of our research are as follows:

- The development of an iterative approach for tuning the parameters by performing an estimation on the data, evaluating the parameters in the simulation, and re-estimating by fixing certain values or adding parameters to converge towards satisfying our desired set of constraints. This process is simplified and shown in Figure 1.2.

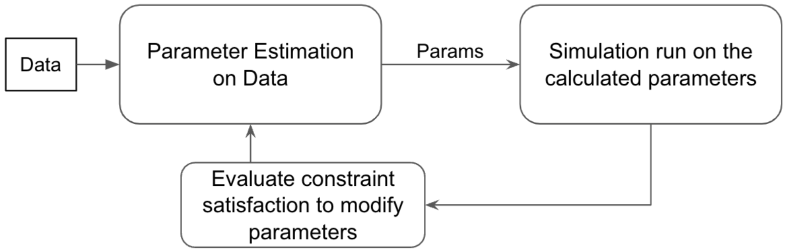


Figure 1.2: Iterative process through which the parameter estimation is performed on the data, the simulation is run with the new parameters, and the results are evaluated to inform the estimation.

- The finding of a set of parameters with fixed price sensitivity that produces results that are both representative of the data and have the desired constraint satisfaction over pricing methods to model how carriers select loads on the platform.

- The evaluation of various pricing methods, namely dynamic pricing, and load presentation methods, through bundling, on real-world data in the Texas Triangle. With the simulation, we experiment with how these methods affect the interactions of carriers with loads on the platform and how the platform can use those insights to improve the efficiency of the long-haul freight network.

Chapter 2

Data

First, we define a few terms related to the large-scale freight network. Shippers refer to the companies who own the goods being shipped (e.g., Walmart), while carriers refer to the companies or individuals that transport the goods on behalf of the shippers. A load is a single shipment from an original location, often a warehouse of the shipper, to the destination, which can be an individual store or separate warehouse. A shipping lane is a path between 2 markets or cities frequently traveled by carriers.

Our goal is to create a simulation that models how a digital freight platform engages with shippers and carriers. For this platform, shippers agree to list individual loads on the platform. When listing a load, the shipper provides the pickup and dropoff location of a load, the pickup and dropoff time windows (usually a range of hours to provide for flexibility), the pickup and dropoff facility ID, and the price that the shipper is willing to pay for the load to be transported. We assume that the price the shipper pays the platform is fixed per each load.

The platform gathers the loads posted by shippers into bundles to present to the carriers and generates a price to list each of these bundles. Bundles are formed by grouping one or more loads together as an ordered package that a carrier can select, and we also consider single loads as a bundle when referring to the carrier options.

On the other side, carriers appear on the platform either in the form of individual carriers using the app or in the form of trucking companies that can control or host multiple carriers. A carrier can log on to the app, view bundles presented by different shippers on the platform, see the prices of how much they would be paid for transporting a load or bundle of loads, and choose to select a bundle to deliver, which would register as a *booking* and they would receive further instructions on how to carry out the load, or they can leave the app and not book a bundle at that moment in time. Each time a carrier opens the app, this action is registered as a *session* for the carrier. Within one session, every bundle that a carrier views and clicks into for more details is an *impression*.

These digital freight platforms make a profit from the marginal difference between the price at which a carrier accepts a bundle and the price at which a shipper lists the bundle to the platform. The platforms typically are in control of setting the prices for bundles on the platform and can use this to make certain loads appear more appealing and to increase or decrease their profit margins. They incur a cost for any loads not delivered under the time constraints given by the shippers, since for a missed load, the platform has to take on

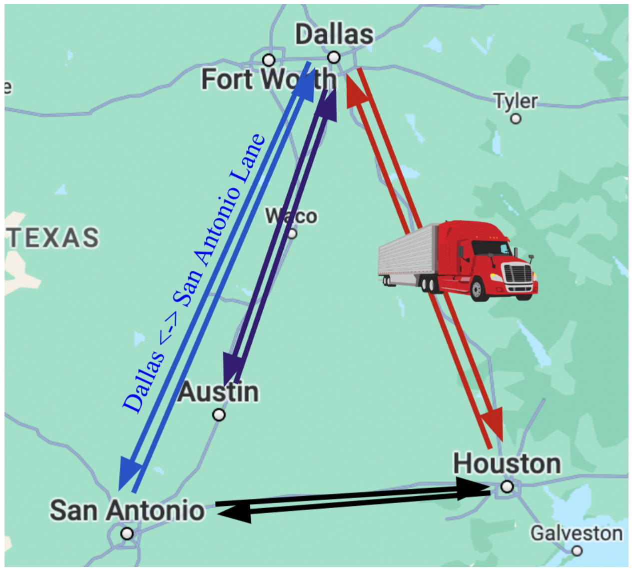


Figure 2.1: Visualization of the Texas Triangle, which is the network we use for the simulation. The arrows represent shipping lanes, such as the labeled Dallas to San Antonio lane and reverse. The truck logo illustrates how carriers drive between markets on lanes to pickup loads and deliver loads.

the operational cost of rescheduling and the carrier cost of the rescheduled load, and there is also a negative impact to the shipper's relationship with the platform. Their systems can be evaluated based on their environmental impact in terms of *empty miles*, or miles where a trucker is driving an empty trailer with no shipments, typically on the way to pick up another load.

2.1 Freight Network Data

For this project, we have partnered with a large-scale freight platform to receive industry data regarding carrier behavior and load information. The data specifically applies to the Texas Triangle network, which is made up of the cities of Houston, San Antonio, Austin, Dallas, and Fort Worth. We consider each city as a separate market within the larger network. Because of their proximity, we group Forth Worth and Dallas together. The Texas Triangle is an ideal market to examine because there is an abundance of freight movement within the network, both inter- and intra-city. The triangle is connected by three main freeways, and it is a dense area where the transport and distribution of raw materials, such as oil, coal, minerals, and construction materials, are an integral part of the economy. The Texas triangle, along with terms like lanes, carriers, and markets, is illustrated in Figure 2.1.

For the data, we have loads and session information. For each load, we have the following statistics:

1. Pickup and dropoff market/city

2. Start and end time for the pickup and dropoff time windows

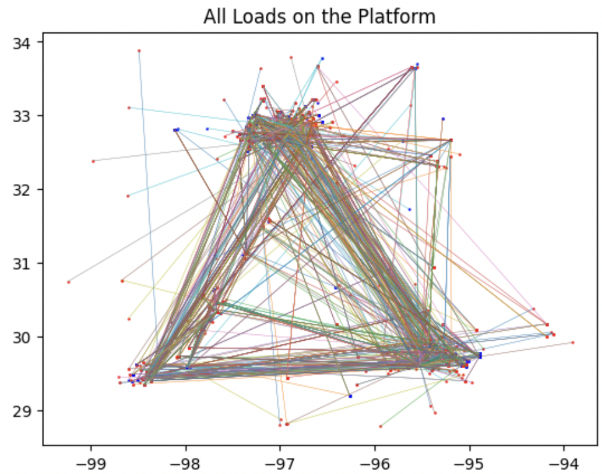


Figure 2.2: All loads from the given data are plotted on a longitude and latitude map, where blue points indicate a pickup and red points indicate a dropoff.

3. Time of creation of the load on the platform

4. Distance in miles for the load

5. Pickup and dropoff facility IDs

6. Pickup and dropoff longitude and latitude

We are also given a list of sessions, where each session contains the session start time and location in latitude and longitude. In addition, we are given distributions for the number of impressions per session and the distribution of sessions per user.

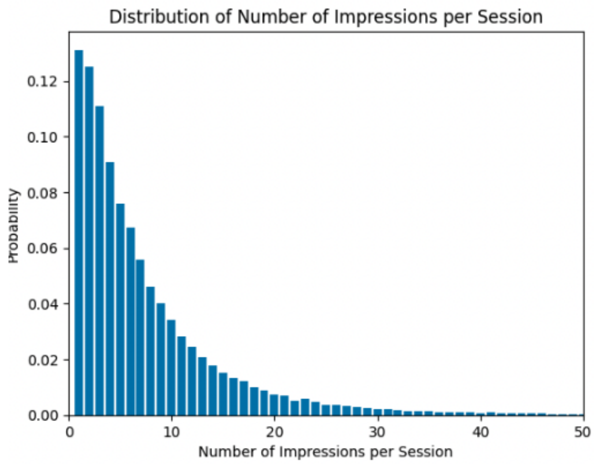
We also have access to a basic price distribution of loads per lane from our collaboration with a large freight platform. We use the average price per lane, or from the starting market to the ending market, as the baseline industry price.

A key feature to note about the data is that although we have a large number of data points available in terms of loads registered by shippers and logins to the app from carriers, we do not have access to specifically which carriers in which sessions accepted what load. Instead, based on the distributions of sessions, load views, and loads, we can understand that a lot of carriers are opening the app and not accepting loads at all, so the data is skewed and we must capture this accordingly in our simulation.

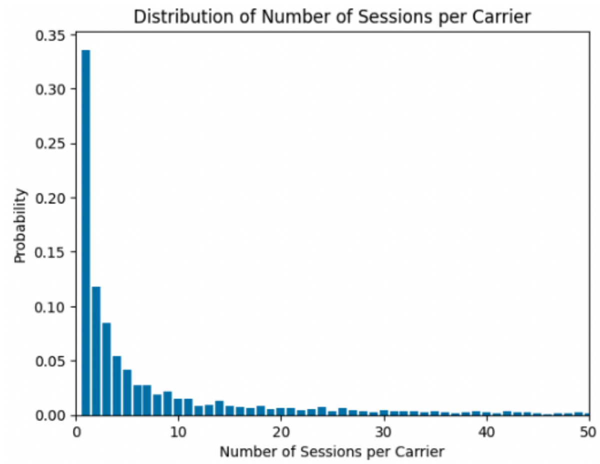
2.1.1 Pre-processing

To make the given data usable for our purposes in the simulation, we perform some pre-processing. Because the data given to us by the major freight platform provider is masked to protect the sensitive information contained in the locations and specific load details, we add in some randomness by perturbing the longitude and latitude of load and carrier locations slightly.

In addition, we filter the given loads to only those that are reasonably satisfiable by getting rid of unrealistic or extreme loads. This involves removing loads where the end of



(a) Number of impressions per session, where an impression is a bundle viewed in the app by one carrier.



(b) Number of sessions per carrier, where a session is one instance of a carrier logging onto the app.

Figure 2.3: Distribution of carrier actions for both number of logins and number of bundles viewed per login.

the dropoff window is before the start of the pickup window and loads in which the pickup and dropoff time are separated by more than 2 days. The resulting loads are mapped on the Texas Triangle in Figure 2.2. This includes a total of 2395 loads.

2.1.2 Carrier Analysis

We analyze the numbers given for the observed frequencies of impressions per session and sessions per carrier to determine the following distributions.

The number of impressions per session, or essentially the number of loads that a carrier will view in one interaction with the app, follows the distribution in Figure 2.3a, which demonstrates how the number of sessions that carriers consider roughly follows an exponential decay, and based on the data, has a pretty high tail end that is cut off in the figure for visibility purposes. We eventually will discuss how we use this distribution, given by data from a major freight service provider, to determine how many bundles to present to the carrier.

The number of sessions per carrier, or the number of times a carrier logs onto the app, can be seen in Figure 2.3b. Based on the distribution, we see that many carriers only log onto the app once, and then there is a sharp drop before it exponentially decays and also has a pretty high tail. This distribution is important for understanding the ratio of supply and demand between loads and carriers and how this affects the prices of loads.

The carrier starting locations are randomly sampled from a distribution given by a major freight service provider. For each carrier, we then calculate its starting market by finding the closest city based on the city centers of our 4 markets (Austin, Dallas, Houston, and San Antonio), as the carriers can be in between cities while completing another shipment when looking for their next. The carrier location distribution is shown in Figure 2.4.

