Congestion Reduction through Efficient Empty Container Movement

Final Report

METRANS Project 15-28

August 15, 2017

Principal Investigator:

Maged M. Dessouky

Ph.D. Graduate Student:

Santiago Carvajal

Daniel J. Epstein Department of Industrial and Systems Engineering

University of Southern California

Los Angeles, California

Disclaimer

Abstract

In 2015 the Ports of Los Angeles and Long Beach moved 15.3 million Twenty-foot Equivalent Units (TEU). There is a significant body of work on moving loaded containers efficiently, however there has been little research on the movement of empty containers. Out of the 15.3 million TEUs about 30% or 4.3 million TEUs were empty containers.

Empty container movement is increasing greatly because of the enormous inconvenience for companies to coordinate with each other to exchange empty containers. This problem is known as the Empty Container Problem. This study proposes a mathematical model that solves the empty container problem using double and single container trucks. The model discretizes time and ensures demand is met. By solving the empty container problem, congestion can be reduced since fewer truck trips would be needed to satisfy demand. Furthermore, since double container trucks can deliver two containers per truck trip, the quantity of trucks needed to satisfy the demand is decreased even more, further reducing congestion.

The model was tested using data from the Ports of Los Angeles and Long Beach. The results are promising and show that the number of miles and trucks can be significantly reduced by increasing the number of street exchanges, and further reduced by using double container trucks. This report shows that using a single container policy instead of the current policy would reduce truck miles by about 12%, and would reduce significant truck trips to and from the port. The double container policy reduces truck miles by about 55% compared to the current policy, which is a noteworthy reduction. This could potentially reduce congestion substantially, lessening the impact of container freight movement on the environment

Disclosure

The project was funded in entirety under this contract to California Department of Transportation.

Acknowledgement

We would like to thank METRANS for funding this research.

Contents

List of Tables

List of Figures

Introduction

Background

In today's world, there is a significant body of work regarding the efficient distribution of loaded containers from the ports to consignees. However, to fully maximize the process and better address environmental concerns, study is needed on allocating empty containers created by consignees. This is an essential aspect 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 follows 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 traffic. 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.

Motivation

In this study, we propose a model that allows empty containers to 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 the coordination required between 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 research augments earlier work by proposing the use of double container trucks. Double container trucks would increase the number of street exchanges that could be made since the possibilities are greater with two container trucks. Currently, double truck containers 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. For example they are prohibited from operating in the Ports of Long Angeles and Long Beach since infrastructure improvements are necessary to accommodate double container trucks. This report presents important benefits of using double container trucks on the impact on the reduction of truck routes if the local infrastructure was expanded to account for double container trucks.

We study the Empty Container Reuse Problem with the added feature of adding double container trucks. Since double container trucks can deliver two containers in a single trip, we show that if the port logistics were to adopt this container movement, the number of truck trips and truck miles would decrease, lessening the ecological impact due to container movement. We also show that empty container reuse using single container trucks will significantly reduce both the number of truck trips and the number of truck miles, over the existing routing strategy, where the road infrastructure cannot support double container trucks.

Structure of Report

The rest of this report 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. In Section 6 a heuristic is presented for the construction of a truck schedule for delivery of the assignment of the container movements. In Section 7, we discuss the implementation and applicability of our work. Finally, in Section 8 conclusions are drawn.

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 paper.

One of the earliest models for this 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. 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.

Bandeira et al (2009) developed a rolling horizon model to coordinate different customer demands as to minimize costs. Their model is solved in two steps. First it meets all demand for that time period. Then it adjusts the solution to allocate containers to minimize costs. Erera et al. (2009) built a robust optimization framework for container allocation. This allowed them to find an approximate optimal solution in a dynamic world where future demand for containers are stochastic.

Braekers et al (2013) tackled the dynamic empty container reuse problem. 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.

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.

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.

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.

Problem Statement and Formulation

Problem Description

We assume container demand at each location is given and deterministic for each day. Our model focuses on satisfying all 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.

Figures 1 and 2 show the current and proposed single container flow for both full and empty containers. 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. Figure 3 shows container movement for the proposed double container flow. The arrows for full or empty containers show potential flow for both single containers or two containers of the same type. For example, in Figure 3 a possible two container route involves going from the Port to an exporter to deliver an empty container, and then going from the exporter to an importer to deliver a full container. It is for this reason that Figure 3 has many more options compared to the possible routes in Figure 2. However, this does not mean that an exporter can supply an importer, since it is actually the Port that supplies containers. For this reason, Figure 2 also shows what locations can supply other locations.

