
Solving the Empty Container Problem using Double-Container Trucks to Reduce Vehicle Miles

Santiago Carvajal

scarvaja@usc.edu

(424) 333-4929

Daniel J. Epstein Department of Industrial and Systems Engineering

University of Southern California

Maged Dessouky*

maged@usc.edu

(213) 740-4891

Daniel J. Epstein Department of Industrial and Systems Engineering

University of Southern California

*Corresponding Author

Abstract

This paper proposes a mathematical model for the empty container reuse problem using double-container trucks. The model discretizes time and makes sure demand is met. Several model properties are presented, along with two heuristics that allow the model to be solved. By solving the empty container reuse problem, truck miles can be reduced since less truck trips are needed to satisfy the demand. Furthermore, since double-container trucks can deliver two containers per truck trip, the amount of trucks needed to satisfy the demand is decreased even more, further reducing truck miles. This will in turn make the system more environmental friendly. The model is then tested using data from the Ports of Los Angeles and Long Beach, and on randomized data sets.

Keywords: Container management, Empty container problem, Integer programming

1. Introduction

In today's world, there is a significant amount of investigation regarding how to efficiently distribute loaded containers from the ports to the consignees. However, to further optimize the process and become more environmentally friendly, one should also study how to allocate the empty containers created by these consignees. This is an essential part in the study of container movement since it balances out the load flow at each location. Currently, most container movement at the Ports of Los Angeles and Long Beach follow a simple movement, going from the port to importers and then back to the port as an empty container. Subsequently, some of these empty containers go from the port to exporters and then return as loaded containers to the port. Finally, both empty and full containers are shipped from the ports to Asia. This creates a lot of unnecessary container movement into and out of the ports. For example, in 2015 the Ports of Los Angeles and Long Beach had 15.3 million Twenty-foot Equivalent Units (TEU). About 30% of this or 4.3 million TEUs were empty containers. This is a significant amount of unnecessary empty container movement.

In this paper, we propose to have some of the empty containers go directly from the importers to the exporters and not return empty back to the port. This movement is usually called a "street exchange". There are several reasons why street exchanges are uncommon in today's container movement process. However, probably the most prominent reason is because of the substantial amount of required coordination between the different companies to make the exchange in a timely fashion.

The problem of coordinating the container movement to increase the number of street exchanges has been studied in the past and is called the "Empty Container Reuse Problem". This

paper augments the earlier work by introducing the use of double-container trucks to the literature, something that to the best of our knowledge has not been done. In addition to the proposed model, we identify some solution properties of the model that partly form the basis for the developed computationally efficient heuristics. Double-container trucks would increase the number of street exchanges that could be made since the possibilities are greater with two container trucks. Additionally, double-container trucks are more efficient than single-container trucks. Currently, double-container trucks are used in multiple countries, including but not limited to Mexico, Argentina, Australia, and Canada. In the United States, double-container trucks are allowed on some roads, but not all.

The rest of this paper is organized as follows. In Section 2, a literature review of the relevant problems is presented. Section 3 formally defines and describes the mathematical model used for the assignment of the container movement. In Section 4 some heuristics are presented to obtain effective feasible solutions to the model since it is computationally prohibitive to obtain optimal solutions for large scale problem sizes. In Section 5 the results for two types of experiments are shown, one using data from the Ports of Los Angeles and Long Beach, and the other one using randomized data sets. We conclude the paper in Section 6.

2. Literature Review

There has been some prior research on the Empty Container Reuse Problem due to the fact that container repositioning has become increasingly more expensive over the years. Historically, the problem has been subdivided into two sub-problems. The first problem focuses on empty container reuse in inland destinations. The second sub-problem focuses on the movement of containers that

are near the port areas, usually no more than 20 miles from the port. It is this second problem that is the focus of this research.

One of the earliest models for the Empty Container Reuse Problem was developed by Dejax and Crainic in 1987. They developed several deterministic, stochastic, and hybrid models as to how empty containers should be repositioned. They proposed successive research with new ideas such as adding a depot center and integrating empty and loaded container movements at an industry level. Shen and Khoong (1995) also studied the problem from the perspective of a single company. In their paper, they develop a decision model that yielded when and how to move a container, as well as when to lease a container. They then performed some constraint relaxations on the model that allows the model to react quickly to decisions made after the model was solved and to supply and demand changes. Our model formulation is similar to these earlier papers by also making the container routes to be the main decision variables but extending it to double-container routes.

Bourbeau et al. (2000) developed a mixed integer model and used a parallel branch and bound approach to optimize the location of the depot and provide a flow of the container allocation problem. In our formulation, we assume the location of all container transfer points (e.g., depot) are given and are not decision variables. Li et al (2014) studied the problem at a more global view. They built a model that maximized profit for the shipping company. Their model was deterministic and operated on a rolling horizon basis. They then tested their model on a real life example using some ports from the east coast of China, and showed that not only is their approach more profitable but also provides a greener solution. Along the same lines, Choong et al. (2002) studied the effect of the length of the planning horizon with regard to the empty container problem. In their paper,

they use a model developed in Choong (2000) and other empty container models to find the effect of the length of the planning horizon and how far in advance does planning need to start.

Probably the most extensive research of container movement in the Ports of Long Beach and Los Angeles was done by the Tioga Group (2002). They did extensive research on container movement in and out of the Port of Long Beach. After compiling extensive data, they suggested a concept of how empty container reuse could be increased in this area. Their work has served as a foundation to various other empty container models that use the Port of Long Beach as their research scenario, especially when using their data. For example, Jula et al. (2006) built a dynamic model that used the Tioga report data to come up with a feasible solution of how to allocate containers on a daily basis. Taking into account that on any single day all demand is deterministic, but the demand for the next day is stochastic. They use dynamic programming to find the best match of a bipartite transportation network. In that way, they meet all the daily demand and try to optimize the containers for future days as well. Chang et al. (2008) studied when and where containers should be substituted with another type. They proposed a heuristic method that divided the problem into dependent and independent parts. They are then able to apply a branch and bound algorithm to arrive at an integer solution relatively fast. They tested their procedure for the Ports of Los Angeles and Long Beach using data from the Tioga report, and on randomized scenarios, comparing it to other commercial mixed integer programming solvers. We also use the Tioga Report as a basis to understand container movement at the Port of Long Beach to set up the test operating scenarios.

Similarly Lam et al (2007) demonstrated how dynamic programming can lead to a competitive approximate solution that improves efficiency. They first built an empty container model, and then used linear approximations to simplify the model. They then used dynamic

programming to arrive at an approximation of the optimal solution. They first tested their approach in a simple example, and for a real life example. They then compared their solution to other heuristics used in industry.

Dam Le (2003) has also assessed from the perspective of the logistics involved to make container reuse possible in Southern California. She conducted several interviews with field experts to make recommendations on where depots would make the most sense according to expected demand from the different importers and exporters. The problem has also been studied in different ports around the world. Islam et al. (2010) conducted an extensive study of the Port of Auckland. They studied how containers were moved and how truck congestion changed throughout the day. They then suggested times at which empty containers should be repositioned to decrease congestion at peak hours.

Braekers et al (2013) tackled the dynamic empty container reuse problem, in which origins and destinations are not known beforehand. They constructed a network flow model to optimize the movement from importers, exporters, depots, and the port. They used a sequential approach and an integrated approach to solve the model. This yielded a sub-optimal result, but decreased the complexity of the model, thus reducing the solving time. They tested their solving methods using a small example that they created, as well as other examples from other papers for comparison. Different than our model Braekers et al (2013) use the containers as decision variables as opposed to the truck trips. Extending the formulation to double-container trucks with the containers as decision variables significantly increases the complexity of the problem due to the exponential growth in the number of binary variables. Therefore, we use truck trips as the decision variables in our model to reduce the complexity of the problem and leave it to future research to

extend the model at the container level which would provide additional system efficiencies in the solution by allowing multiple loaded locations for each truck.

3. Single and Double Container Movement Model

3.1 Problem Description

The objective of the Empty Container Problem is to find a container assignment that meets all demands, while minimizing truck miles. Our model does this indirectly by minimizing the total cost for all container movement which has a correlation to truck miles. However, using cost allows us to differentiate between single-container truck trips and double-container truck trips. We assume container demand at each location is given and deterministic for each day. Our model focuses on satisfying all the demand, both for loaded and empty containers, at all the locations throughout the day. First, time is discretized. The decision variables are integer variables that correspond to the number of containers sent from location i to location j at each point in time. There are three main types of variables. A truck carrying two containers is divided into two variables. The first variable corresponds to the container that the truck delivers first. The second variable corresponds to the container that the truck delivers second. Lastly, the third variable corresponds to a truck delivering a single container.