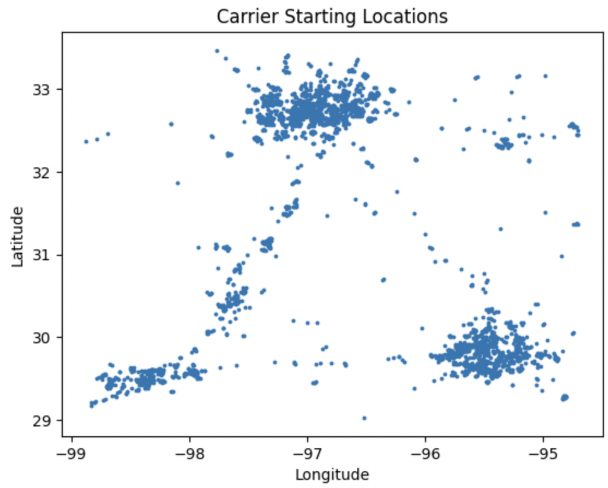


Figure 2.4: Distribution of locations of carriers when they log onto the app.

We can also analyze the starting locations of sessions per market. This analysis is not super exact as many carriers log onto the app and start a session while they are still between cities, possibly while they are still delivering their previous load or returning home after finishing their last delivery. As a result, we map session starting locations to market cities by grouping the location with the city whose center point is the closest in distance. With this method, we see the distribution of session start locations described in Table 2.1.

Table 2.1: Repartition of Sessions Starting Locations

San Antonio	Austin	Dallas	Houston
10.2%	7.98%	47.7%	34.1%

2.1.3 Loads Analysis

We examine the distribution of loads between the different markets across dropoffs and pickups. The percentage of loads that start and end in each market are detailed in Table 2.2. Based on this distribution, we can see that there is an imbalance between the pickups and dropoffs of loads, which may result in the prices of loads that end in a certain market, say Austin, being higher valued to incentivize a carrier to make what would otherwise be a less-desirable journey, as it may be harder to get a load that starts in Austin after the carrier has completed the first load due to the imbalance in the market trends.

It is notable that the distribution of sessions follows roughly the same trends as the pickup locations of loads when comparing Table 2.1 and Table 2.2, with Dallas having both the most carriers logging on and the most loads starting, followed by Houston, San Antonio,

Table 2.2: Pickup and Dropoff Load Percentages by Market

	San Antonio	Austin	Dallas	Houston
pickup	11.5%	5.59%	44.3%	38.6%
dropoff	12.6%	8.72%	39.3%	39.3%

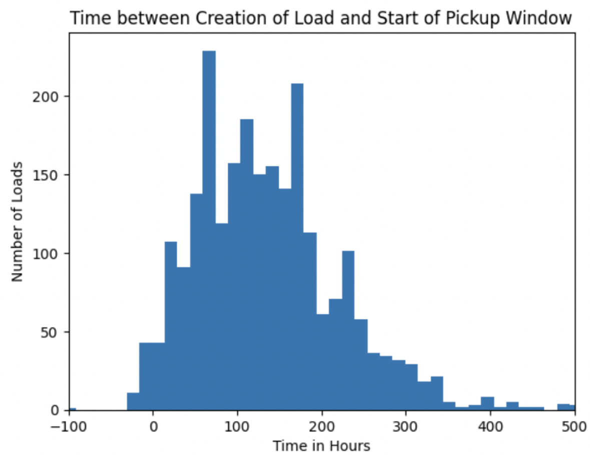


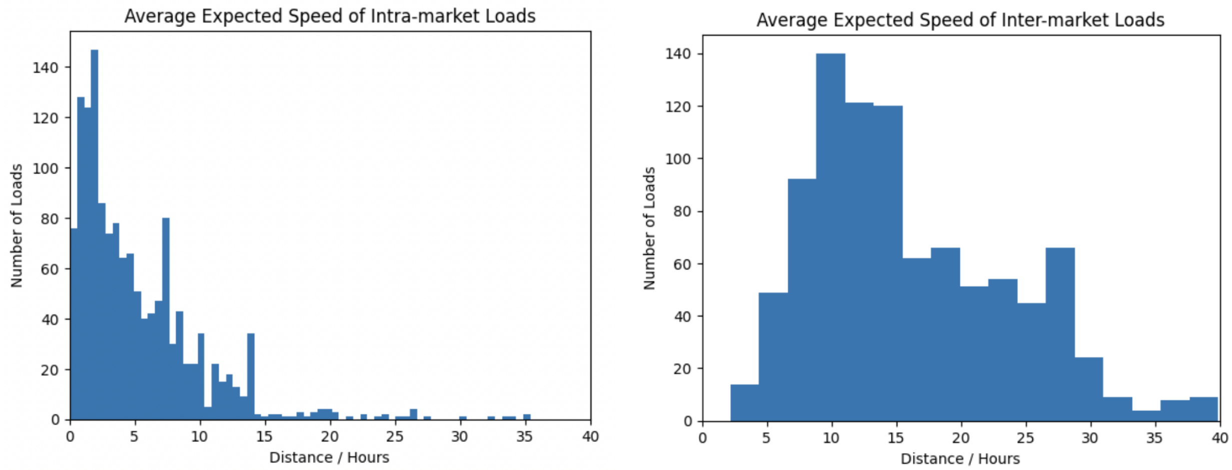
Figure 2.5: Distribution of the time between creation and the start of the pickup window for loads.

then Austin. The relative percentages that each city sees of loads and carriers are roughly the same, which illustrates how the market of supply reflects demand.

We also take a closer look at the difference between inter- and intra-market loads, where inter-market loads are between cities and intra-market loads are within cities. We observe that approximately 60% of the loads are intra-market, and the remaining 40% are inter-market. It is worth noting that intra-market loads can also be relatively long, as the distance between opposite ends of a city can be rather large in Texas. To account for this, we also do an analysis based on the distance in miles of the loads. We find that inter-market loads are an average distance of 187 miles while intra-market loads are an average of 26 miles between pickup and dropoff. When we analyze by distance, we find that 55.6% of loads are less than 50 miles in distance, while the remaining 44.4% are longer than 50 miles.

This separation is important in analysis because we expect the price per mile of shorter loads to be much higher than the price per mile of longer-haul loads, as there is some base price of loads, meaning that the relationship between price and distance is likely to be non-linear.

In the context of the digital freight platform, the relationship between the shipper and the platform is also important because the amount of lead time given between when the shipper posts a load onto the platform and the start time of the load affects how long carriers have to view and potentially book the load. We can calculate the average time between the creation of the load and the start of the pickup window to be 137.3 hours or roughly 5.7 days. This



(a) Distribution of average speed on intra-market loads

(b) Distribution of the average speed on inter-market loads

Figure 2.6: Distribution of average speed for intra- and inter-market loads, where the average speed is calculated as the distance between pickup and dropoff for the load divided by the time from the start of the load pickup window to the end of the load dropoff window.

value roughly stays the same between intra- and inter-market loads. The distribution of the value can be seen in Figure 2.5.

The interpretation of this distribution is that loads can generally be viewed by carriers about a week before they actually begin, although this ranges from loads being created during the pickup window to loads being created even 20 days in advance. This is impactful because in order to bundle loads together, the platform must know of the existence of both loads before presenting the bundle to the carrier, which means both loads must have been created. Longer periods between creation time and load start time on average can result in better formed bundles that are more optimized for carrier preferences and reduce empty miles.

There has also been research done by Uber Freight that has shown that more relaxed time windows for loads result in a more efficient platform and better results for both shippers and carriers [5]. We investigate this in the data by looking at the time windows relative to the distance of a load, or essentially the average speed for loads, by finding the distance from the pickup to the dropoff location of the load and dividing this by the full time-span of the load, or the time between the start of the pickup window and the end of the dropoff window.

$$s = \frac{d}{t} = \frac{d}{\text{end}_{\text{dropoff}} - \text{start}_{\text{pickup}}}$$

We calculate the mean speed of intra-market loads to be 5.86 miles per hour and the mean speed of inter-market loads to be 17.50 miles per hour. The intra-market load speeds follow the distribution in Figure 2.6(a) while the inter-market speeds follow the distribution in Figure 2.6(b).

We can notice that the speed is typically much lower for intra-market loads, which makes sense as the route to deliver these loads potentially would traverse on local roads and have to deal with more traffic lights or other traffic-dependent conditions. This is in contrast to inter-market loads that can have a much higher speed expectation because the driving route is most likely primarily on highways or freeways, which have much higher speed limits than the local roads of intra-market deliveries. However, we can see for both cases that the range is pretty wide, with some loads having much stricter time windows, while others in the long tail of the distribution have very relaxed speed expectations.

Chapter 3

Simulation Design

From the perspective of a digital freight platform, important aspects of the freight network include how loads are presented to carriers and how loads are priced. For load presentation, the most straightforward method results in presenting the loads on their own, and presenting all available loads to carriers. However, there has been recent work done to explore the benefit of bundles, or grouping loads together in an effort to reduce empty miles driven. By booking loads one at a time, carriers have to spend significant amounts of time planning ahead to secure their next load, and booking a load while delivering the previous one raises some logistical challenges. As a result, carriers can end up accepting loads that are relatively far away in both time and distance, resulting in wasted time and empty miles. The idea behind bundling is that the platform can offer loads to carriers in bundles or groups of 2 or more, simplifying the process of booking, better optimizing for empty miles, maximizing truck utilization, and minimizing off-duty driving. In Figure 3.1, we can see how suggesting loads to a carrier in a pair can allow them to make the drive back to their original location profitable, as they would not drive back with an empty trailer. In our research, using algorithms to generate bundles results in a more efficient method of finding better bundles compared to the manual process done by carriers.

We now proceed to describe the general steps of the simulation. We first determine the location and time of carriers that log onto the app in sessions. Then, given a set of loads that represent deliveries, we generate the set of bundles to present to carriers. For each carrier that logs onto the app in time order, we gather the bundles that are valid for this carrier, satisfying the conditions that the carrier's appearance time is within the first load's pickup window and that the carrier is in the same market as the first load. We generate the prices for all of these bundles and use this price, along with the carrier and bundle information, to calculate the utility of each bundle. This utility is then used with the choice model presented in Section 4.1 to select an action for the carrier, whether that be booking a bundle or leaving the app. The process is repeated until all loads are handled or the simulation ends, meaning there are no more carriers. Overall, the process can be described in Figure 3.3.

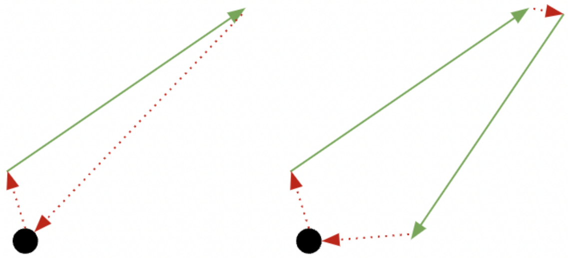


Figure 3.1: Illustration of bundling affects: the black dot is the carrier's start and end location, the solid green lines are loads, and the red dotted lines are empty miles.

