CONTAINER RELOCATION IN A LOGISTIC SERVICES CONTAINER TERMINAL

Juan Pablo Cavada, Department of Civil Engineering, Universidad de Chile – jucavada@ing.uchile.clm
Cristián E. Cortés, Department of Civil Engineering, Universidad de Chile – ccortes@ing.uchile.com
Pablo A. Rey, Department of Industry and Programa Institucional de Fomento a la Investigación, Desarrollo e Innovación, Universidad Tecnológica Metropolitana – prey@utem.cl

Palabras clave: Logistics , Simulation , Container Transport

ABSTRACT

This research is focused on the analysis of a container terminal common in industrial zones in urban areas, called Logistic Services Container Terminal (LSCT), particularly the allocation, relocation, and removal of containers during operations. We define an LSCT as a small to medium size inland terminal with no intermodal facilities whose main function is to provide services to his hinterland market. The material handling equipment used is limited to reach stacker cranes and front loaders, and the ground transportation in and out is performed using trucks exclusively. Moreover, many operations are related to servicing customers in special service areas within the yard. In this research, we explicitly recognize the limitations and features associated with the operations within LSCT yards, which means that the problems and solutions have to be conceived and addressed in a very different way, considering in addition that services must be coordinated with the strategies proposed for managing containers in and out. This paper's contribution is twofold, first, we characterize the different aspects present in the treatment of an LSCT, and propose adapted versions of three known greedy decision rules for determining the location for an arriving container and then extended them to consider the cost of moving relocating containers. We evaluate their performances using a discrete-event simulator. Results show that one rule outperforms the others and when using cost-aware rules while increasing the expected number of relocations reduces the total expected cost significantly.

1. INTRODUCTION

In the development of containerization worldwide, the treatment of the terminals' features where the containers are stored has been an issue of deep study over the last years in the specialized literature. Inland terminals' function can be classified by their relationship with their hinterland in a three-tier system (Rodrigue et al., 2010; Rodrigue, 2020). Tier I considers pure transport-related functions. Tier II corresponds to logistic functions, which are those functions that add value to the cargo (like consolidation, transloading, or storage). Finally, tier III functions are those focused on balancing the inbound and outbound flows, namely market and empty container management. Therefore, according to their primary function, a container terminal would go from an intermodal terminal in one extreme to pure market services in the other.

With regard to the main operations performed in a container terminal, in the literature we mostly find the description, analysis and study of standard intermodal transportation terminals located in ports. The two basic container yard problems found in the literature are the *Container Stacking Problem* (Borgman et al., 2010; Casey y Kozan, 2012; Yu y Qi, 2013; Jang et al., 2013; Lin y Chiang, 2017; Kim y Kim, 1999; Kim y Park, 2003; Kim et al., 2000; Güven y Eliiyi, 2014) and the *Container Relocation Problem* CRP (Caserta et al., 2012; Petering y Hussein, 2013; Expósito-Izquierdo et al., 2015; Zehendner y Feillet, 2014; Ting y Wu, 2017).

In this paper we study these problems in the context of a hinterland terminal of containers, that we have called a Logistic Services Terminal (LSCT), whose functions are focused on accommodating freight in containers, which are received from, serviced within and dispatched to ports, retailers, and in general, they center their operation on customers that require to handle, perform services and store their freight in a strategic location close to urban areas, and therefore close to final customers warehouses. As far as we are aware, LSCT as an integrated concept has not been studied in the specialized literature before. In this research, we focus the efforts in recognizing the limitations associated with the operations within LSCT yards, mainly related to the stacking and retrieval problems that are now subject to accessibility constraints (side blocking and reach), 3-D and inter-block movements problems, which in the context of LSCT have to be conceived and addressed in a completely different way from what has been developed in the literature so far. In addition, the services provided for the customers in an LSCT (for example deconsolidation/consolidation, transloading, cross-docking) have to be coordinated with the strategies proposed for managing the containers in and out.

This paper shows a first effort in characterizing the different aspects present in the treatment of an LSCT, proposing, an adapted version of well-known greedy rules to determine a container's new position, in the context of the LSCT Container Operations Problem. Moreover, our approach considers a consistent way to compute the costs associated with the movements and includes all the extra conditions and constraints arising in these particular yards, which make the entire integrated optimization problem very attractive and challenging. The new strategies are tested and compared using a discrete-event simulation platform (validated using real data) built ad-hoc.

This paper's contribution is twofold, first, we characterize the different aspects present in the treatment of an LSCT, and then, propose adapted versions of three known greedy decision rules for determining the location for an arriving container and then extended them to consider the cost of moving relocating containers. A discrete event simulator is used to evaluate the performarce of each studied rule.

2. LOGISTIC SERVICES TERMINALS LSCT: DEFINITION AND CHARACTERIZATION

A Logistic Services Container Terminal (hereinafter denoted as LSCT) is a type of inland terminal usually located within urban areas. These terminals are defined by three key elements: (1) *Location and accessibility*, (2) *Function and hinterland relationship* and (3) *Size and infrastructure* In what follows, we explain how each one of these features affects the container yard operations in the context and treatment of LSCTs.

