Equipping Nolan Transportation Group for Rapid Growth: Providing Tools to Assist the Atlanta Enterprise, Less-Than-Truckload, and Drayage Departments

Alexander Bolen
Sean Hartrich
Zoya Mahmood
Sarah Stevens
Victor Tonelli Rohlfs
James Michael Wiggins
Jonathan Elijah Williams
Nathanael Williams

April 20, 2018

Georgia Institute of Technology
School of Industrial & Systems Engineering

Liaison (Jonathan Elijah Williams) email: jwilliams408@gatech.edu
Team 20 email: team20gatech@gmail.com

Client: Nolan Transportation Group
Client contact: Bert Brunsting, Corporate Analyst
bert.brunsting@ntgfreight.com, 615-557-2612

Advisor: Andrea Laliberte
alaliberte@gatech.edu

Executive Summary

Nolan Transportation Group (NTG) offers third-party logistics services to the shipping industry in three main areas: full-truckload (FTL), less-than-truckload (LTL), and drayage. Since NTG is experiencing rapid growth, they require an infrastructure upon which to grow their business.

Atlanta Enterprise is the specialized area of FTL that services NTG's top customers. They provide specialized care to the company's top customers and ensure they receive the best possible service. The current system uses a suboptimal staffing distribution, thus inhibiting the number of possible loads serviceable during the day. Using the optimal ratio of staffing distribution and implementing alternative methods for structuring their team will increase the number of loads that NTG is able to service. To further increase overall customer service, NTG can implement this model in the entire FTL department.

The Less-Than-Truckload department handles shipments that require only a fraction of a truck's capacity. NTG currently outsources their LTL business to fourth-party carriers. Combining multiple shipments into pooled routes will allow NTG to bring LTL business in-house and provide their customers with their exceptional FTL service. Implementation of the proposed LTL routing model will give NTG a 5.27% increase in profit margin, or an annual net profit increase of $467,000. This valuation accounts for the expenses necessary for NTG to incorporate the proposed changes. As NTG continues to expand their LTL department, the increase in number of shipments available for matching will increase both the quantity and quality of combined routes, resulting in continued improvement to profit margin.

The Drayage department at NTG is a brokerage department that services import and export shipments. Both imports and exports are shipped as individual trips, which results in carriers' transporting an empty container for half of each trip. Combining an import shipment with an export shipment produces cost savings by reducing the transportation of empty containers. Currently, NTG is manually matching loads based on experience. However, automatically generating potential combinations for NTG to use in matching shipments gives them the potential to realize more profit by finding more import-export matches. Implementation of this model will find NTG 55 import-export matches per year and increase their annual net profit by $300,000. Once this model is in place, NTG can encourage their customers to ship drayage loads along lanes commonly used in matching, which will further increase this value.

This project has given NTG the tools needed to prepare for future growth in their Full Truckload, Less-Than-Truckload, and Drayage departments.

Table of Contents

1. Client Description

Nolan Transportation Group (NTG) is a transportation brokerage firm that works with contract partners by matching available trucks from their transportation providers with client delivery requests. They serve as a third-party logistics service provider to over 8,000 customers across North America. While NTG deals primarily with full-truckload (FTL) shipments, they also provide less-than-truckload (LTL), partial, and drayage shipping services for their clients. NTG recently launched a new customer service-oriented FTL business model called Atlanta Enterprise, which services NTG's top five customers and follows the shipments from order placement to delivery.

NTG has been experiencing rapid growth over the past 5 years: since 2012, they have increased both the number of loads serviced and number of workers employed. Figure 1.1 and Figure 1.2 illustrate the growth in these areas.

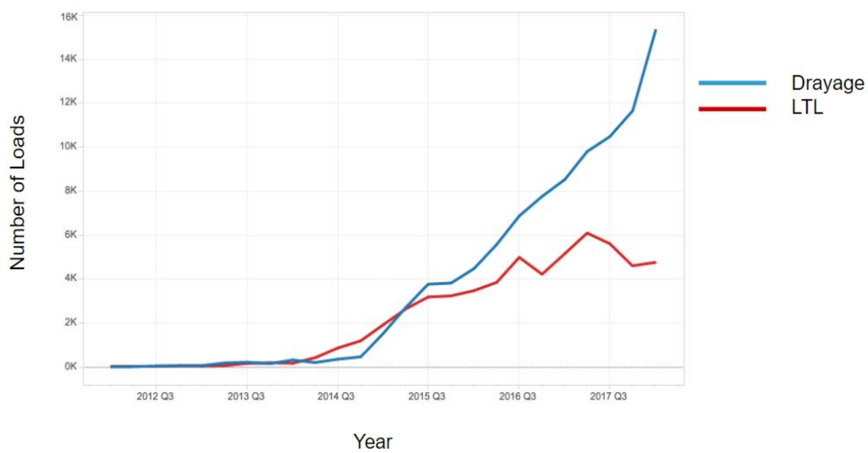

Figure 1.1. Number of drayage and less-than-truckload shipments from 2012 to 2017.

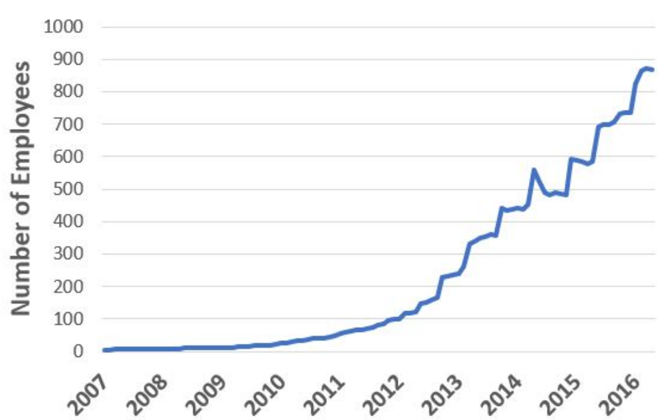

Figure 1.2. Total number of employees at Nolan Transportation Group from 2007 to 2016.

2. Goals

The overall goal for our project is to improve NTG's infrastructure to support their continued growth in the Atlanta Enterprise, Less-Than-Truckload, and Drayage departments. The main goals for each of the departments, respectively, are to:

- Maximize the throughput of loads handled per day by Atlanta Enterprise
- Increase net profit by consolidating LTL and partial shipments into larger, pooled-route shipments
- Reduce operating costs by eliminating empty import-export drayage travel legs

3. Atlanta Enterprise

3.1 Current System

Atlanta Enterprise is NTG's new business model to service their top five customers. Before Atlanta Enterprise was implemented, an NTG office would have many Account Managers that would build, schedule, and book each load with minimum support. Atlanta Enterprise reduces the number of Account Managers and adds support staff comprised of these roles: Builder, Scheduler, Booker, and Tracker. The following are the job descriptions for each role (expanded in Appendix B.1):

- Account Manager - Interacts with a designated customer and solicits orders from the customer
- Builder - Receives a copy of the order from the Account Manager via email, reviews the document, and inputs the load data into Freight Hawk Dispatch (NTG's in-house platform for freight management)
- Scheduler - Contacts each warehouse to schedule pick-up and delivery times for freight
- Booker - Negotiates a lane price with potential carriers to find a transporter for each load
- Tracker - Communicates regularly with the carrier and provides NTG with updates on the location and estimated arrival time for a load

The general movement of a load through the system is shown in Figure 3.1. The following steps are required to service a customer order throughout its life:

1. A load is initiated when an Account Manager receives an order request from a customer, in a form standard to the customer but not to NTG.
2. The load is assigned to a Builder who inputs the load order into the Freight Hawk Dispatch system.

3. A Scheduler reviews the information entered by the Builder and sets pick-up and delivery times for the load.
4. A Booker finds a carrier for the load and may communicate with the Scheduler to change pick-up or delivery times to accommodate the potential carrier. The time needed to book a load varies greatly depending on the shipping lane - booking time can be anywhere from five minutes to two hours.
5. From dispatch to delivery, the Trackers communicate with the carrier to provide updates on the location and estimated time of arrival for the load. After checking on a load, a Tracker sets a new follow up time, usually in the range of 30 minutes to several hours, depending on how soon the delivery is to be made. If the carrier is unable to complete the delivery on time, the Tracker will send the load back to the Schedulers or Bookers to set a new pick-up and drop-off time or to find a new carrier.

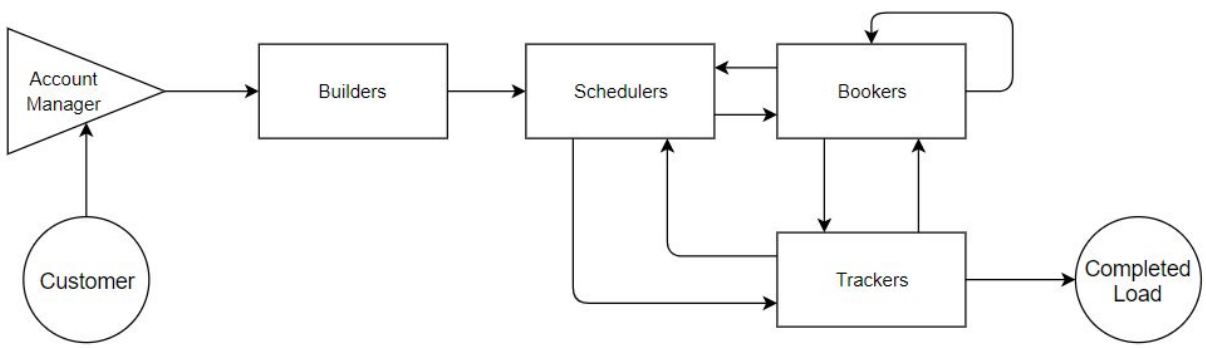

Figure 3.1. Flow diagram showing the work interactions for an order in Atlanta Enterprise.

