Static and Dynamic Online Stochastic Container Relocation Problem under random permutation

Virgile Galle [*] Cynthia Barnhart [†] Patrick Jaillet [‡]

Abstract

The Container Relocation Problem (CRP) is a very well studied NP hard problem that has its major application in port yard operations. In this paper, we introduce a new version of the CRP called the Online Stochastic Container Relocation Problem (OSCRP) and its dynamic version (DOSCRP). In the OSCRP we assume that the retrieval order is drawn uniformely at random among all permutations while no new container is arriving. In the DOSCRP the setting is identical but new containers can arrive to be stacked. First we provide theoretical lower bounds on the expected optimal number of relocations. Secondly we run computational experiments using these bounds to support our intuition that the "Leveling" policy, which aims to makes the configurations as even as possible, is optimal in both the static and dynamic cases.

1 Introduction and Literature Review

Given a configuration with C columns and maximum height H (a common assumption is to take $H \leq C$) and N containers intitally present, each of them associated with a number from 1 to N, the classical CRP is concerned with finding the sequence of container moves that minimizes the number of moves while retrieving the containers in the given order. In this paper we consider a variant of this problem where no order is known a priori. We suppose that given a configuration with C columns and maximum height H (see Figure 2) each container is equally likely to be retrieved at the next step. In addition for the DOSCRP we suppose that a new container can arrive with a given probability dependent on the current state. If so the container has to be stacked; otherwise each container is equally likely to be retreived.

In both these cases, it seems intuitive that the optimal policy should be the one that always places a container on the column with the fewest containers in it, referred to as *'Leveling'* policy (denoted L).

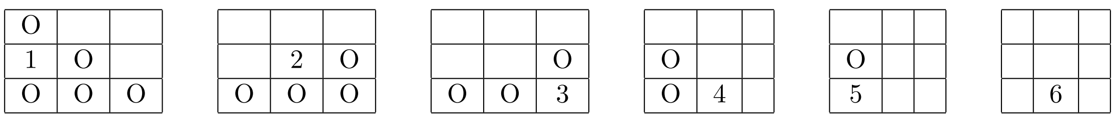

Figure 1: An example of realisation of the OSCRP using the L policy. The process starts with the configuration on the left and container 1 is revealed. We move the blocking container using L on the most empty column (third one). The process is repeated until the configuration is empty.

Literature Review Many papers have studied the classical CRP and proposed various solutions to the problem (including integer programming formulations ([6], [14], [1]) or heuristics such as tabu search, A* and index-based heursitics ([10], [5], [12], [3], [17], [19], [15])). The CRP with incomplete information, where only a fraction of the departure order is known, has recieved less attention ([18], [11]). In this setting, the value of information and the impact of an appointmenent system is studied ([7], [2]). Most papers investigating the management of containers with incomplete information address the stacking problem on its own ([13], [8]) Finally the dynamic CRP even with full information has also been studied in a few papers ([1], [4]).

*ORC, MIT. Email: vgalle@mit.edu.
†CEE, MIT. Email: cbarnhar@mit.edu.
‡EECS and ORC, MIT. Email: jaillet@mit.edu.

To the best of the authors' knowledge, the OSCRP has not been studied by many researchers. The major paper is [16] which introduces the L policy and proves its feasibility. It also shows that the tight competitive ratio of the L policy is $2\lceil \frac{N}{C} \rceil - 1$ and provides some experimental results on expectations. Before that, [9] gave an exact formula that estimates the expected number of relocations for the L policy that is intractable and hence proposed a regression model and an approximation formula as another alternative.

Despite the fact that the solution seems extremely intuitive, it appears that there is no 'easy' proof of the optimality of the L policy. We believe that this paper provides the first theoretical analysis on expectation of the online problem that could lead to future research in this area. Furthermore the DOSCRP has not been defined yet. It is a novel problem that the authors believe is worth studying both from a theoretical and practical point of view. The DOSCRP represents the case of a facility with no appointment system which gets information about retrievals and arrivals only as customers are coming to pick up their products. Proving that L policy is optimal would ensure to operators that there is nothing better to do than this intuitive myopic policy.