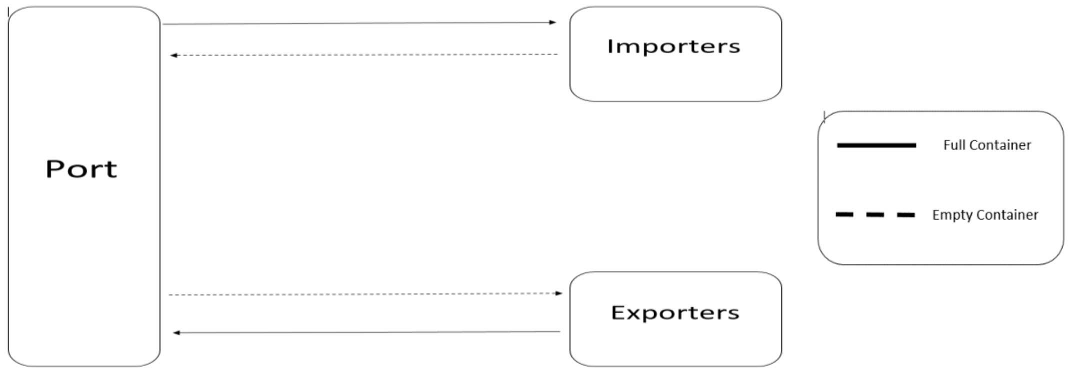

Figure 1. *Current container flow*

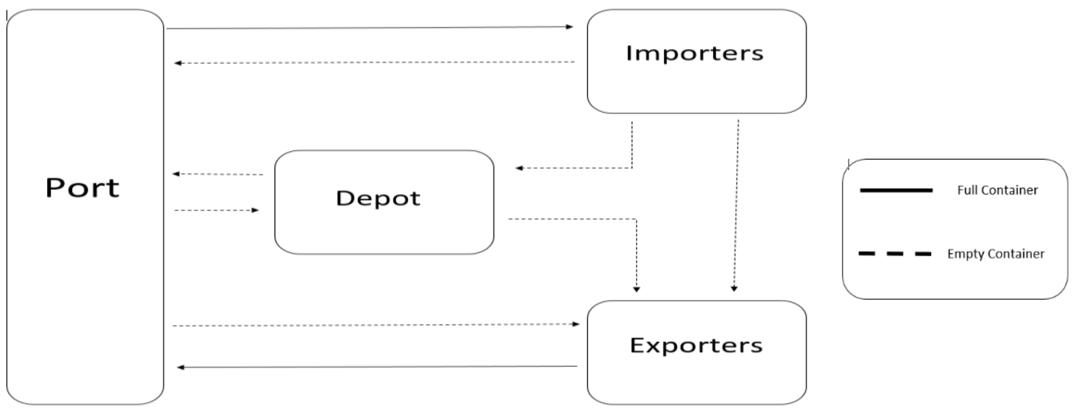

Figure 2. *Proposed single container flow*

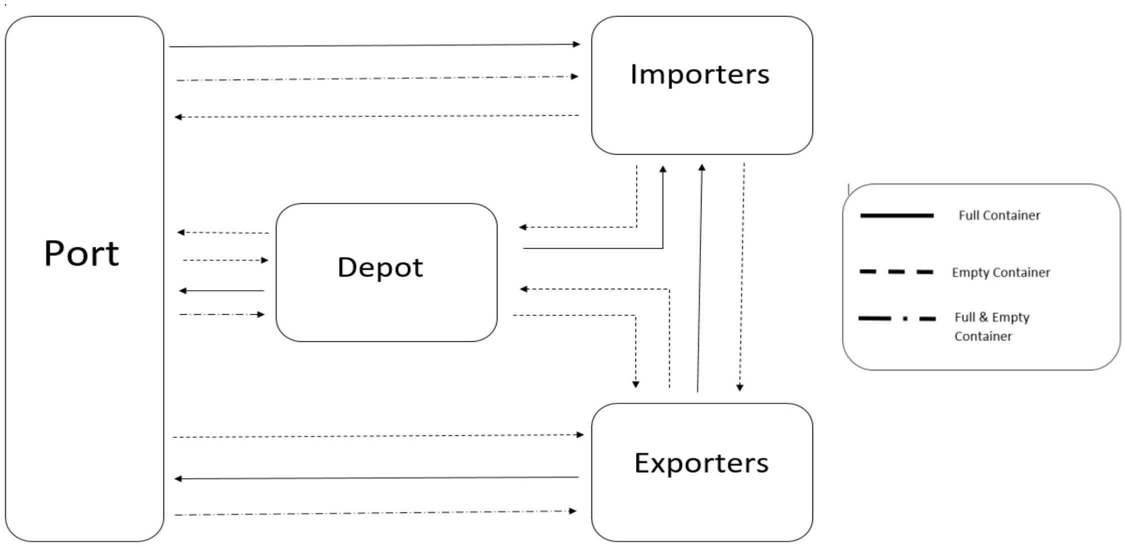

Figure 3. *Proposed double container flow*

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.

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.

Mathematical Formulation

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

Parameters:

$I = Total\ number\ of\ importers$

$E = Total\ number\ of\ exporters$

$D = Total\ number\ of\ Depots$

$T = Number\ of\ time\ discretizations$

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

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

$r_i = Container\ turnover\ time\ at\ location\ i$

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

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

$c_i = Capacity\ of\ location\ i$

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

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

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

Sets:

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

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

$SD = \{I + E + 1, \dots, I + E + D\}$ $(locations\ of\ all\ depots)$

$SP = \{ I + E + D + 1 \}$ $(location\ of\ the\ port)$

$SA = \{SI \cup SE \cup SD \cup SP\}$ $(all\ locations)$

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

Decision Variables:

$x_{i,j,t} = Number\ of\ first\ leg\ two\ container\ trucks\ going\ from\ location\ i\ to\ location\ j\ at\ time\ t$

$y_{i,j,t} = Number\ of\ second\ leg\ two\ container\ trucks\ going\ from\ location\ i\ to\ location\ j$

$at\ time\ t$

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

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

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

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

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

Objective:

$$min \sum_{t\in ST} \sum_{i\in SA} \sum_{j\in SA} (e_{i,j,t} * x_{i,j,t} + f_{i,j,t} * y_{i,j,t} + g_{i,j,t} * z_{i,j,t})$$

s.t.

Containers provided at time t:

$$2 \sum_{j\in SE\cup SD\cup SP} x_{i,j,t} + \sum_{j\in SE\cup SD\cup SP} z_{i,j,t} = m_{i,t} \quad \forall i \in SI \quad \forall t \in ST \quad (Importers) \tag{1}$$

$$2 \sum_{j\in SP} x_{i,j,t} + \sum_{j\in SP} z_{i,j,t} = m_{i,t} \quad \forall i \in SE \quad \forall t \in ST \quad (Exporters) \tag{2}$$

$$2 \sum_{j\in SE\cup SD\cup SP} x_{i,j,t} + \sum_{j\in SE\cup SD\cup SP} z_{i,j,t} = m_{i,t} \quad \forall i \in SD \quad \forall t \in ST \quad (Depots) \tag{3}$$

$$2 \sum_{j\in SI\cup SE\cup SD} x_{i,j,t} + \sum_{j\in SI\cup SE\cup SD} z_{i,j,t} = m_{i,t} \quad \forall i \in SP \quad \forall t \in ST \quad (Port) \tag{4}$$