3.2 Opportunity

Atlanta Enterprise has not been operating as efficiently as other NTG offices. One key performance indicator used to measure an office's efficiency is margin per employee, which is the gross profit of an office divided by the number of employees. Atlanta Enterprise's margin per employee is 40% lower than the company average, which creates an opportunity for system improvement.

Initial analysis of the office showed low worker utilization rates in the current system (listed in Table 3.1) indicating that Atlanta Enterprise is currently operating below their full capacity. The Trackers' utilization is low because they are overstaffed and much of their day is spent waiting for the next checkup time on a load.

Role	Utilization in Role
Builders	40%
Schedulers	47%
Bookers	69%
Trackers	25%

Table 3.1. The utilization (percent of time working on loads) for each role.

NTG is hesitant to give Atlanta Enterprise more work for fear of making their performance worse. They can handle more orders, but their current distribution of employees is not optimal for load processing.

3.3 Approach

Modeling the current system in Simio as a simulation showed both the throughput of the current system and the maximum capacity of the current system. The Simio OptQuest add-on with incrementally increased arrival rates showed the optimal distribution of employees and the maximum throughput of the system. For a more thorough explanation on the Simio simulation model, see Appendix B.2.

3.4 Results

Atlanta Enterprise currently consists of 3 Builders, 2 Schedulers, 20 Bookers, and 18 Trackers. They handle, on average, 84 orders per day, but are capable of handling 184. Reorganizing to have 6 Builders, 4 Schedulers, 18 Bookers, and 15 Trackers results in a system that can process up to 280 orders per day. A summary of these results is listed in Table 3.2.

Role	Current System	Proposed System
Builder	3	6
Scheduler	2	4
Booker	20	18
Tracker	18	15
Total	**43**	**43**

Table 3.2. The current and proposed number of employees for each role in Atlanta Enterprise.

The reorganization of the system to the suggested distribution of employees will increase their daily potential to service loads from 184 to 280, which is a 52% increase. For details of the calculations that produced these numbers, refer to Appendices B.2 - B.3.

3.5 Deliverable

NTG will have the option of expanding this optimal employee distribution to all load-tracking departments company-wide. They can also cross-train employees to complete multiple tasks, which would increase throughput even more by eliminating transfer time between roles.

3.6 Value

Atlanta Enterprise will be able to handle 195 more loads per day once the staffing reorganization is implemented.

3.7 Additional Observations

The following are additional improvements to the system based on observation:

- *Improved communication*: The employees at NTG use several channels of communication to fulfill their daily responsibilities. For example, a Tracker who receives an update will enter the information into Freight Hawk and copy it into both Slack and any relevant emails. Streamlining communication so that an update reaches all applicable parties will reduce confusion and save time.
- *Cross-training*: Both Bookers and Schedulers perform very similar functions in Atlanta Enterprise, and there is no need for the specialization between the two roles. If employees were cross-trained in these areas, a load order could be processed more quickly.

- *Updated booking software*: Employees reported having to spend time re-entering data into Freight Hawk due to technical glitches or unresponsiveness. Adding features to Freight Hawk to make it more user-friendly and less susceptible to glitches will simplify employee entry.

4. Less-Than-Truckload

4.1 Current System

NTG's Less-Than-Truckload (LTL) and Partial departments handle shipments that do not fill trucks up to capacity. LTL deliveries are not as large as FTL or partial deliveries, and take up much less space in the truck, so they are shipped at a higher cost than the other two. Partial shipments do not fill a truck to capacity, but take up a large enough space to warrant pricing at the FTL lane rate multiplied by the percentage of truck space required. Partial shipments receive the same level of service as FTL shipments, however, LTL shipments do not. Currently, NTG coordinates delivery of both types of loads by finding fourth-party carriers that are willing to transport the individual shipments.

4.2 Opportunity

To reduce shipping costs, LTL and partial shipments can be consolidated into pooled loads containing multiple shipments. This reduces cost by reducing travel distance and causing loads to be delivered on shipping lanes with cheaper rates than those for the individual shipments. The left map in Figure 4.1 displays an example representing the current method for delivering LTL and partial shipments, and the right map shows the associated combined routes that would result from combining orders.

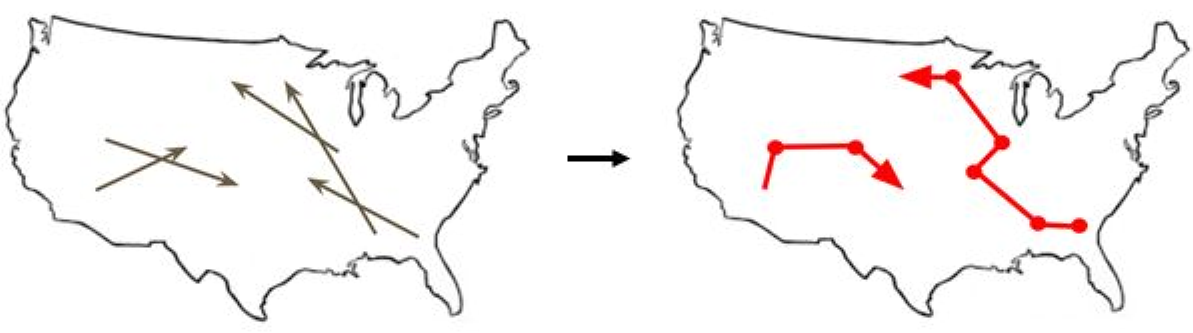

Figure 4.1. Illustration of current individual LTL and partial deliveries (left) and corresponding proposed combined routes (right).

Two major advantages of combining LTL and partial shipments into partial or FTL deliveries are the improved customer service for FTL over LTL and the cost savings that NTG will generate by consolidating loads. These savings can be passed on to their customers in the form of reduced LTL shipping costs, which will encourage more customers to book LTL loads with NTG.

4.3 Approach

Our approach for finding shipments to combine into routes is to first cluster orders that are along the same path of travel, then order the stops in each cluster. Determining the best routing for all shipments at once would require significant computing power and time to solve. Instead, our method uses iterative match-finding techniques to identify clusters within a reasonable runtime. Once the loads are clustered, an optimization model orders the origins and destinations into a feasible route, minimizing travel distance subject to the following constraints:

- Each origin and destination must be visited exactly once.
- Each shipment origin must be visited before its corresponding destination.
- The capacity of a truck may never be exceeded.

Once the optimal order of stops has been found, the amount of truck space required to complete the route is calculated. The amount of space required is equal to the maximum volume of a truck needed on any leg of the trip. This is used to determine whether a combined route will be shipped as a partial or FTL delivery. The shipment method determines the cost associated with completing the combined route.

Because matches are found iteratively, the resulting combinations of routes can change depending upon iteration order. To find the optimal set of combinations, the model repeats the clustering and routing process for ten different orderings of the initial shipments to determine the greatest value without greatly increasing runtime.

The combination of routes found by the model that results in the most savings is kept and displayed in a user interface. Unprofitable routes are left ungrouped to be sent as individual LTL or partial shipments. For a more detailed explanation of the analysis used to combine LTL and partial routes, refer to Appendices C.1 - C.3.

4.4 Results

Running the routing model on the 2018 January and February data finds an average of 14.9 combined routes per day, of which 7.4 (or 49.9%) are determined to be profitable. The average cost savings for these routes compared to the shipping costs for the individual loads is $287.35. An average of 3.5 individual loads are combined into a combined route. The calculations of these values are included in Appendix C.4.

4.5 Deliverable

For NTG to implement this product, they will need to connect their database to the Python program that produces the user interface. They could do this either by querying their database directly from the Python program or by downloading the data from their database into CSV files that can be easily read into the program. Another step that NTG should take to implement this model is to use PC*MILER, a tool that NTG uses to determine exact distances, instead of using the distance matrix between 3-digit zip code areas that this model uses.

Implementation of the LTL routing model will also require NTG to hire a Negotiator, who will run the program, analyze the results, and arrange for carriers to deliver the combined routes. Because the model takes approximately 30 minutes to find matches for a day's worth of data, the Negotiator should run the program once per day to determine profitable combined routes. Since most of customers' LTL orders are placed in the morning, this model can be run on a day's worth of loads in the early afternoon.

The user interface includes cost and distance information for both the combined route and the individual shipments it would replace. It also displays the stops and delivery dates for the proposed route, which are useful pieces of information for the Negotiator to know when booking the combined route. Figure 4.2 shows an example of the user interface displaying possible route combinations.