The rest of the paper is structured as follows: Parts 2 and 3 introduce lower bounds respectively for the static and dynamic OSCRP and show experimental results. Finally, Part 4 concludes and suggests future research.

2 Static Online Stochastic Container Relocation Problem

For the OSCRP, the decision maker is required to choose θ a certain policy that is restricted (i.e., a policy that only moves containers blocking the container to be retrieved) and Markovian i.e., only depends on the current state and does not take into account previous decisions (for example chooses the leftmost column that is not full). We consider Θ_s the set of restricted Markovian policies. Given an initial configuration with N containers, a container is picked uniformly at random to be the next container retrieved. Relocations are performed based on θ and the selected container is retrieved (which leaves a bay with a different configuration and $N-1$ containers in it). Then another container is picked at random and relocations are performed (according to θ). This process is repeated N times which leads necessarily to an empty configuration. The goal is to find the optimal policy θ^* which minimizes the expected number of relocations for all possible initial configurations.

In this problem, conlumns are interchangeable which makes some configurations such as in Figure 2 equivalent to the OSCRP. We describe any class of equivalent configurations by a vector of size $H+1$ defined as $s = (s_0, \ldots, s_H)$ such that s_i is the number of columns with i containers in it. Finally we denote $s > 0$ if class s represents non empty configurations (i.e. $s_0 < C$). In this paper, we will refer interchangeably to a state, a configuration or class of equivalent configurations. Finally notice that we have $s_i \geq 0$, $\sum_{i=0}^{H} s_i = C$ and $\sum_{i=1}^{H} i s_i = n$ if there are n containers in the configuration.

O		
O	O	
O	O	O

		O
O		O
O	O	O

Figure 2: two equivalent bays with 3 tiers, 3 columns and 6 containers. Their state is $s = (0, 1, 1, 1)$. Here containers are noted by O since we do not have any information on their retrieval order.

First, let us denote $X(s, c)$ the number of relocations when we know the configuration s and the next container to be retrieved c (note that by definition $X(s, c)$ does not depend on the policy but only on s and c as it is the number of containers blocking c in s). We denote $R(s)$ such that $R(s) = \mathbb{E}_{c \in \mathcal{U}}[X(s, c)]$. $R(s)$ is the expected number of number of relocations for a configuration s when the container to be retrieved is uniformly taken at random. Then let $\xi_\theta(s, s^0)$ be the probability that given an initial configuration s^0 and using policy θ we reach configuration s. We denote $\mathcal{S}_n$ the set of configurations with n containers. Finally, we define $f(\theta, s^0)$, the total expected number of relocations given the initial configuration s^0 and a policy θ. So we have $f(\theta, s^0) = \sum_{n=1}^{N} \sum_{s \in \mathcal{S}_n} R(s) \xi_\theta(s, s^0)$. Solving OSCRP is finding θ^* for all initial configuration s^0 such that

$$\theta^* = \underset{\theta \in \Theta_s}{Argmin} \left(f(\theta, s^0) \right) \ , \forall s^0 \in \mathcal{S}_N \tag{1}$$

Fact 2.1. *Let $s > 0$ with $n = \sum_{i=1}^{H} is_i$ and R be the random variable defined in the previous paragraph, we have*

$$R(s) = \frac{\sum_{i=1}^{H} i(i-1)s_i}{2n} \tag{2}$$

Moreover if $s_0 = C$ then $R(s) = 0$.

The comes from the following observation: If $s > 0$, we know that the configuration has $n > 0$ containers. The probability that the container to be retrieved belongs to a column with i containers in it is $\frac{s_i}{n}$. Moreover the expected number of blocking containers in a column with i containers is $\sum_{j=1}^{i-1} j = \frac{i(i-1)}{2}$ therefore $R(s) = \sum_{i=1}^{H} \frac{i(i-1)}{2} \frac{s_i}{n} = \frac{\sum_{i=1}^{H} i(i-1)s_i}{2n}$. If $s_0 = C$ it is clear that $R(s) = 0$.

Computing ξ is very complex as it involves conditioning on every scenario of retrievals. Notice that there are a priori $N!$ possible scenarios. Therefore we introduce three lower bounds on the optimal cost to insure some performance of the L policy.