2.1. Location and accessibility

The first element that differences an LSCT is that is an inland terminal that is located near industrial parks or distributions centers. Unlike inland ports and intermodal terminals, these installations are accessible by truck only. This fact has several consequences for the operation: first, as all containers arrive one at a time. Second, inbound and outbound flows of containers are not clearly defined. This means that the operations of allocating a new container and retrieving containers from the yard cannot be decoupled; this is known as the Dynamic Container Relocation Problem (Akyüz y Lee, 2014). A third consequence is that expedite gate operations become much more important, as all the movement in the terminal must pass through it.

2.2. Function and hinterland relationship

Using some definitions from Rodrigue et al. (2010), an inland terminal has transport and supply chain functions. An LSCT is a terminal where these last functions, which we denote as logistic services, are the main focus of the terminal operations. The most common services provided are *Temporal storage* of containers whose cargo is not yet needed by the owner becoming a buffer into the client's supply chain. *(De)consolidation* of cargo that occurs as part of a freight forwarding service. *Transloading* the cargo between containers into smaller load units to fit transport needs and vice versa or between a container owned by the shipping company to one owned by the customer and *Maintenance* services to the containers in order to have them ready for being utilized again. The focus on providing these services means that LSCTs are located very close to their customers in the supply chain, forcing the LSCT facilities to have the ability to quickly respond to their customers' requirements. An effect of this is that these terminals face a great degree of uncertainty in containers' arrivals and on what services will be required for them, unlike the operations of a regular port where the arrival time of the next ship along with the information of what to do with the containers assigned to it are both in advance.

2.3. Size and Infrastructure

The last key aspect that defines an LSCT is the size and the available equipment associated. An LSCT is smaller than a container yard associated with a port because it is generally located inside industrial parks close to the customers where the cost of the land is normally expensive, normally with a total capacity between 3,000 and 10,000 TEU. Therefore, an LSCT will favor the use of smaller and flexible material handling equipment (MHE) like reach stacker cranes (RSC), which are preferred over the more efficient but also more expensive gantry cranes (GC) and straddle carriers (SC) commonly observed in larger terminals.

3. CONTAINERS OPERATIONS IN A LSCT

On any container yard there are two main container operations: Stacking or allocation and retrieval a detailed description of these operations and relevant literature associated can be found in (Caserta et al., 2011; Carlo et al., 2014; Lehnfeld y Knust, 2014).

In most terminals, containers are usually stacked in rectangular blocks, built by adjacent rows of containers stacks separated in bays, the position in each stack is called a tier (see figure 1). The actual length, width, and height of each container block will be defined by the specific MHE used (Lee y Kim, 2010, 2013). While efficient in the use of space, stacking containers means that only the topmost one of each stack is accessible at any given time. Therefore, the most important objective in container operations is to minimize the number of re-handlings, that moving a container to a new position in order to access a target container.

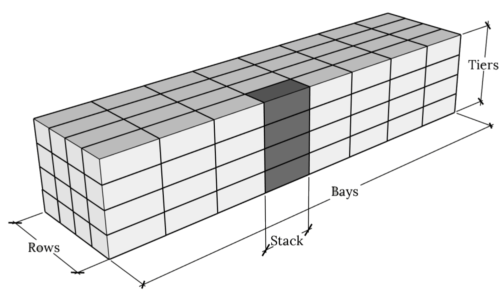

Figura 1: A representation of a container block.

The reachability to containers of RSC is more limited than that of GC, which can access all containers at the top of any stack. Most RSC can only stack up to five containers high and 2 containers deep. This means that the maximum size of a bay would be 5 by 4 containers, with cranes accessing the bay from both sides. Secondly, even when a container is on top of a stack, an RSC may not be able to reach that container if there were a higher stack in front of him, we call this “side blocking’.

When a container arrives at the yard, it should be located in a position such that the number of future rehandlings is minimized. This is generally achieved by choosing a location for the arrived container that should not block the access to other containers that have to leave the yard before the arrived one. Over the years, several measures have been proposed for estimating the number of rehandlings (Watanabe, 1991; Murty et al., 2005; Kim y Hong, 2006; Caserta y Voß, 2009; Wu y Ting, 2010; Caserta et al., 2011; Forster y Bortfeldt, 2012; Zhu et al., 2012).

The problem of retrieving a given sequence of containers from a bay is known as the Container Relocation Problem (CRP). The CRP was proven to be NP-hard in (Caserta y Voß, 2009). A number of methods have been proposed to find solutions: priority-based heuristics (Kim, 1997; Murty et al., 2005; Kim y Hong, 2006; Wu y Ting, 2010). Search metaheuristics: tree search (Kim y Hong, 2006; Forster y Bortfeldt, 2012; Unluyurt y Aydin, 2012), beam search (Expósito-Izquierdo et al., 2014; Ting y Wu, 2017), Iterative deepening (Zhu et al., 2012) and A$*$ search Expósito-Izquierdo et al. (2014). Exact methods developed are MIP (Caserta et al., 2012; Petering y Hussein, 2013; Expósito-Izquierdo et al., 2015), dynamic programming (Caserta et al., 2011) and branch-and-price (Zehendner y Feillet, 2014).