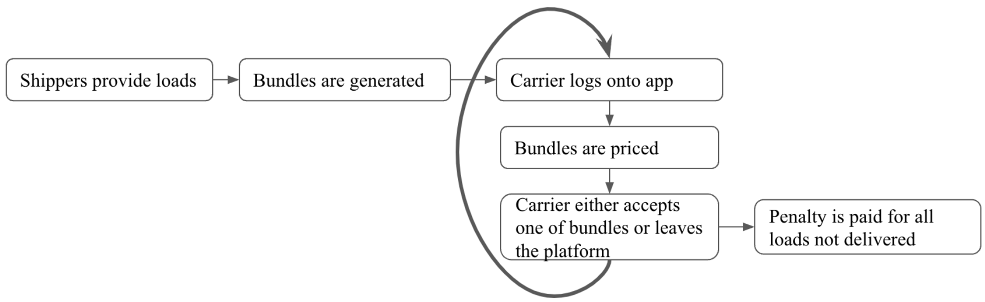


Figure 3.2: Timeline of the simulation

3.1 Key Assumptions

In setting up the simulation, we make a few key assumptions and simplifications. For the carrier, we assume that the average speed is 40 miles per hour. For the loads, we make the following assumptions, as this data was not provided to us:

1. The maximum allowed idle time for a bundle, or the time between dropping off a load and picking up the second one in a bundle, is 10 hours. We make this assumption because having long time windows between loads in a bundle means that if this bundle were to be accepted, the carrier would be forced to wait at a location before the next load can be started, resulting in wasted time and reduced efficiency that counteracts the benefits of bundles. Different platforms or companies may enforce different lengths on the exact value of the maximum idle time in a bundle, so we make a ballpark estimate. In addition, this also increases the efficiency of the simulation because it results in less bundles to consider.

2. The load and unload time of a shipment is assumed to be 2 hours. This is based on industry standards of the time to empty and fill a trailer.

3.2 Setup

To setup the simulation, we define some basic initial guidelines. We maintain a date for the carrier and load start times, as well as a date for the simulation end time. The carrier start

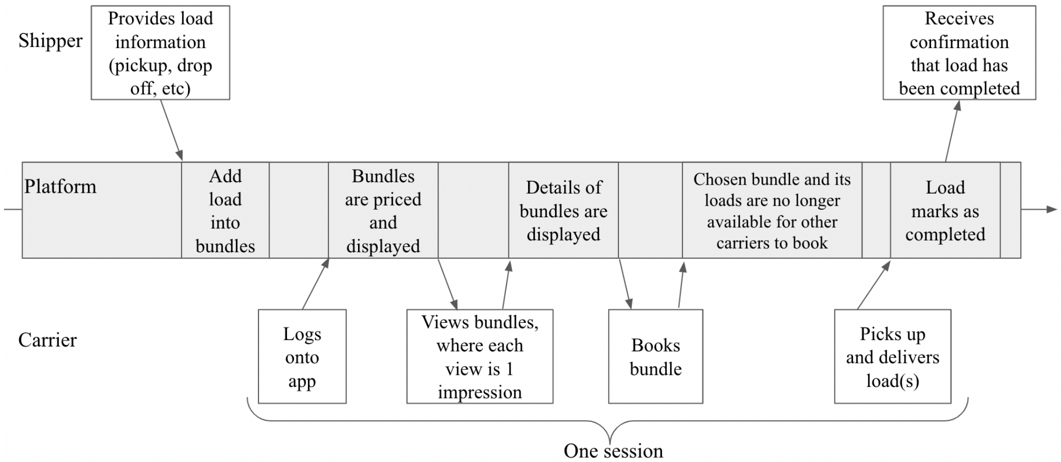


Figure 3.3: Timeline of the interactions between the carrier, shipper, and platform

time represents when carriers begin logging onto the app to view loads, and the load start time represents the pickup time window start of the earliest load. This is meant to model how carriers book loads often a week in advance, but can continue to book as the pickup window for the load gets closer.

At the end of the simulation, when there are no more carriers logging on, the platform incurs a penalty for any load that is not delivered. This represents either a loss of faith from the shipper in the platform, the refunding of a shipper, or allocating a dedicated driver to fulfill the load from the platform's own fleet [8].

We model the simulation in an event-based manner, specifically within the Texas triangle network, which includes Dallas, Austin, Houston, and San Antonio.

3.3 Bundling and Pricing

In this section, we detail how bundles are generated given a set of loads, and how bundles are priced given its characteristics. The bundles are generated once at the start of the simulation, and the bundles are re-priced whenever a carrier logs onto the app to view the bundle options.

We implement an industry-based bundling method that generates all possible bundles of size 1 or 2 loads that satisfy a set of industry constraints.

Once we have generated the bundles, we integrate them into the simulation. At every time stamp of a carrier logging onto the app and beginning a session of viewing bundles, we determine which bundles are feasible for this carrier based on the pickup time windows and the creation time of the bundles. We ensure that the carrier's log on time is before the end of the pickup window of the first load, the carrier and the first load begin in the same market, and the carrier's log on time is after all loads in the bundle have been created on the platform. Given these feasible bundles, we calculate the utility of these bundles and use the utility to select somewhere from 1-8 bundles to display based on the data of impressions per session discussed in Section 2.1.2. Then, a bundle or the reject option is selected for the given carrier according to the choice model. This process is described more broadly in Figure 3.3.

3.3.1 Industry-based Bundling Method

In the industry-based bundling method, we create bundles of size 2 and present these to carriers alongside the standalone loads. We iterate over all combinations of 2 loads, and if the following conditions are satisfied, loads i and j are bundled together and possibly presented to carriers.

1. Same market constraint: the destination of load i must be the same as the origin of load j.

2. Maximum deadhead constraint: the deadhead distance between them must be less than the expected deadheading distance for the unbundled loads

3. Minimum time between loads: the driver must have enough time to pick up load j after dropping off load i:

$$A_{j_1} - A_{i_0} \geq T_i + t_{ij} + \text{loadTime} + \text{unloadTime}$$

   where $[A_{i_0}, A_{i_1}]$ and $[A_{j_0}, A_{j_1}]$ are the [start, end] pickup time windows of loads i and j, T_i is the driving time required to deliver load i, and t_{ij} is the deadhead driving time from dropping off load i to picking up load j.

4. Maximum time between loads: there must not be too much idle time between dropping off load i and picking up load j:

$$A_{j_0} - A_{i_1} \leq T_i + t_{ij} + \text{loadTime} + \text{unloadTime} + \text{maxIdleTime}$$

3.3.2 Baseline Industry Pricing

The baseline method for pricing in our simulation is using the average of the range given by the major freight service provider for each lane. Given a starting and an ending market of a load, we can instantaneously get the average of the range and use that as the pricing of the load. In this simple scenario, to get the price of a bundle containing two loads, we add the prices of each individual load together to get the price of the bundle.

We also run the simulation two times the average price and three times the average price to test the sensitivity of parameters and bundling algorithms.

However, the disadvantages of this method are that the prices of the loads are not dependent on the distance of the load within the lane, as each market is pretty large for the range they cover. In addition, the pricing is static over time and between carriers, which means that as the pickup time for the load gets closer, the price does not change to the carrier. This is not realistic or ideal because to incentivize a carrier to accept a load as the deadline approaches, platforms typically increase the price of a load as the time to pickup decreases.

Chapter 4

Sequential Estimation and Evaluation Approach

In Section 3, we have illustrated how we model the appearance of loads for the shipper-platform interaction and how carriers log onto the app to book loads on the carrier-platform interaction. In Section 3.3.1 and 3.3.2, we have delved into how bundles are generated on the platform and how loads are priced. To create an accurate simulation that can be used to test other algorithms without having access to all the bookings data, we must tune the parameters and weights of the carrier choice model of how carriers select loads to book depending on the carrier's and the load's characteristics.

4.1 Carrier Choice Model

First, we gather all valid bundles for a given carrier. For a carrier that is not in the process of completing a bundle when they log onto the app, a valid bundle is defined by the following conditions: the end time of the first pickup falls after the time at which the carrier logs on, all loads in the bundle have not yet been booked by any carrier, all loads in the bundle have been created by the time of log in, and the first pickup location is in the same market as the carrier. If the carrier is in the process of completing a bundle, then the last condition changes such that the first pickup location of the bundle being considered must be in the same market as the drop off location of the carrier's in progress bundle. In addition, the time of first pickup must be after the time of drop off of the current bundle. In general, these are the conditions for a bundle to be compatible with a carrier and its potential current delivery.

We calculate the utility of each bundle as a weighted combination of the bundle size, price, distance, location, and carrier information in a multinomial logit model. Including the basic of the parameters, the utility of a bundle i for a carrier c is calculated as:

$$U_i = \beta_d d_i + \beta_p p_i + \beta_t t_i + \beta_{\texttt{bundle}} + \epsilon_{i,c}$$

where β_d is the distance sensitivity, β_p is the price sensitivity, β_t is the time-to-pickup sensitivity, $\beta_{\texttt{bundle}}$ is the booking constant and parameter from the size of the bundle, and $\epsilon_{i,c}$ is the term that encapsulates the effect or the origin and destination markets of the bundle and location of the carrier. Also, d_i represents the distance traveled by delivering the loads in bundle i, p_i is the price of bundle i, and t_i is the time in hours between when the carrier views the load and the pickup start time for the load. In more complex parameter sets, there are more that we are not listing in this equation.

Once the utility is calculated for each valid bundle for a given carrier, we use these utilities to randomly select an action for the carrier. We first select the number of loads to display to the carrier by sampling from the distribution of impressions per carrier, or the number of loads a carrier will typically view before making a decision based on the data. Carriers arriving onto the platform can either select one of the displayed bundles or leave the platform without selecting any bundle, where we model it as each load can either be accepted or rejected with some probability set by the model. We use the softmax activation function to convert utilities into probabilities for each function. The steps to choose an action for a carrier are described in Algorithm 1.

Algorithm 1 Selecting a Bundle for a given Carrier

$U \leftarrow []$
for `bundle` $i \in$ `all bundles` **do**
  $U_i \leftarrow \beta_d d_i + \beta_p p_i + \beta_t t_i + \epsilon_{i,c}$
end for
$U' \leftarrow U.sort(desc)[:n]$ ▷ Get top n utilities to display
for $i \in$ `all actions: bundle and reject` **do**
  if `random`$(0,1) > p_{reject}$ **then**
    $U'_0 \leftarrow 0$ ▷ U'_0 is the utility of the rejection case
    $p_i = \frac{e^{U'_i}}{e^{U'_0} + \sum_j e^{U'_j}}$
    $b \leftarrow$ `random weighted choice`$([i, 0], p_i)$
    if $b = i$ **then**
      Return i ▷ If the current bundle is not chosen, continue to the next bundle.
    end if
  end if
end for

We chose to adopt this model where each bundle is considered one at a time, and while considering each bundle, there is a probability that the carrier will reject the bundle that is calculated independently of the bundle's utility because there are factors that we are unable to be account for that could cause a carrier to immediately dismiss a bundle. For example, if the bundle does not end up where the carrier wants to go, or the start of the bundle pickup time is too far away, among other reasons, the carrier could rule it out for reasons unrelated to the utility calculation parameters. This probability of direct rejection is accounted for in the 2-way latent class model for each load. The overall choice process for one carrier is detailed in Figure 4.1.

When a bundle is chosen by the carrier, we remove it from consideration for further carriers and from price calculations.

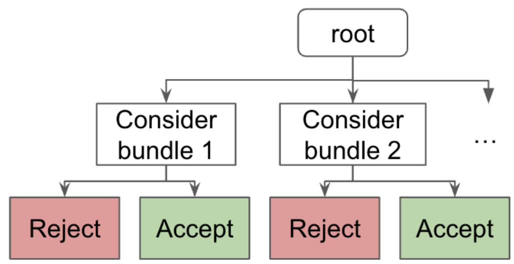


Figure 4.1: Illustration of the choice model of the carriers

4.2 Goals of the Simulation

By analyzing the results of this simulation, we hope to first benchmark the base pricing and bundling schemes and then evaluate the different existing and newly proposed bundling and pricing algorithms to determine what effect they have on the system.