2.1 First lower bound

Before introducing the lower bound and for the sake of completness of the paper we state the following proposition for the function R.

Proposition 2.1. *[16] Let $R(s)$ be the function defined in Equation 2 and $n = \sum_{i=1}^{H} is_i > 0$. Then $R(s)$ is minimized for $s^{bal}(n)$ (balanced bay with n containers) defined as $s^{bal}_{k_0(n)+1}(n) = n - Ck_0(n)$ and $s^{bal}_{k_0(n)}(n) = C(k_0(n) + 1) - n$ where $k_0(n) = \lfloor \frac{n}{C} \rfloor$.*

Using this property we have

$$\begin{aligned} f(\theta^*, s^0) &= \sum_{n=1}^{N} \sum_{s \in \mathcal{S}_n} R(s)\xi_\theta(s, s^0) \geq R(s^0) + \sum_{n=1}^{N-1} \min_{s \in \mathcal{S}_n} (R(s)) = R(s^0) + \sum_{n=1}^{N-1} R(s^{bal}(n)) \\ &\geq R(s^0) + \sum_{n=1}^{N-1} k_0(n) \left(1 - \frac{C(k_0(n)+1)}{2n}\right) = \eta_1(s^0) \end{aligned} \tag{3}$$

Proposition 2.2. *Let θ^* be the optimal solution of equation 1 and η_1 defined in equation 3, then for all initial configuration s^0 we have*

$$f(\theta^*, s^0) \geq \eta_1(s^0)$$

2.2 Second and third lower bounds

The idea is to refine our previous lower bound. For η_1 we considered that the 'balanced' configuration was realized with probability $\xi_\theta(s^{bal}(n), s^0) = 1$ for each stage n (i.e., when there is n containers left in the bay). But due to the definition of the problem, there is some positive probability for any policy that the configuration gets more unbalanced i.e., having at least one empty column. Conditioned on this event for $n \geq C$ (if $n < C$ the balanced state necessarily has an empty column), the configuration that minimizes $R(s)$ is $s^{bal2}(n)$ defined as follows: let $l_0(n) = \lfloor \frac{n}{C-1} \rfloor \geq 1$ and then $s^{bal2}_{l_0(n)+1}(n) = n - (C-1)l_0(n)$ and $s^{bal2}_{l_0(n)}(n) = (C-1)(l_0(n)+1) - n$ and $s_0 = 1$. We have $R(s^{bal2}(n)) = l_0(n) \left(1 - \frac{(C-1)(l_0(n)+1)}{2n}\right)$. This comes from Property 2.1 applied to a configuration with $C - 1$ columns.

Let denote $u_n(s^0)$ a lower bound on the probability of having at least one empty column in the n^{th} stage. So $1 - u_n(s^0)$ is an upper bound on the probability of not having a empty column (so potentially being in the balanced state). Therefore we derive a new lower bound:

$$\begin{aligned} f(\theta^*, s^0) &\geq \sum_{n=1}^{C-1} R(s^{bal}(n)) + \sum_{n=C}^{N-1} R(s^{bal}(n))(1 - u_n(s^0)) + \sum_{n=C}^{N-1} R(s^{bal2}(n))u_n(s^0) + R(s^0) \\ &= \eta_1(s^0) + \sum_{n=C}^{N-1} (R(s^{bal2}(n)) - R(s^{bal}(n)))u_n(s^0) \end{aligned}$$

The improvement with $\eta_1(s^0)$ is directly linked with $u_n(s^0)$. The larger $u_m(s^0)$ are, the tighter the new lower bound is. Here we give a recurrence relation for $u_n(s^0)$. First notice that $u_N(s^0) = 1$ if $s_0 > 0$ and $u_N(s^0) = 0$ otherwise. Let us define the event $A_n = \{s_0 > 0 | s \in \mathcal{S}_n \text{ and } \exists \theta, \xi_\theta(s, s^0) > 0\}$. A_n represents the event that a configuration reachable from s^0 with some policy θ has at least one empty column. Now we have

$$\mathbb{P}(A_n) = \mathbb{P}(A_n|A_{n+1})\mathbb{P}(A_{n+1}) + \mathbb{P}(A_n|A_{n+1}^c)\mathbb{P}(A_{n+1}^c) = \frac{C}{n+1} + \left(\mathbb{P}(A_n|A_{n+1}) - \frac{C}{n+1}\right)\mathbb{P}(A_{n+1})$$