Figure 1 shows the current flow for both full and empty containers. As seen in Figure 1 the movement of containers is very inefficient going from the port to importers, back to the port, then out to exporters, and finally back to the port to be shipped out. Figure 2 shows the proposed container flow if only single-container trucks were used, and Figure 3 shows the proposed container flow if both single and double-container trucks were used. As can be seen in the figures the network locations are separated into four groups: importers, exporters, depots, and the port.

The depots are currently not being fully utilized; however, our model proposes that depots need to be added to make street exchanges easier to schedule. Each location has a demand for either loaded or empty containers, or both. Also, each location yields empty or loaded containers, or both. For example, an importer requests loaded containers and yields empty containers that can be used to satisfy other locations. Not all locations can satisfy the demand for other locations. For example, an importer's demand can only be satisfied by loaded containers coming from the port; however, it can satisfy empty container demand for exporters and the port.

In Figure 3 the solid line could be a container movement of either one full container or two full containers. Similarly, the dashed line could be either one empty container or two empty containers. However, the dashed dotted line symbolizes the movement of one full container and one empty container. For example, a truck may carry a full container and an empty container from the port to an exporter. It is for this reason that the proposed double-container truck system has many more options compared to the possible routes in the other two systems. However, this does not mean that an exporter can supply an importer, since it is actually the port that supplies containers.

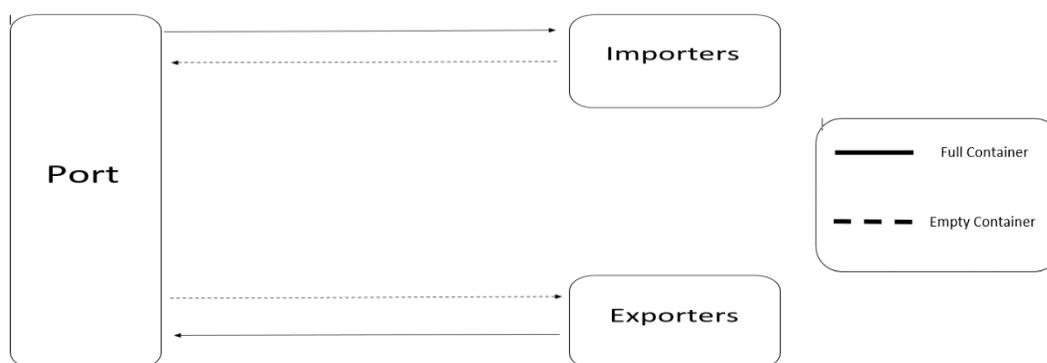


Figure 1. Current container flow

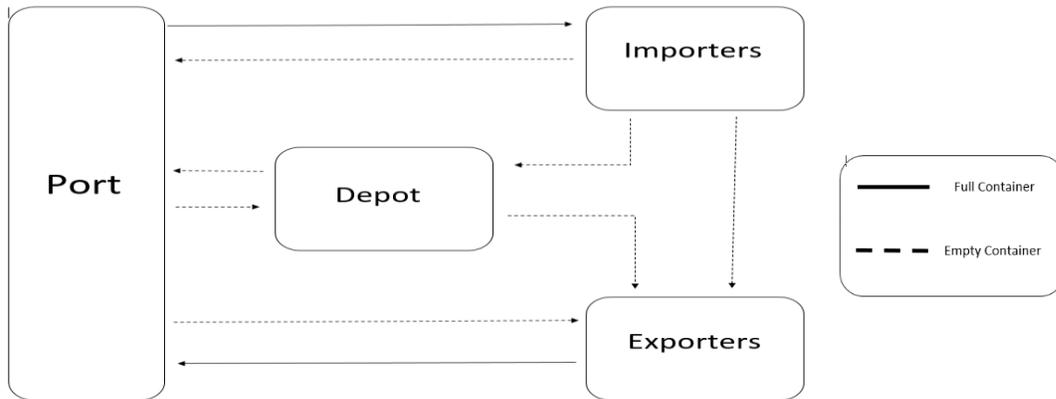


Figure 2. Proposed single container flow

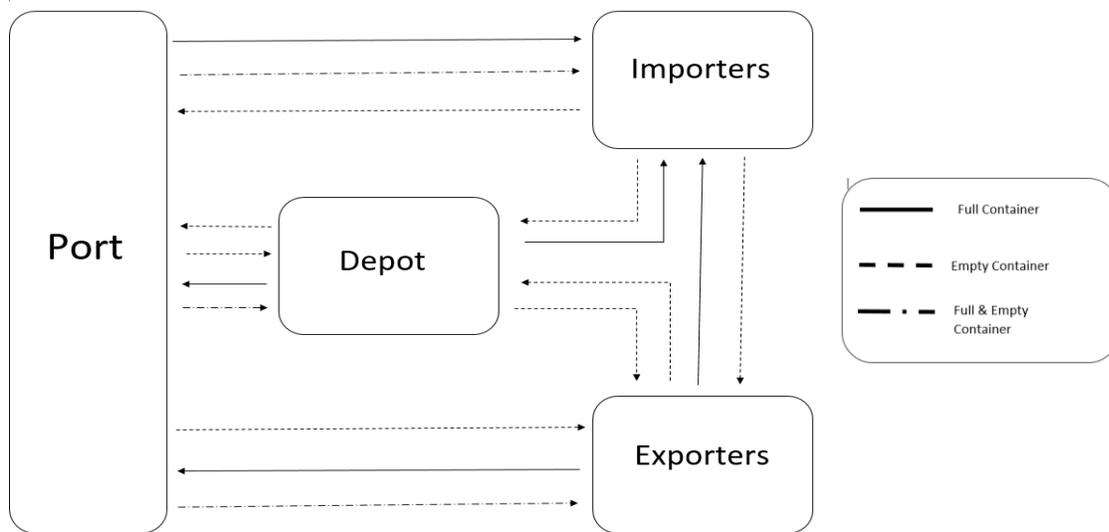


Figure 3. Proposed double container flow

As stated above at each discretization of time, the model allows for containers to be moved from one location to another. We then introduce two new variables. The first variable records the number of containers received at each location at each point in time. The second variable records the number of containers provided by each location at each point in time. It is these two variables that allow the model to ensure demand is met at each time period. We also divide the port into 3 locations. The first location holds the loaded containers that need to be delivered to an importer. The second location holding empty containers that can either be used to supply exporters or be

shipped out. Finally, the third location holds loaded containers coming from exporters that need to be shipped out.

The model also assumes trucks are not a limiting resource since there are a good deal of trucks around the port area waiting for a job. Thus, we do not have to balance the number of trucks, and we assume that trucks are on standby waiting for a job.

3.2 Mathematical Formulation

We next present the mathematical formulation of the double container reuse model. The notation for the formulation is as follows:

Parameters:

$l_{i,j,t}$ = time it takes to go from location i to location j leaving at time t

$o_{i,j,t}$ = time it takes to go from location i to location j arriving at time t

r_i = Container turnover time at location i

p_i = Number of containers available at the beginning of the day at location i

$d_{i,t}$ = Number of containers demanded at location i by time t

c_i = Capacity of location i

$e_{i,j,t}$ = Cost of first leg of a two container route going from location i to location j starting at time t

$f_{i,j,t}$ = Cost of second leg of a two container route going from location i to location j starting at time t

$g_{i,j,t}$ = Cost of a one container route from location i to location j starting at time t

Sets:

$SI = \{1, \dots, |SI|\}$ (locations of all importers)

$SE = \{|SI| + 1, \dots, |SI| + |SE|\}$ (locations of all exporters)

$SD = \{|SI| + |SE| + 1, \dots, |SI| + |SE| + |SD|\}$ (locations of all depots)

$SP1 = \{|SI| + |SE| + |SD| + 1\}$ (location of the importer container port)

$SP2 = \{ |SI| + |SE| + |SD| + 2 \}$ (location of the empty container port)

$SP3 = \{ |SI| + |SE| + |SD| + 3 \}$ (location of the exporter container port)

$SA = \{ SI \cup SE \cup SD \cup SP1 \cup SP2 \cup SP3 \}$ (all locations)

$ST = \{ 1, \dots, |ST| \}$ (times of the day)

Decision Variables:

$x1_{i,j,t}$ = Number of first leg two container trucks going from location i to location j at time t of a truck carrying two loaded containers

$x2_{i,j,t}$ = Number of first leg two container trucks going from location i to location j at time t of a truck carrying two empty containers

$x3_{i,j,t}$ = Number of first leg two container trucks going from location i to location j at time t of a truck carrying one loaded container and one empty container

$y1_{i,j,t}$ = Number of second leg two container trucks going from location i to location j at time t of a truck carrying two loaded containers

$y2_{i,j,t}$ = Number of second leg two container trucks going from location i to location j at time t of a truck carrying two empty containers

$y3_{i,j,t}$ = Number of second leg two container trucks going from location i to location j at time t of a truck carrying one loaded container and one empty container

$z_{i,j,t}$ = Number of single container trucks going from location i to j at time t

$m_{i,t}$ = Number of containers supplied by location i at time t

$n_{i,t}$ = Number of containers delivered to location i at time t

$a_{i,t}$ = Number of containers that have been supplied by location i by time t

$b_{i,t}$ = Number of containers that have been delivered to location i by time t

Objective:

$$\min \sum_{t \in ST} \sum_{i \in SA} \sum_{j \in SA} (e_{i,j,t} * x1_{i,j,t} + f_{i,j,t} * y1_{i,j,t} + g_{i,j,t} * z_{i,j,t} + e_{i,j,t} * x2_{i,j,t} + f_{i,j,t} * y2_{i,j,t} + e_{i,j,t} * x3_{i,j,t} + f_{i,j,t} * y3_{i,j,t})$$

s.t.