The limitations introduced by the reachability of an RSC change the feasible movements when stacking or retrieving a container, imposing additional adjacency-constraints to the possible movements: (1) When stacking a container, some stacks may not be accessible even if they are not full. (2) When retrieving a target container, in addition to relocating those containers stacked on top of the target, it may be necessary to relocate some containers in adjacent stacks that may be side blocking the target container. (3) When relocating a container to another position on the same bay to gain access to a target container, choosing some stacks may produce new side blocks to the target container making those relocations unfeasible. Figure 2 illustrates these three additional constraints.

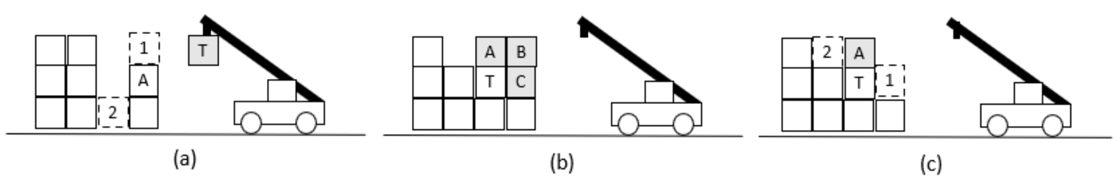

Figura 2: Examples of adjacency constraints: (a) The only possible position to put container T is position 1 as position 2 is side blocked by container A. (b) To retrieve container T, containers A, B and C must be relocated. (c) Container A can only be relocated to position 2, if moved to 1 in would side block the target container.

The cost of rehandling a container changes considerably with the distance. Moving a container to an adjacent bay can be performed fast. However, moving a container to a bay located on the other side of the block will require the use of a truck. Therefore, when evaluating the cost of relocation, one must consider that the movement of a container to a different block may be cheaper than a relocation of that container to a farther position in the same block. For simplicity, we classify the movement cost in three ranges: short-range, medium-range, and long-range. The most relevant consequence of allowing inter-block movements is that the whole yard must be considered in the process of planning the terminal operations, or at least several blocks at the same time.

4. THE LSCT CONTAINER OPERATIONS PROBLEM

We formalize the problem behind the dynamic day-to-day operation within an LSCT and provide the proper notation for the resulting algorithms. In these developments, we are making the following assumptions: (A1) All movements are performed by RSC, and therefore, they are all subject to side constraints. (A2) Arrival times for containers are unknown. (A3) Only once a container arrives at the terminal, its departure time and all the required services are known.

We define the LSCT Container Operations Problem as the combination of three basic operations: allocation, retrieval and relocation, and services.

Allocation of new containers When a new container arrives at the storage yard, either from the gate or from a service location, we have to find the best position in the yard to allocate that container. For doing this, we propose to search over all accessible stacks and use one of the rules explained in detail in next section **??**, to find its new position in the yard.

Retrieval and relocation The process for retrieval of a target container from the storage yard is may involve rehandling operations if it is not accessible. First, we look for all containers blocking the target container and choose one that is accessible to be relocated. Then, we build the list of all accessible stacks where it can be relocated (Avoiding all the stacks that may further block the target container or another container blocking it. Then, we choose a stack from this list using any of the decision rules in from the following section **??**). Finally, either select the next blocking container or proceed to retrieve the target container.

Services Note that the services in the context of the described LSCT are time-dependent processes. A service is defined by: (a) a set of participating containers, (b) the earliest and latest time when it can begin and (c) a time required to complete the service. If a service requires a single container ($|N_e| = 1$), then it is called a simple service; otherwise, it is denoted a complex service. Only when all involved containers are in the service area, the proper service operation may begin. After the service is concluded, each container is allocated again in the storage yard or leaves the terminal.

4.1. Basic decision rules

All the described operations require a clear criterion to choose the position for a container to be located or relocated during the operation. While there are many methods to find good –even optimal– movement sequences, these methods require near-perfect information and sometimes a long computational time. Therefore, in practice, greedy policies are often used instead. In this section, we review three well-known greedy rules to determine a container's new position as found in the literature, adapted to the context of the LSCT Container Operations Problem.

Reshuffle index (RI) and Reshuffle Index with look ahead (RIL) The reshuffle index was introduced by Murty et al. (2005) and extended by Wu y Ting (2010). The reshuffle index is a simple heuristic that chooses the stack where the least number of containers will be blocked by the new one. In our context, we additionally must considerer the side blockings. We calculate the reshuffle index r_{sj} of each stack s when moving container j by first counting those containers in the stack that have lower retrieval time. Then, we look at all containers that will become side-blocked and count those with a lower retrieval time. Finally, the heuristic will choose the stack with the lowest reshuffle index to move the container. If there are multiple candidates the lowest stack is chosen.