Containers received at time t:

$$\sum_{i\in SP} x_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SE\cup SD} y_{i,j,t-o_{i,j,t}} + \sum_{i\in SP} z_{i,j,t-o_{i,j,t}} = n_{j,t} \quad \forall j \in SI \quad \forall t \in ST \quad (Importers) \tag{5}$$

$$\sum_{i\in SI\cup SD\cup SP} x_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SE\cup SD} y_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SD\cup SP} z_{i,j,t-o_{i,j,t}} = n_{j,t} \quad \forall j \in SE \quad \forall t \in ST \ (Exporters) \tag{6}$$

$$\sum_{i\in SI\cup SD\cup SP} x_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SE\cup SD} y_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SD\cup SP} z_{i,j,t-o_{i,j,t}} = n_{j,t} \quad \forall j \in SD \quad \forall \in ST \quad (Depots) \tag{7}$$

$$\sum_{i\in SI\cup SE\cup SD} x_{i,j,t-o_{i,j,t}} + \sum_{i\in SA} y_{i,j,t-o_{i,j,t}} + \sum_{i\in SI\cup SE\cup SD} z_{i,j,t-o_{i,j,t}} = n_{j,t} \quad \forall j \in SP \quad \forall t \in ST \quad (Port) \tag{8}$$

Demand and Feasibility constraints:

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

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

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

$$b_{i,t} \geq d_{i,t} \qquad \forall i \in SA \quad \forall t \in ST \quad (Demand\ at\ location\ i\ must\ be\ met\ by\ time\ t) \tag{12}$$

$$b_{i,t} + p_i - a_{i,t} \leq c_i \qquad \forall i \in SA \quad \forall t \in ST \quad (Capacity\ at\ i\ cannot\ be\ exceeded) \tag{13}$$

$$\sum_{i \in SA} x_{i,j,t} = \sum_{k \in SA} y_{j,k,t+l_{i,j,t}} \qquad \forall j \in SA \quad \forall t \in ST \quad (Two\ container\ trucks\ must\ provide\ two\ containers) \tag{14}$$

$$x_{i,j,t}\ , y_{i,j,t}\ , z_{i,j,t}\ \geq 0 \qquad \forall i \in SA \quad \forall j \in SA \quad \forall \in ST \quad (Non-negative\ Constraint) \tag{15}$$

$$x_{i,j,t}\ , y_{i,j,t}\ , z_{i,j,t}\ \in \mathbb{Z} \qquad \forall i \in SA \quad \forall j \in SA \quad \forall \in ST \quad (Integer\ Constraint) \tag{16}$$

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 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. The x 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. The y 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. 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 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 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. 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.

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 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. Therefore all single and double truck trips can only originate from the port. However, the y variables do not need to necessarily originate from the port. There are several ways in which the second leg of a double container truck trip can come from either importers, exporters or depots. In fact, the second leg of a trip cannot originate from the port because logistically it would make no sense to have a trip go from a non-port location to the port, and then return to a non-port location.

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 (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.

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. Our first observation is that the Linear Program (LP) relaxation of the model gives an integer solution when (1) $d_{i,j}$ and c_i are both even numbers for all *i* and *t*, and (2) the costs for single container trips ($g_{i,j,t}$) and double container trips ($e_{i,j,t} + f_{j,k,t}$) are unique for all *i*, *j*, and *k*. Although we were unable to prove this mathematically, this held true under all our experimental settings. The model is similar to solving t basic transportation models, with the added feature being that double container trucks are possible. Now, it is well known that the classic transportation model yields integer solutions when demand is integer. This happens because at every stage a variable (which represents a movement of demand from one location to another) is chosen with the smallest cost, and the value of this variable is increased as much as possible. New variables are chosen until all the demand is met. Because our model assimilates the transportation problem we conjecture that it has similar properties. At every stage the model needs to satisfy a certain demand. The model then finds the variable with the least cost and sends as many containers as possible until no more containers can be sent, or the demand is already satisfied. Next, constraint (14) means that there needs to be a balance between variables $x_{i,j,t}$ and variables $y_{i,j,t}$ (the two parts of a two container truck trip). The model identifies the route with the least cost and sends as many containers as possible. However, if demand is odd this means that in order for the variables $x_{i,j,t}$ and the variables $y_{i,j,t}$ to sum to an odd number, they will have to be half numbers. If demand is even, then constraint (14) is not a problem and the model is conjectured to yield only integer solutions. The second part of the observation requires that the costs be unique because if the costs are equal the model might divide the flow between the different routes, and this division does not necessarily have to be integer.

If the demand ($d_{i,t}$) or capacity (c_i) is odd for any combination of i and t, then the LP relaxation is likely to return either an integer or half integer solution. As discussed before, this is due to constraint (14), and therefore the half integer solutions will always come in pairs. This means that for every half integer $x_{i,j,t}$ there is another half integer $y_{i,j,t}$. The sum of the variables has to be an integer, since demand is an integer. For this reason, a search can be done to pair variables going to the same location that are not integer and rounding them down, then adding a single container truck to that location. By doing this, an integer solution can be recovered, although this solution is not guaranteed to be optimal.

Another observation is that if the cost for a single truck container is strictly greater than double the cost for the double truck container for every segment, then the model will return a solution that uses only double container trucks. On the other hand, if the cost is strictly less, then the model will return a solution that uses only single container trucks. If for some segments the cost for single containers is less than half of the double containers, but in other segments it is the other way around, then the model might return a solution that gives a combination of single and double containers. Also, this solution is not guaranteed to be integer, but it will be half integer.

Heuristics

Under general conditions solving the model as a LP will not yield a feasible solution, since the optimal solution may yield fractional values for the decision variables. In order to get a feasible solution two heuristics are introduced. These heuristics use the result given by the Linear Relaxation Program and yield an approximate solution to the problem.

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*, were *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 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.

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 meet. 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).

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.

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 was aggregated according to container demand over small regions. We use the centroid of the region to represent the location for the aggregated demand of that region. This resulted in a total daily demand of 200 containers by the importers from the Port to the locations. Meanwhile, the amount of containers demanded by the exporters to the Port in this area is about 90 containers daily. We then use a representative location to account for all the demand for that region. In total, we use eleven locations with five importer locations, three exporter locations, and two depots. Table 1 shows the location number in the model either as an importer, an exporter, a depot, or the Port. Table 2 shows the distances between the different locations. Figure 3 is a map of the Port and the surrounding area showing the positioning of the representative locations.