Containers provided at time t :

$$2 \sum_{j \in SEUSD \cup SP2} x2_{i,j,t} + \sum_{j \in SEUSD \cup SP2} z_{i,j,t} = m_{i,t} \quad \forall i \in SI \quad \forall t \in ST \quad (\text{Importers}) \quad (1)$$

$$2 \sum_{j \in SP3} x1_{i,j,t} + \sum_{j \in SP3} z_{i,j,t} = m_{i,t} \quad \forall i \in SE \quad \forall t \in ST \quad (\text{Exporters}) \quad (2)$$

$$2 \sum_{j \in SEUSD \cup SP2} x2_{i,j,t} + \sum_{j \in SEUSD \cup SP2} z_{i,j,t} = m_{i,t} \quad \forall i \in SD \quad \forall t \in ST \quad (\text{Depots}) \quad (3)$$

$$2 \sum_{j \in SI} x1_{i,j,t} + \sum_{j \in SI \cup SEUSD} x3_{i,j,t} + \sum_{j \in SI} z_{i,j,t} = m_{i,t} \quad \forall i \in SP1 \quad \forall t \in ST \quad (\text{Port 1}) \quad (4a)$$

$$2 \sum_{j \in SEUSD} x2_{i,j,t} + \sum_{j \in SI \cup SEUSD} x3_{P1,j,t} + \sum_{j \in SEUSD} z_{i,j,t} = m_{i,t} \quad \forall i \in SP2 \quad \forall t \in ST \quad (\text{Port 2}) \quad (4b)$$

Containers received at time t :

$$\begin{aligned} & \sum_{i \in SP1: t - o_{i,j,t} \geq 1} x1_{i,j,t - o_{i,j,t}} + \sum_{i \in SP1: t - o_{i,j,t} \geq 1} x3_{i,j,t - o_{i,j,t}} + \sum_{i \in SI: t - o_{i,j,t} \geq 1} y1_{i,j,t - o_{i,j,t}} \dots \\ & \dots + \sum_{i \in SEUSD: t - o_{i,j,t} \geq 1} y3_{i,j,t - o_{i,j,t}} \dots + \sum_{i \in SP1: t - o_{i,j,t} \geq 1} z_{i,j,t - o_{i,j,t}} = n_{j,t} \quad \forall j \in SI \quad \forall t \in ST \quad (\text{Importers}) \quad (5) \end{aligned}$$

$$\begin{aligned} & \sum_{i \in SI \cup SD \cup SP2: t - o_{i,j,t} \geq 1} x2_{i,j,t - o_{i,j,t}} + \sum_{i \in SP1: t - o_{i,j,t} \geq 1} x3_{i,j,t - o_{i,j,t}} + \sum_{i \in SEUSD: t - o_{i,j,t} \geq 1} y2_{i,j,t - o_{i,j,t}} \dots \\ & \dots + \sum_{i \in SI: t - o_{i,j,t} \geq 1} y3_{i,j,t - o_{i,j,t}} \dots + \sum_{i \in SI \cup SD \cup SP2: t - o_{i,j,t} \geq 1} z_{i,j,t - o_{i,j,t}} = n_{j,t} \quad \forall j \in SE \quad \forall t \in ST \quad (\text{Exporters}) \quad (6) \end{aligned}$$

$$\begin{aligned} & \sum_{i \in SI \cup SD \cup SP2: t - o_{i,j,t} \geq 1} x2_{i,j,t - o_{i,j,t}} + \sum_{i \in SP1: t - o_{i,j,t} \geq 1} x3_{i,j,t - o_{i,j,t}} + \sum_{i \in SEUSD: t - o_{i,j,t} \geq 1} y2_{i,j,t - o_{i,j,t}} \\ & \dots + \sum_{i \in SI: t - o_{i,j,t} \geq 1} y3_{i,j,t - o_{i,j,t}} \dots + \sum_{i \in SI \cup SD \cup SP2: t - o_{i,j,t} \geq 1} z_{i,j,t - o_{i,j,t}} = n_{j,t} \quad \forall j \in SD \quad \forall t \in ST \quad (\text{Depots}) \quad (7) \end{aligned}$$

$$\begin{aligned}
& \sum_{i \in SI \cup SD: t - o_{i,j,t} \geq 1} x_{2,i,j,t - o_{i,j,t}} + \sum_{i \in SE \cup DUP2: t - o_{i,j,t} \geq 1} y_{2,i,j,t - o_{i,j,t}} \dots \\
& \dots + \sum_{i \in SI \cup SD: t - o_{i,j,t} \geq 1} z_{i,j,t - o_{i,j,t}} = n_{j,t} \quad \forall j \in SP2 \quad \forall t \in ST \quad (\text{Port 2}) \quad (8a)
\end{aligned}$$

$$\begin{aligned}
& \sum_{i \in SE: t - o_{i,j,t} \geq 1} x_{1,i,j,t - o_{i,j,t}} + \sum_{i \in SP3: t - o_{i,j,t} \geq 1} y_{1,i,j,t - o_{i,j,t}} \dots \\
& \dots + \sum_{i \in SE: t - o_{i,j,t} \geq 1} z_{i,j,t - o_{i,j,t}} = n_{j,t} \quad \forall j \in SP3 \quad \forall t \in ST \quad (\text{Port 3}) \quad (8b)
\end{aligned}$$

Demand and Feasibility constraints:

$$a_{i,t} = \sum_{q=1}^t m_{i,q} \quad \forall i \in SA \quad \forall t \in ST \quad (\text{Number provided at } i \text{ by time } t) \quad (9)$$

$$b_{i,t} = \sum_{q=1}^t n_{i,q} \quad \forall i \in SA \quad \forall t \in ST \quad (\text{Number received at } i \text{ by time } t) \quad (10)$$

$$b_{i,t-r_i} + p_i - a_{i,t} \geq 0 \quad \forall \{i, t \mid i \in SA, t \in ST, t - r_i \geq 1\} \quad (\text{Number provided cannot be more than received}) \quad (11a)$$

$$p_i - a_{i,t} \geq 0 \quad \forall \{i, t \mid i \in SA, t \in ST, t - r_i < 1\} \quad (\text{Number provided cannot be more than received}) \quad (11b)$$

$$b_{i,t} \geq d_{i,t} \quad \forall i \in SI \cup SE \cup SD \quad \forall t \in ST \quad (\text{Demand at location } i \text{ must be met by time } t) \quad (12a)$$

$$\begin{aligned}
& \sum_{i \in SP2 \cup SP3} b_{i,t} + p_j - n_{j,t} \geq \sum_{i \in SP2 \cup SP3} d_{i,t} \quad \forall j \in SP2 \quad \forall t \in ST \\
& \quad \quad \quad (\text{Demand for port location must be met by time } t) \quad (12b)
\end{aligned}$$

$$b_{i,t} \geq d_{i,t} \quad \forall i \in SP3 \quad \forall t \in ST \quad (\text{Demand the third port location must be met by time } t) \quad (12c)$$

$$b_{i,t} + p_i - a_{i,t} \leq c_i \quad \forall i \in SI \cup SE \cup SD \quad \forall t \in ST \quad (\text{Capacity at } i \text{ cannot be exceeded}) \quad (13a)$$

$$\begin{aligned}
& \sum_{i \in SP2 \cup SP3} b_{i,t} + \sum_{i \in SP1 \cup SP2 \cup SP3} p_i - \sum_{i \in SP1 \cup SP2} a_{i,t} \leq \sum_{i \in SP1 \cup SP2 \cup SP3} c_i \quad \forall t \in ST \\
& \quad \quad \quad (\text{Capacity at } i \text{ cannot be exceeded}) \quad (13b)
\end{aligned}$$

$$\sum_{i \in SP1} x_{1,i,j,t} = \sum_{k \in SI: t + l_{i,j,t} \leq |ST|} y_{1,j,k,t + l_{i,j,t}} \quad \forall j \in SI \quad \forall t \in ST \quad (\text{Balance containers in double trucks}) \quad (14a)$$

$$\sum_{i \in SE} x_{1,i,j,t} = \sum_{k \in SP3: t + l_{i,j,t} \leq |ST|} y_{1,j,k,t + l_{i,j,t}} \quad \forall j \in SP3 \quad \forall t \in ST \quad (\text{Balance containers in double trucks}) \quad (14b)$$

$$\sum_{i \in SI \cup SD \cup SP2} x2_{i,j,t} = \sum_{k \in SE \cup SD \cup SP2: t+l_{i,j,t} \leq |ST|} y2_{j,k,t+l_{i,j,t}} \quad \forall j \in SE \cup SD \cup SP2 \quad \forall t \in ST$$