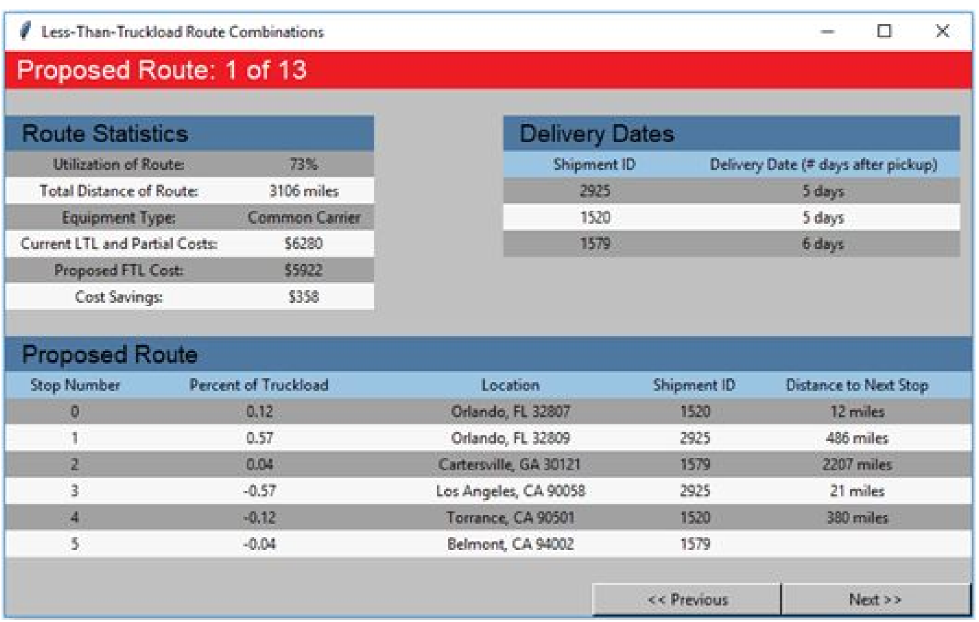

Figure 4.2. Example user interface displaying results for the less-than-truckload routing model.

4.6 Value

Simulation of the design model over LTL and partial data from 2018 January and February results in a 5.27% increase in LTL profit margin, which translates to an annual gross profit increase of $527,000. Implementation of the LTL routing model will require NTG to purchase a Gurobi optimization software license and to hire a Negotiator to interact with carriers to book combined routes. Accounting for these expenses, the net value of the LTL model is $453,000 for the first year and $467,000 for each subsequent year.

There is additional saving potential for NTG if they focus on customers placing orders along popular routes, and actively growing their LTL and partial business. The detailed calculations of the valuation of the Less-Than-Truckload part of the project are in Appendix C.4.

5. Drayage

5.1 Current System

NTG's Drayage department deals with import and export shipments, loads coming into and going out of the country via ocean containers. These containers are owned by the steamship lines and are stored at the seaports where they operate. For that reason, regardless of the type of the load (import or export), a carrier will always have to begin a drayage trip at a port. For import loads, the port pick-up is a full container ready to be delivered, and for export loads, the port pick-up is an empty container ready to be filled. This process is pictured in Figure 5.1.

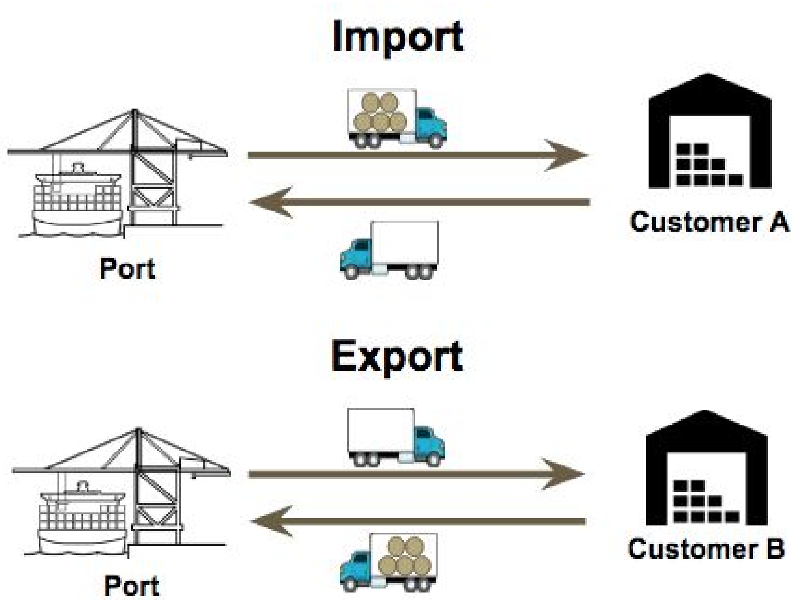

Figure 5.1. Diagrams representing the current drayage system for import and export deliveries.

Drayage line-hauls are round-trips dedicated to each independent import or export delivery. Since the ocean steamship lines own the containers and require each container to return to the port where its trip begins, the current system contains the inefficiency of every round-trip having a leg in which the container is empty.

5.2 Opportunity

NTG has recognized the possibility of combining import orders with export orders to produce cost savings by having carriers perform the route shown in Figure 5.2. When NTG recognizes two loads are feasible to pair, based on experience and constant market research, they can arrange for the shipments to be matched. Due to the difficulty of matching loads, NTG has only been able to realize these pairings five or fewer times per month, which, on average, saved them $500 per match (value determined by NTG's internal metrics).

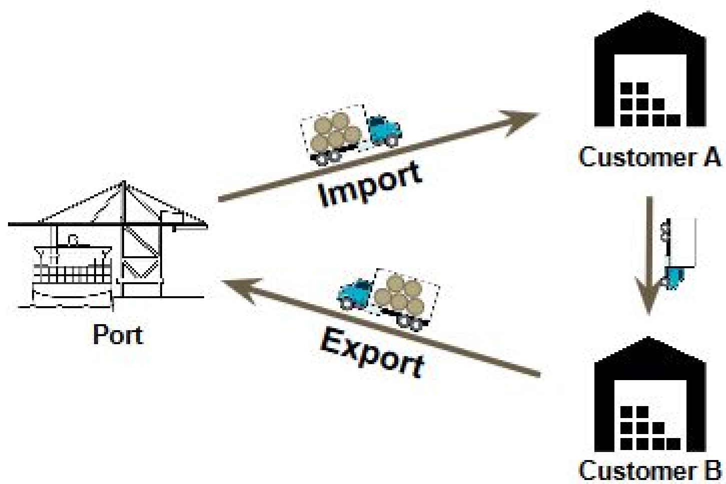

Figure 5.2. Diagram representing the proposed drayage system for Import-Export matches.

A tool that NTG can use to automatically identify crucial parameters among all active loads in their system and display all possible pairings will make finding import-export matches easier. NTG needs a software solution that will seamlessly connect to NTG's in-house platform, Freight Hawk Dispatch, and non-invasively find drayage matches. Developing a pairing algorithm to systematically display the best matches allows the Bookers at NTG to have the freedom to choose which loads to pair and then issue a divergence document, which legally binds an import load to an export load and is necessary for carriers to pick up and release orders.

5.3 Approach

Our approach to design this software solution considers all parties involved in drayage trips: NTG, drayage clients, partner carriers, and steamship companies. Specifically, each of the following items must be satisfied:

- This solution must be user-friendly for NTG's current staff.
- Drayage clients should experience the same excellent service that they currently receive.
- Partner carriers must agree to combined routes: all divergences must be reasonable.
- Steamship company authorities must approve each pairing.

This solution for finding import-export matches connects to Freight Hawk Dispatch and displays possible drayage matches for approval. The following steps describe the process for finding possible matches for a single drayage load:

1. Retrieve information for the current load
2. Determine basic key parameters for matching loads
3. Use basic match parameters to build an SQL query and return a table of possible matches
4. Filter the returned list using a date constraint
5. Sort the remaining matches by deadhead distance
6. Return matching load information

The basic matching requires an import load be matched with an export load because the goods for an import are located at the port, and the goods for an export are not at the port. Two imports cannot be matched with each other because that combination of loads would exceed the capacity of a truck during the leg from the port to the drop-off location. For that same reason, two exports cannot be paired, either. Also, the port city and steamship line must be the same because steamship companies own the containers and require that they be delivered back to their port of origin.

The complex matching requires the import drop-off and export pick-up dates to be within two days of each other, which allows NTG to keep their current level of customer service. Additionally, the overall distance of the paired loads must be less than the distance of completing each load independently. Presented with the filtered matches sorted by deadhead distance, the distance between import drop-off and export pick-up, NTG can choose to book the best load combination. Refer to Appendix D.1 for a detailed analysis of this matching algorithm.

Implementing data validation procedures is necessary to guarantee a successful use of this pairing tool. To find import-export pairings, the following load parameters must exist and be correctly entered in NTG's database:

- Import/Export field
- Port city zip code
- Steamship line
- Type of shipping equipment
- Dates:
    - Import: estimated time of arrival (ETA) and last free day (LFD)
    - Export: earliest receiving date (ERD) and last day to drop-off load (CUT)

The first four parameters are necessary for the basic matching, and the date parameters are required for the complex matching. Appendix D.2 contains the details explaining the importance of the date parameters, and Appendix D.3 explains the data validation implementation.

5.4 Results

Simulation of this model on 2017 and early 2018 data showed that implementation of this model will give NTG an average of 55 matches per month, which is approximately 50 more matches per month than NTG has currently been finding.

5.5 Deliverable

This model is designed as a Python RESTful API, which allows the model to be implemented in Python and still communicate with NTG's C# framework. This design keeps the solution as a module outside NTG's framework, so changes can be made if necessary without interfering with NTG's other Freight Hawk Dispatch operations.

NTG also has the freedom to pick which loads to pair when a new load is entered into the system. Once a match is finalized, NTG files a divergence document to legally bind the import and export, and finds a carrier who will agree to transport the combined trip.

This API had been demoed to the NTG developer team, and NTG's developers will be including this tool in their next IT sprint in May 2018. The data validation recommendations needed for this model are foundational changes, and will not be implemented until NTG's biannual major Freight Hawk Dispatch update in August 2018.

5.6 Value

The current annualized cost savings of this tool are estimated to start at $300,000 (50 matches per month at $500 per match); however, the number of matches will scale proportionally with load volume. As NTG continues its dramatic growth, drayage pairing will increase. Additionally, NTG plans to identify commonly matched shipping lanes and steamship lines, and will push more clients to use those. Finally, the improved data validation necessary for this program will also improve the overall quality of NTG's database and capability to conduct future analytics.