The RIL rule refined this tie-break criteria, if there are multiple candidates instead it chooses the stack where the earliest retrieval time is the highest among all blocked containers. The idea behind is that when the container needs to be relocated, more containers would have been moved and better positions may be found

Min-max heuristic (MM) This rule was first proposed by Caserta y Voß (2009). When evaluating a new position for a container you choose a stack where it does not produce any blocking. If there are multiple choices, the rule will choose the stack where the earliest retrieval time is the earliest. If a block is unavoidable, the rule chooses the stack where the earliest retrieval time is the latest instead.

4.2. Considerations for container positioning decision rules

Earlier in our experimental tests, we found that the implementation of the different rules for selecting new positions of containers, under highly congested scenarios, led to configurations with no accessible stacks left in the storage yard for allocating new containers, even though, some stacks were not at full capacity. This happened when the decision rule chooses a stack for a container that renders another stack inaccessible, in cases where the container allocated is not side blocking any container having an earlier retrieval time. In order to avoid these situations, we established a simple accessibility rule:

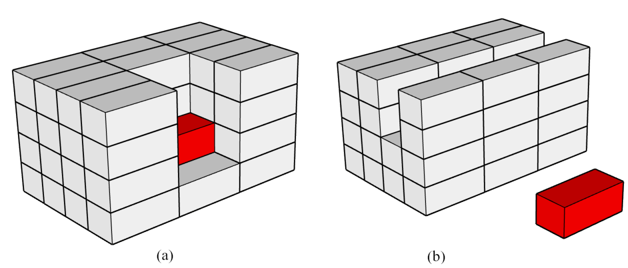

Figura 3: Examples of inaccessibility situations in container allocation-retrieval. In (a) the red containers can not be relocated to allow the container under it to be retrieved. In (b) the red containers can not be allocated even thought there is space available in the second row.

Accesibility rule. *A container cannot be placed in a stack if it would make another non-full stack inaccessible unless there is no other possible location.*

Another issue is avoiding additional blocks when relocating side blocking containers of a target container. Assume we need to retrieve a container, that is not immediately accessible because it is side blocked. When relocating this container, we will choose among the accessible stacks in the yard excluding those positions where would side block the target container or any other container blocking it. To take this into account, we define the following rule.

Non-blocking rule. *When relocating a container that blocks the target container, the new position should not block the target container or any other container blocking the target container.*

4.2.1. Incorporating hauling cost

The latter rules consider only the number of future relocations. In the context of an LSCT, where movements options inside a single container block are more limited and instead is often necessary to haul containers from one block to another it is important to incorporate the hauling cost into the decision process. We present two strategies to incorporate this hauling cost into the latter decision rules.

Combined weighted cost ("-C") The first strategy is to consider the weighted cost of the haul plus the expected cost of future relocations. The total cost of relocating a container c from stack s to s' is calculated as follows:

$$PC^c_{ss'} = \alpha D^c_{ss'} + (1 - \alpha)E^c_{s'}$$

where $D^c_{ss'}$ is the cost of hauling the container from the stack s to s' and $E^c_{s'}$ will be the expected number of future relocations if container c is relocated to stack s'. Parameter α will weight both terms according to the specifics of each terminal. Decision rules using this strategy will use the value of $PC^c_{ss'}$ instead of the expected number of reshuffles r_{sc}. Hereinafter, these rules are identified by the "-C" suffix in their notation.

Staggered tiers cost ("-S") In this strategy, we define N cost tiers, sorted by cost with $n = 1$ being the cheapest and $n = N$ the most expensive. Then for each stack s, we classify all other stacks in one of these tiers. Let T^n_s be the set of stacks in the n-th cost tier for stack s. When looking for a container's destination we first look only at those accessible stacks in the first (cheapest) tier, if there are no accessible stacks in the first tier, then we check those in the second tier. This process continues until a suitable stack is located. In our experiments, this strategy is identified with the "-S" suffix.

5. IMPLEMENTATION AND RESULTS

In the present section, we present our findings and interesting analysis when applying our proposed rules over a simulation platform of an LSCT. This section aims to build a simulation tool designed to evaluate the performance of the previous rules in the context of an LSCT. First, we present the structure of the simulation. Next, we succinctly describe the case study, to finish this section with the development of experiment and posterior analysis and insights obtained from the simulation results.

5.1. LSCT Simulation tool

A discrete event simulator was built to simulate the LSCT operation. This system simulates the three main areas of the terminal: Gate, Container yard and Service Area. The simulator includes all the main operations between these areas and their associated resources. The use of internal and external trucks is not explicitly included. The simulator was build using the Python language and the Simpy library[1]. A general description of the simulator is as follows:

Entities: The simulator sole entity is the container. Each container is characterized by a retrieval time, a date, and a service time. Containers arrive at the terminal using a non-homogeneous Poisson process that varies with the time of day.

Resources: There are three resources modeled in the system: *Cranes, Service Slots* and *Corridors*. The most important resource is the cranes that are required to perform all the movements in the yard. Service slots represent the number of containers that can be simultaneously allocated in the service area. A Corridor is a lane between two blocks; then, to simulate a safety measure used by terminal managers is the following: "it is forbidden to have two cranes operating in the same corridor at the same time." Wherever a crane resource is seized, the simulator must also seize all corridors involved in the operation. All resources are assigned using a first-come-first-serve rule.