(Balance containers in double trucks) (14c)

$$\sum_{i \in SP1} x3_{i,j,t} = \sum_{k \in SE \cup SD: t+l_{i,j,t} \leq |ST|} y3_{j,k,t+l_{i,j,t}} \quad \forall j \in SI \quad \forall t \in ST$$

(Two container trucks must provide two containers) (14d)

$$\sum_{i \in SP1} x3_{i,j,t} = \sum_{k \in SI: t+l_{i,j,t} \leq |ST|} y3_{j,k,t+l_{i,j,t}} \quad \forall j \in SE \cup SD \quad \forall t \in ST$$

(Two container trucks must provide two containers) (14e)

$$x1_{i,j,t}, y1_{i,j,t}, x2_{i,j,t}, y2_{i,j,t}, x3_{i,j,t}, y3_{i,j,t}, z_{i,j,t} \geq 0 \quad \forall i \in SA \quad \forall j \in SA \quad \forall t \in ST$$

(Non – negative Constraint) (15)

$$x1_{i,j,t}, y1_{i,j,t}, x2_{i,j,t}, y2_{i,j,t}, x3_{i,j,t}, y3_{i,j,t}, z_{i,j,t} \in \mathbb{Z} \quad \forall i \in SA \quad \forall j \in SA \quad \forall t \in ST$$

(Integer Constraint) (16)

The objective of the model is to minimize the transportation costs needed to meet all the demand. There is a cost associated with each possible single-container truck trip which depends on the locations for pickup and drop-off of the container, as well as the time of day. We have separate transportation costs for the first container on a double-container trip, and the second container on a double-container trip. We divided this cost into two because depending on the destination of the second container the price to hire a double-container truck can vary. For example, if both containers are going to the same location, the price is most likely going to be less than if the containers are going to different locations.

As stated before, the model has three main integer types of variables x , y , and z . Additionally, the x and y variables are subdivided into 3 more variable types. The $x_{i,j,t}$ variables correspond to a double-container truck going from location i to location j starting at time t to drop off its first container at j . Its first subtype is $x1$ and corresponds to a truck carrying two loaded

containers. Similarly, x_2 corresponds to a truck carrying two empty containers, and x_3 corresponds to a truck carrying one loaded and one empty container. The $y_{i,j,t}$ variables correspond to a double-container truck (now with only one container) travelling from location i to location j starting at time t to drop off its second container. The subtypes here are similar to their x counterparts. Finally, the z variables represent a single-container truck trip from location i to location j starting at time t . Note that i and j cannot be the same for any x or z variable since it does not make sense that a location can provide itself with containers; however the y variables can have i and j be the same since that means the second container is being dropped off at the same location as the first container. The rest of the variables only serve to record the total number of received and delivered containers at each location for each time period, and are determined by specific summations of the main three variables.

Constraints (1)-(4) sum all the containers provided by a specific location at a specific point in time. It then does this for all locations at all points in time and equals them to the $m_{i,t}$ variables which represent all the containers provided by location i at time t . Single-container truck trips only add one container since there is only one container involved. However, double-container truck trips count double since there are two containers involved if they are carrying just loaded or empty containers, thus the x_1 and x_2 variables are multiplied by two. For example, constraint (1) sums up all the containers provided by the importers. That is, importers can only provide empty containers. Therefore, the destination for the empty containers are exporters, depot, and the port. This does not include other importers since they have no demand for empty containers. Similarly, constraint (2) sums up all the containers provided by the exporters. Unlike importers, exporters can only provide full containers and thus these containers have to go to the third port location. Additionally, constraint (4a) does this for the first port location, which has loaded containers for

importers, while constraint (4b) deals with the second port location which has empty containers. It is worth noting that all x_3 variables need to originate from the port since it is the only location that has both empty and loaded containers. Specifically, it needs a container from the first port location and one container from the second port location. For this reason, it is not multiplied by 2 in any of the constraints, but instead appears in both constraints (4a) and (4b) and takes a container from each location each time.

Constraints (5)-(8) sum all the containers received by a specific location at a specific point in time. It then does this for all locations at all points in time and equals them to the $n_{i,t}$ variables which represent all the containers received by location i at time t . Since each variable represents the drop-off of a single container, all variables only add one in this sum. For example, constraint (5) sums all the containers received by the importers, which can only receive loaded containers. In this case there are 5 ways (summations) an importer can receive a loaded container. The first sum represents the first loaded container being dropped off at importer j in a truck route carrying two loaded containers (i.e. $x_{1,j,t}$). This truck route could only have originated at the port since loaded containers for importers can only be found at the port. The second sum represents a loaded container being dropped off at importer j in a truck route carrying one loaded container and one empty container (i.e. $x_{3,j,t}$). In this case the first container is being dropped off in the two-container truck route. Therefore, the truck route will originate from the port as well. The third sum represents the second loaded container being dropped off at importer j in a double-container truck route carrying two loaded containers (i.e. $y_{1,j,t}$). In this case the second leg of the route should have originated at an importer, since the other loaded container could only have been dropped off at an importer, possibly the same importer. The fourth sum represents the loaded container being dropped off second at importer j in a double-container route carrying a loaded container and an

empty container (i.e. $y_3^{i,j,t}$). In this case the second leg of the route could only have come from either an exporter or a depot where it dropped off the empty container first at either an exporter or a depot. Finally, the last sum represents a loaded container being dropped off at location importer j for a truck route carrying only a single loaded container from the port. Another example, can be found in constraint (8b) there are three sums each determining the possible ways the third port location can receive loaded containers, with the first sum representing the first loaded container of a double-container truck route coming from an exporter. The second sum represents the second loaded container of the same route. Finally, the last sum represents a loaded container from a single-container route coming from an exporter.

The next set of constraints deal with meeting the demand, and ensuring the feasibility of the solution. Constraint (9) aggregates all the provided containers that a location has provided by time t . It then does this for all time periods and all locations. Constraint (10) does the same but aggregates all the containers that a location has received by time t .

Constraint (11) is a feasibility constraint that deals with the fact that the number of containers received minus the number of containers provided, plus the number of containers at the start of the day cannot be a negative number. Notice that the a variables are all containers provided until time t , while the b variables are all the containers received by time t . They have to be offset by time r_i which is the turnover time at location i . The idea is that when a container arrives at a location there is a certain time that is needed to either unload or load the container. Constraint (12) ensures demand is met. Constraint (12a) ensures that the demand at all non-port locations is met. Constraint (12b) ensures that there are enough containers to load the ships that are leaving the port. Constraint (12c) ensures that all the loaded containers needed from the exporters arrive at the third port location.

Constraint (13) deals with the fact that a location only has a certain amount of space or capacity. This constraint makes sure that at every point in time the amount of containers that are in a location does not exceed this capacity. Finally, constraint (14) makes sure that a double-container truck delivers two containers. The x variables represent a truck going from location i to location j at time t . After some delay, given by the parameter l . This truck must go to another location (this can be the same location) to deliver the second container. This is represented by the y variable. This constraint says that all the x variables that arrive at a certain location by time t must have a corresponding y variable associated with them. This constraint is subdivided into 5 different constraints, to correctly match the x variables with their corresponding y variables.