First, we must determine that our baseline model is realistic in order to further build on it and treat it as a stable and reliable benchmark. We evaluate the baseline on the following conditions and whether they match the values given by our collaboration with the digital freight platform:

1. The percentage of loads that are accepted at an extremely low price (eg $0) is very low

2. The percentage of loads accepted and the average acceptance probability of a load at a high price of double or triple the average price given by Uber Freight is relatively high

3. The price-per-lane of accepted loads fits into the ranges given by Uber Freight

After implementing the different bundling and pricing algorithms, we evaluate the parameters of the system by examining the following metrics:

1. The efficiency of the platform in terms of reducing the number of empty miles driven and improving the negative environmental impact of the long-haul freight.

2. The costs to the platform, computed as the sum of the prices of accepted loads and the penalty incurred for all non-accepted loads.

We hope to use these clearly outlined metrics to determine quantitative improvements between the different parameters and algorithms and better understand how the large-scale freight network reacts to these different parameters. With this simulation, we hope to provide a well-investigated and data-driven proposal to large freight companies of how to best reduce their negative carbon emissions while still maintaining their desired profit.

4.3 Simulation Execution

The simulation is set up over a 2 week period, where carriers start logging on to the platform 1 week before the first load's pickup time window. Once the loads begin, the carriers still continue to log onto the app uniformly for the next week. This is meant to model how carriers book loads often a week in advance, but can continue to book as the pickup window for the load gets closer. For our case specifically, we have carriers beginning on 9/5/2022, loads beginning on 9/12/2022, and the simulation ends on 9/20/2022.

4.3.1 Defining Metrics

The measured acceptance probability from the simulation is the average probability calculated by the softmax function applied to the utilities over all valid bundles, not including the option to reject all bundles. We anticipate this probability to be relatively low, as there are many more impressions of loads of carriers than there are loads to accept, meaning that many carriers open the app and view loads but do not book for a variety of reasons, ranging from the bundle not aligning with their preferred route or none of the routes have satisfactory conditions or prices.

We calculate the cost to the platform as the sum of the price of all the loads when booked, as well as the summed penalties that the platform must pay for all unfulfilled loads. From a business perspective for the platforms, the most optimal method would minimize the cost to the platform, as this means loads are priced at a lower cost and the platform can gain more of a profit between what the shippers pay the platform and what platforms pay the carriers. However, the platforms also incur a cost when they are unable to fulfill a load, as they either have to pay the shipper back, which can result in lost trust, or they have to use their own resources of a platform owned fleet if this exists to fulfill the load. As a result, the platform also wants to reduce the number of loads that are not accepted by the end of the simulation, which we model with the penalty term.

4.4 Parameter Tuning Results

To ensure our simulation is realistic and matches the expected percentage of loads accepted at an extremely low price and at a high price, we iterate over different approaches to modeling the choice of carriers. We perform this iterative exploration through the partnership with our industry collaborator. Because we do not have the exact data of which loads are booked when by which carrier, we rely on our collaborator to perform our requested abstraction of parameters as they have full access to the booking data. The different sets of parameters are obtained by specifying which variables we want to include in the estimation and performing a maximum likelihood estimation (MLE) to get a set of parameters that fits the observed data.

To evaluate each set of parameters, we examine the probability of acceptance and the average number of accepted loads over a range of different pricing methods. For both the average probability of acceptance and the number of accepted loads, we compare methods where all loads are priced at zero, the average industry price, 2 times the average industry price, and 3 times the average industry price to get the whole range. We look at these two graphs specifically because they illustrate how sensitive the model is to different pricing mechanisms and how many carriers will still book loads in each case.

We execute these tests on a randomly selected subset of 100 loads that varies across runs, as the simulation is somewhat slow on larger instances of data and we found 100 loads to be representative of larger datasets as well.

In the following section, we walk through the different parameters in the timeline through which we explored them to follow the logical process of developing each model.

4.4.1 Initial Baseline Model

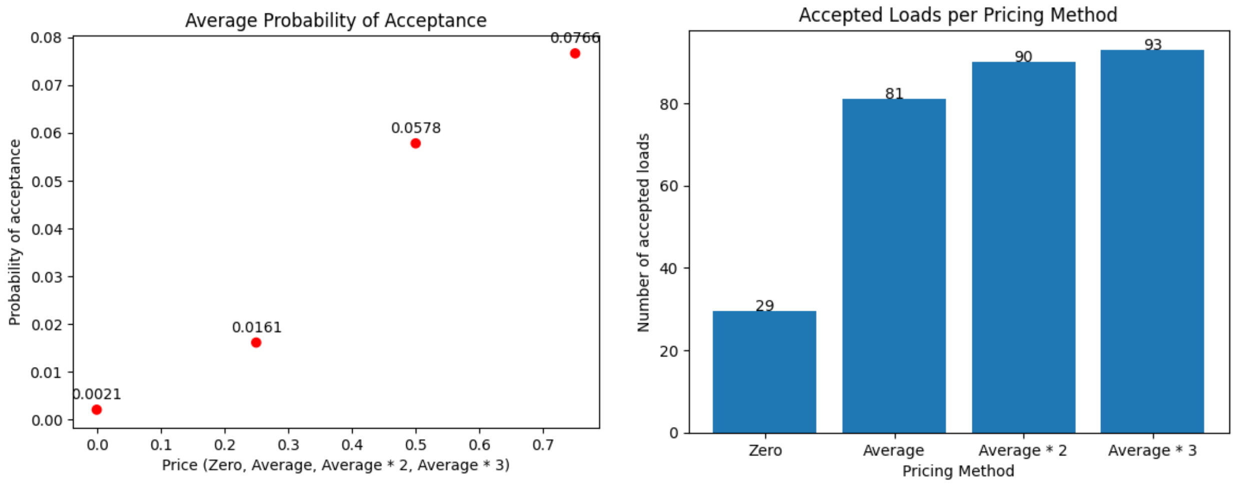
We first estimated the parameters of the model with the most straightforward set of parameters. We included indicator variables for the location of the carrier, the starting location of the bundle, and the ending location of the bundle per each market. We also included a booking constant and a constant negative utility for bundles of size 2 and size 3+ because carriers tend to lean away from bundles and the booking constant aims to capture the existence of competition between bundles and platforms. Then for the value dependent parameters, we include the total distance in the bundle, the deadhead distance driven to deliver the bundle (which includes the distance from carrier starting to first load pickup and any distance between the dropoff of one load to the pickup to the next), the price, and the time until the load pickup starts.

Figure 4.1 details the parameters used for the weights in calculating the utility. For terminology, "load from" refers to the origin market of the load, while "load to" refers to the destination market of the load, and "carrier in" refers to the market of the carrier when they log on for their session. For these 3 sets of values, all parameters are relative to Austin, which has a weight of 0 in these cases. In addition, the distance, price, and hours are all in hundreds of units.

To interpret and better understand these parameters, we see that based on this initial set of parameters, carriers seem to give more importance to the total distance in a bundle than the deadhead they must drive to get to the bundle. The weights of both of these parameters are negative, which illustrates that longer bundles have a relatively negative utility that must be offset by a higher price for the load to appear attractive.

When we run these carrier choice model parameters on a subset of the data containing 100 loads with 150 carriers, we obtain the results illustrated in Figure 4.2.

In Figure 4.2(a), we see that overall, the acceptance probabilities are pretty low. With very high prices at three times the average, we see that the average probability of acceptance per impression viewed by the carrier is 7.67%. However, this number is reasonable in context because, with these model parameters, the probability of rejection for each impression is 92%. This means that at most, the probability of acceptance of each load on average is 8%. We see that the achieved value of 7.67% is very high in comparison to the max, and in fact, with these prices, 93% of loads are accepted. We also see that at the price of zero, the acceptance probability is smaller, at .2%. However, even at the price of zero, 29% of loads are booked as seen in Figure 4.2(b), which is more than a third of the loads booked at the industry average price. This is unsatisfactory, as having very few loads accepted at price zero is one of our simulation constraints.



(a) (b)

Figure 4.2: Probability of acceptance and number of accepted loads for the initial carrier choice model over 100 loads.

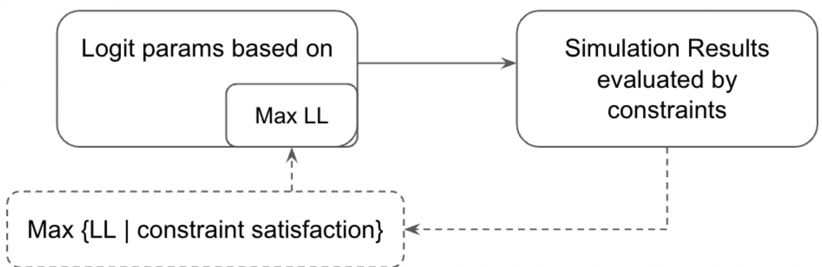


Figure 4.3: Feedback loop illustrating how the logit model params are estimated with maximum log-likelihood (LL), and the results they generate in the simulation are used to constrain the estimation of parameters.

4.4.2 Tuning Feedback Loop

From this initial attempt at a straightforward estimation of parameters, we recognized that we faced significant challenges in achieving parameters that both fit the given data and produce the desired simulation results that are expectations based on data. As a result, we define our problem as a constrained objective parameter search, as we must take into account the results of the simulation in fine-tuning and constraining the parameter estimates.

This feedback loop is described in Figure 4.3. As illustrated, the carrier choice logit model parameters are estimated by performing a maximum log-likelihood estimate on the booking data that captures interactions between carriers, loads, and the platform. Then, we test these parameters by running the simulation and producing results with respect to the constraints. We then use the results' constraint satisfaction in conjunction with the maximum likelihood estimate to generate new logit-model parameters to iteratively approach a more realistic model. We follow this approach in the next sections.

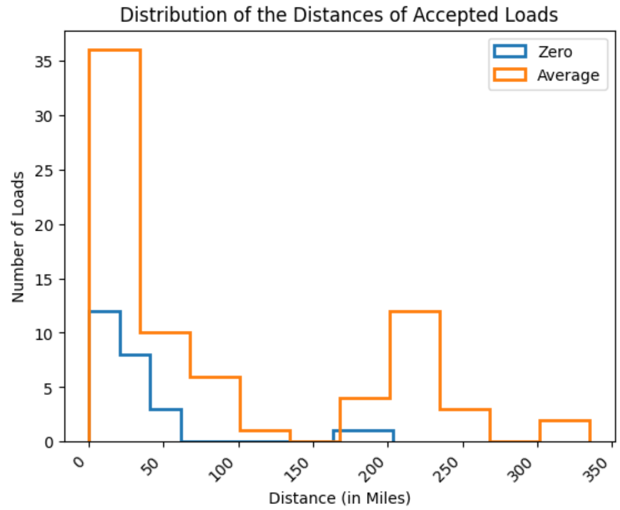


Figure 4.4: Distribution of the distances of accepted loads for both the zero pricing and the average pricing cases.

4.4.3 Relationship Between Price and Distance

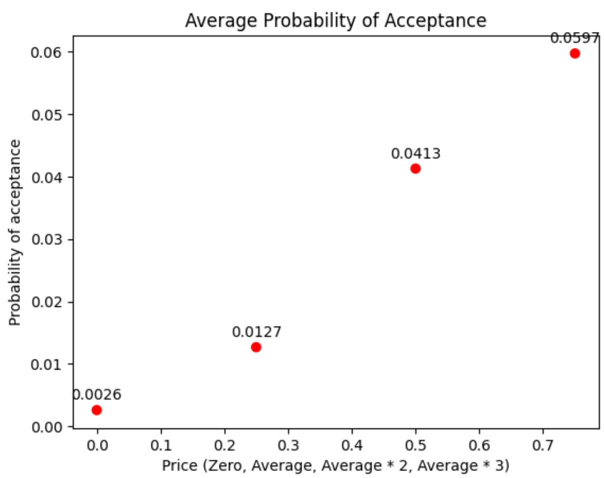
When we investigated which loads were being booked at a price of zero in the baseline parameters described in Section 4.4.1, we found that the average distance of loads booked to be around 30 miles, whereas the average distance of loads booked at the industry average price was around 85 miles. The distribution of the distance of accepted loads is shown in Figure 4.4. It is worth noting that this is purely a count of the number of loads accepted, and there are almost 3 times more loads accepted at the average price when compared to the zero price, so the bars in the histogram are much higher and sum to many more loads. However, it highlights that the distribution of distances for loads accepted at price zero is concentrated at the low end of less than 50 miles.