Table 1. *Location types*

Location Number	Location Name
1	Importer 1
2	Importer 2
3	Importer 3
4	Importer 4
5	Importer 5
6	Exporter 1
7	Exporter 2
8	Exporter 3
9	Depot 1
10	Depot 2
11	Port

Table 2. *Distance between locations in miles*

Locations	1	2	3	4	5	6	7	8	9	10	11
1	0	8.2	1.8	6	4	2	2.5	3.9	3.2	6.1	2.3
2	8.2	0	6.7	5.9	5	8	5.1	6.1	4.8	5	13
3	1.8	6.7	0	5.6	3.6	2.4	0.7	3.5	1.7	5.7	5.3
4	6	5.9	5.6	0	3.1	6.9	6.6	3	5.6	3.1	10
5	4	5	3.6	3.1	0	3.9	3.4	1.4	3.6	3.2	8.5
6	2	8	2.4	6.9	3.9	0	3.1	2.7	3.3	7.2	5
7	2.5	5.1	0.7	6.6	3.4	3.1	0	4.1	1	6.7	7
8	3.9	6.1	3.5	3	1.4	2.7	4.1	0	3.8	3.3	7.3
9	3.2	4.8	1.7	5.6	3.6	3.3	1	3.8	0	5.7	6.2
10	6.1	5	5.7	3.1	3.2	7.2	6.7	3.3	5.7	0	10.5
11	2.3	13	5.3	10	8.5	5	7	7.3	6.2	10.5	0

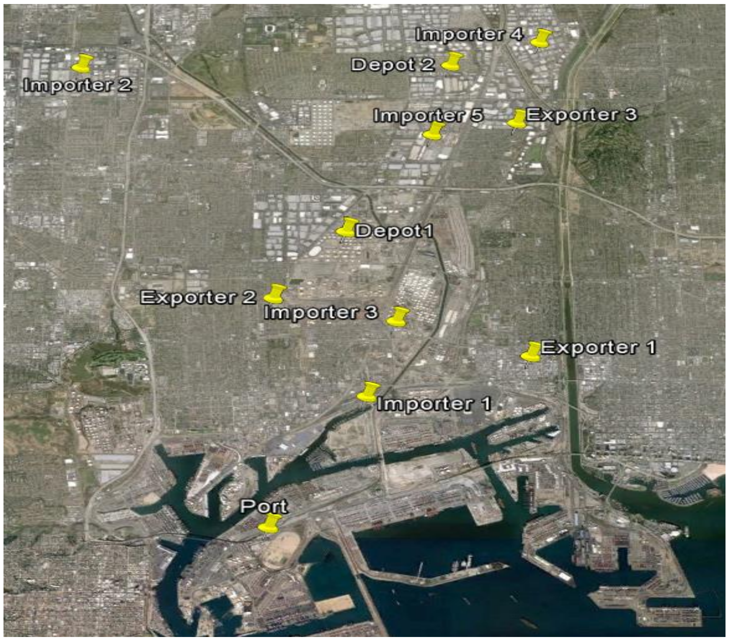

Figure 4. *Position of the locations*

For these set of experiments we assume a 12-hour day in which each of the five importer locations has a demand of 40 by time 9, and each exporter has a demand of 30 by time 9. We 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. Table 3 shows the summary of the parameters used for these set of experiments. 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.

Table 3. *Summary of the parameters of the model*

Parameter name	Parameter value
# of importers (I)	5
# of exporters (E)	3
# of depots (D)	2
Loading and unloading of containers	1 hour
Truck turnover time at port	2 hours
Daily time horizon	12 hours
Time discretization size	1 hour

The model was built in Julia and solved using the Gurobi solver. It is worth noting that double container trucks are currently not allowed to enter the Ports of Los Angeles and Long Beach. There are also not allowed on some roads which means that not all double container truck trips would be possible. Thus, the following experiments help to measure what would be the gain if double container trucks would be allowed in the future. However, the experiments also show the potential gains of using a reuse policy for single container trucks. 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 trucks for every route. Another assumption as well was that it was cheaper to have a double 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 Table 4.

Table 4. *Results from the data of the Ports of Los Angeles and Long Beach*

Scenario	# Double Truck Trips	# Single Trucks Trips	Double Truck Miles	Single Truck Miles	Total Truck Miles
Double Container Reuse	245	0	1558	0	1558
Single Container Reuse	0	490	0	3116	3116
Single Direct (Current)	0	500	0	3702	3702
Double Container (Port Forbidden)	45	400	200.5	2717	2917.5
Double Container (Second leg allowed to Port)	90	310	845	2189	3034

There are some interesting results from these experiments. 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. We therefore prohibited any part of a double 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 truck container 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 Table 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 containers 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 travel. 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 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 truck trips, is more efficient than the current movement of containers, and that both the number of trucks and truck miles are reduced.

Randomly Generated Data Instances

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 are 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. We then test 3 parameters that could have an influence on the quality of the heuristics. These are the position of the locations, demand size, and storage capacity. The parameter settings are summarized in Table 5 below. Finally, for all experiments the IPH is run for 15 CPU minutes.

Table 5. *Parameter settings*

Parameter name	Parameter value
# of importers (I)	7
# of exporters (E)	5
# of depots (D)	2
Loading and unloading of containers	1 hour
Location of port	bottom center
Truck turnover time at port	2 hours
Daily time horizon	12 hours
Time discretization size	15 mins

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). We ran 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 6. 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 6. *Sensitivity of the results for the location parameter*

Grid Size	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
10x10	1.010	1.124
15x15	1.011	1.123
20x20	1.011	1.124
25x25	1.011	1.121
30x30	1.012	1.126
Avg.	1.011	1.124
Std.	0.0007	0.002

From this set of experiments we can see that the IPH heuristic performs extremely well and is within 2% of the lower bound. The STH does not perform as well and is within 12.6% of the lower bound. The tradeoff between both heuristics is that the IPH takes 15 mins to get a solution but gets a good solution, while the STH takes less than a second but yields a worse solution. The location parameter does not really have an impactful effect on either heuristic. For this reason, a 25x25 square with random locations are used for the rest of the experiments.

The next parameter that can 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. As stated before a 25x25 square with random locations is 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 demand parameter*

Importer Demand	Exporter Demand	Capacity	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
Unif(65-85)	Unif(50-70)	17	1.007	1.120
Unif(85-105)	Unif(65,85)	17	1.003	1.122
Unif(95-115)	Unif(80-100)	17	1.010	1.120

Unif(105-125)	Unif(95-105)	17	1.012	1.121
Unif(110-130)	Unif(100-120)	17	1.009	1.124
		Avg.	1.008	1.121
		Std.	0.003	0.002

As seen in Table 7 the IPH heuristic performs extremely well within 2% of the lower bound and the STH heuristic performs within 12.1% on average. The demand size does not seem to have any significant impact on either heuristic. The STH seems to decrease only slightly when the demand size increases.