3.3 Model Properties

Although the worst-case complexity of the model is not known, in this section we focus on pointing out some interesting observations of the model. These observations partly form the basis for our proposed heuristics. Our first observation is that the Linear Program (LP) relaxation will yield an integer or half integer solution if (1) moving containers is cheaper by either using only single-container trucks or double-container trucks (i.e. $e_{i,j,t} + f_{j,j,t} > 2g_{i,j,t}$ or $e_{i,j,t} + f_{j,j,t} < 2g_{i,j,t}$ for all locations i and j , and all times t). This assumption translates to a trucking company charging more for a double-container route than two single-container routes starting at the same location, or the opposite. The second assumption (2) is that dropping of the second container at the same location is always cheaper than dropping it off at any other location ($f_{i,i,t} < f_{i,j,t} \forall i, j \in SA \forall i \neq j$). This assumption is reasonable because dropping off the second container at another location means that the truck must drive to another location, wasting more time instead of simply dropping the second container at the same location.

A second observation of the model is that if both demand and capacity is even, then the model will always yield an integer solution. The half integrality only comes in play when either the demand or capacity at any location is odd. Proofs for both observations can be found in Appendix 1.

4. Heuristics

Under general conditions solving the model as a LP will not yield a feasible solution, since the optimal solution may yield half integer values for the decision variables. In order to get a feasible solution, we introduce two heuristics. These heuristics use the result given by the Linear Relaxation Program, make use of the half integer solution property identified in Section 3.3, and yield an approximate solution to the problem. Both heuristics could potentially be affected by three factors, position of locations, demand, and location capacity. In this section we introduce the heuristics, and we will test the effect of the factors in the following experimental section.

4.1 Single Truck Heuristic

The first heuristic is what we would call the Single Truck Heuristic (STH). This is a very simple heuristic that takes advantage of the half integer solution that is found when solving the linear program relaxation of the model. As previously discussed, the model only uses double-container trucks if they are cheaper than the single-container trucks. This heuristic takes any double-container truck trip (i.e. the $x_{i,j,t}$ and the corresponding $y_{j,j,t}$) and rounds both of them down. It then adds a single-container truck trip from location i to location j , where i and j correspond to the variable $x_{i,j,t}$ that was rounded down. This then yields a feasible solution. It is worth noting that after solving the LP model this heuristic is a greedy algorithm and that its running time is $\Theta(N)$, where N is the number of truck trips yielded by the LP relaxation.

4.2 Integer Programming Heuristic

For the second heuristic we first solve the model using the LP relaxation. We then round all fractional solutions down to the nearest integer. These variables are then fixed, reducing the total demand that must be met. We then solve the model using Integer Programming techniques, and because the problem size is significantly smaller, this can be done in a reasonable amount of time. This then yields a feasible solution to the problem. We will refer to this heuristic as the Integer Programming Heuristic (IPH).

5. Experimental Analysis

In this section, we first run the model using data from the Ports of Los Angeles and Long Beach. We first test the model under specific parameters such that the linear program yields a feasible solution. The purpose for the first set of experiments is to show the degree of effectiveness of empty container reuse both with single and double-container trucks, by reducing the number of trucks and truck miles needed to fulfill demand. The second set of experiments test the effectiveness of the heuristics (STH and IPH) on randomly generated problems where the LP relaxation may not necessarily yield a feasible solution.

5.1 Ports of Los Angeles and Long Beach

The model was first tested on data from the Ports of Los Angeles and Long Beach. We used real data for container demand in the Southern California area of containers going from/to the Port of Long Beach and Terminal Island. We focused on the locations that are near the Port area (no more than 15 miles), since these are the locations where street exchanges are most likely to occur. The data available to us was aggregated in small regions. Since we did not have information about how many locations were inside each location we ran three separate experiments, with different number

of location in each region. The locations in each region were chosen using Google maps at locations where container terminals are located. The total aggregated daily demand is 200 containers by the importers from the Port to the locations. Meanwhile, the total aggregated demand for exporters to the Port is 90 containers. The number of importer regions was five, while the number of exporter regions was three. In our first set of experiments we consolidate the demand in each region to a single location within the region. In our second set of experiments we consolidate the demand for each region to two locations within the region. Finally, in our third set of experiments we consolidate each region to three locations within the region. In all experiments we have two depots and the Port. Table 1 shows the distances between the different locations for the first set of experiments. We then used Google maps to calculate to calculate the travel time between locations (i.e. the $o_{i,j,t}$ and $l_{i,j,t}$ variables).

Table 1. Distance between locations in miles

Locations	Importer 1	Importer 2	Importer 3	Importer 4	Importer 5	Exporter 1	Exporter 2	Exporter 3	Depot 1	Depot 2	Port
Importer 1	0	8.2	1.8	6	4	2	2.5	3.9	3.2	6.1	2.3
Importer 2	8.2	0	6.7	5.9	5	8	5.1	6.1	4.8	5	13
Importer 3	1.8	6.7	0	5.6	3.6	2.4	0.7	3.5	1.7	5.7	5.3
Importer 4	6	5.9	5.6	0	3.1	6.9	6.6	3	5.6	3.1	10
Importer 5	4	5	3.6	3.1	0	3.9	3.4	1.4	3.6	3.2	8.5
Exporter 1	2	8	2.4	6.9	3.9	0	3.1	2.7	3.3	7.2	5
Exporter 2	2.5	5.1	0.7	6.6	3.4	3.1	0	4.1	1	6.7	7
Exporter 3	3.9	6.1	3.5	3	1.4	2.7	4.1	0	3.8	3.3	7.3
Depot 1	3.2	4.8	1.7	5.6	3.6	3.3	1	3.8	0	5.7	6.2
Depot 2	6.1	5	5.7	3.1	3.2	7.2	6.7	3.3	5.7	0	10.5
Port	2.3	13	5.3	10	8.5	5	7	7.3	6.2	10.5	0

For these set of experiments, we assume a 12-hour day. In the first set of experiments, each of the five importer locations has a demand of 40 by time 9, and each of the three exporter locations has a demand of 30 by time 9. While in the second set of experiments each of the ten importer locations has a demand of 20 by time 9, and each of the six exporter locations has a demand of 15

by time 9. We therefore also assume that all 200 containers are ready for transport at the Port at the beginning of the day, and need to return to the Port (either empty or full) by the end of the day. We also assume that each importer or exporter location has a capacity of 10 containers. Meanwhile each depot has a capacity of 26 containers. The loading and unloading time of containers (r_i) is 1 hour at all locations. Finally, we also assume that the truck turnover time at the Port is 2 hours. It is worth noting that because of these specific set of parameters the LP relaxation will yield an integer solution, because of the properties previously discussed.

The model was built in Julia and solved using the Gurobi solver. The first experiment we performed involved solving the Double Container Reuse model. For this experiment, we made the assumption that it was cheaper to use one double-container truck rather than two single-container trucks for every route. Another assumption as well was that it was cheaper to have a double-container truck deliver both containers to the same location, rather than two different locations. We then set all $x_{i,j,t}$ and $y_{i,j,t}$ variables to zero and ran the same experiment. We called this trial the Single Container Reuse. Third, to have a baseline, we ran the experiment using only single-container trucks going from the port to non-port destinations. This experiment would mostly resemble the current situation. The results for these experiments are shown in Tables 2-4. Table 2 shows the results where there is only importer or exporter in each region while Tables 3 and 4 contain the results for the 2 and 3 cases, respectively.

Table 2. Results from the data of the Ports of Los Angeles and Long Beach (11 total locations)

Scenario	# Double-Container Truck Trips	# Single-Container Trucks Trips	Double-Container Truck Miles	Single-Container Truck Miles	Total Truck Miles	Total Cost
Double Container Reuse	245	0	1558	0	1558	\$35,294
Single Container Reuse	0	490	0	3116	3116	\$96,725
Single Direct (Current)	0	500	0	3702	3702	\$105,845
Double Container (Port Forbidden)	45	400	200	2717	2917	\$85,147
Double Container (Second leg allowed to Port)	90	310	845	2189	3034	\$68,547

Table 3. Results from the data of the Ports of Los Angeles and Long Beach (19 total locations)

Scenario	# Double-Container Truck Trips	# Single-Container Trucks Trips	Double-Container Truck Miles	Single-Container Truck Miles	Total Truck Miles	Total Cost
Double Container Reuse	245	0	1539	0	1546	\$34,876
Single Container Reuse	0	490	0	3092	3092	\$95,586
Single Direct (Current)	0	500	0	3702	3681	\$104,595
Double Container (Port Forbidden)	45	400	195	2705	2900	\$84,120
Double Container (Second leg allowed to Port)	90	310	832	2172	3004	\$67,725

Table 4. Results from the data of the Ports of Los Angeles and Long Beach (27 total locations)

Scenario	# Double-Container Truck Trips	# Single-Container Trucks Trips	Double-Container Truck Miles	Single-Container Truck Miles	Total Truck Miles	Total Cost
Double Container Reuse	245	0	1528	0	1528	\$34,641
Single Container Reuse	0	490	0	3092	3056	\$94,958
Single Direct (Current)	0	500	0	3702	3681	\$103,899
Double Container (Port Forbidden)	45	400	192	2699	2891	\$83,569
Double Container (Second leg allowed to Port)	90	310	827	2165	2992	\$67,285

There are some interesting results from these experiments. First, we can conclude that the number of locations within each region has no effect on the general trends. One noticeable detail is that the Double Container Reuse and the Single Container Reuse solutions yield the same movement of containers, with the only difference being that the Double Container Reuse uses only

double-container trucks, while the other experiment uses only single-container trucks. This means that the number of trucks and miles is exactly double for the Double Container Reuse compared with the Single Container Reuse. Now, comparing the Single Container Reuse versus the current situation there is about a 16% reduction in truck miles.