This illustrates that the majority of the loads booked at a low price were short-haul loads, and in fact, when we did a closer analysis on the origin and destination markets of these loads, most were intra-market. This led us to believe that we needed to better capture the relationship between distance and price in the model, as the parameters in the simulation were not correctly capturing how the price per mile should be much higher in the case of short-haul trips when compared to long-haul trips. We can capture this in the concept of a "willingness to pay" of carriers, which represents their acceptable level of a rate per mile as it varies across distances. We calculate the willingness to pay as

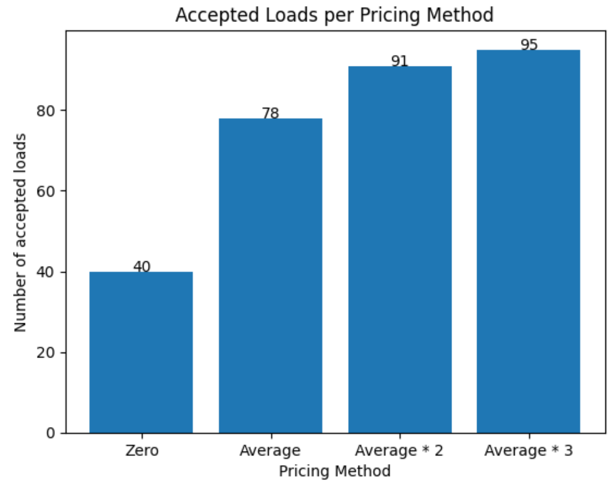
$$WTP = \frac{dU/dP}{dU/dD}$$

or the change in utility with response to price divided by the change in utility with respect to distance. We expect the curve to be somewhat of an exponential decay graph or a $\frac{1}{x}$ graph, as the rate per mile should become more steady for long-haul distances and be less sensitive to distance.

To model this in the carrier choice parameter estimation, we attempted to add the loga-



(a)



(b)

Figure 4.5: Probability of acceptance and number of accepted loads for choice model with log(distance) and log(deadhead) over 100 loads.

rithm of distance and deadhead, as

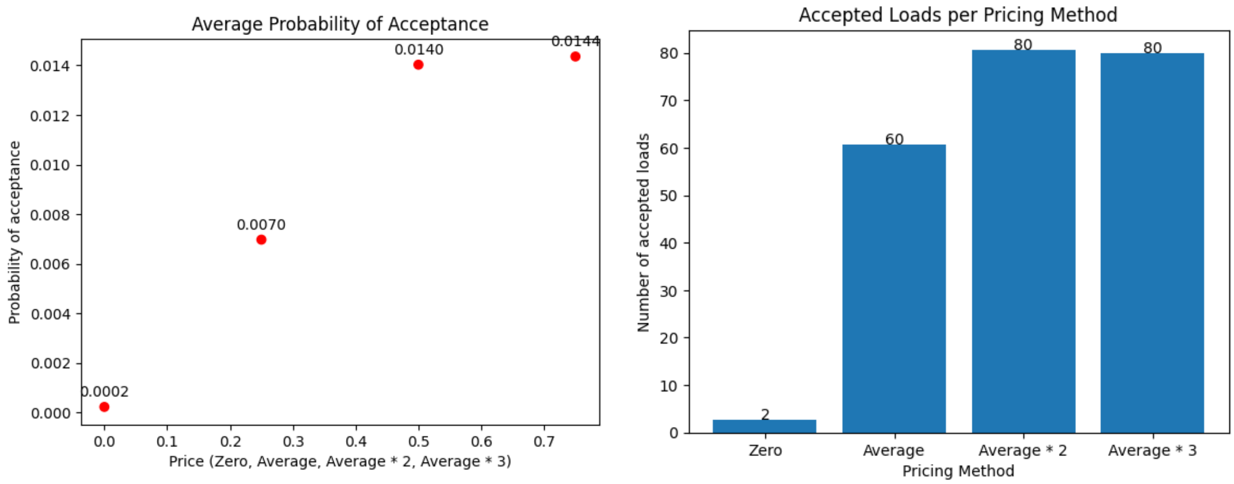
$$\frac{d}{dx}\log x = \frac{1}{x}$$

and a rate per mile parameter in hopes that these will capture the expectation of higher price per miles at lower distances and vice versa.

We first tried adding just the logarithm of distance and deadhead into the parameters to be estimated, and achieved the results shown in Figure 4.5. We see that the ratio of the number of loads accepted at the price of zero to the number of loads accepted at the industry average price has increased greatly, as it is now more than half. While we can increase the overall number of loads accepted by increasing the ratio of average sessions per carrier, the probability of acceptance and thus the ratio of the loads accepted at different prices cannot be tweaked, so we use this as a metric to judge our systems. When we look at the average distance of loads booked at price zero, it has increased to around 36 miles and the distribution seems more spread out to include longer loads, but the large percentage of loads booked at price of zero still illustrates that this model does not result in the desired simulation constraints.

Next, we attempted to directly incorporate the price per mile parameter into the model estimation to capture the inverse effect of distance in a load with price per mile. This yielded the results shown in Figure 4.6. The probability of acceptance at price zero is a factor of 10 smaller than the previous of .2%, as it is now 0.02% as seen in Figure 4.6(a). This results in a much smaller proportion of loads being booked, with the ratio of the number of loads booked at price zero to the number of loads booked at the average price being almost zero.

However, we notice that at the average price, only 60% of loads are accepted, and even when we increase the prices significantly by doubling or tripling the average, still only 80% of loads are accepted. This shows that in adopting this methodology, we may have focused too much on lowering the number of loads accepted at price zero and in the meantime achieved



(a) (b)

Figure 4.6: Probability of acceptance and number of accepted loads for the choice model with log(distance), log(deadhead), and price per mile over 100 loads.

results that do not behave as expected at the average price or high prices. We expect at least 80-90% of loads to be booked at the average price, and near or above 90% of loads to be booked when prices are doubled or tripled. To test this hypothesis, we increased the ratio of sessions per carrier while maintaining the same general distribution that the values are sampled from. This allows us to see how the acceptance probabilities that are a result of this set of parameters scale to the desired number of accepted loads.

The results are shown in Figure 4.7, which we achieved by increasing the ratio of sessions to carriers by a factor of five. We see that in this case, the ratio of the number of loads booked at price zero to the number of loads booked at the average price is still relatively small, at $\frac{1}{8}$th, which is acceptable, and the overall number of accepted loads at average and high prices fits within our desired range. We have found with some modifications, this method works fairly well in both fitting to the given data and producing the desired results in the simulation.

We also experimented with handling the non-linearity between price and distance with indicator variables for certain distance thresholds. For example, because we found earlier that the average distance for accepted loads at price zero was 30 miles, and in general they all fell under 50 miles except for a few outliers, we added indicator variables for the distance cutoffs of less than 10 miles, less than 25 miles, and less than 50 miles. We reasoned that this would result in the shorter loads being handled separately from the rest of the loads, as they would have larger negative utilities for their length, resulting in fewer bookings when they're priced low. However, the results are significantly worse as seen in Figure 4.8. The acceptance probability of the loads at price zero has increased by a factor of 3 from the highest we had seen before to the value of .6%, and there are $\frac{2}{3}$ as many loads booked at price zero than at the average price, which is much too high.

We recognized that the reason for this was that for shorter loads, the negative utilities associated with distance thresholds were being balanced out by the utility from the

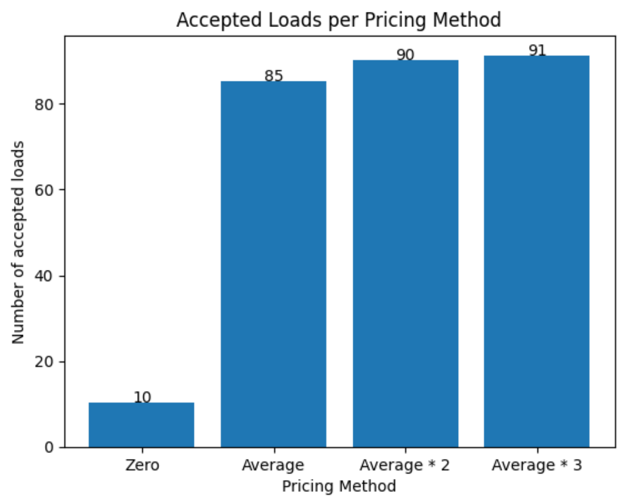


Figure 4.7: Accepted number of loads across different pricing mechanisms with the carrier choice model including log(distance), log(deadhead), and rate per mile with 5 times the sessions to carriers ratio and 100 loads.

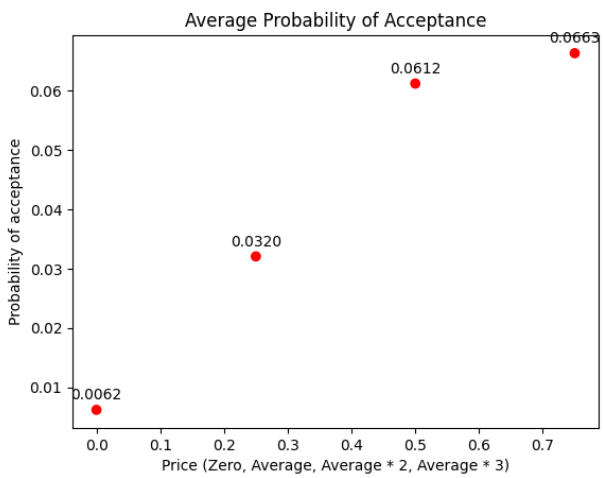
log(distance). For example, if there are about 20 miles from the pickup to the dropoff of the load, then the utility will be affected by both the less than 25 and the less than 50 mile thresholds, but $\log(d) = \log(20/100) \approx -0.7$. Because the log(distance) parameter is negative at $-.908$, then the utility will actually be affected significantly and positively by the log(distance) term. We thus deemed that this exploration of using indicator variables for distance thresholds was not viable.

4.4.4 Fixing Price Sensitivity

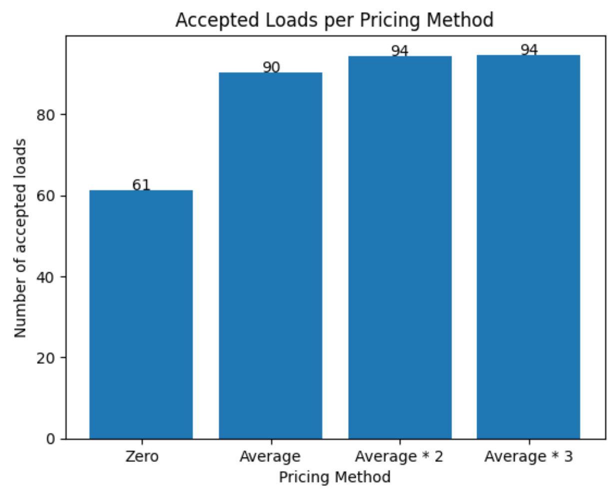
From the problem areas of previous experiments, we found that the model appeared to be not sensitive enough to price, for example in Figure 4.5 where almost half as many loads are booked at price zero than at the average price.