Main Processes: Three main processes are modeled: *Arrivals, Retrievals, Services* and correspond to the basic operations defined in section **??**.

5.2. Description of case study

We built a test yard using realistic data obtained from an LSCT located near Santiago, Chile. The data set contained two years of container movements in the terminal (arrivals, relocations, services, and retrievals), with a total of over 200,000 containers tagged in the whole period. We used this data to adjust probability distributions to container arrival and leaving times, as well as probability and time of service. Our study scenario consists of a container yard made by 4 blocks of $4 \times 4 \times 12$. In this scenario, all movements are performed by a single crane and the service area has the capacity for at most four containers simultaneously.

[1] `https://simpy.readthedocs.io`

Containers arrive at the yard following a non-homogenous Poisson distribution, where the arrival rate varies through the day in three periods: morning, midday, and afternoon, being the highest, early in the morning. For the base scenario, the average time between arrivals are 3.6/7.0/8.9 minutes for the three periods, respectively. Finally, to find the exact time of retrieval for a container, we adjusted an empirical distribution with two peaks around midday and the end of the day.

In the case of services, we have included only single-container services in our analysis. From historical data, we have estimated that each container has a 0.2 probability of having a service. The specific service time is distributed as follows: 0.4 chance of being in the first week since arrival, 0.3 chance of being on the same day that its pick-up and 0.3 is occurring at any other time. Within these time windows, the service's specific day is chosen uniformly, all services starting at the beginning of the day.

Each simulation day starts at 7 a.m. Arrivals and retrievals can occur at any time until 5 p.m. After this moment, no new orders are received but any pending operation must still be completed on the same day considering overtime.

As a modeling horizon, we choose to simulate 180 days, starting with an empty container yard. The warm-up time before reporting solutions is 90 days. This allows the yard to reach a stable occupancy level. It is expected that the performance of different sorting strategies will be affected by different congestion levels. We present results for tree congestion scenarios, low congestion (0.6 average occupancy), normal congestion (0.77 avg. occupancy) and high congestion (0.85 avg. occupancy).

Regarding cost parameters, we normalized the relocation cost to the values 1/3/5 for close-, medium- and long-range relocations. The transportation cost for new containers coming to the yard was set to zero to avoid introducing any particular block or position bias. The same was conducted in the case of containers moving from the service and into the service area. For the tiered cost strategies, we used the same 3-tier cost structure. Using experimental data, we found that the number of expected future relocations for a container after the first one was very close to zero, making the transportation cost the main driver of the combined cost strategies.

5.3. Results

We present a comparison in performance of the following combination of rules and cost strategies presented previously: RI, RIL, MM, RI-C, RIL-C, RI-S, RIL-S and MM-S. Note that we did not include an MM-C variant because, considering how the MM rule chooses a new position together with the normalization of the cost mentioned earlier, the use of either tiered cost or combined cost will lead to the same solutions. Each scenario simulation was run with 100 replications.

In figure 4 we present a comparison of the number of relocations for each rule and congestion level, showing the mean value as well as the confidence intervals for a 95 % confidence level. The most important result is that the Min-Max rules greatly outperform the other rules regarding the number of relocations at all congestion levels. Under a low congestion scenario, the MM and MM-S rules eliminate the need for relocations. The RIL-C and RIL-S have very similar performances, being the RI rules the ones that perform worse in all scenarios. As we increase the congestion level, the difference between different rules decreases, mainly in the MM and RIL families' cases. Overall, the MM rule has been well documented in the literature as being the best in performance; it is interesting to observe that even with the additional constraints related to the LSCT, it still performs extremely well.

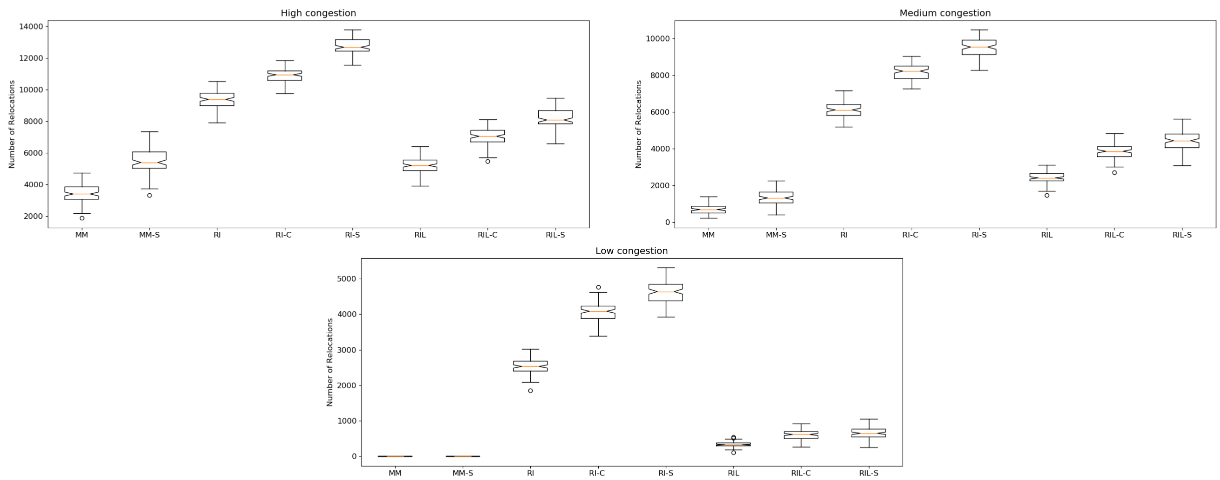

Figura 4: Number of relocations for each proposed rule for three congestion levels.