6. Final Summary

Our mission was to refine and expand NTG's infrastructure to accommodate their rapid growth. We accomplished this by attacking major objectives across each of NTG's services in the following ways:

- **Atlanta Enterprise**: Revised NTG's staffing model to an optimal one, which NTG can implement in all 12 NTG offices across North America
- **Less-Than-Truckload**: Constructed a new product for NTG to consolidate shipments and actively seek out both LTL and partial shipments from their customers
- **Drayage**: Provided NTG with a tool to easily capture cost savings by combining imports and exports

We have provided NTG with the necessary information for Atlanta Enterprise to reorganize and accept an additional 195 loads per day. Our solution will give NTG annual cost savings in the Less-Than-Truckload and Drayage departments of $767,000, both of which will increase as NTG continues growing and actively targeting popular shipping lanes.

Appendix A: Glossary

C#: object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications (source: https://www.webopedia.com)

Freight Hawk Dispatch: NTG's main internal platform used to input, edit, and track load information

FTL: full truckload, a type of trucking where the freight takes up the entire trailer

JSON: lightweight data-interchange format. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others; these properties make JSON an ideal data-interchange language (source: https://www.json.org/)

LTL: less-than-truckload, a type of trucking where the freight takes up less than the entire trailer.

NTG: Nolan Transportation Group, https://ntgfreight.com/

Pandas: Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive (source: https://pandas.pydata.org/pandas-docs/stable/index.html)

Python: an interpreted, object-oriented, high-level programming language with dynamic semantics (source: https://www.python.org)

RESTful API: an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data (source: https://searchmicroservices.techtarget.com/definition/RESTful-API)

Slack: team collaboration tool where conversations are shared, organized, and accessible (source: https://slack.com/)

SQL: standardized query language for requesting information from a database (source: https://www.webopedia.com)

Appendix B: Atlanta Enterprise Process Improvement

The current distribution of employees in Atlanta Enterprise is suboptimal. Simulation using a Simio optimization model shows the optimal distribution of employees to maximize load capacity.

B.1 Current System

Account Managers: The Account Managers are the first connection between NTG and the customer. They spend time talking to customers, developing good relationships, and addressing any concerns the customer has. An Account Manager will get a request for a shipment either via email or phone and then send the request information to the Builder who creates the load in Freight Hawk.

Builders: The Builder's job is to take the information from the Account Manager and create the load in Freight Hawk, which is a time-intensive process because there is a lot of information that needs to be manually entered. If multiple loads need to be built, a Builder will input the one with the shortest lead time first.

Schedulers: The Scheduler's job is to set up appointment times for the load to be picked up and dropped off, which involves correspondence with both the shipper (origin) and the consignee (destination). Schedulers also prioritize loads based on the shortest lead time. Unlike Account Managers and Builders, the Schedulers are not necessarily finished with a load after working on it. Problems can occur at the Trackers or the Bookers that require a load to go back to the Schedulers to be reworked. For example, if a truck is running late and not going to make the appointment time, a Tracker will send the load back to the Schedulers to call the shipper to make sure that it is acceptable for the load to arrive late.

Bookers: The Bookers are tasked with finding a carrier to run each of the loads. They do this by identifying potential carriers, using the company contact database, and calling them to negotiate a lane rate. It is also possible that a carrier will call NTG inquiring about taking a load. In this scenario, the processing time for a load is less than 5 minutes. Some loads are harder to book either due to appointment times or the origin and destination being undesirable locations for carriers to drive to. As a result, these loads are still not booked late in the day and require the Bookers to rework them until a carrier is found.

Trackers: The Trackers call the carrier to make sure that there are no problems with the load. Problems can include the carrier being late, not having the right equipment for transport, or not showing up to pick up the load. Trackers will call each carrier anywhere from 5 to 10 times until a load has been delivered.

B.2 Simio Implementation

Account Manager: Sending a customer load from the Account Manager to a Builder initiates the system, and is thus represented as a Source object in Simio. The time-varied arrival rate is calculated by using averages of the number of loads requested each hour of the day for the past 60 days. On average, the Account Managers receive 84 loads/day.

The following are modeled as Server objects in Simio:

Builders: The process time for a Builder is based on data collection of how much time a Builder needed to create a load. Once a load is built, it moves on to the Schedulers.

Schedulers: The processing time for a Scheduler working on a load is based on time needed to schedule a load. Once a load is processed by the Scheduler server, it travels to either the Waiter server or the Bookers so that they can find a carrier to transport the load. If the lead time is greater than zero, the load goes to the Waiter.

Waiter: The Waiter server in the Simio model is not a physical person or process but is included in the model because Bookers only work on loads that have zero lead time (loads that are scheduled to be picked up that day). The Waiter server acts as a buffer to hold the loads until their pick-up day by deducting a day from the lead time once per day.

Bookers: Loads vary in their booking difficulty dependent upon aspects of the load such as the lane or time of day. Bookers process loads by increasing processing time. Once they find a carrier to take a load, the Bookers send it to the Trackers.

Trackers: An Entity state called "Times Tracked" is introduced when the load enters the Trackers server. This represents the number of times the load is serviced by the Trackers. Each time a load gets tracked, its number of Times Tracked is increased by one. If a truck will be late, the load will be sent to the Schedulers to make sure the lateness is acceptable with the point of pick-up or drop-off. If not, the load will be sent to the Bookers for rebooking.

Service times for Builders, Schedulers, and Trackers are modeled by spline interpolated distributions, and the service time for Bookers is modeled by a Gamma distribution with mean inversely proportional to the number of available loads.

The distribution of lead times for orders in Atlanta Enterprise are shown in Figure B.1.

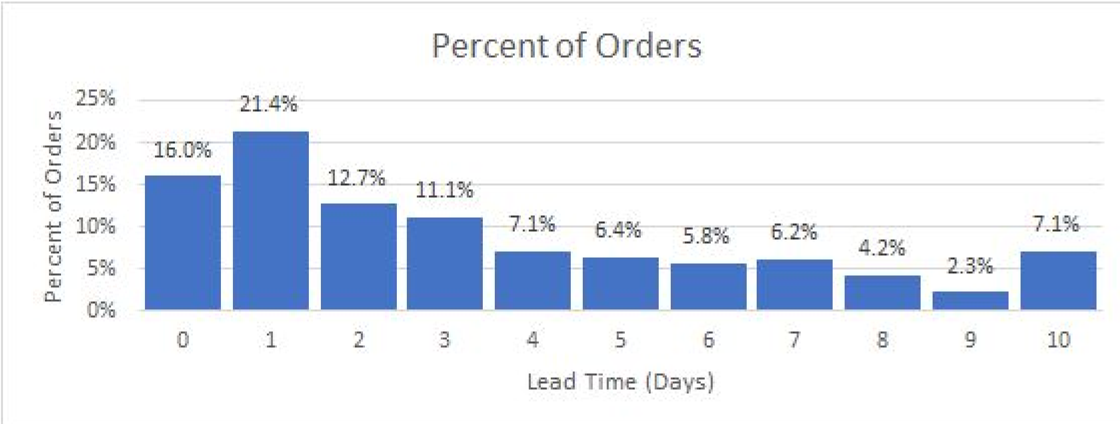

Figure B.1. Distribution of lead times for orders processed by Atlanta Enterprise.

NTG supplied the following travel relations for a load, which represent the workflow in the model:

- If the load is at Builders:
  - 100% to Schedulers
- If the load is at Schedulers:
  - If first time at Schedulers:
    - 100% to Bookers
  - If the load has been to Bookers, but not to Trackers (the load is hard to book and the Bookers want to see if there is any flexibility):
    - 100% to Bookers
  - If the load has already been to Trackers and Bookers (load was originally booked but a problem occurs, like lateness):
    - 85% to Trackers (lateness is acceptable with customer)
    - 15% to Bookers (truck is not going to pick-up and they must find a new carrier)
- If the load is at Waiter:
  - If the wait time is greater than one day
    - 100% to Waiter
  - If the wait time is less than one day (day of delivery)
    - 60% to Bookers
    - 40% to Trackers
- If the load is at Bookers:
  - If first time at Bookers:
    - 75% to Trackers (booked the load)
    - 25% to Schedulers (need more flexibility)
  - If second time at Bookers, and has not been to Trackers (could not book it the first

time and got more flexibility from Schedulers):
        - 97% to Trackers
        - 3% to Schedulers (still need more flexibility)
    - If already been to Trackers (the original truck is going to be late):
        - 100% to Trackers (priority to book)
- If the load is at Trackers:
    - Number of times being tracked is 1-4 (before pick-up):
        - 85% back to Trackers (for follow up)
        - 10% to Schedulers (verify late pick-up is acceptable)
        - 5% to Bookers (late pick-up is not acceptable and load needs to be rebooked)
    - Number of times being tracked is 5-9 (after the pick-up):
        - 90% back to Trackers (for follow up)
        - 10% to Schedulers (verify late delivery is okay)

B.3 Proposed System

Running an optimization on the current system using Simio's built-in optimization software determined the optimal number of employees at each role. The following constraints are used in this solution:

- The current number of staffers in Atlanta Enterprise (43) is the upper bound for the number of employees.
- The arrival rate is set to infinity to determine the capacity of the system.
- The number of employees at each position is treated as a decision variable.
- The objective is to maximize the total throughput of the system.
- Ten trials will be run for each scenario, which will provide an accurate analysis while keeping the runtime of the optimization small.