For the next set of experiments, we use the same parameter settings, except we change the capacity. 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. Table 8 shows the results, and we also plot the results in Figure 5.

Table 8. *Sensitivity of the results for the capacity parameter*

Cap	Total Cost Ratio IPH/LP	Total Cost Ratio STH/LP
15	1.015	1.125
16	1.002	1.009
19	1.010	1.106
20	1.002	1.008
21	1.008	1.090
22	1.001	1.006
25	1.005	1.076
26	1.0009	1.006
29	1.0010	1.062
30	1.0007	1.005
Avg.	1.005	1.049
Std.	0.005	0.048

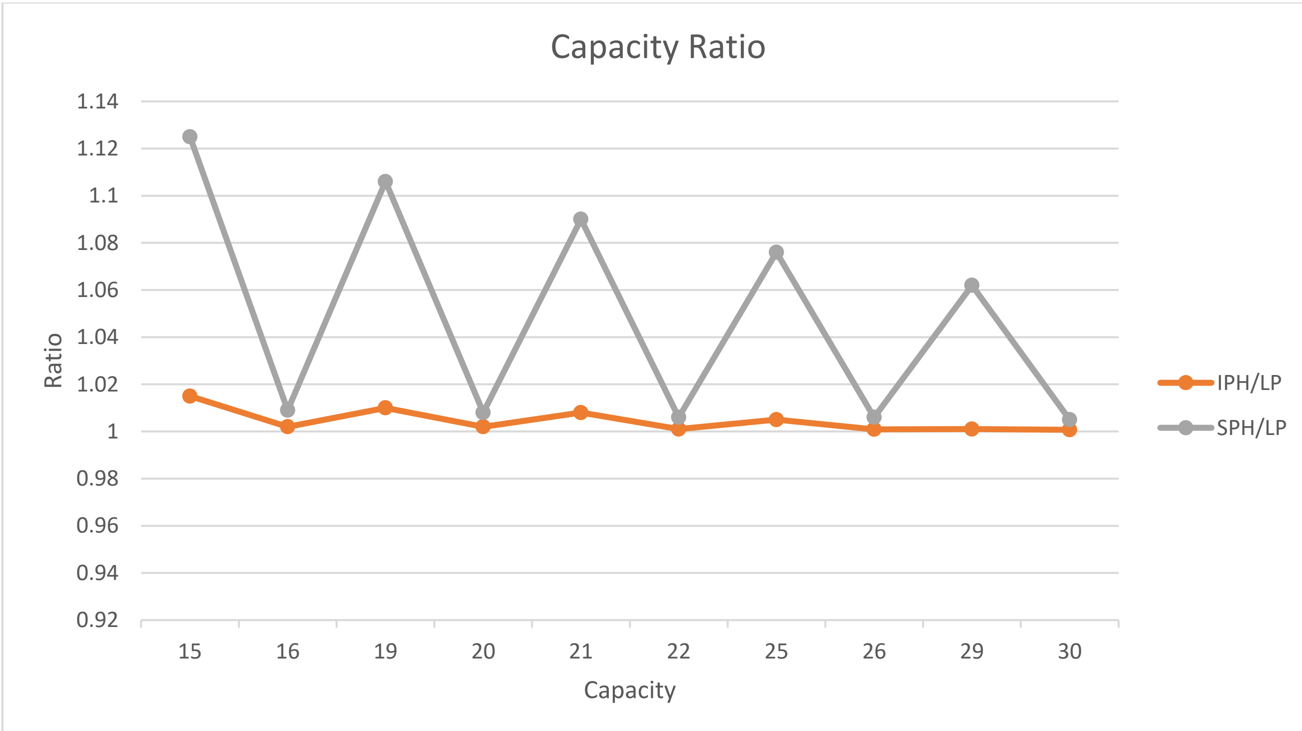

Figure 5. *Sensitivity of the results to capacity*

As seen in Table 8 and Figure 5 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 truck to meet the demand, it pairs multiple locations in such a way that it uses a double truck to meet demand.

The other noticeable effect of Figure 5 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.

Truck Routing

Up to this point, this report has made the assumption that trucks are not a limiting resource and that there are enough trucks in the area ready to respond to any container movement. We now relax this assumption. This means that trucks are a limiting resource and that a truck routing plan is needed to direct trucks throughout the day. The construction of a vehicle route is a well-known problem called "The Vehicle Routing Problem" (VRP). To add this complexity to the Double Container Reuse Model we will add one parameter, one variable, and four constraints to the model, as well as modifying the objective. Below we introduce the parameter, variables, and constraints that must be added to the double container reuse model in order to yield feasible truck routes. We will call this model the Double Container Truck Route Model (DCTRM).

Double Container Truck Route Model

Added Parameter:

$$\pi_{i,j,t} = Cost\ of\ a\ truck\ with\ no\ containers\ moving\ from\ location\ i\ to\ location\ j\ starting\ at\ time\ t$$

Added Variable:

$\tau_{i,j,t}$
$= Number\ of\ trucks\ with\ no\ contianers\ going\ from\ \ location\ i\ to\ location\ j\ starting\ at\ time\ t$

Modified Objective:

$$min \sum_{t\in ST} \sum_{i\in SA} \sum_{j\in SA} (e_{i,j,t} * x_{i,j,t} + f_{i,j,t} * y_{i,j,t} + g_{i,j,t} * z_{i,j,t}) + \sum_{t\in ST} \sum_{i\in SA+\{0\}} \sum_{j\in SA} \pi_{i,j,t} * \tau_{i,j,t}$$