The inclusion of cost into the decision process will naturally increase the number of relocations, this holds true for all cases with the more restricted tiered strategy leading to the highest number of relocations. Full results are presented in table 1.

In figure 5 below, we show the total relocation cost on average, for each rule (top graph in the figure), as well as the total number of relocations split by distance (bottom graph in the figure), presenting only the results for the medium congestion scenario. We observe that one more time, the MM rules are the ones performing the best because the number of relocations for such rules is the minimum, although the incorporation of a cost strategy reduces the total cost in half. The reason is that the pure MM rule will choose the best position for a new container without considering the cost effect, leading to more long-range relocations, while the MM-S, which chooses close-range relocations if possible, leads to more but much cheaper relocations.

We observe a similar pattern in the results with the other two rule families, in which the introduction of a simple cost strategy greatly reduces the total cost of the operation. A final consideration regarding cost, in the context of our case study, is that the results achieved by the combined and tiered cost strategies are very similar, with the tiered strategy being more restricted but easier to be implemented; it is, in fact, a very promising rule to use in a real-world application.

Congestion Scenario (ρ)	Rule	Average relocations	STD	Conf. Interval (5 %)		Mean / Arrivals	Mean / Retrievals	Avg. Close reloc.	Avg Medium reloc.	Avg. Long reloc.
Medium (0.77)	MM	699.1	262	[646.8 ,	751.3]	0.12	0.12	14.1	44.9	640.1
	MM-S	1328.8	408.8	[1247.2 ,	1410.3]	0.24	0.24	1047.1	263.2	18.4
	RI	6118.8	407.6	[6037.5 ,	6200.1]	1.08	1.08	122.9	401.3	5594.6
	RI-C	8186.2	406	[8105.2 ,	8267.2]	1.45	1.45	4611.6	3226.3	348.4
	RI-S	9521.6	479.7	[9425.9 ,	9617.2]	1.69	1.69	7025.6	2361.8	134.2
	RIL	2426.1	327.3	[2360.8 ,	2491.4]	0.43	0.43	42.3	136.8	2247
	RIL-C	3853.1	433.6	[3766.6 ,	3939.5]	0.68	0.68	2294.2	1411.5	147.3
	RIL-S	4425.4	531	[4319.5 ,	4531.3]	0.78	0.78	3357.4	984.6	83.4
High (0.85)	MM	3416.9	637.9	[3289.6 ,	3544.1]	0.55	0.55	65.9	222.2	3128.8
	MM-S	5437	904.5	[5256.6 ,	5617.3]	0.87	0.87	3043.9	2038.8	354.2
	RI	9360.9	546.8	[9251.8 ,	9469.9]	1.50	1.50	187.2	610.6	8563.1
	RI-C	10872.8	470	[10779 ,	10966.5]	1.74	1.74	4623	4882.4	1367.4
	RI-S	12757.1	513.8	[12654.7 ,	12859.6]	2.04	2.04	6689.8	5264.3	803
	RIL	5183.3	559.3	[5071.7 ,	5294.8]	0.83	0.83	96.2	314.6	4772.4
	RIL-C	7022.1	566.6	[6909.1 ,	7135.1]	1.12	1.12	3093.5	3118.2	810.4
	RIL-S	8153.1	683.5	[8016.8 ,	8289.4]	1.30	1.31	4470.1	3117.6	565.4
Low (0.61)	MM	0	0	[0 ,	0]	0.00	0.00	0	0	0
	MM-S	0	0	[0 ,	0]	0.00	0.00	0	0	0
	RI	2545.2	211.7	[2502.9 ,	2587.4]	0.56	0.56	51.6	165.7	2327.9
	RI-C	4064.6	272.4	[4010.3 ,	4118.9]	0.90	0.90	3003.3	1034.9	26.4
	RI-S	4621.7	307.3	[4560.4 ,	4683]	1.02	1.02	4242.9	372.4	6.4
	RIL	339	73.7	[324.3 ,	353.7]	0.07	0.07	5	16.8	317.2
	RIL-C	606.8	136.1	[579.7 ,	634]	0.13	0.13	488.9	114.5	3.5
	RIL-S	664.7	160.5	[632.7 ,	696.7]	0.15	0.15	616	47.2	1.5

Tabla 1: Detailed results of simulations for the different rules and cost strategies