The optimal distribution of staffers determined by running this Simio optimization model through a substantially large number of scenarios is 6 Builders, 4 Schedulers, 18 Bookers, and 15 Trackers. This takes employees from roles with a low utilization and moves them to those with a higher utilization, ultimately increasing the system capacity. Using this distribution, Atlanta Enterprise will be able to process 280 loads per day, which is a 52% increase in production over the current system. Table B.1 illustrates the comparison between the current system and the proposed system.

	Load Counts with Current Staffing	Load Capacities with Current Staffing	Load Capacities with Reorganized Staffing
Input Loads/Day	84	184	280
Output Loads/Day	84	184	280
Utilizations (# of Servers)			
Builders	40% (3)	99% (3)	74% (6)
Schedulers	47% (2)	100% (2)	71% (4)
Bookers	69% (20)	96% (20)	95% (18)
Trackers	25% (18)	52% (18)	92% (15)

Table B.1. Current and proposed staffing distributions, load capacities, and utilizations.

Appendix C: Less-Than-Truckload Routing

The less-than-truckload routing model consolidates less-than-truckload (LTL) and partial shipments into combined loads. Depending upon amount of truck space needed, these grouped shipments can be sent as either partial or full-truckload (FTL) deliveries. This model uses a three-step process for combining shipments: clustering, shortest path integer program, and cost comparison.

C.1 Clustering

The clustering portion of the model uses a decision tree to quickly group shipments based on the following factors:

1. The routes that can be matched are limited to those currently in the database as "active loads" (loads that NTG has agreed to deliver but has not found a carrier to deliver). This reduces the complications of trying to re-route drivers who have already committed to certain routes.
2. The loads being tested for matching are not contained in other routes.
3. The type of equipment is the same for all loads in a combined route. For example, goods needing refrigeration can only be delivered in refrigerated vehicles.
4. Two routes, Route 1 and Route 2, are compatible for matching if all stops on Route 2 are along the travel path of Route 1. Each stop can satisfy this constraint in one of two ways:
    a. A stop on Route 2 is within a 100-mile radius of a stop on Route 1.
    b. The distance increase by adding a stop to Route 1 between Stop A and Stop B does not increase the total route distance by more than 30% of the distance between Stop A and Stop B, the travel path it would replace.

The travel distances used in this model came from Google Maps Distance Matrix API, which gives the distance between 3-digit zip code areas. Since NTG uses to PC*MILER to determine exact distances, they can determine more accurate travel distances when they implement this model in their system.

C.2 Shortest Path Integer Program

Each feasible cluster of shipments is then routed using a linear integer program to determine the optimal shortest travel path. Figure C.1 shows an illustration of the model. Each order corresponds to two nodes: an origin and a destination. Dummy start and end nodes are added to each cluster so the model can solve for the shortest path without being constrained to starting or ending at a specified node. The arcs between nodes represent the cost (travel distance) between locations. The costs of the arcs going from the dummy start node to any origin are 0, and the arc costs from the destination nodes to the dummy end node are 0.

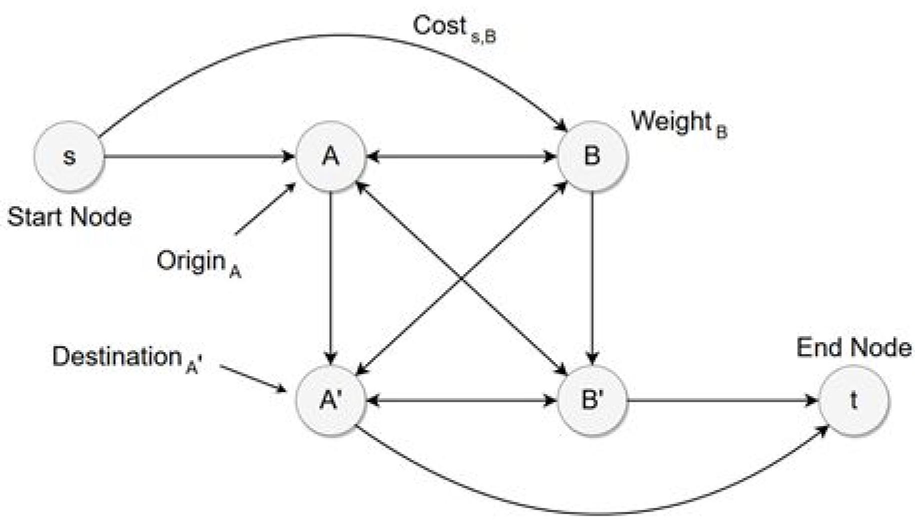

Figure C.1. Graph representing the model implemented as a shortest path integer program used to find the optimal order of stops.

Arcs that are not listed in the model are added to the integer linear program with large costs (represented by M in the following model) to simplify the definition of constraints without having any chance of including these arcs in the optimal solution. Running an integer linear programming optimization model to determine the shortest path between nodes in a route requires definitions for the node sets, decision variables, and parameter values.

Node Set Definitions:

n : number of deliveries (origin / destination pairs) satisfied in a single route

s : start node (inserted before the beginning of the route)

t : end node (inserted after the end of the route)

$V = \{1, 2, \dots, n, n+1, n+2, \dots, 2n, s, t\}$: shortest path nodes ordered such that $(i, n+i)$ is the origin / destination pair for a delivery that needs to be routed, $\forall\, i = 1, \dots, n$

$m = 2n + 2$: total number of nodes

$O = \{0, 1, \dots, m-2, m-1\}$: orders for the nodes in the route

Decision Variables:

$y_{i,j}$: 1 if the arc i to j is used in the route, 0 otherwise, $\forall\ i, j \in V$

$x_{i,k}$: 1 if node i in the route is visited in order k in the route, 0 otherwise, $\forall\ i \in V, \forall\ k \in O$

Parameter Values:

M : arbitrarily large constant (can be thought of as infinite, but must be represented as a finite number when included in programming)

$c_{i,j}$: arc cost to go from node i to node j, $\forall\ i, j = 1, \dots, 2n, \quad i \neq j$

$c_{s,j} = 0$: allows any origin node to start the route, $\forall\ j = 1, \dots, n$

$c_{i,t} = 0$: allows any destination node to end the route, $\forall\ i = n+1, \dots, 2n$

$c_{i,i} = M$: eliminates the possibility of going from one node to itself $\forall\ i \in V$

$c_{i,s} = M$: does not allow an arc entering the start node to be selected, $\forall\ i = 1, \dots, 2n$

$c_{i,t} = M$: does not allow an origin node to end the route, $\forall\ i = 1, \dots, n$

$c_{s,j} = M$: does not allow a destination node to start the route, $\forall\ j = n+1, \dots, 2n$

$c_{t,j} = M$: does not allow an arc departing the end node to be selected, $\forall\ j = 1, \dots, 2n$

$c_{s,t} = M$: does not allow an arc from the start node to the end node to be selected

$c_{t,s} = M$: does not allow an arc from the end node to the start node to be selected

Q : capacity of a truck

q_i : quantity picked up at a node (negative if dropped off)

Integer Linear Program:

$$min \quad \textstyle\sum_{i \in V} \sum_{j \in V} c_{i,j}\, y_{i,j} \tag{1}$$

$$s.t. \quad \textstyle\sum_{i \in V} x_{i,k} = 1 \qquad \forall\ k \in V \tag{2}$$

$$\textstyle\sum_{k \in O} x_{i,k} = 1 \qquad \forall\ i \in V \tag{3}$$

$$x_{i,k} + x_{j,k+1} - 1 \leq y_{i,j} \qquad \forall\ i, j \in V, \quad \forall\ k = 1, \dots, m-1 \tag{4}$$

$$x_{i+n,k} \leq \textstyle\sum_{\ell=1}^{k-1} x_{i,\ell} \qquad \forall\ k \in O \tag{5}$$

$$\textstyle\sum_{\ell=1}^{k} \sum_{i \in V} q_i x_{i,\ell} \leq Q \qquad \forall\ k \in O \tag{6}$$

$$\textstyle\sum_{j \in V} y_{s,j} = 1 \tag{7}$$

$$\textstyle\sum_{i \in V} y_{i,t} = 1 \tag{8}$$

$$x_{s,0} = 1 \tag{9}$$

$$x_{t,m-1} = 1 \tag{10}$$

$$x_{i,k} \in \{0,1\} \qquad \forall\, i \in V, \quad \forall\, k \in O \tag{11}$$

$$y_{i,j} \in \{0,1\} \qquad \forall\, i,j \in V \tag{12}$$

The objective function (1) minimizes the total cost for the routing of the nodes. Constraint (2) requires that each order in the route is used exactly once. Constraint (3) requires that all nodes receive orders in the route. Constraint (4) forces an arc to be included from node i to node j in the order in the route if node i comes immediately before the order for node j. Constraint (5) forces all route origins to be visited before their corresponding destinations. Constraint (6) keeps the capacity of a truck from being violated during the trip. If the total volume of goods being delivered on a route exceeds the capacity of a truck, some goods must be dropped off before all goods are picked up to keep the volume of necessary space in the truck within capacity. Constraint (7) forces exactly one arc to depart from the start node. Constraint (8) forces exactly one arc to enter the end node. Constraints (9) and (10) make the start node and end node be the first and last nodes in the ordering. Constraints (11) and (12) require that each order and each arc to either be used once or not included.

C.3 Cost Comparison

The cost of a combined route produced by the shortest path integer program is compared to the costs associated with sending each load as if it would have been sent otherwise, either as an LTL or partial delivery. Because truck drivers want to receive the revenue associated with delivering a full load, they fill up their trucks with as many loads as they can find on their routes by finding shipments from various providers. The cost of each combined route from this model is determined by the "utilization" of that route, which is the maximum volume of a truck that is required to complete that route.