As a result, we attempted to measure the effect of the price sensitivity, or the price elasticity, on the results of the simulation. The price elasticity can be calculated as the change in booking probability over the change in price. It is important to note that models with low price elasticities will result in unrealistically aggressive price curves with dynamic pricing, as carriers will be willing to accept loads at both low and high extremes of prices if the booking probability does not change much relative to the price [8]. Because this is unrealistic, we weigh lower price elasticities negatively as we tune the model.

To conduct this analysis in a more controlled manner, we perform the log-likelihood estimation of parameters on the data while fixing the price sensitivities. Based on Figure 4.9, we see that with price sensitivities of less than 1, the number of loads booked at a price of zero is relatively high, ranging from 25-40% of loads being booked compared to the 80% of loads booked at the average price. However, with the higher price sensitivities of 2 or 2.5, we notice that while the overall booking probabilities are significantly lower because the probability of direct rejection is much higher, the number of loads booked across all pricing



(a)



(b)

Figure 4.8: Probability of acceptance and number of accepted loads for the choice model with log(distance), log(deadhead), price per mile, and threshold indicator variables for shorter distances over 100 loads.

methods produces our expected result - less than 5% of loads booked at the price of zero, and about 80-90% of loads booked at average or higher prices.

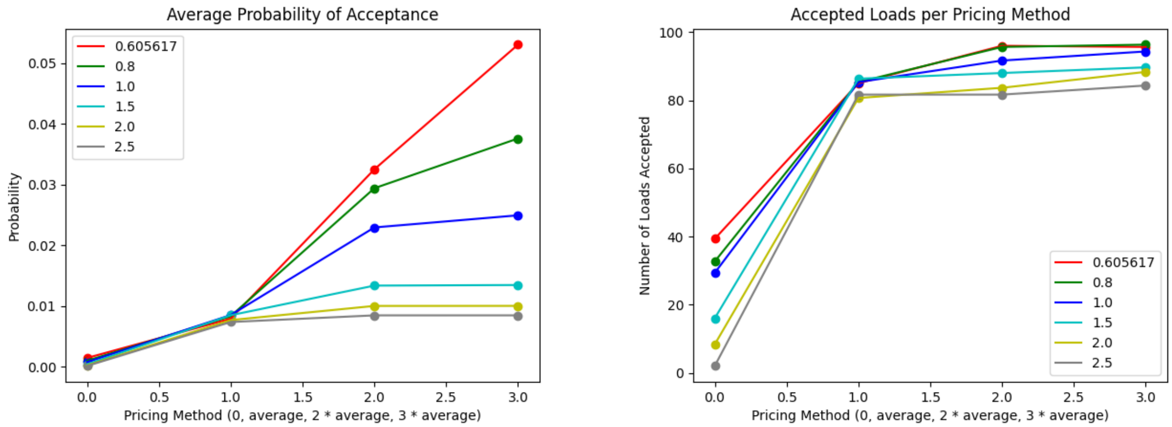
This illustrates that by fixing the price sensitivity at around 2, we are able to perform an analysis over its effect on the simulation and use this to make an informed decision that the higher price sensitivities produce more realistic results.

4.5 Key Takeaways

Based on this iterative parameter tuning results, we have landed on a set of parameters that both represents the data and produces the desired results in the simulation in terms of responding to different pricing methods.

We found that the initial baseline model of a naive maximum log-likelihood estimation with basic parameters gave unrealistic results where 30% of loads were accepted at a price of zero, which is a very high percentage especially when considering that only 80% of loads were accepted at the average price. We then introduced parameters representing the logarithm of distance and deadhead to capture the non-linear relationship of price and distance, but this result was not an improvement as 40% of loads were accepted at a price of zero, which was almost half as many loads as accepted at the average price. We attempted to directly add the rate per mile parameter in to again capture this non-linearity and achieved better results but overall fewer than expected loads were booked at an average price. We also experimented with indicator random variable parameters for distance of the load, but the results were not ideal as well because the logarithm parameters counteracted the distance threshold parameters.

We then fixed the price sensitivity parameter and found that a price sensitivity of 2 allowed us to most accurately capture how carriers respond to low and high prices, so this is



(a) (b)

Figure 4.9: Probability of acceptance and number of accepted loads for model over different price sensitivities.

the model we settled on as the most accurate and representative of the data and real-world responses of carriers.

Table 4.1: Choice Model Parameters for Different Tested Models

	Baseline	Log dst	RPM	Dst Thresholds	Price Sensitivity = 2
Booking constant	-4.362	-5.245	-4.769	-5.118	-5.297
Bundle size = 2	-3.141	-2.939	-4.168	-3.769	-6.722
Bundle size = 3+	-3.094	-3.353	-5.696	-4.234	-9.420
Total deadhead	-0.297	0.421	0.551	-0.293	-0.237
In load distance	-1.310	-0.932	-0.910	-0.335	-3.131
Bundle Price	0.738	0.636	0.780	0.652	2.0
Time to Pickup	0.547	-	-	-	-
Load from SAN	0.323	0.292	0.128	0.285	0.809
Load from DAL	0.411	0.411	0.233	0.276	1.664
Load from HOU	0.198	0.128	-0.147	0.100	0.973
Carrier in SAN	0.145	-0.102	-0.223	0.185	0.117
Carrier in DAL	-0.142	-0.242	-0.431	-0.141	-0.241
Carrier in HOU	0.003	-0.169	-0.290	0.013	0.062
Load to SAN	0.887	0.815	0.770	0.714	2.242
Load to DAL	0.424	0.393	0.374	0.305	0.858
Load to HOU	0.704	0.59	0.321	0.466	1.364
Log(dist)	-	-0.233	0.158	-0.908	-
Log(deadhead)	-	-0.543	-0.641	-	-
RPM	-	-	0.233	-	-
$d < 10$	-	-	-	-1.732	-
$d < 25$	-	-	-	-0.171	-
$d < 50$	-	-	-	0.084	-
Prob direct reject	0.92	0.933	0.986	0.933	0.989

The 1st is the baseline as used in this section. The 2nd set, Log dst, introduces the new parameters of log(distance) and log(deadhead). The 3rd set introduces the rate per mile or RPM parameter. The 4th introduces new parameters for shorter distance thresholds, and the 5th fixes the price sensitivity to 2.

Chapter 5

Evaluating Pricing and Bundling Algorithms

Now that we have demonstrated a realistic and consistent method of simulating the interactions between carriers and loads through a digital freight platform, we can work towards using this simulation to evaluate other methods. More specifically, the digital freight platform has leverage over the carrier-load interactions through its presentation of loads to the carriers and the pricing of loads after receiving a cost from the shipper. We use our previously defined simulation to evaluate how different methods can impact the freight network and potentially have positive effects on reducing costs and empty miles.

5.1 Methods

In the following sections, we describe more optimized methods for bundling and pricing the loads on a platform, which we hope can improve the efficiency of the network and reduce costs to the platform while maintaining the same or better level of load fulfillment. The mixed-integer linear programming bundling algorithm we propose in section 5.1.1 is used in place of the industry-based bundling described in Section 3.3.1, and this method will ideally result in carriers picking better bundles among fewer but more well-formed options as only suggesting a few bundles to each carrier means that there is less random, uncontrollable variation in the selection process. The dynamic pricing method proposed in Section 5.1.2 is evaluated in place of the static, industry average-based pricing described in Section 3.3.2, and should better represent how prices are dynamic over time, as a bundles can increase in price as the delivery time nears, making it more attractive to carriers.

5.1.1 MILP Bundling Algorithm

To generate a more efficient and selective set of bundles, we write our constraints as a mixed-integer linear program (MILP) to determine which loads to bundle together to get the highest benefit. The objective function of our MILP is to maximize the total utility of the bundles that are proposed by the platform to the carriers. In our case, the utility calculation assumes that the bundle is priced at the industry average prices and that the empty miles for a load are static based on market averages. The optimization problem takes the form of a set packing problem where the number of packs is upper bounded as we want to limit the number of bundles to display, and we also add constraints to ensure that there are no overlapping bundles, as having the same load appear in multiple bundles is inefficient when only selecting a small number of bundles. As the MILP can provide a set of bundles that do not include all loads, the obtained set of bundles is then combined with the set of single loads to ensure that all loads are included in at least one bundle.

In our situation, a feasible bundle of loads satisfies that a single carrier can deliver all loads in order in the bundle while satisfying their delivery windows and accounting for driving time, the idle time between two delivers does not exceed a set maximum, the number of loads in a bundle fits within the desired amount, and the time to deliver the entire bundle does not go over a certain value. We add additional constraints to guarantee that each load is contained in at most one pack so that each load can only be involved in one bundle resulting in no overlapping bundles.

The potential benefit of this methodology when compared to the industry-based bundling is that we consider bundles of size greater than 2, whereas the size of 2 is a strict constraint in the method described in Section 3.3.1. While this creates more flexibility in generating bundles, there is a potential downside, as Figure 4.1 illustrates how there is an greater negative value associated with bundles of size 3 or more when compared to bundles of size 2.

In addition, instead of presenting all possible bundles that satisfy certain time window constraints, we are able to provide a much more specialized bundle suggestion mechanism where each load is only included once in each bundle. This is beneficial because carriers typically only look at and seriously consider a few loads when opening the app as seen in Figure 2.3(a), so it is important to not overcrowd the suggestions with feasible but suboptimal bundles. Instead, we are able to reduce the amount of recommended bundles and only leave the most optimal, which increases the chance that they are viewed in an *impression* and thus booked.

We solve this MILP with Gurobi, a software for optimization.

5.1.2 Dynamic Programming Pricing Method

We also developed a pricing algorithm that is a reflection of the number of miles in a bundle, the time until the bundle, and the other bundles available on the platform. This pricing method is inspired by the work of Gallego and Ryzin, in which they generate prices for consumer goods based on the problem of selling a set amount of items by a certain deadline. The pricing is thus both supply, demand, and time sensitive, so it can be similarly applied to our situation where load pricing is load characteristic specific, depends on the rest of the market, and is time-sensitive [4].

We can model this as a dynamic programming problem where we want to calculate the expected cost to the platform at a given time t with remaining loads L left to satisfy. For the base case, we know that if we have L loads left at the end that are unfulfilled, then the platform will have to pay the penalty for all of these L loads. Then, if there's one carrier remaining, we can determine the cost using the probability of the carrier accepting the load and the base case of penalty with L or $L-1$ loads.

Given this model, we can formulate the problem as

$$\text{cost}_t(L) = \text{penalty}(t, L) + \rho_{\text{reject}} * \text{cost}_{t+1}(L) + \sum_{i \in B} \rho_i * [p_i + \text{cost}_{t+1}(L \setminus B_i)]$$

where ρ_i represents the probability of selecting bundle or action i, p_i represents the cost of fulfilling bundle i, t is the current time, L represents all available loads, and B_i represents the loads included in bundle i. Essentially, the cost of having L loads left at time t is a combination of three terms:

- The carrier rejects all the bundles, so at time step $t + 1$ there are still L loads, represented by the second term.
- The carrier selects one of the bundles at a certain price, leaving the carrier with all L loads except the ones in the selected bundle at time step $t + 1$, represented by the third term.
- The penalty of loads expiring at time t, represented by the first term.

This gives us a pricing algorithm that is easily adaptable in that it takes into account time, distance and market conditions of the load, and eventually can also take into account carrier information. We also experiment with homogeneous pricing, where prices are not carrier-dependent but can still vary between different markets, and heterogeneous pricing, where different carriers at the same time can see different prices depending on their exact location.

However, we found when solving for this dynamic programming formulation with larger instances of even 100 loads and 150 carriers, the problem is intractable. As a result, we estimate the costs at future time steps by assuming that the bundles are priced according to the static market average. With this price assumption, we can compute the probability $\rho_{i,\text{avg}}$ that bundle i will be selected before it expires, which allows us to estimate the cost at time step $t+1$ as the simplified sum over all remaining bundles of $p_{i,\text{avg}} * \rho_{i,\text{avg}} + \text{penalty} * (1 - \rho_{i,\text{avg}})$. We then use this approximation of $\text{cost}_{t+1}(L)$ to compute the optimal price p_i at the current time step, and this is the price displayed to the carriers.