Added Constraints:

$$\sum_{j\in SA} x_{i,j,t} + \sum_{j\in SA} z_{i,j,t} = \sum_{j\in SA+\{0\}} \tau_{j,i,t-o_{i,j,t}} \qquad \forall i \in SA \quad \forall t \in ST \tag{17}$$

$$\sum_{i\in SA} \sum_{q=1}^{t} y_{i,j,q-o_{i,j,k}} + \sum_{i\in SA} \sum_{q=1}^{t} z_{i,j,q-o_{i,j,k}} \geq \sum_{i\in SA} \sum_{q=1}^{t} \tau_{j,i,q} \quad \forall j \in SA \quad \forall t \in ST \tag{18}$$

$$\tau_{i,j,t} \geq 0 \qquad \forall i \in SA \quad \forall j \in SA \quad \forall t \in ST \qquad (Non-negative\ Constraint) \tag{19}$$

$$\tau_{i,j,t} \in \mathbb{Z} \qquad \forall i \in SA \quad \forall j \in SA \quad \forall t \in ST \qquad (Integer\ Constraint) \tag{20}$$

To keep track on the number of trucks used, we have also added the zeroth {0} location. This location is an artificial location and determines whether a truck is initiated for use on a given route. This location can be thought as the truck depot. All the added constraints revolve around the introduction of the new variable $\tau_{i,j,t}$. This variable corresponds to a truck without a container going from location *i* to location *j* starting at time *t*. The cost of moving a truck without a contianer from location *i* to location *j* starting at time *t* is $\pi_{i,j,t}$, in other words the cost of one unit of $\tau_{i,j,t}$. However, $\pi_{0,j,t}$ is always the cost of using one extra truck for any *j* and *t*. This cost is usually much higher than any possible distance reduction that could be made by any truck route. Therefore, the model first prioritizes reducing the number of trucks, and then reducing the total truck miles.

The new added constraints (17) and (18) can be thought as enforcing the conservation of trucks. Constraint (17) equals the number of trucks that leave location *i* with containers to the number of trucks without containers that must arrive at location *i* at time *t*. Constraint (18) enforces the constraint that the number of trucks with containers that end their job at location j ($y_{i,j,t}$ for double truck trips and $z_{i,j,t}$ for single truck trips) by time t must be greater than or equal to the total number of trucks without containers that leave location *j* by time *t*. Constraints (19) and (20) state that the new VRP variable must be integer and non-negative.

Once the model is solved we can build truck routes. Notice that the Double Container Truck Route Model will yield both the movement of trucks with containers (either loaded or empty) variables ($x_{i,j,t}$, $y_{i,j,t}$, or $z_{i,j,t}$), and the movement of trucks without container variables ($\tau_{i,j,t}$). Truck routes are stored in ordered sets called R_p. These ordered sets contain tuples that will direct the trucks how to move. For example, the tuple $\{i,\ j,\ k,\ t\}$ means that a double container truck *p* should go from location *i*, to location *j,* to location *k*, starting at time *t*. These tuples represent two types of trips: truck movements without a container and movement of trucks with container (either loaded or unloaded) also referred to as jobs. For each truck *p*, the ordered set R_p alternates between a truck movement with no containers and a truck movement with containers. Truck movement without containers are stored in the order set R_p only at odd positions. Thus, a tuple $\{i,\ j,\ j,\ t\}$ found in an odd position in the ordered set R_p represents that truck *p* moves without any containers from location *i* to location *j* at time *t*. By definition the first tuple in any route is $\{0,\ i,\ i,\ t\}$, which means that truck *p* starts its route at location *i* at time *t*.

Truck movements with containers are referred to as jobs, which can be either a single container job or a double container job. For a double container truck movement, a job is defined by the tuple $\{i,\ j,\ k,\ t\}$, where the movement of the first leg of the double container is from location *i* to location *j*, and the second leg is from location *j* to location *k* at time *t.* Similarly, a single container truck job is defined by the tuple $\{i,\ j,\ j,\ t\}$ where the truck moves one container from location *i* to location *j* starting at time *t*. These jobs are also stored in the ordered sets R_p but only in even positions. Finally, let $w_{i,j,k,t}$ be the number of job tuples of the form $\{i,\ j,\ k,\ t\}$ in R_p for all *p*. We next describe how to construct the ordered sets R_p, which will yield the truck routes. We call this algorithm "Truck Route Construction".

Truck Route Construction

1. Solve the Double Container Truck Model to get the job variables ($x_{i,j,t}$, $y_{i,j,t}$, or $z_{i,j,t}$) and the truck movement without any containers variables ($\tau_{i,j,t}$).
2. Set all $w_{i,j,k,t}$ such that:
    a. $w_{i,j,k,t} = (x_{i,j,t} + y_{j,k,t+l_{i,j,t}})/2$
    b. $w_{i,j,j,t} = z_{i,j,t}$
3. Set $p = 1$
4. Choose any positive $\tau_{0,j,t}$. Suppose we choose $\tau_{0,i',t'}$.
5. Add the truck movement without container tuple $\{0, i', i', t'\}$ to the ordered set R_p
6. Set $\tau_{0,i',t'} = \tau_{0,i',t'} - 1$
7. Set $q' = t' + l_{0,i',t'}$
8. Choose any positive $w_{i',j,k,q'}$. Suppose we choose $w_{i',j',k',q'}$.
9. Add the truck job tuple $\{i', j', k', s'\}$ to the ordered set R_p
10. Set $w_{i',j',k',q'} = w_{i',j',k',q'} - 1$
11. Set $r = q' + l_{i',j',q'} + l_{j',k',q'+l_{i',j',q'}}$
12. If there are no positive $\tau_{k',\alpha,u}$ go to Step 16, where $u \geq r$ and $u \in ST$. Otherwise, choose a positive $\tau_{k',\alpha,u}$ with the smallest u. Suppose we choose $\tau_{k',\alpha',u'}$.
13. Add the truck movement without container tuple $\{k', \alpha', \alpha', u'\}$ to the ordered set R_p
14. Set $\tau_{k',\alpha',u'} = \tau_{k',\alpha',u'} - 1$
15. Go back to Step 8 with $q' = u' + l_{k',\alpha',u'}$ and $i' = \alpha'$
16. If there is a positive $\tau_{0,j,t}$ set $p = p + 1$ and go back to Step 4. Otherwise STOP and Return R_p for all p.