Routes with a high utilization take up too much room on the truck for the driver to find another shipment to deliver at the same time and must be booked as full-truckload deliveries. However, routes with moderate utilization can be booked as partial deliveries because they leave enough space on the truck for the driver to find another shipment to deliver at the same time. In this routing model, if the utilization of a route is greater than 70%, the combined route is priced as a full-truckload delivery. If, instead, the utilization is between 40% and 70%, inclusive, the combined route is priced as a partial delivery. Any route with a utilization less than 40% is not combined, and each shipment of this type is broken apart to be booked as an individual LTL or partial delivery. The cost of sending a combined load as a full-truckload or partial shipment is determined by the following formulas:

$$\textit{FTL Cost} = \textit{Lane Rate Per Mile} * \textit{Total Distance} + \textit{Per-Stop Cost}$$

$$\textit{Partial Cost} = \textit{FTL Cost} * \textit{Utilization}$$

The lane rate is the cost of an FTL delivery being shipped from one city to another. Lane rates are based on both distance and the ability of a truck driver to find a load into and out of each city. Thus, the appropriate cost for a FTL delivery in this model depends on the lane rate per mile for the lane associated with the origin and destination of the combined route. A per-stop cost is then added to account for the time and effort associated with the additional stops the driver would need to make to complete the route. The cost of a partial delivery is equivalent to the cost of an FTL delivery multiplied by the amount of truck space needed for that load.

C.4 Results and Value Calculations

Table C.1 shows the daily statistics for the LTL routing model. “Routes” are combined routes that are left after the shortest path integer program, and “Negotiations” are routes that are left after the cost comparison. “Ships in Negs” is the number of shipments consolidated into combined routes. Table C.2 shows the calculations of the result statistics.

2018 Improvement Data

	Day 1					Day 2				
Week	Shipments	Savings	Routes	Negotiations	Ships in Negs	Shipments	Savings	Routes	Negotiations	Ships in Negs
2	63	$3,324	14	7	21	80	$485	16	9	29
3	55	$3,333	10	5	16	77	$5,498	19	7	24
4	82	$2,647	14	7	27	99	$5,776	21	15	57
5	76	$7,399	15	9	35	87	$2,646	19	9	25
6	78	$4,538	16	10	29	102	$7,398	23	8	35
7	67	$6,257	14	6	21	85	$5,017	20	11	39
8	54	$1,621	11	5	22	61	$2,390	12	4	15

	Day 3					Day 4				
Week	Shipments	Savings	Routes	Negotiations	Ships in Negs	Shipments	Savings	Routes	Negotiations	Ships in Negs
2	57	$5,191	12	9	33	71	$2,840	15	7	23
3	53	$2,561	10	3	15	89	$4,990	21	8	29
4	95	$4,852	23	17	57	96	$3,535	19	11	41
5	102	$3,818	24	13	46	82	$9,441	18	9	30
6	80	$5,883	18	8	32	75	$7,516	15	3	8
7	72	$4,145	16	6	17	69	$4,436	15	8	26
8	69	$8,963	17	9	30	82	$1,916	21	10	31

	Day 5				
Week	Shipments	Savings	Routes	Negotiations	Ships in Negs
2	86	$6,362	19	8	29
3	98	$16,912	21	14	53
4	75	$2,100	19	6	20
5	94	$6,915	20	9	27
6	68	$2,883	16	8	28
7	80	$5,611	16	9	39
8	67	$1,776	16	10	31

Table C.1 Table showing daily less-than-truckload statistics.

Total Savings:	$170,975		Average Savings per Route:	(Savings/Negotations)	$287.35
Total Routes:	595		% of Routes that are Profitable:	(Negotiations/Routes)	49.90%
Total Negotiations:	297		Average # Shipments per Route:	(Shipments in Neg/Negs)	3.5
			Average # Negotiations per Day:	(Total negs/# of days)	7.4
Total Number of Shipments:	2,726		Average # Shipments per Day:	(Shipments/# of days)	68
Total Shipments in Negotiations:	1,040		Average # Routes per Day:	(Routes/# of days)	14.9

Table C.2 Table displaying less-than-truckload route statistics calculations.

NTG had a $400 million revenue run rate in 2017, of which $10 million, or 2.5%, was LTL revenue (value determined by NTG internal metrics). NTG scaled the data provided for FTL lane rates and both LTL and partial costs by a common factor to keep data manipulation accurate while preserving proprietary data. Running the LTL routing model on the 2018 January and February scaled data gives a 5.27% increase in profit margin, from 20.62% to 25.88%. The calculations for these values are shown in Table C.3 and Table C.4.

	2018 Improvement Data									
	Day 1					Day 2				
Week	Shipments	Revenue	Cost	Savings	Cost After	Shipments	Revenue	Cost	Savings	Cost After
2	63	$69,900	$54,500	$3,324	$51,176	80	$78,300	$64,500	$485	$64,015
3	55	$55,900	$46,400	$3,333	$43,067	77	$84,600	$63,600	$5,498	$58,102
4	82	$90,500	$66,700	$2,647	$64,053	99	$89,600	$70,700	$5,776	$64,924
5	76	$78,200	$62,400	$7,399	$55,001	87	$115,110	$88,700	$2,646	$86,054
6	78	$76,400	$59,100	$4,538	$54,562	102	$143,500	$105,800	$7,398	$98,402
7	67	$88,400	$67,500	$6,257	$61,243	85	$97,000	$78,100	$5,017	$73,083
8	54	$67,300	$53,300	$1,621	$51,679	61	$89,000	$71,100	$2,390	$68,710
	Day 3					Day 4				
Week	Shipments	Revenue	Cost	Savings	Cost After	Shipments	Revenue	Cost	Savings	Cost After
2	57	$58,600	$48,400	$5,191	$43,209	71	$84,800	$69,500	$2,840	$66,660
3	53	$58,400	$44,700	$2,561	$42,139	89	$112,300	$89,200	$4,990	$84,210
4	95	$95,300	$79,400	$4,852	$74,548	96	$100,200	$79,000	$3,535	$75,465
5	102	$104,700	$83,000	$3,818	$79,182	82	$91,400	$71,500	$9,441	$62,059
6	80	$100,300	$82,200	$5,883	$76,317	75	$92,400	$72,300	$7,516	$64,784
7	72	$92,200	$76,300	$4,145	$72,155	69	$101,900	$79,300	$4,436	$74,864
8	69	$90,500	$70,700	$8,963	$61,737	82	$118,300	$88,300	$1,916	$86,384
	Day 5									
Week	Shipments	Revenue	Cost	Savings	Cost After					
2	86	$93,500	$82,200	$6,362	$75,838					
3	98	$115,000	$97,400	$16,912	$80,488					
4	75	$104,300	$85,300	$2,100	$83,200					
5	94	$124,100	$102,100	$6,915	$95,185					
6	68	$87,500	$67,600	$2,883	$64,717					
7	80	$116,300	$91,600	$5,611	$85,989					
8	67	$80,200	$64,300	$1,776	$62,524					

Table C.3. Table displaying daily cost savings of the less-than-truckload model for 2018 data.

	2018 Summary			
	Revenue	Cost	Profit	Margin
Current	$3,245,910	$2,576,700	$669,210	20.62%
Improved		$2,405,725	$840,185	25.88%
	Applied to 2017 Summary			
	Revenue 2017	Cost 2017	Profit 2017	Margin
Current	$10,000,000	$7,938,298	$2,061,702	20.62%
Improved	$10,000,000	$7,411,558	$2,588,442	25.88%

Valuation Calculation	
Margin Increase:	5.27%
Total Savings:	$526,740
Other Costs	
Negotiator Salary:	$60,000
Final Savings	
$466,740	

Table C.4 Table displaying less-than-truckload valuation calculations.

Figure C.2, displays the weekly number of LTL shipments for 2017; Figure C.3 shows the same data with outliers removed, corresponding to reduced shipping during the weeks including Thanksgiving and Christmas. These figures show that seasonality does not apply to this portion of the shipping industry, so using 2018 January and February data is accurate for analysis of the entire year.

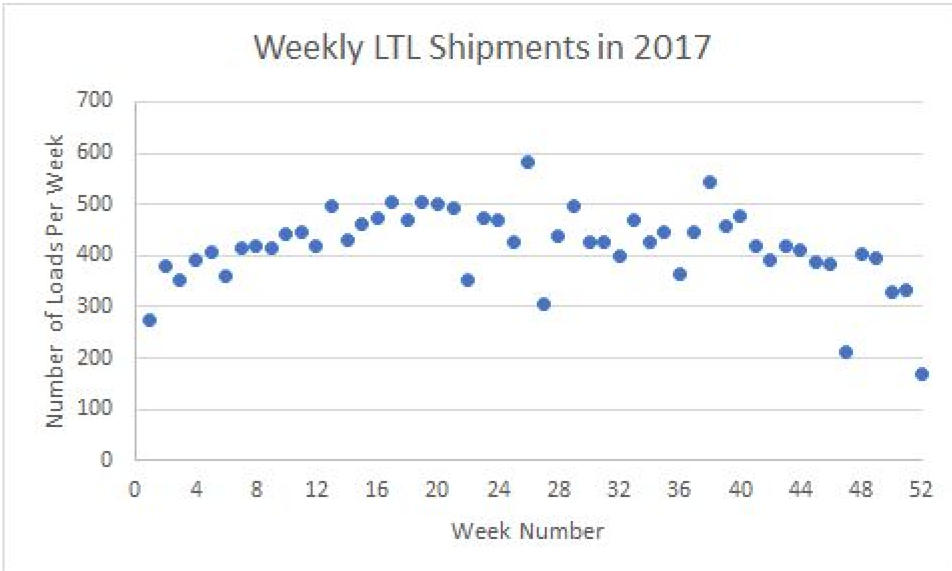

Figure C.2. Graph showing the weekly number of Less-Than-Truckload shipments for 2017.