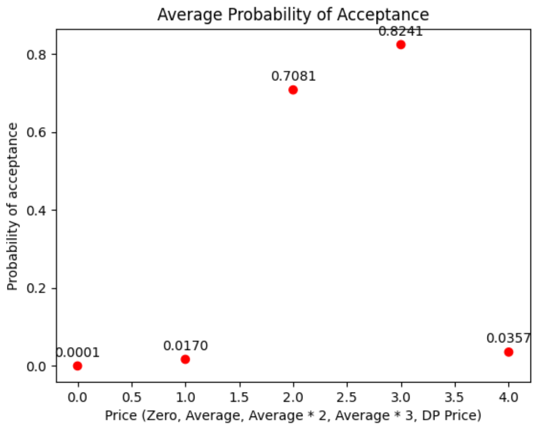
5.2 Evaluation

We analyze the proposed methods by integrating the algorithms into the simulation and examining the produced results. More specifically, we use the most realistic set of parameters found in Section 4.4.4, namely with the price sensitivity fixed to 2 and without the log (distance) parameter.

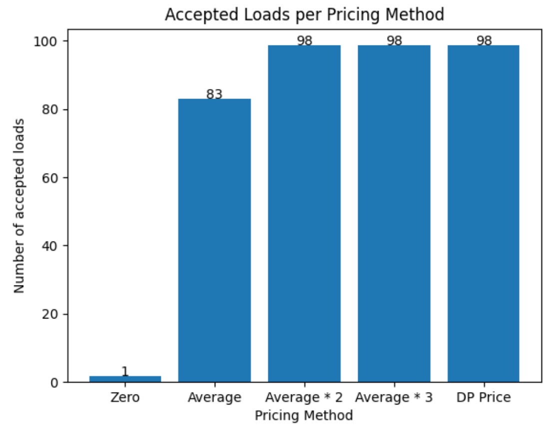
5.2.1 Pricing

In evaluation, we use the following pricing methods.

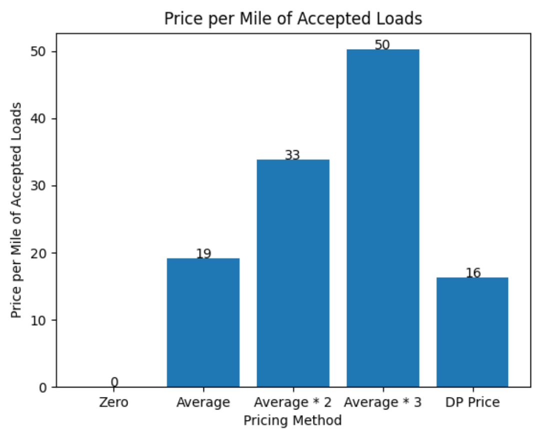
1. Static price of $0 for every load, used to make sure our simulation parameters are as expected.



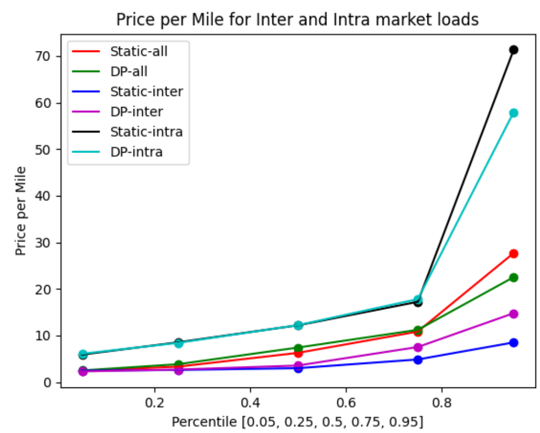
(a) Probability of Acceptance



(b) Number of Accepted Loads



(c) Price per Mile of all Accepted Loads

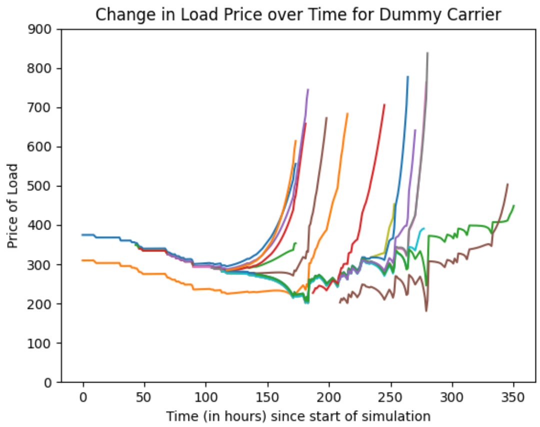


(d) Price per Mile Percentiles for inter-market and all Loads

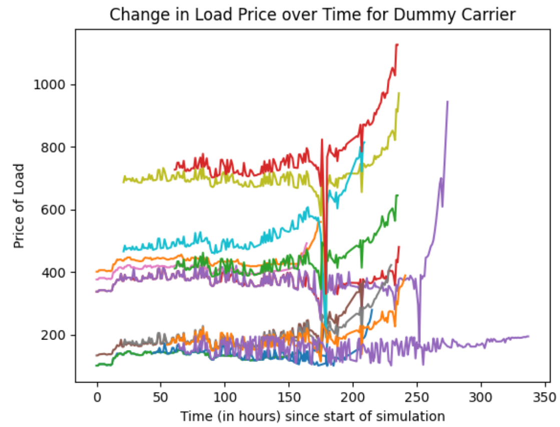
Figure 5.1: Results from the simulation for accepted loads and price per mile for 100 loads total.

2. Static price of the average ranges per lane of directional city to city loads given by Uber Freight.
3. Static price of two times the average ranges per lane of directional city to city loads given by Uber Freight.
4. Static price of three times the average ranges per lane of directional city to city loads given by Uber Freight.
5. Dynamic pricing where the price of a load changes over time and the price of a load differs at a certain time depending on the carrier viewing the load.

We see in Figure 5.1a that the average baseline model has an acceptance probability of 1.7%, which results in 83% of loads being accepted as seen in Figure 5.1b. In comparison, the dynamic pricing model has an slightly higher acceptance probability of 3.57%, resulting



(a)



(b)

Figure 5.2: Price over time for a dummy carrier in scenario where (a) no loads are ever accepted and (b) the simulation runs as normal.

in 98% of loads accepted. However, when we compare the prices of the static average and dynamic pricing models, we see that the pricing model is able to achieve the higher percentage of accepted loads while maintaining a lower price per mile on average of accepted loads. As seen in Figure 5.1c, the average price per mile of all accepted loads at the static average price is \$19 per mile, whereas the average price per mile of all accepted loads at the dynamic programming pricing method is \$16.

We can also can perform a more fine-grained analysis into the price per mile by investigating the pricing difference between inter-market and all loads. In Figure 5.1d, the 5%, 25%, 50%, 75%, and 95% percentiles for the different types of loads are plotted for both the baseline static pricing and the dynamic pricing. We notice that for inter-market loads, the dynamic pricing has a higher price per mile, and for intra-market loads, the dynamic pricing has a lower price per mile. This results in the overall price per mile of all loads to be slightly lower in the dynamic pricing model. The price per mile is much higher for intra-market loads for both pricing methods because for shorter loads within one city, carriers still have a somewhat fixed minimum price to accept a load that inflates the price per mile of shorter loads. In addition, long-haul inter-market loads are less sensitive to slight variations in distance.

This possibly indicates that the dynamic pricing model has not fully captured how carriers respond to long-hual loads and how much prices must be increased to make a long load more incentivized as the deadline for pickup approaches. Overall, it shows that the dynamic pricing model can respond better to differences in time, carrier, and load to result in loads being accepted at lower prices overall, which lead to higher profit margins for the platform.

To analyze how the dynamic pricing is set over the course of the simulation, we generate the prices over time per bundle for a dummy carrier. Because the prices are carrier-dependent, we fix the location of the carrier to be in the center of Dallas arbitrarily. We expect to see a relatively steady price in the first few days after the load is created, and then once the load is relatively close like within a day away, we expect the price to exponentially increase over time to incentivize carriers to book the load. To control the experiment, we run the simulation with the same number of carriers, loads, impressions per carrier, and parameters as our base model, except we do not let any carriers accept any loads. The resulting prices are seen in Figure 5.2a.

The load price over time illustrated in Figure 5.2a is run on a reduced version of the data, where we only have 100 loads, to be able to better visualize the trends. As a control, this is run with only single loads presented on the platform, without bundles. For readability, we are also only plotting the curves that have a significant amount of data points, meaning that there is enough time between the load creation and the load expiration point. Each curve in the graph represents the price over time of a different load. The x length of the curve represents the time from when the load was created on the platform to the end of the pickup time of the load, as this is when we deem that a load has expired and can no longer be booked.

We see that each line follows a relatively steady pricing before changing to be on a dramatic exponentially increasing trajectory when the load is about 1-2 days away from being booked. The fluctuations in the price when the simulation time is still far from the pickup expiration time can be explained by the interaction of loads on the platform, as one load's price is lowered to make another load's price higher if the latter load's expiration date is soon. In addition, the price of a load decreases when new loads arrive in the network to keep the load competitive with more options, which explains why the prices are decreasing at the beginning of the simulation as more loads are created by shippers on the platform. The few loads in the figure that do not experience an exponential increase in price are loads whose pickup start times fall within the 2 week simulation time frame, but whose pickup expiration time falls after.

We take a look at the price over time for loads in the case where the simulation is running regularly, in the case of Figure 5.2b, which illustrates the prices for a dummy carrier over the course of the simulation for 500 loads, where carriers do accept loads, thus removing them from being priced by the platform. In this scenario, we see much larger fluctuations, which occur when another load is booked and the prices are adjusted to reflect this. For this case, we see a few loads reach the stage where the price begins to exponentially increase, but most loads are booked before this time is reached.

We also investigate the lead times for the different pricing methods to better understand how the pricing methods interact with carrier choices. We measure lead time as the time between when a load is booked and the start of the pickup time window for the load. As we can see in Figure 5.3, the lead times for static pricing defined by the blue histogram is relatively flat across with not many loads being booked with more than 225 hours to spare, but that is likely due to not many loads being created with that much lead time. In the dynamic pricing case, we see that there are a lot of loads booked within the last 50 hours of the load. As prices are lower in the dynamic case on average, fewer loads are booked many days in advance, but as the prices spike closer to the starting time of a load, carriers are much more likely to book.

Overall, we do see in Figure 5.4b that the cost to the platform is lower when comparing the dynamic pricing method to the baseline average pricing method. The cost is calculated as the sum of the prices for all accepted loads plus the sum of the penalty incurred for all loads not accepted by the end of the simulation. We see a decrease of about 20% in the cost

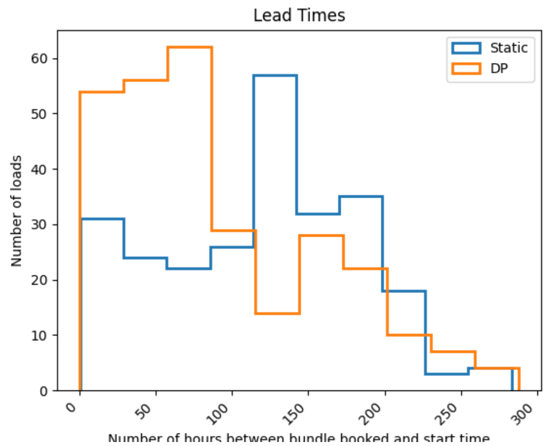


Figure 5.3: Lead times for Static and Dynamic Pricing

to the platform, which can be explained by the reduced price per mile of accepted loads and the slight increase in the number of accepted loads.