After these experiments, we ran two other experiments on the Double Container Reuse by changing the cost parameters. This allowed us to simulate different situations. We first took into account that double-container trucks are not allowed in the Ports since infrastructure improvements are necessary to accommodate double-container trucks. However, the comparison of these results with the previous results can show the benefits of using double-container trucks on the impact on the reduction of truck miles if the local infrastructure was expanded to account for double-container trucks. We therefore prohibited any part of a double-container truck from entering or leaving the port by assigning a large cost for both the first and second leg of the double-container trip. This forbade double-container truck trips from entering the port, but allowed double-container truck trips for the street exchanges. Afterwards, we allowed the second leg of a double-container truck to be able to enter the Port since it would only carry one container during this part of the trip. We therefore lowered the cost of the second part of a truck container going from a non-port location to the port. The results for these two experiments are shown in the last two rows of Tables 2-4.

As it can be observed, the amount of truck miles and trucks does go up in these two experiments, compared to the Double Container Reuse. However, this is still a reduction on the Single Container Reuse. When comparing these two experiments where double-container trucks are not allowed into the port, there are some advantages and disadvantages to each. By allowing the second leg of the truck trip to go into the Port the number of truck miles goes up, but the

number of trucks goes down, compared to when no double-container trips can go into the Port. This tradeoff between truck miles and number of trucks, is due to the fact that when the second leg of a truck trip is allowed into the Port, the model will choose to send a second leg of a truck into the Port. Even if this increases the number of miles the truck must go. By doing so it increases the number of double-container truck trips, thus reducing the total number of trips. The policy that is most beneficial will thus depend on the cost of an extra truck compared to the cost of having longer trips.

In conclusion, we can say that double-container trucks are more efficient than single-container truck trips, even when further restrictions are implemented on where double-container trucks can go. This was somewhat expected since double-container trucks carry more capacity than single-container trucks. It is also concluded that implementing the empty container reuse, even with only single-container truck trips, is more efficient than the current movement of containers, and that both the number of trucks and truck miles are reduced.

5.2 Experimental Design

We next test the effectiveness of the heuristics for a more general setting of parameters where the LP relaxation may yield fractional values to test the quality of the two heuristics (STH and IPH). In the previous experiments only even numbers were used, both for demand and the capacity at each location. This was done so that the LP relaxation yielded a feasible solution. In the next set of experiments, we test the STH and IPH heuristics to see how well they perform under more general conditions. We study three parameters that can have an influence on the solution. These being the position of the locations, demand size, and location capacity. For all the experiments in this section we use a 12-hour day, with time discretized into 15 minute intervals. We also assume that all locations can process one container in 1 hour, and that getting into and out of the Port takes

2 hours. We also use rectilinear distances between any two locations, with the port always being in the center at the bottom of the area. There are always 7 importers and 5 exporters. To analyze each parameter, we will use a two-way layout experiment, with a replication of 10 for each factor. The two factors in each experiment will be the heuristic being used, and the factor being tested (location, demand, and capacity). To be able to compare their performance we will use the ratio of the heuristic over the linear program lower bound. We were unable to use the optimal solution since when we ran the integer model in Gurobi for 8 CPU hours we were unable to even find a feasible solution. Thus we used the LP relaxation as our lower bound. For each trial we stopped the IPH after 20 CPU minutes. The STH took seconds to run in all cases.

The first parameter we test is the position of the locations. More specifically we test how close or spread out they are from each other. That is, the locations are randomly generated from a square of varying size. The port is located at the bottom center of the square. For example, an experiment may have each location be uniformly distributed on a 25x25 square (locations can only be on integer coordinates), with the Port being located on coordinate (13,0). As mentioned before we run 10 replications for each square size, each with a new set of locations in the same square. Demand was fixed with each importer demanding 115 containers and each exporter demanding 95 containers. The capacity for each location was also fixed at 17 containers. The results are shown in Table 5. In order to compare the results of the heuristics, we use the ratio between the heuristic and the solution to the LP relaxation. Note that LP stands for the solution for the Linear Program Relaxation, which is a lower bound of the problem and in general is not a feasible solution.

Table 5. Sensitivity of the results for the location parameter

Grid Size	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
10x10	1.031421	1.151461
15x15	1.023611	1.141502
20x20	1.013781	1.127415
25x25	1.014377	1.123801
30x30	1.031285	1.134819
Avg.	1.022895	1.1358
Std.	0.008649	0.011102

Table 6. ANOVA table for the location parameter

Source	SS	df	MS	F	P-value	F crit
Location	0.006872	4	0.001718	2.700146	0.035491	3.534992
Heuristic	0.31567	1	0.31567	496.1566	2.18E-38	6.925135
Interaction	0.001161	4	0.00029	0.456151	0.76767	3.534992
Residual	0.057261	90	0.000636			

From this set of experiments, we can see that the IPH heuristic performs extremely well and is within 4% of the lower bound. The STH does not perform as well and is within 16% of the lower bound. The tradeoff between both heuristics is that the IPH takes 20 CPU mins to get a solution but gets a good solution, while the STH takes less than a second but yields a worse solution. This is further supported by Table 6 which indicates that the heuristic chosen is statistically significant with a very small P-value, while we fail to reject the null hypothesis that the location is not a statistically significant factor at a p-value critical value of .01 or smaller.

The next parameter that could have an impact on the quality of the heuristics is the demand size. To experiment on this parameter, the demand was set uniformly. The range of these numbers

was changed for each trial and on each trial 10 replications were made. A 25x25 square with random locations was used, with the Port at coordinate (13,0). Also, the capacity of each location is fixed at 17 containers. The results are shown on Table 7.

Table 7. Sensitivity of the results for the demand parameter

Importer Demand	Exporter Demand	Capacity	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
Unif(65-85)	Unif(50-70)	17	1.009472	1.125203
Unif(85-105)	Unif(65,85)	17	1.012793	1.12301
Unif(95-115)	Unif(80-100)	17	1.014162	1.120227
Unif(105-125)	Unif(95-105)	17	1.011246	1.122243
Unif(110-130)	Unif(100-120)	17	1.011848	1.119382
		Avg.	1.011904	1.122013
		Std.	0.001749	0.00231

Table 8. ANOVA table for the demand parameter

Source	SS	df	MS	F	P-value	F crit
Demand	5.88E-05	4	1.47E-05	0.312509	0.86894	3.534992
Heuristic	0.303098	1	0.303098	6445.716	1.51E-85	6.925135
Interaction	0.000277	4	6.93E-05	1.472841	0.217053	3.534992
Residual	0.004232	90	4.7E-05			

As seen in Table 7 the IPH heuristic performs extremely well within 2% of the lower bound and the STH heuristic performs within 13%. In Table 8 we are able to see that once again we fail to reject the null hypothesis that demand is not a statistically significant factor.

For the next set of experiments, we use the same parameter settings, except we change the capacity at each non-port location. The demand for importers is set as a uniform variable ranging from (95-115) while the demand for exporters is set at (80-100). We then ran 10 different scenarios,

each with a different capacity setting for the locations. We ran 10 replications for each scenario. We show our results in Tables 9 and 10.