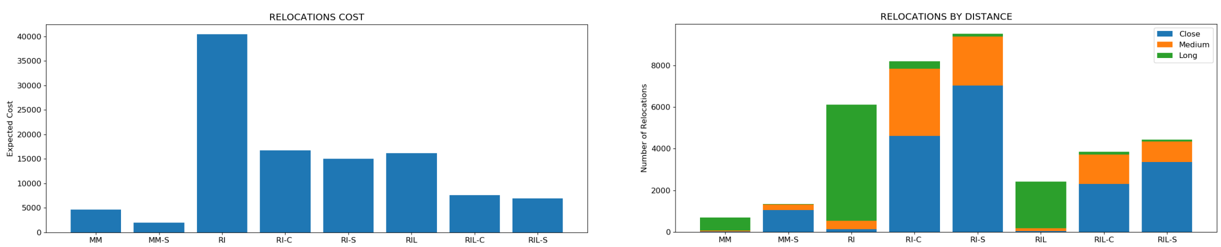

Figura 5: Total relocation cost and number of relocations by distance for each rule and strategy (medium congestion scenario).

6. COMMENTS AND CONCLUSIONS

In this work, we study a class of the dynamic container allocation-and-relocation problem. This problem is similar to the well-studied container allocation problem and involves both arrivals and departures instead of only arrivals of containers to the terminal.

We have applied the logic of three known greedy decision rules for allocating and relocating containers to the logic of an LSCT and evaluated their performances using a discrete-event simulator. We have found that the min-max rule greatly outperforms the other tested rules at each congestion level. Furthermore, we have extended these basic rules with two simple strategies to consider the cost of moving relocating containers: a combined-cost one and a tiered-cost one. With the help of simulation, we have found that using any of these cost-aware rules while increasing the expected number of relocations reduces the total expected cost significantly.

Acknowledgements

The authors want to acknowledge the support of project ANID/FONDECYT/REGULAR 1191200 and the Complex Engineering Systems Institute ANID PIA/APOYO AFB180003. Juan Pablo Cavada would like to thank the support of ANID BECA DOCTORADO NACIONAL 21171505.

REFERENCIAS

Akyüz, M. H., y Lee, C. (2014). A mathematical formulation and efficient heuristics for the dynamic container relocation problem. **Naval Research Logistics (NRL)**, 61 (2), 101–118. doi: 10.1002/nav.21569

Borgman, B., Asperen, E. v., y Dekker, R. (2010). Online rules for container stacking. **OR Spectrum**, 32 (3), 687–716. doi: 10.1007/s00291-010-0205-4

Carlo, H. J., Vis, I. F., y Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. **European Journal of Operational Research**, 235 (2), 412–430. doi: 10.1016/j.ejor.2013.10.054

Caserta, M., Schwarze, S., y Voß, S. (2011). Handbook of Terminal Planning. **Operations Research/Computer Science Interfaces Series**, 247–269. doi: 10.1007/978-1-4419-8408-1_13

Caserta, M., Schwarze, S., y Voß, S. (2012). A mathematical formulation and complexity considerations for the blocks relocation problem. **European Journal of Operational Research**, 219 (1), 96–104. Descargado de `http://dx.doi.org/10.1016/j.ejor.2011.12.039` doi: 10.1016/j.ejor.2011.12.039

Caserta, M., y Voß, S. (2009). A Corridor Method-Based Algorithm for the Pre-marshalling Problem. En **Applications of evolutionary computing** (pp. 788—797). Berlin, Heidelberg: Springer Berlin Heidelberg. Descargado de `http://link.springer.com/10.1007/978-3-642-01129-089` doi: 10.1007/978-3-642-01129-0

Casey, B., y Kozan, E. (2012). Optimising container storage processes at multimodal terminals. **Journal of the Operational Research Society**, 63 (8), 1126–1142. doi: 10.1057/jors.2011.113

Expósito-Izquierdo, C., Melián-Batista, B., y Moreno-Vega, J. M. (2014). A domain-specific knowledge-based heuristic for the Blocks Relocation Problem. **Advanced Engineering Informatics**, 28 (4), 327–343. Descargado de `http://dx.doi.org/10.1016/j.cie.2014.06.010` doi: 10.1016/j.aei.2014.03.003

Expósito-Izquierdo, C., Melián-Batista, B., y Moreno-Vega, J. M. (2015). An exact approach for the Blocks Relocation Problem. **Expert Systems with Applications**, 42 (17-18), 6408–6422. doi: 10.1016/j.eswa.2015.04.021

Forster, F., y Bortfeldt, A. (2012). A tree search procedure for the container relocation problem. **Computers & Operations Research**, 39 (2), 299–309. Descargado de http://dx.doi.org/10.1016/j.cor.2011.04.004 doi: 10.1016/j.cor.2011.04.004

Güven, C., y Eliiyi, D. T. (2014). Trip Allocation and Stacking Policies at a Container Terminal. En **Transportation research procedia** (Vol. 3, pp. 565–573). Descargado de http://linkinghub.elsevier.com/retrieve/pii/S2352146514001987 doi: 10.1016/j.trpro.2014.10.035