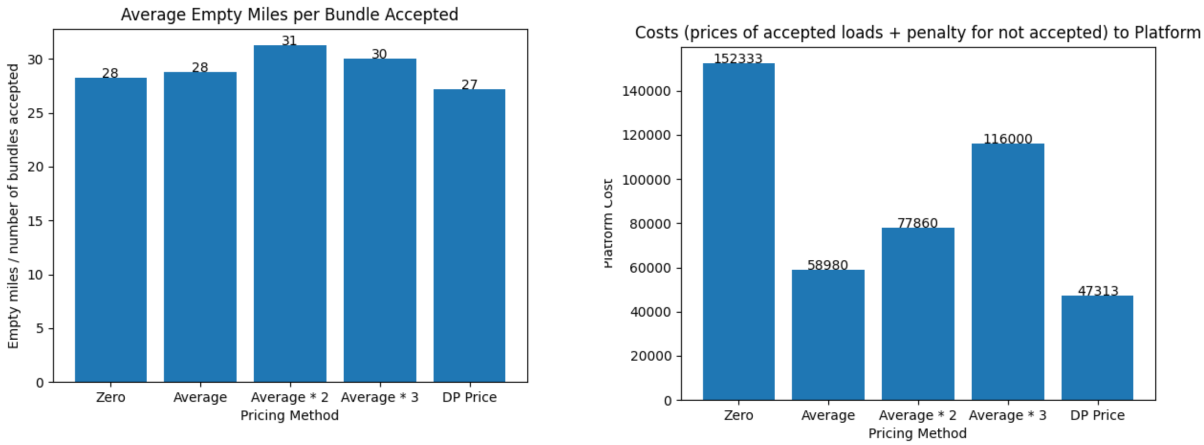
5.2.2 Emissions

We evaluate the emissions of the system by summing up the empty miles driven by all carriers that accept loads from the platform. We can calculate this given the position of the carrier when they open the app and book the load, the pickup location of the load, and in the case of a bundle, the dropoff location of the first load and the pickup location of the second load. We hope to see that our bundling and pricing methods result in similar or slightly lower costs to the platform with lower emissions.

With the industry-based bundle generation methods, we see in Figure 5.4a that the average empty miles per bundle accepted, which is calculated as the distance driven from the carrier's starting location to the first load pickup location plus the distance between the dropoff and pickup of the next load if there are multiple loads in the bundle, is not reduced between the average pricing and the dynamic pricing methods. In fact, the average empty miles increases by a small amount, possibly due to the randomness in the process of assigning bundles to carriers and the effect of lower prices on the interactions between deadhead miles and price in the choice model.

Table 5.1: Percentage of Loads that are Accepted in a Bundle

	Static Pricing	Dynamic Pricing
Industry-based Bundling	1.5%	6.1%
Industry-based Bundling, $\beta_{\texttt{bundle}} = 0$	32.5%	33.7%
MILP Bundling, $\beta_{\texttt{bundle}} = 0$	15.7%	30.6%



(a) Average empty miles for each accepted bundle

(b) Cost to the platform

Figure 5.4: Results for empty miles and costs for the industry-based bundling method with dynamic pricing.

However, we notice in this model that there are barely any loads accepted in a bundle, as seen in Table 5.1, with 1.5% of loads accepted in a bundle of size 2 with the static market average prices, and 6.1% of loads being accepted with the dynamic pricing model. Because of these low percentages, we can better analyze the effect of bundles if carriers are willing to consider them. To alter this in the simulation, we set the previously negative weight of a bundle ($\beta_{\texttt{bundle}}$) to 0 in the carrier choice model to represent a scenario in which carriers are unaffected by the bundle type and size. We see the results of this in Table 5.1, where around 30% of loads are being accepted in a bundle in both the static and dynamic pricing cases.

Table 5.2: Average Empty Miles per Load over Pricing and Bundling

	Static Pricing	Dynamic Pricing
Industry-based Bundling	27.8	30.2
Industry-based Bundling, $\beta_{\texttt{bundle}} = 0$	30.0	27.9
MILP Bundling, $\beta_{\texttt{bundle}} = 0$	27.0	28.7

We can now analyze the methodology of the MILP-based bundling algorithm, where we only generate half as many bundles as there are loads, so if there are a 100 loads in the simulation, we generate 50 bundles. We display these as options to the carrier, in the setup with the modified carrier choice model. With the MILP-based bundling, there are about 15% of loads accepted in a bundle with static pricing, and 30% of loads accepted in a bundle with dynamic pricing. When we then evaluate the results in terms of empty miles, we see in Table 5.2 that with the introduction of bundles, the average empty miles per accepted load generally decreases when we remove the carrier dislike of bundles, but there is not really a large change or anything significant enough to draw conclusions.

5.3 Key Takeaways

Based on our experimentation with more optimized pricing and bundling algorithms in our simulation, we have found that our dynamic pricing method accurately captures the exponential increase in prices of loads as their pickup time nears. We find that over all loads, the price per mile of accepted loads is slightly lower in the dynamic pricing case when compared to the baseline, which means that carriers are willing to accept loads are a lower price and thus the platform is able to gain more profits. When we break this down into a inter- vs. intra- market comparison, we see that the lower prices come from the intra-market cases, as inter-market loads are on average priced higher in the dynamic pricing method, which can be interpreted to mean that the model has not fully captured the interaction between carriers and prices for loads of longer distances. This pricing method, when compared to the baseline static industry-average pricing method, results in more carriers booking loads within the last 75 hours before a load, whereas in the baseline case, more carriers booked a load with more lead time, typically between 100-200 hours in advance.

In terms of bundling to reduce emissions, we find that with our generated parameters, the negative weight associated with accepting a bundle is so large that very few bundles are booked at all. When we discard this weight to model a scenario where carriers are unaffected by the bundle size, we find that more bundles are accepted but they have a very small effect on the average empty miles emitted, so we do not draw any conclusions on this methodology.

Chapter 6

Concluding Remarks

Overall, we have seen that through the sequential estimation and evaluation technique we describe in this paper, we are able to leverage the simulation and its results to produce a model that accurately fits the given data and satisfies our desired constraints. The implications of this success lay beyond just the transportation world, as this method of learning models can be applied in other contexts as well where human choice behavior needs to be understood.

Specifically in our research, by creating a simulation based on industry data to realistically model how carriers interact with the platform and select bundles to deliver, we were able to perform a detailed and insightful analysis on where the platform can improve. We have found that fixing the price sensitivity of a simpler model gives the most realistic results.

We use this model to analyze novel pricing and bundling methods, which illustrate how digital freight platforms can leverage their role in pricing bundles to achieve the same amount of loads booked with slightly lower prices in the case of intra-market deliveries. This suggests that our model is able to better capture the difference in pricing between long and short haul.

However, we were not able to demonstrate a significant improvement in the reduction of empty miles with our current simulation and proposed bundling methods, as with the data-driven carrier choice model parameters, the acceptance rate of bundles is extremely low, as there is a large negative constant associated with the number of loads in the bundle as seen in Table 4.1. This means that although we are able to propose better-formed and more tailored bundles to the carriers, the chance that they even consider a bundle with multiple loads is relatively slim. As a result, even though we can improve the optimization of how bundles are formed to reduce empty miles, these methods fail to have a significant impact on the number of empty miles driven by carriers overall.

6.1 Future Work

To improve on this in the future, we can continue to optimize the bundling method, but in order to have a tangential change in the number of empty miles, we first must figure out a way to incentivize carriers to select bundles. While this is not necessarily a change possible in the simulation, as the choice model parameters we have are based on the industry data, we can work with freight networks to better understand what causes bundles to be so unattractive to carriers and how to mitigate this negative effect. If carriers remain relatively unwilling to accept bundles, then the only empty miles improvements that can be made would result from pushing carriers to accept loads closer to their starting point, which is already done in the simulation but has an element of randomness and carrier variability to it.

Another element to consider adding into the simulation in the future would be to better model the competition among platforms. We are accounting for this right now in the option where the carrier can immediately dismiss a bundle independently of its characteristics and leave the platform without accepting any loads. However, the aspect of competition between platforms perhaps should reflect more on the pricing of loads - if a load is priced too low, then carriers could look elsewhere to book loads and perhaps lose trust in the current platform and never return for more sessions. This loss of faith between carriers and the platform is an important consideration, as the lowering of prices for bundles is a slippery slope. We have shown that there is some wiggle room to lower the prices of bundles and still have roughly the same average acceptance probability, but because of the competition between platforms and load boards, the price cannot be lowered too far. If this competition can be more rigorously and directly modeled in the simulation, then this concern of prices being too low can be resolved.

Once we have a completely robust simulation for modeling the interaction between shippers, carriers, and loads on a digital freight network, we can in future work apply this simulation to gain insights about situations that are more abstract. For example, in the space of adopting electric freight vehicles (EVs), there is lots of uncertainty regarding the feasibility of charging along long-haul routes. The adoption of EVs would lead to huge improvements in reducing the amount of carbon emissions in the freight industry. Given the simulation setup we have, we can accurately map the routes taken by carriers and their interactions with loads in the Texas Triangle. We can leverage this information to propose a plan for charging station infrastructure placement while also revealing how much of an impact the need to charge would have on the speed of delivering loads and satisfying time windows. Charging a long-haul vehicle is much slower than stopping to fill the tank of gas, but because of the amount of downtime that carriers typically have between booking a load and picking up a load, we believe that the simulation could demonstrate that there are optimal ways to use the downtime to charge, allowing EVs to be adopted without having a major impact on the delivery times of loads.

We have taken the first major steps to provide an accurate and realistic model of simulating carriers, bundles, and bookings on a digital freight platform that is rooted in industry data, and the use cases of a simulation have lots of potential in improving the efficiency of the freight industry and reducing emissions.

References

[1] Fast facts on transportation greenhouse gas emissions, 2023. URL https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions.

[2] Uber freight load bundles reduce deadhead by almost 23URL https://www.uberfreight.com/blog/how-uber-freight-reduces-deadhead/.

[3] Omar Abed, Tom Bellemans, Gerrit K. Janssens, Davy Janssens, Ansar-Ul-Haque Yasar, and Geert Wets. An agent based simulated goods exchange market; a prerequisite for freight transport modeling. *Procedia Computer Science*, 52:622–629, 2015. ISSN 1877-0509. doi:10.1016/j.procs.2015.05.054. URL https://www.sciencedirect.com/science/article/pii/S1877050915008546. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015).

[4] Guillermo Gallego and Garrett J. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020, 1994. doi:10.1287/MNSC.40.8.999.

[5] Kilian Heilmann. Information frictions, load matching, and route efficiency in the trucking industry. 03 2020. doi:10.2139/ssrn.3545019.

[6] Anh-Tuan Nguyen, Sigrid Reiter, and Philippe Rigo. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113: 1043–1058, 2014. doi:10.1016/j.apenergy.2013.08.061.

[7] Carolina Osorio and Michel Bierlaire. A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345, 2013. doi:10.1287/opre.2013.1226.

[8] Guillaume De Roo. Freight pricing with a controlled markov decision process, 2021. URL https://www.uber.com/blog/freight-markov/.

[9] Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4): 500–520, March 2014. doi:10.1287/trsc.2013.0490.

[10] Stefan Schroeder, Michael Zilske, Gernot Liedtke, and Kai Nagel. A computational framework for a multiagent simulation of freight transport activities. 39, 01 2012. URL https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2011/11-19/zz_archive/2011-08-18_freight_transport_activities_submitted_trb.pdf.

[11] AJ Todd. What is a freight broker?, January 2020. URL https://www.flockfreight.com/

blog/what-is-a-freight-broker/#:~:text=A%20freight%20broker%20is%20a,freight%20arrives%20safely%2C%20on%20time.