Table 9. Sensitivity of the results for the capacity parameter

Capacity	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
15	1.018324	1.135487
16	1.000356	1.007962
19	1.008232	1.107865
20	1.000364	1.009226
21	1.004747	1.09701
22	1.000415	1.007719
25	1.001118	1.086323
26	1.000371	1.007213
29	1.000467	1.072623
30	1.000408	1.006602
Avg.	1.00348	1.053803
Std.	0.005842	0.051082

Table 10. ANOVA table for the capacity parameter

Source	SS	df	MS	F	P-value	F crit
Capacity	0.139997	9	0.015555	298.8889	3E-103	2.507228
Heuristic	0.126619	1	0.126619	2432.941	1.7E-106	6.777786
Interaction	0.097919	9	0.01088	209.0537	2.36E-90	2.507228
Residual	0.009368	180	5.2E-05			

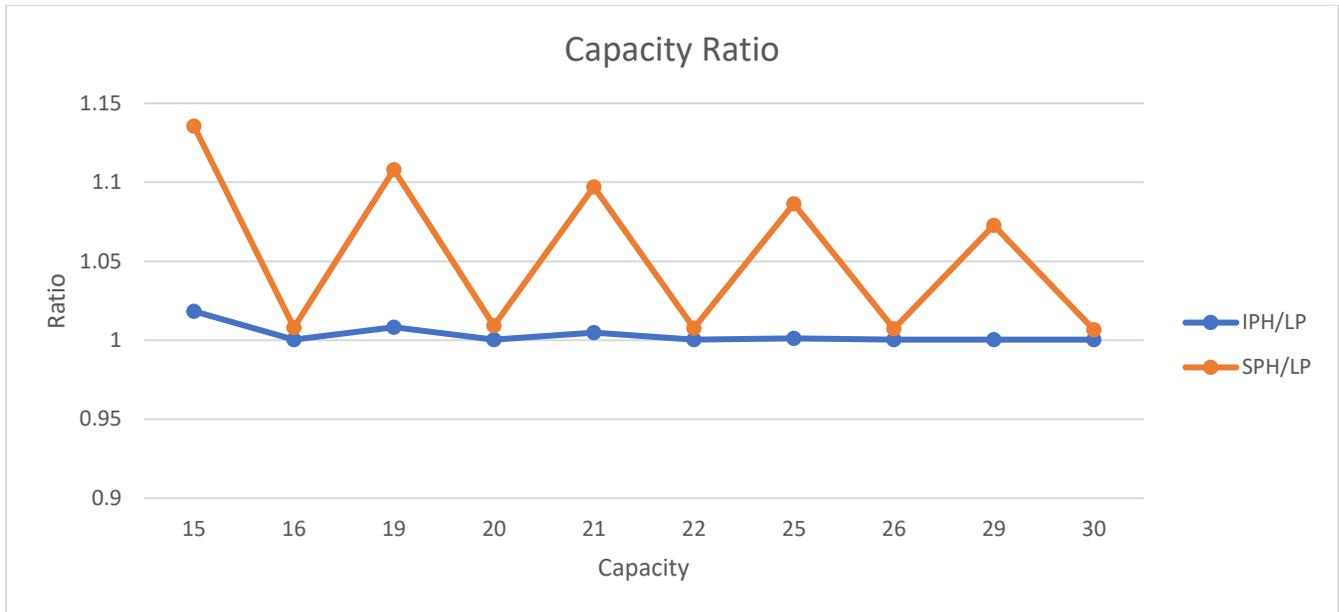


Figure 4. Results of capacity experiment

As seen in Figure 4 the capacity has a big effect on the STH solution, and a smaller effect on the IPH solution. There is also a much bigger effect when capacity is odd as compared to even. Capacity has an effect in the STH solution because of how the LP relaxation assigns the flow of the containers. All the containers start at the Port and then move to an importer, then to an exporter and finally back to the Port. The LP relaxation pairs up a particular importer to an exporter, depending on how costly it is to move a container from that importer to that exporter. It does this for all exporters such that every exporter is assigned to a particular importer while minimizing the total cost. It is for this reason that the total cost increases so much when the capacity is odd. When capacity is even at all locations, the LP relaxation is usually an integer solution. This is why the ratio keeps increasing and decreasing when capacity is odd versus when it is even. For the IPH however the impact of the capacity changes is not as much (both when it is even or odd) because instead of simply using a single-container truck to meet the demand, it pairs multiple locations in such a way that it uses a double-container truck to meet demand. This conclusion is further

supported by the Tables 9 and 10, which shows that both the capacity and the heuristic used have very significant effects on the cost ratio.

The other noticeable effect of Figure 4 is the downward slope especially for the STH heuristic. This downward slope is caused by the fact that as capacity increases the total number of times that the heuristic needs to adjust the flow is decreased because the total number of times that the location capacity needs to be filled goes down, and the “tightness” of the problem also goes down. From this result we can conclude that the capacity does have an effect on the STH heuristic solution quality, and they both perform better when the capacity is even than when it is odd. It also suggests that the demand to capacity ratio is also a factor. As the demand to capacity ratio decreases the heuristic to linear programming ratio goes down. If the ratio is taken all the way to 1 the heuristic will tend to go towards the same result as the linear programming solution. With only a minor difference if the demand is even or odd, which only affects the last unit of demand.

6. Conclusions

This paper introduces a model for the “Empty Container Reuse Problem” that uses double-container trucks. The model was solved using the Gurobi solver for an example based on actual data from the Ports of Los Angeles and Long Beach. The results show that the number of miles and trucks can be significantly reduced by increasing the amount of street exchanges through the use of double-container trucks. This could potentially reduce significant congestion and reduce the impact of container freight movement on the environment. This reduction is especially impactful in big port areas, such as Los Angeles which already has a significant congestion problem.

Experiments were also performed to test the heuristics proposed to solve the model under general cases on randomized data sets. It was determined that location and demand did not have a

significant impact on the heuristics, and they were only impacted by the container capacity at each location. The STH performance is around 15% of the lower bound, but has an extremely fast computation speed while IPH always performed extremely well within 2% of the lower bound, but takes longer to run.

In this paper, we restricted double-container movements to pick up its two containers at the same location. This restriction was done to reduce the complexity of the model. In future work dropping this restriction would lead to further system improvements but at the expense of computational speed. Furthermore, future research could study the stochastic demand case which could consider the optimal positioning of containers for tomorrow's stochastic demand as well as today's known demand.

Acknowledgements

We acknowledge METRANS for their kind support of this research.

References

- Bandeira, D. L., Becker, J. L., & Borenstein, D. (2009). A DSS for integrated distribution of empty and full containers. *Decision Support Systems*, 47(4), 383-397.
- Bourbeau, B., Crainic, T. G., & Gendron, B. (2000). Branch-and-bound parallelization strategies applied to a depot location and container fleet management problem. *Parallel Computing*, 26(1), 27-46.
- Braekers, K., Caris, A., & Janssens, G. K. (2013). Integrated planning of loaded and empty container movements. *Operation Research Spectrum*, 35(2), 457-478.
- Choong, S. T. (2000). *Container Management for Multimodal Transportation Networks*. M.S. Thesis, Department of Industrial Engineering, University of Arkansas, Fayetteville, AR.

-
- Choong, S. T., Cole, M. H., & Kutanoglu, E. (2002). Empty container management for Intermodal transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 38(6), 423-438.
- Hanh, D. L. (2003). The logistics of empty cargo container in the Southern California region. *Mettrans Research Project*.
- Dejax, P. J., & Crainic, T. G. (1987). Survey paper-a review of empty flows and fleet management models in freight transportation. *Transportation Science*, 21(4), 227-248.
- Erera, A. L., Morales, J. C., & Savelsbergh, M. (2009). Robust optimization for empty repositioning problems. *Operation Research*, 57(2), 468-483.
- Islam, S., Arthanari, T., & Olsen, T. (2010). *Empty container-truck movement problem: at Ports of Auckland*. 45th Annual Conference of the ORSNZ.
- Jula, H., Chassiakos, A., & Ioannou, P. (2006). Port dynamic empty container reuse. *Transportation Research Part E: Logistics and Transportation Review*, 42(1), 43-60.
- Lam, S. W., Lee, L. H., & Tang, L. C. (2007). An approximate dynamic programming approach for the empty container allocation problem. *Transportation Research Part C: Emergency Technology*, 15(4), 265-277.
- Li, L., Wang, B., & Cook, D. P. (2014). Enhancing green supply chain initiatives via empty container reuse. *Transportation Research Part E: Logistics and Transportation Review*, 70, 190-204.
- Shen, W. S., & Khoong, C. M. (1995). A DSS for the empty container distribution planning. *Decision Support Systems*, 15(1), 75-82.

The Tioga Group (2002). *Empty ocean logistics study*. Technical Report, Submitted to the Gateway Cities Council of Governments, CA, USA.