This algorithm initializes a truck route p by first choosing a positive $\tau_{0,j,t}$. After it has chosen an initial starting location, constraint (17) guarantees that there is a job waiting at that location. It then services one of those jobs. It will end the job at location k'. Then constraint (18) states that if truck p needs to service another job then there will be a positive $\tau_{k',\alpha,u}$. If there isn't then truck's p route ends there. If there is, then we route truck p to the start of its next job. One property of the algorithm is that at the end we will be left with P number of truck routes were $P = \sum_{j \in SA} \sum_{t \in ST} \tau_{0,j,t}$. Finally, also when the algorithm stops because of the combination of constraints (17) and (18), all $w_{i,j,k,t}$ will equal zero meaning that all jobs are satisfied.

Double Container Truck Route Algorithm

The Double Container Truck Route Model is computationally hard to optimally solve since it is a combination of two large scale problems (the Empty Container Reuse Problem and a VRP). Thus, our approach to solve the problem separates both problems. We first solve the Double Container Reuse Model (presented in section 3.2) which solves the Empty Container Reuse Problem. Solving this problem will only yield the truck movement with container variables ($x_{i,j,t}$, $y_{i,j,t}$, or $z_{i,j,t}$) and not the truck movement without container variables ($\tau_{i,j,t}$). We then use the truck movement with container variables to generate truck jobs in order to solve a VRP problem using a modified version of Ropke and Pisinger's Adaptive Large Neighborhood Search Heuristic (ALNS) (2006).

ALNS is a genetic algorithm that iteratively modifies a solution. However, to start using the algorithm we must first build an initial feasible solution. Similar to the previous section we must first convert the truck movement with container variables ($x_{i,j,t}$, $y_{i,j,t}$, or $z_{i,j,t}$) into truck jobs ($w_{i,j,k,t}$). We also use ordered sets L_p to store tuples with the truck routes. However, unlike R_p, L_p only holds job tuples, and does not hold any tuples for truck movement without containers because in this section we do not have the truck movement without container variables ($\tau_{i,j,t}$). In the ordered set L_p it is assumed then that trucks move without containers between jobs. For example, if in the ordered set L_p the first tuple is $\{i,\ j,\ k,\ t\}$ and the second tuple is $\{\alpha,\ \beta,\ \gamma,\ s\}$. Truck p will go to location i service the first job at time t, which will end at location k. It will then move without containers from location k to location α and service job two starting at time s. By construction there will be enough time for the truck to move from the end of one job, to the beginning of the next job. We also use

σ_p to represent the ending time of the last job in truck route p. To be more specific $\sigma_p = t + l_{i,j,t} + l_{j,k,t+l_{i,j,t}}$ where all the subscripts come from the last tuple $\{i,\ j,\ k,\ t\}$ inserted into the ordered set L_p ($l_{i,j,t}$ is the parameter introduced in section 3.2 and it is the travel time from location i to location j). If the ordered set L_p is empty, set σ_p to be 0. Finally, we let $\delta_{\alpha,i,t}$ be the travel time from the last location of the last job in the ordered set L_p (α) to location i arriving at time t. To be more specific if the last tuple in the ordered set L_p is $\{\beta, \gamma, \alpha,\ s\}$ and we are considering adding the tuple $\{i,\ j,\ k,\ t\}$, then $\delta_{\alpha,i,t}$= $o_{\alpha,i,t}$. We now present our algorithm to get an initial feasible solution below. We call this algorithm "VRP Initial Solution Construction".

VRP Initial Solution Construction

1. Solve the Double Container Reuse Model to get the job variables ($x_{i,j,t}$, $y_{i,j,t}$, or $z_{i,j,t}$).
2. Set all $w_{i,j,k,t}$ such that
    a. $w_{i,j,k,t} = x_{i,j,t} + y_{j,k,t+l_{i,j,t}}$
    b. $w_{i,j,j,t} = z_{i,j,t}$
3. Set $p = 1$
4. Set $\alpha = 0$
5. Choose a positive $w_{i,j,k,t}$ with the smallest t subscript such that:

    $t - \delta_{\alpha,i,t} \geq \sigma_p$. Break ties based on the smallest distance between α and i. Suppose we choose $w_{i',j',k',t'}$
6. Add the tuple $\{i', j', k', t'\}$ to the ordered set L_p.
7. Set $w_{i',j',k',t'} = w_{i',j',k',t'} - 1$
8. Set $\alpha = i'$
9. Repeat Steps 5 to 8 until no more $w_{i,j,k,t}$ can be chosen in Step 5.
10. If there is at least one positive $w_{i,j,k,t}$ set $p = p + 1$ and go back to Step 4. Otherwise STOP.

The above algorithm is a greedy algorithm and uses a heuristic that tries to minimize the idle time. It tries to accomplish this by inserting jobs that minimize the time between the last job added to a truck route p and the new time of the job being added to truck route p. The algorithm starts by adding the job with the earliest starting time. It then calculates the time that it takes to service this job. We choose the next job, in such a way that we minimize the idle time of truck p. We do this iteratively until we cannot add any more jobs to truck p. At this time, if there are more jobs to be serviced we add another truck and repeat the process. The algorithm will always yield a feasible truck schedule, because the algorithm keeps adding trucks until there are no more jobs and a new truck can at least complete one job. The algorithm (excluding solving the Double Container Reuse Model) can be implemented in $O(n^2)$ time, were n is the total number of jobs. This algorithm yields a feasible solution that can be used to perform a modified ALNS.

Before we present our modified version of ALNS we need to introduce some parameters. The first parameter ζ determines how many single job truck routes each iteration will try to eliminate. The second parameter Δ determines how many jobs will be removed and reinserted at every iteration. Notice that $\Delta \geq \zeta$ since removing one truck means that one job is also removed. The third parameter Ψ determines how many iterations of the heuristics will be performed. Conversely, the variable ψ gives the current iteration number. Furthermore, let p_{max} represent the maximum number of trucks that are currently being used. We introduce a new set G which will hold the removed jobs that later will need to be reinserted back to some route in order to preserve feasibility. Next, let ƨ represent the minimum cost of adding a job. Finally, ƕ holds the place where the minimum cost of inserting a job appears. We now introduce our modified ALNS.