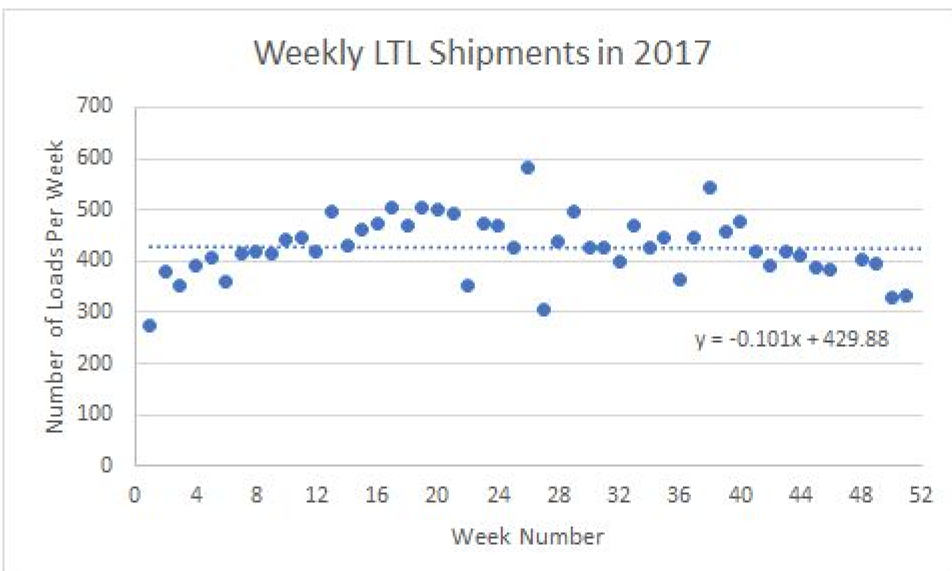

Figure C.3. Graph showing Less-Than-Truckload shipments for 2017, with outliers removed.

Applying the 5.27% increase in profit margin to the $10 million revenue for 2017 gives a yearly gross profit increase of $527,000. Implementation of the LTL routing model will require NTG to purchase a Gurobi optimization software license and to hire a Negotiator to interact with carriers to book combined routes. The Gurobi license will be a $14,000 one-time expense, and the annual salary for a Negotiator will be $60,000, which includes healthcare and benefits. This translates to an annual net profit increase of $467,000. Accounting for the purchase of the Gurobi license, the net profit increase for the first year is $453,000.

C.5 Example Results

The following shipments are representative of individual routes that are processed by the LTL routing model:

Shipment IDs: ['768']
Ship Dates: [datetime.date(2018, 1, 19)]
Delivery Dates: [datetime.date(2018, 1, 22)]
Equipment: ['LTL - COMMON CARRIER']
Origins: ['98158']
Destinations: ['98424']
Percentage: [(0.12, 1)]
Distance: 30.44
Optimal Order:None
Utilization:0.12
Costs:[400]
Revenues:[500]
Cost:0
Cities:[('SEATTLE', 'WA'), ('FIFE', 'WA')]
Actions:
Date: 2018-01-19, Location: 98158, Percentage: 0.12, ID: 768
Date: 2018-01-22, Location: 98424, Percentage: -0.12, ID: 768

Shipment IDs: ['763']
Ship Dates: [datetime.date(2018, 1, 19)]
Delivery Dates: [datetime.date(2018, 1, 22)]
Equipment: ['LTL - COMMON CARRIER']
Origins: ['35210']
Destinations: ['35674']
Percentage: [(0.04, 1)]
Distance: 99.1208154932
Optimal Order:None
Utilization:0.04
Costs:[300]
Revenues:[300]
Cost:0

Cities:[('IRONDALE', 'AL'), ('TUSCUMBIA', 'AL')]
Actions:
Date: 2018-01-19, Location: 35210, Percentage: 0.04, ID: 763
Date: 2018-01-22, Location: 35674, Percentage: -0.04, ID: 763

Figure C.4 and Figure C.5 show the individual shipments and combined route for the following combined FTL route example that results from running the routing model on a single day's data:

Shipment IDs: ['826', '766', '803', '2804', '775']
Ship Dates: [datetime.date(2018, 1, 19), datetime.date(2018, 1, 19), datetime.date(2018, 1, 19), datetime.date(2018, 1, 19), datetime.date(2018, 1, 19)]
Delivery Dates: [datetime.date(2018, 1, 24), datetime.date(2018, 1, 25), datetime.date(2018, 1, 26), datetime.date(2018, 1, 25), datetime.date(2018, 1, 22)]
Equipment: ['LTL - COMMON CARRIER', 'LTL - COMMON CARRIER', 'LTL - COMMON CARRIER', 'VAN PARTIAL', 'LTL - COMMON CARRIER']
Origins: ['85043', '92121', '91746', '91761', '17050']
Destinations: ['17406', '07753', '01840', '18914', '02861']
Percentage: [(0.15, 1), (0.23, 1), (0.12, 1), (0.47, 1), (0.04, 1)]
Distance: 3114.93310191
Optimal Order:[0, 5, 7, 3, 1, 2, 9, 8, 4, 10, 6, 11]
Utilization:0.97
Costs:[600, 5500, 3900, 3200, 300]
Revenues:[600, 5900, 2100, 4500, 300]
Cost:7304.63760581
Cities:[('PHOENIX', 'AZ'), ('YORK', 'PA'), ('SAN DIEGO', 'CA'), ('NEPTUNE', 'NJ'), ('CITY OF INDUSTRY', 'CA'), ('LAWRENCE', 'MA'), ('ONTARIO', 'CA'), ('CHALFONT', 'PA'), ('MECHANICSBURG', 'PA'), ('PAWTUCKET', 'RI')]
Actions:
Date: 2018-01-19, Location: 85043, Percentage: 0.15, ID: 826
Date: 2018-01-24, Location: 17406, Percentage: -0.15, ID: 826
Date: 2018-01-19, Location: 92121, Percentage: 0.23, ID: 766
Date: 2018-01-25, Location: 07753, Percentage: -0.23, ID: 766
Date: 2018-01-19, Location: 91746, Percentage: 0.12, ID: 803
Date: 2018-01-26, Location: 01840, Percentage: -0.12, ID: 803
Date: 2018-01-19, Location: 91761, Percentage: 0.47, ID: 2804
Date: 2018-01-25, Location: 18914, Percentage: -0.47, ID: 2804
Date: 2018-01-19, Location: 17050, Percentage: 0.04, ID: 775
Date: 2018-01-22, Location: 02861, Percentage: -0.04, ID: 775

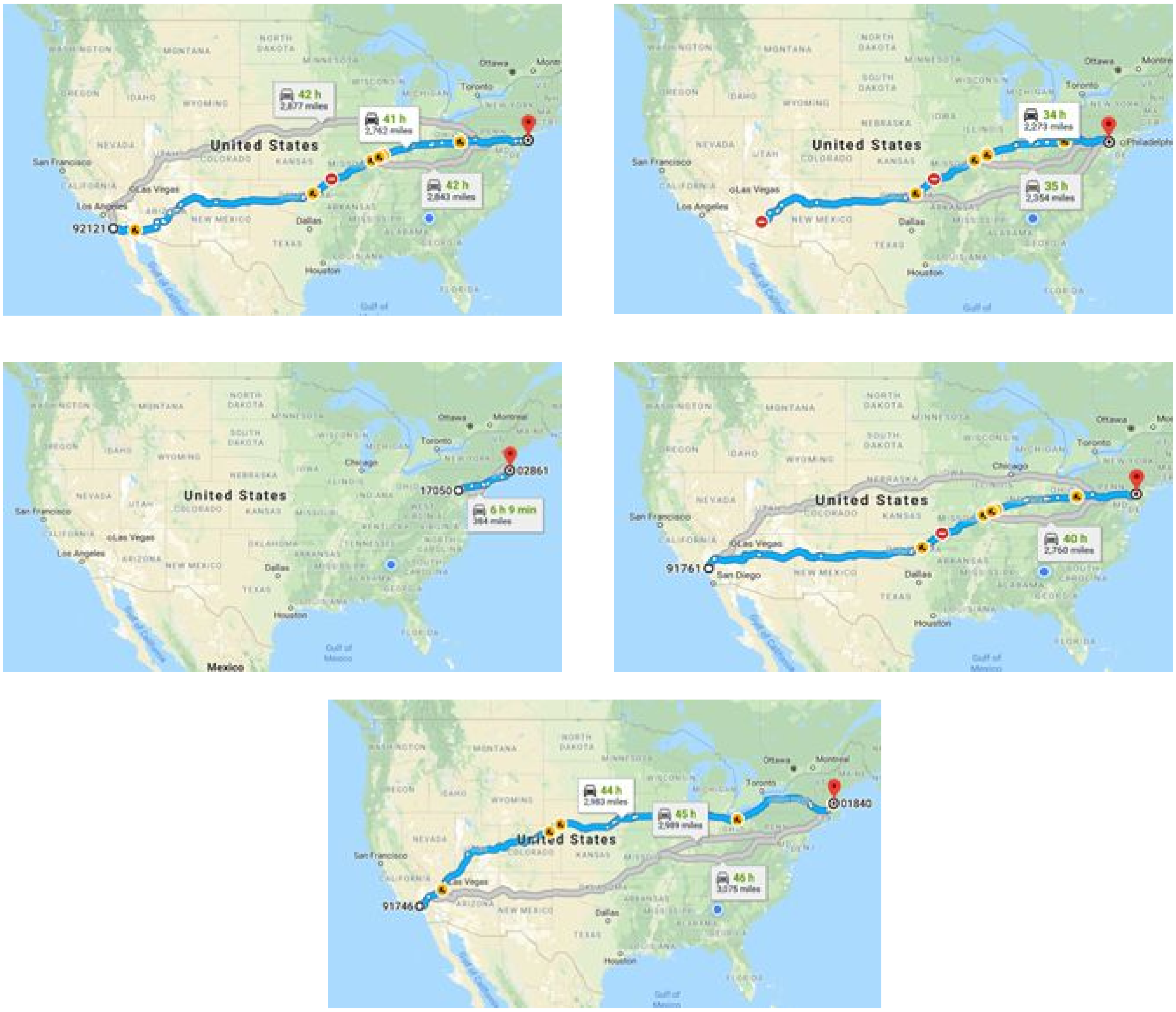

Figure C.4. Maps showing individual less-than-truckload and partial shipments for the example.