Jang, D.-W., Kim, S. W., y Kim, K. H. (2013). The optimization of mixed block stacking requiring relocations. **International Journal of Production Economics**, 143 (2), 256–262. Descargado de http://dx.doi.org/10.1016/j.ijpe.2012.06.033 doi: 10.1016/j.ijpe.2012.06.033

Kim, K. H. (1997). Evaluation of the number of rehandles in container yards. **Computers Industrial Engineering**, 32 (4), 701—711. doi: 10.1016/s0360-8352(97)00024-7

Kim, K. H., y Hong, G.-P. (2006). A heuristic rule for relocating blocks. **Computers & Operations Research**, 33 (4), 940–954. doi: 10.1016/j.cor.2004.08.005

Kim, K. H., y Kim, H. B. (1999). Segregating space allocation models for container inventories in port container terminals. **International Journal of Production Economics**, 59 (1-3), 415–423. doi: 10.1016/s0925-5273(98)00028-0

Kim, K. H., y Park, K. T. (2003). A note on a dynamic space-allocation method for outbound containers. **European Journal of Operational Research**, 148 (1), 92–101. doi: 10.1016/s0377-2217(02)00333-8

Kim, K. H., Park, Y. M., y Ryu, K.-R. (2000). Deriving decision rules to locate export containers in container yards. **European Journal of Operational Research**, 124 (1), 89–101. doi: 10.1016/s0377-2217(99)00116-2

Lee, B. K., y Kim, K. H. (2010). Optimizing the block size in container yards. **Transportation Research Part E: Logistics and Transportation Review**, 46 (1), 120–135. Descargado de http://dx.doi.org/10.1016/j.tre.2009.07.001 doi: 10.1016/j.tre.2009.07.001

Lee, B. K., y Kim, K. H. (2013). Optimizing the yard layout in container terminals. **OR Spectrum**, 35 (2), 363–398. doi: 10.1007/s00291-012-0298-z

Lehnfeld, J., y Knust, S. (2014). Loading, unloading and premarshalling of stacks in storage areas: Survey and classification. **European Journal of Operational Research**, 239 (2), 297–312. Descargado de http://dx.doi.org/10.1016/j.ejor.2014.03.011 doi: 10.1016/j.ejor.2014.03.011

Lin, D.-Y., y Chiang, C.-W. (2017). The Storage Space Allocation Problem at a Container Terminal. **Maritime Policy & Management**, 44 (6), 1–20. Descargado de https://www.tandfonline.com/doi/full/10.1080/03088839.2017.1335897 doi: 10.1080/03088839.2017.1335897

Murty, K. G., Liu, J., Wan, Y.-w., y Linn, R. (2005). A decision support system for operations in a container terminal. **Decision Support Systems**, 39 (3), 309–332. doi: 10.1016/j.dss.2003.11.002

Petering, M. E., y Hussein, M. I. (2013). A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem. **European Journal of Operational Research**, 231 (1), 120–130. Descargado de http://dx.doi.org/10.1016/j.ejor.2013.05.037 doi: 10.1016/j.ejor.2013.05.037

Rodrigue, J.-P. (2020). The Geography of Transport Systems. , 208–247. doi: 10.4324/9780429346323-6

Rodrigue, J.-P., Debrie, J., Fremont, A., y Gouvernal, E. (2010). Functions and actors of inland ports: European and North American dynamics. **Journal of Transport Geography**, 18 (4), 519–529. Descargado de http://dx.doi.org/10.1016/j.jtrangeo.2010.03.008 doi: 10.1016/j.jtrangeo.2010.03.008

Ting, C.-J., y Wu, K.-C. (2017). Optimizing container relocation operations at container yards with beam search. **Transportation Research Part E: Logistics and Transportation Review**, 103, 17–31. doi: 10.1016/j.tre.2017.04.010

Unluyurt, T., y Aydin, C. (2012). Improved rehandling strategies for the container retrieval process. **Journal of Advanced Transportation**, 46 (4), 378—393.

Watanabe, I. (1991). Characteristics and analysis method of efficiencies of container terminal: an approach to the optimal loading/unloading method. **Container Age**, 3, 36–47.

Wu, K.-C., y Ting, C.-J. (2010). A beam search algorithm for minimizing reshuffle operations at container yards. **International Conference on Logistics and Maritime Systems** (2000), 703—710.

Yu, M., y Qi, X. (2013). Storage space allocation models for inbound containers in an automatic container terminal. **European Journal of Operational Research**, 226 (1), 32–45. Descargado de http://dx.doi.org/10.1016/j.ejor.2012.10.045 doi: 10.1016/j.ejor.2012.10.045

Zehendner, E., y Feillet, D. (2014). A branch and price approach for the container relocation problem. **International Journal of Production Research**, 52 (24), 7159–7176. Descargado de http://dx.doi.org/10.1080/00207543.2014.965358 doi: 10.1080/00207543.2014.965358

Zhu, W., Qin, H., Lim, A., y Zhang, H. (2012). Iterative Deepening A* Algorithms for the Container Relocation Problem. **IEEE Transactions on Automation Science and Engineering**, 9 (4), 710–722. doi: 10.1109/tase.2012.2198642