Appendix 1

This appendix provides a proof of observation one which states that the Linear Program (LP) relaxation will yield an integer or half integer solution if (1) moving containers is cheaper by either using only single-container trucks or double-container trucks (i.e. $e_{i,j,t} + f_{j,j,t} > 2g_{i,j,t}$ or $e_{i,j,t} + f_{j,j,t} < 2g_{i,j,t}$ for all locations i and j , and all times t). The second assumption (2) is that dropping of the second container at the same location is always cheaper than dropping it off at any other location ($f_{i,i,t} < f_{i,j,t} \forall i, j \in SA \forall i \neq j$). Finally, the third assumption (3) is that we have infinite capacity at each location. We note that in all of our experiments we still get integer and half integer solutions without this assumption, but we need this assumption to prove our observation.

We will prove this observation by using several networks. We will start by noting that because of our first assumption the LP will use only single or double containers, depending on what cost dominates the other. We will first assume that it is cheaper to use single-container trucks and so our solution will only involve single-container truck movements. We will also temporally assume that the turnover time at each location is 0 ($r=0$) for all locations.

We will start by building a graph in which each level represents a time discretization. There will be T discretizations. At each level, there are $2n$ nodes where n is the number of locations. That means that there are two copies of each location, we will refer to the first copy as i and the second copy as i' . Figure 5 shows a sample representation of the graph. At each level t each node i is connected to node j at level $t+l_{i,j,t}$ if location i can supply location j . Node i is also connected to node i' at level $t+1$. Moreover, each node i' is also connected to node j at level $t+l_{i,j,t}$ if location i can supply location j . It is also connected to node i' at level $t+1$. We do this for all i and i' nodes and for all levels t (except for $t=T$). We then set the demand for each node i at each time t (no

demand for nodes i'). Notice that we can add a starting node s at the beginning connected to all locations at i' for $t=1$. We will then add a flow from s to all i' equal to the number of starting containers at location i (p_i). Finally, we will add a terminal node e and add an edge going from all nodes at level $t=T$ to this terminal node. Now, notice that we have just built a network with no loops and all the demands are integer numbers. Thus, the problem can be solved in polynomial time using linear programming, and the solution will be an integer solution.

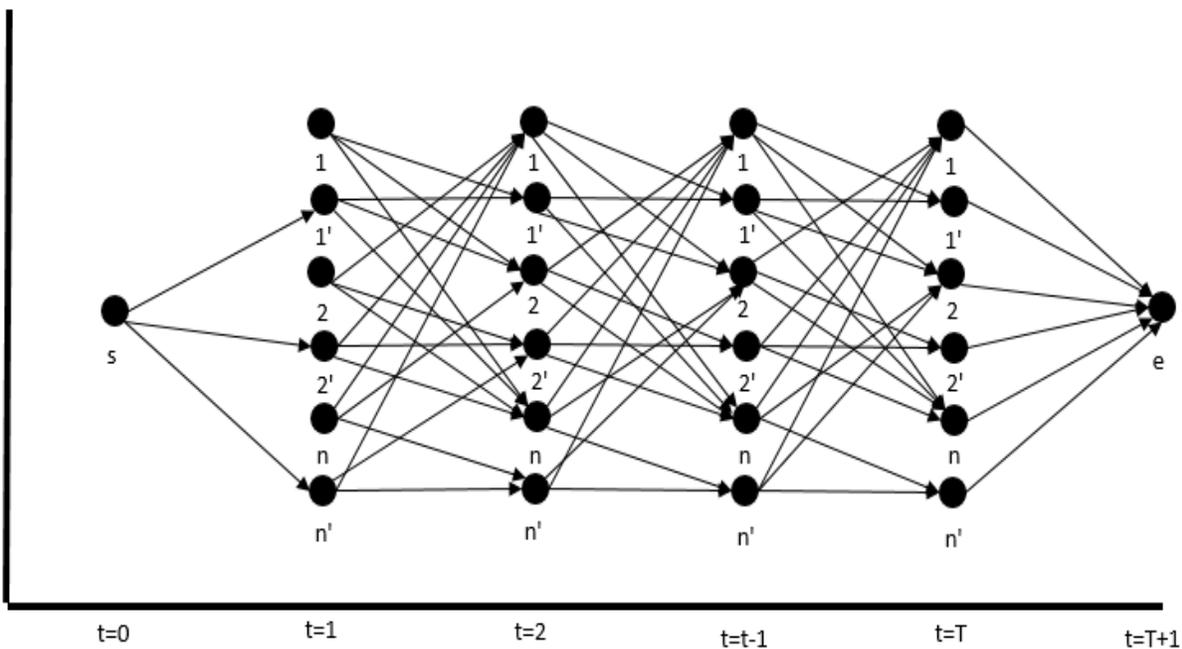


Figure 5. Example of graph for single-container trucks, $r=0$, and unlimited capacity

We now build a second graph dropping the assumption that $r=0$. We will use a similar graph as before. However, at each t level we will have $r+1$ copies of each location labeled $i_1, i_2, i_3, \dots, i_r, i_{r+1}$. Furthermore, we will have an edge going from i_a at level t to i_b at level $t+1$. We will do this for all subscripts $a \in 1, 2, 3, \dots, r-1$ copies (connecting them to the next copy at the next time level), and all time levels. Then, we connect i_r at level t to j_1 at level $t+l_{i,j,t}$, if location i can supply

location j . We will also connect node i_r at time level t to node i_{r+1} at time level $t+1$ (similar to the previous graph). We will then connect i_{r+1} at level t to j_1 at level $t+l_{i,j,t}$ if location i can supply location j . Finally, we will also connect node i_{r+1} at time level t to node i_{r+1} at time level $t+1$. We do this for all nodes i , and all-time levels t (except for $t=T$). Finally, we add a starting node s and connect it to all nodes i_{r+1} at time level $t=1$, and we will add a terminal node e and connect all nodes in time level $t=T$ to node e . Now, if we assume infinite capacity (or M where M is a really large number), then we have a network with no loops and all the demands are integer numbers. Thus, the problem once again can be solved in polynomial time using linear programming and the solution will be an integer solution. We have thus proven the conjecture that the LP will yield an integer when we use only single-container trucks.

We will now assume the other possibility of our first assumption, and assume that double-container trucks are cheaper than single-container trucks. Thus, we will only use double-container trucks. We will also start by assuming that $r=0$. The graph for double-container trucks can become complex. First, we will create a similar graph to the previous graph but we will have exactly 3 copies (i_1, i_2, i_3) of each location for each time level. We will then have an edge from node i_2 at level t to node j_1 at level $t+l_{i,j,t}$, if location i can supply location j . We will then have another edge from node j_1 at level $t+l_{i,j,t}$ to node k_2 at level $t+l_{j,k,t+l_{i,j,t}}$, if location k is a possible second location drop off location of i . For both cases only add the edge if there could be a possible truck route with a double-container route starting at location i then going to location j and ending at location k starting at time t . We will then have an edge from node i_2 at level t to node i_3 at time $t+1$. Furthermore, we will have an edge from node i_3 at time t to node i_3 at time $t+1$. We will also have an edge from node i_3 to node j_1 at time $t+l_{i,j,t}$, if location i can supply location j . We then do this for all nodes i and all levels t (except for $t=T$). Now, notice that there is always an edge going

from node i_1 to node i_2 in the same time level t . This edge represents the second container being dropped off at the same location as the first container. Now, if we assume that dropping of the second container at the same location is always cheaper than dropping it off at any other location ($f_{i,i,t} < f_{i,j,t} \forall i, j \in SA \forall i \neq j$), then we can prove that a linear program will only use routes that deliver the second container to the same location as the first container. This is because $e_{i,j,t} + f_{j,j,t+l_{i,j,t}} < e_{i,j,t} + f_{j,k,t+l_{i,j,t}} \forall i, j \in SA \text{ s. t. } i \neq j$. Thus, we have proven that we will never use the different routes and we can eliminate all the edges from all node i_1 except for the edge going from i_1 to i_2 in the same time level t . Now at this node all the flow that comes into i_1 needs to go to i_2 , because it is the only option. Thus, we can combine i_1 and i_2 into a single node i , knowing that any solution given by the linear program will have to be divided by 2 since each flow unit represents 2 containers arriving at location i . Now, notice that the graph now is exactly the same as the first graph we built for single-container trucks with $r=0$. We have shown that this problem could be solved in polynomial time using linear programming and it will yield an integer solution. Thus, we can solve this problem in polynomial time using linear programming; however in this case we may not necessary get an integer solution, because we must divide the integer solution by 2. Thus, we will get a solution that is either integer or a half integer solution. Similarly, we can drop the assumption that $r=0$ and the resulting graph becomes the second graph we built. Thus, for the $r>0$ case we will also obtain an integer or half integer solution.

Note that the second observation discussed in Section 3.3 of the model is that if both demand and capacity is even, then the model will always yield an integer solution. The half integrality only comes in play when either the demand or capacity at any location is odd. This observation is a direct result from the above proof.