This paper proposes a mathematical model for the empty container problem using double container trucks. The model discretizes time and makes sure demand is met. By solving the empty container problem, congestion 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 congestion. 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 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 fully maximize the process and become more environmental friendly in the process, 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.

In this study, 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 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.

2. Literature Review

There has been some prior research on the Empty Container Reuse Problem. 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. Bourbeau et al. (2000) developed a mixed integer model and used a parallel branch and bound approach to optimize the location of the depot as well as suggesting a container flow. Bandeira et al. (2009) developed a rolling horizon model to coordinate different customer demands as to minimize costs. Erera et al. (2009) built a robust optimization framework for container allocation, used for stochastic future demand.

Braekers et al. (2013) tackled the dynamic empty container reuse problem. 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. Li et al. (2014) studied the problem at a more global view. They built a model that maximized profit for the shipping company and tested it using ports in China.

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. 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.
3. Single and Double Container Movement Model

3.1 Problem Description

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 and proposed single container flow for both full and empty containers. As can be seen in the figure 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 1 also 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 the proposed double container flow 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 this last policy has many more options compared to the possible routes in the other two policies. However, this does not mean that an exporter can supply an importer, since it is actually the Port that supplies containers. For this reason, the single container flow also shows what locations can supply other locations.
Figure 1. Container flow for different policies
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.

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:**

- \( I \) = 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 \( t \)
- \( a_{i,j,t} \) = 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:**
\( SI = \{1, \ldots, I\} \) (locations of all importers)
\( SE = \{I + 1, \ldots, I + E\} \) (locations of all exporters)
\( SD = \{I + E + 1, \ldots, I + E + D\} \) (locations of all depots)
\( SP = \{I + E + D + 1\} \) (location of the port)
\( SA = \{SI \cup SE \cup SD \cup SP\} \) (all locations)
\( ST = \{1, \ldots, T\} \) (times of the day)

**Decision Variables:**

\( x_{i,j,t} = \text{Number of first leg two container trucks going from location } i \text{ to location } j \text{ at time } t \)
\( y_{i,j,t} = \text{Number of second leg two container trucks going from location } i \text{ to location } j \text{ at time } t \)
\( z_{i,j,t} = \text{Number of single container trucks going from location } i \text{ to } j \text{ at time } t \)
\( m_{i,t} = \text{Number of containers supplied by location } i \text{ at time } t \)
\( n_{i,t} = \text{Number of containers delivered to location } i \text{ at time } t \)
\( a_{i,t} = \text{Number of containers that have been supplied by location } i \text{ by time } t \)
\( b_{i,t} = \text{Number of containers that have been delivered to location } i \text{ by time } t \)

**Objective:**

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

**s.t.**

Containers provided at time \( t \):

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

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

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

\[
2 \sum_{j \in SP} x_{i,j,t} + \sum_{j \in SE} z_{i,j,t} = m_{i,t} \quad \forall i \in SP \quad \forall t \in ST \quad \text{(Port)} \tag{4}
\]
Containers received at time $t$:

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

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

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

$$\sum_{i \in USEUSe} x_{i,t-o_{i,t}} + \sum_{i \in USEUSe} y_{i,t-o_{i,t}} + \sum_{i \in SP} z_{i,t-o_{i,t}} = n_{j,t} \quad \forall j \in SP \forall t \in ST \quad \text{(Port)} \quad (8)$$

Demand and Feasibility constraints:

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

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

$$b_{i,t} - a_{i,t} + p_{t} - c_{i} \geq 0 \quad \forall i \in SA \quad \forall t \in ST \quad \text{(Capacity at i cannot be exceeded)} \quad (13)$$

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

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

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

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-
on 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.

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. 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.
4. 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.

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 $y_{j,i,t}$) 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.

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 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)

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 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 distances between the different locations.

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

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. Loading and unloading of containers (\(r_l\)) is 1 hour at all locations. Finally, we also assume that truck turnover rate 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 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 2.

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

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 2.

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 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 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.

5.2 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. Finally, for all experiments the IPH is run for 15 CPU minutes.

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 3. 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 3. Sensitivity of the results for the location parameter**

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Total Cost Ratio IPH/LP</th>
<th>Total Cost Ratio STH/LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10x10</td>
<td>1.010</td>
<td>1.124</td>
</tr>
<tr>
<td>15x15</td>
<td>1.011</td>
<td>1.123</td>
</tr>
<tr>
<td>20x20</td>
<td>1.011</td>
<td>1.124</td>
</tr>
<tr>
<td>25x25</td>
<td>1.011</td>
<td>1.121</td>
</tr>
<tr>
<td>30x30</td>
<td>1.012</td>
<td>1.126</td>
</tr>
<tr>
<td>Avg.</td>
<td>1.011</td>
<td>1.124</td>
</tr>
<tr>
<td>Std.</td>
<td>0.0007</td>
<td>0.002</td>
</tr>
</tbody>
</table>
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 4.

<table>
<thead>
<tr>
<th>Importer Demand</th>
<th>Exporter Demand</th>
<th>Capacity</th>
<th>Total Cost Ratio IPH/LP</th>
<th>Total Cost Ratio STH/LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unif(65-85)</td>
<td>Unif(50-70)</td>
<td>17</td>
<td>1.007</td>
<td>1.120</td>
</tr>
<tr>
<td>Unif(85-105)</td>
<td>Unif(65,85)</td>
<td>17</td>
<td>1.003</td>
<td>1.122</td>
</tr>
<tr>
<td>Unif(95-115)</td>
<td>Unif(80-100)</td>
<td>17</td>
<td>1.010</td>
<td>1.120</td>
</tr>
<tr>
<td>Unif(105-125)</td>
<td>Unif(95-105)</td>
<td>17</td>
<td>1.012</td>
<td>1.121</td>
</tr>
<tr>
<td>Unif(110-130)</td>
<td>Unif(100-120)</td>
<td>17</td>
<td>1.009</td>
<td>1.124</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td></td>
<td>1.008</td>
<td>1.121</td>
</tr>
<tr>
<td>Std.</td>
<td></td>
<td></td>
<td>0.003</td>
<td>0.002</td>
</tr>
</tbody>
</table>

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. We show our results in Figure 2.
As seen in Figure 2 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 2 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

A model that meets all the demand 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 trucks can be significantly reduced by about 58% 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.

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.
References