Modified ALNS

1. Set $\psi = 1$
2. Set $p = 1$

3. If ordered set L_p contains only one tuple. Remove it from the ordered set L_p and add the tuple to set G. Then set $p_{max} = p_{max} - 1$
4. If set G has ζ elements go to Step 6. Otherwise, CONTINUE.
5. If $p = p_{max}$, CONTINUE. Otherwise, set $p = p + 1$ and go back to Step 3.
6. Randomly remove any tuple from a random truck route (L_p) and add it to set *G.*
7. If G has less than Δ elements go back to Step 6. Otherwise CONTINUE.
8. Sort the tuples in G based on their starting time (t).
9. Remove the first tuple $\{i', j', k', t'\}$ from *G.*
10. Set $p = 1$
11. Set $ƨ = \infty$ and $ƕ = \{0\}$
12. If tuple $\{i', j', k', t'\}$ can be inserted on truck route L_p. Calculate the additional cost of inserting the job on route L_p. If this cost is less than ƨ. Set $ƕ = p$. Otherwise, CONTINUE.
13. If $p = p_{max}$, CONTINUE. Otherwise, set $p = p + 1$ and go back to Step 12.
14. If $ƨ < \infty$ insert tuple $\{i', j', k', t'\}$ to truck route ƕ. Otherwise, set $p_{max} = p_{max} + 1$ and add tuple $\{i', j', k', t'\}$ to truck route $L_{p_{max}}$.
15. If G is empty, CONTINUE. Otherwise, go back to Step 9.
16. If $\psi = \Psi$, STOP. Otherwise, set $\psi = \psi + 1$ and go back to Step 2.

The idea of this algorithm is that at each iteration some jobs will be removed along with some trucks which only have one job. The jobs are then reinserted such that the total cost is reduced in the long run, although it may increase at a particular iteration, since increasing the cost at some iterations will allow the heuristic from getting stuck at a local minima. As shown in the paper by Ropke and Pisinger the algorithm in practice does tend to perform very well compared to other well-known algorithms, although no theoretical performance is shown.

Experiments

In this section, we use our truck routing heuristic on the Ports of Los Angeles and Long Beach data set introduced in Section 5.1. The same parameters are used as the original set of experiments. We assume the cost of using an additional truck is much greater than any mileage cost. That is, minimizing the number of trucks is more important than any mileage reduction that could take place when using an extra truck. We also set the parameter ω to 10, which represents the number of jobs to be removed in every iteration of the ALNS. We also set the parameter ρ to 2, which is the number of trucks that are attempted to be removed at every iteration. Finally, we set ϑ to 1000, which is the number of ALNS iterations that will be performed. These extra parameters are summarized below in Table 9.The results are shown below in Table 10.

Table 9. *Summary of parameters for VRP experiment for the Ports of Los Angeles and Long Beach*

Parameter name	Parameter value
# of importers (I)	5
# of exporters (E)	3
# of depots (D)	2
Loading and unloading of containers	1 hour
Truck turnover time at port	2 hours
Daily time horizon	12 hours
Time discretization size	1 hour
Number of ALNS iterations (Ψ)	1000
Number of jobs to remove at each iteration (Δ)	10
Number of trucks to be removed at each iteration (ζ)	2

Table 10. *Truck routing results for the Ports of Los Angeles and Long Beach*

	# Double Trucks	# Single Trucks	Double Truck Miles	Single Truck Miles	Empty Truck Miles	Total Truck Miles
Double Container Reuse	100	0	1555.7	0	341	1896.7
Single Container Reuse	0	200	0	3113.7	615	3728.7
Single Direct (Current)	0	200	0	3699.7	546.8	4246.5

As seen from the results it is preferable to use double container trucks. Once again the routes are the same as using single container trucks with reuse, but the single trucks must do everything twice. Therefore, twice as many trucks are needed and twice as many truck miles are needed to fulfill the demand. Meanwhile the single container truck reuse model uses the same number of trucks as the single container direct. However, the trucks miles are reduced by about 14%. This means that there is a lot of savings to be made even when only using the single container reuse policy as opposed to using the direct policy currently in practice.

Implementation

This problem addresses how to efficiently move empty containers to reduce the number of total trucks and truck miles that are required to meet demand. As more and more containers pass through ports every year, it becomes increasingly more important to efficiently move these containers. As shown in this report empty container reuse helps improve the container movement, and reduce congestion at the port.

Furthermore, it has been shown that if laws and infrastructure were to be modified to allow double container trucks to operate, there would be a lot of efficiency gained. We ran experiments, both on randomized data sets and using data from the Ports of Los Angeles and Long Beach to show that these gains can be significant. Additionally, the approach that this paper developed can be implemented to yield truck routes for both loaded and empty container movements. The implementation of our approach will require a programming language, such as Julia, and an optimization solver, such as Gurobi.

Conclusions

A model that meets all demands for containers using both single and double container trucks is proposed. 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 look promising and show that the amount of miles and number of trucks can be significantly reduced by increasing the amount of street exchanges, and further reduced by using double container trucks. This could potentially reduce significant congestion and reduce the impact of container freight movement on the environment. Furthermore, we showed that using a single container policy instead of the current policy would reduce truck miles by about 12%, and reduce significant truck trips to and from the port. The double container policy reduces truck miles by about 55%, compared to the current policy, which is a significant reduction in congestion.

Experiments were also performed to test the heuristic on randomized data sets. In the following experiments, it was determined that the Single Truck Heuristic solution's quality was not affected by the locations of the importers and exporters, but was highly affected by the ratio of demand over location capacity. However, this heuristic experimentally performs within 15% of the lower bound, and is a very fast heuristic to implement. The second heuristic that was tested is not affected by any parameter, and performs extremely well under all conditions. This heuristic however takes a little longer to find a solution than the previous heuristic, and may have some scalability problems. These findings leads us to believe that the model proposed in the report is robust and could potentially be adapted for other ports or container rail yards.

One future research direction could be to relax the assumption that all demand is deterministic. This is a reasonable assumption if only one day is being modeled. Nevertheless, to become even more efficient and use the depots to their maximal potential, a stochastic model might be used where today's demand is still deterministic, but containers can be allocated in such a way as to anticipate future demand.

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. OR spectrum, 35(2), 457-478.

Dam Hanh, Le. 2003. The logistics of empty cargo container in the southern california region. Metrans 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. Operations Research 57(2) 468-483.

Jula, H., Chassiakos A., & Ioannou, P. (2006). Port dynamic empty container reuse. Transportation Research Part E: Logistics and Transportation Review 42(1) 43-60.

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

Ropke, S., & Pisinger, D. (2006, November). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40(4), 455-472.

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