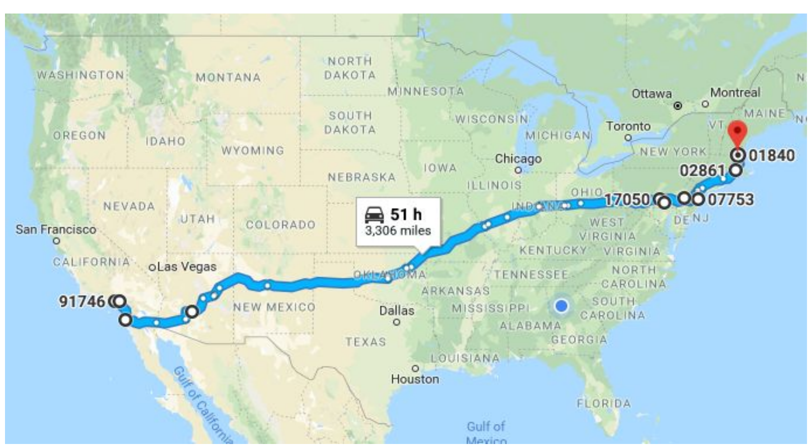

Figure C.5. Map displaying the combined full truckload route for the example.

Appendix D: Drayage Import-Export Matching

The drayage pairing model finds possible matches for a load that is entered into NTG's system. This algorithm implements both basic and complex constraints, and it requires data validation for key parameters.

D.1 Drayage Algorithm in Detail

The following is a brief description of the drayage matching algorithm:

1. Retrieve information for the current load
2. Determine basic key parameters for matching loads
3. Use basic match parameters to build SQL query and return table of possible matches
4. Filter the returned list using a date constraint
5. Sort the remaining matches by deadhead distance
6. Return matching load information

The following is a more thorough explanation of each step in the algorithm:

1. To obtain necessary information about the current load, this program queries NTG's database to determine all the load details.
2. Using the load details, the program sets the pair's import/export to the opposite value. The equipment field, steamship line, and port city required for matches are set to the target trip's respective values.
3. All of these elements are inserted into a single SQL statement to find basic matches from NTG's database. These results are temporarily stored in a pandas dataframe for manipulation.
4. The following date parameters must be met to keep potential loads for matching:
    a. The estimated time of arrival of import (ETA) must be before the last day to drop-off an export load (CUT) to ensure the import is the first of the two trips. This is mandatory because all drayage trips originate at the port city. This order is the only way to ensure the empty export leg is eliminated.
    b. The last free day for the import (LFD) must be no more than two days before the earliest receiving date for the export (ERD). The LFD is the last date that a shipyard will not charge a client for holding a shipping container at the port. For an import, the carrier must pick up the container by this date or the client will lose money to holding fees. The ERD is the first day an export container may be returned to the port city. If the difference between these is any more than two days, a carrier would be required to independently hold a full container, which is not acceptable for NTG's level of customer service.

5. All remaining loads in the dataframe are feasible pairs for the current load. These possible matches are sorted from least to greatest deadhead distance, where the deadhead distance represents the mileage from the import drop-off to the export pick-up. This newly constructed empty container leg replaces the empty legs that the individual import and export shipments would have had.
6. The sorted results are exported via JSON to Freight Hawk Dispatch, NTG's in-house platform for managing loads. NTG uses a template to display each possible pair and its information so the Booker can choose a match. Each pair of loads that are matched requires a divergence document to legally pick-up an export on the return trip of an import. Once a divergence document has been filed, the import and export have now been merged into a single trip, reducing the overall cost of the two trips by ~$500.

D.2 Load Type Date Constraints

Figure D.1 show the timeline for important dates for matching import and export loads. The ETA is when an import load should be at the port, and the container may stay at the port until the LFD date to avoid any fees. For an export load, the ERD is the earliest date that a container can be returned to the port, and CUT is the last day to drop-off the container before the vessel leaves the port.

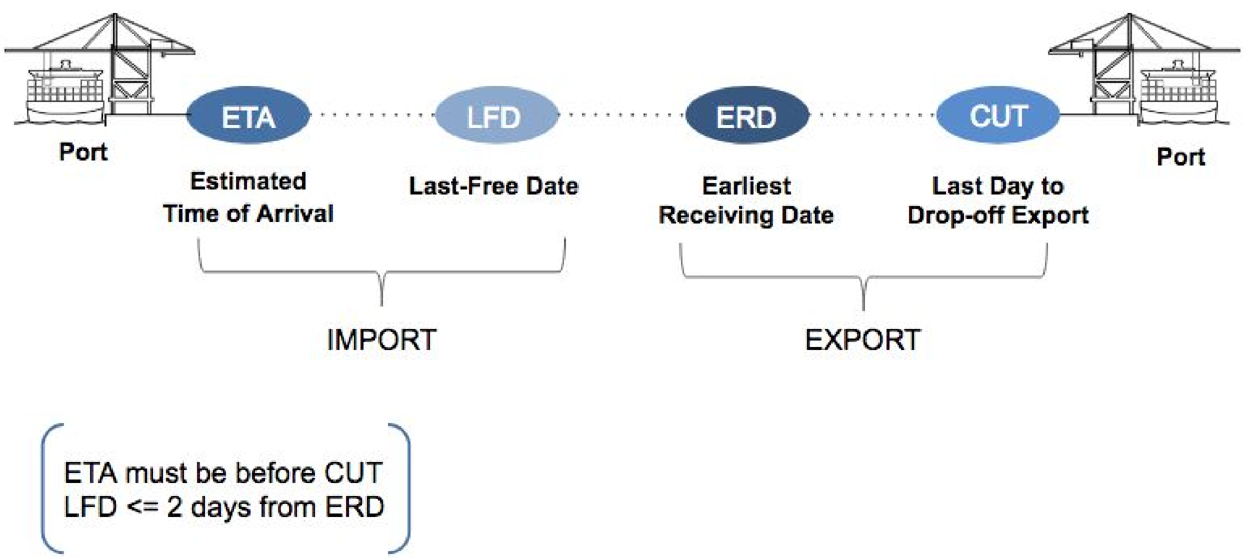

Figure D.1. Timeline of date parameters considered in the import-export pairing tool.

D.3 Data Validation Implementation

Table D.1 specifies key constraints necessary for drayage matchmaking, how they exist in the current system, and how they will be modified in the future system to guarantee data validity. Figure D.2 shows the API interface corresponding to these constraints.

PARAMETER	CURRENT	IMPLEMENTED
Load Type	Manually Typed	Toggle Button
Port City	Manually Typed	Manually Typed but cross-checked with Zip Code and State internal database
Delivery City	Manually Typed	Manually Typed but cross-checked with Zip Code and State internal database
Steamship Company	Manually Typed	Drop-down option
Equipment	Manually Typed	Drop-down option
Import Dates	Ship Date and Delivery Date Manually Typed	ETA and LFD field entry appears as a calendar option once Import is toggled on Load Type field
Export Dates	Ship Date and Delivery Date Manually Typed	ERD and CUT field entry appears as a calendar option once Export is toggled on Load Type field

Table D.1. Load parameters input data validation procedures implemented.

Figure D.2. Freight Hawk Dispatch example showing data validation.

D.4 Example Results

The following JSON output articulates data items that will be communicated to NTG once a load is matched:

{
"Potential Matches for 1463943": [
{
"LoadID": 1476509,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/19/2018",
"CUT": "01/23/2018"
},
{
"LoadID": 1476514,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/19/2018",
"CUT": "01/23/2018"
},
{
"LoadID": 1478675,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/27/2018",
"CUT": "01/31/2018"
},
{
"LoadID": 1478681,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/25/2018",
"CUT": "01/29/2018"
},
{
"LoadID": 1488028,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{

"LoadID": 1488030,
"DeadHead": 0,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "LA PORTE, TX 77571",
"Cons_City_Full": "HOUSTON, TX 77055",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474058,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474059,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474060,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474061,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474062,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{

"LoadID": 1474063,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474064,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
},
{
"LoadID": 1474065,
"DeadHead": 147,
"ImpExp": "EXPORT",
"SteamShipLine": "COSCO",
"Equipment": "40 HC",
"Ship_City_Full": "BARBOURS CUT, TX 77571",
"Cons_City_Full": "AUSTIN, TX 78724",
"ERD": "01/26/2018",
"CUT": "01/30/2018"
}
] }