We want to get a lower bound on $\mathbb{P}(A_n|A_{n+1})$. Let us define $a_n = \min\left(\frac{C-1+\lceil\frac{n+2-C}{H-1}\rceil}{n+1}, 1\right)$. Note that we have $\mathbb{P}(A_n|A_{n+1}) \geq \min_{1 \leq i \leq C-1}\left(\min\left(\frac{C-i+\lceil\frac{n+1-(C-i)}{H-1}\rceil}{n+1}, 1\right)\right) = a_n$ and $a_n \geq \frac{C}{n+1}$ for $n \geq C$. Therefore we have $\mathbb{P}(A_n) \geq \frac{C}{n+1} + \left(a_n - \frac{C}{n+1}\right)\mathbb{P}(A_{n+1}) \geq \frac{C}{n+1}$. We can define a second lower bound:

$$\eta_2(s^0) = \eta_1(s^0) + \sum_{n=C}^{N-1}(R(s^{bal2}(n)) - R(s^{bal}(n)))\frac{C}{n+1} \tag{4}$$

Finally if $u_n(s^0)$ is defined by recurrence such that $u_N(s^0) = 1$ if $s_0 > 0$ and $u_N(s^0) = 0$ otherwise and $u_n(s^0) = \frac{C}{n+1} + \left(a_n - \frac{C}{n+1}\right)u_{n+1}(s^0)$ for $1 \leq n \leq N-1$, we can define a third lower bound:

$$\eta_3(s^0) = \eta_1(s^0) + \sum_{n=C}^{N-1}(R(s^{bal2}(n)) - R(s^{bal}(n)))u_n(s^0) \tag{5}$$

Note that since $R(s^{bal2}(n)) - R(s^{bal}(n)) \geq 0$ and $u_n(s^0) \geq \frac{C}{n+1} \geq 0$, we have $\eta_3(s^0) \geq \eta_2(s^0) \geq \eta_1(s^0)$.

Proposition 2.3. *Let θ^* be the optimal solution of equation 1 and $\eta_2(s^0)$ and $\eta_3(s^0)$ defined in equations 4 and 5, then for all initial configuration s^0 we have*

$$f(\theta^*, s^0) \geq \eta_3(s^0) \geq \eta_2(s^0)$$

2.3 Experimental Results

The first experiment uses the three lower bounds to show the effecitvness of the L policy. We consider several values for H and C in the range for typical problem size ($4 \leq H \leq 6$ and $6 \leq C \leq 10$). We consider to have $N = (H-1)C$ containers in the initial configuration. Table 1 summarizes the results.

C / H	6	7	8	9	10
4	23.4%	**24.5%**	25.3%	25.9%	26.3%
5	18.6%	18.9%	18.9%	19.1%	19.0%
6	16.0%	15.7%	15.6%	15.4%	15.2%

Table 1: Guaranteed performance of the L policy for OSCRP: for each problem size, we give the best relative guaranteed gap to optimality $\left(\frac{f(\theta_L, s^{bal}(N)) - \eta_3(s^{bal}(N))}{\eta_3(s^{bal}(N))}\right)$. Note that in order to compute $f(\theta_L, s^{bal}(N))$ we consider 100,000 realisation of a random permutation of N elements. A common size of the problem is believed to be $H = 4, C = 7$; in this case, the guaranteed gap with optimality for L policy is 23.1% for regular sized instances which means that our best lower bound could be improved to give better evidence of the optimality of L. In addition, η_3 gets closer to L as H increases but there is no clear relation between η_3 and C.

The second experiment tries to identify the part for which the lower bound could be improved in order to give a better guaranteed gap for L. Given $H = 4$, $C = 7$, $N = (H-1)C = 21$, let us look at the evolution of the lower bound depending on the stage n.

Stage	Gap	Stage	Gap	Stage	Gap	Stage	Gap
21	0 (1.00)	15	0.11 (0.71)	9	0.13 (0.31)	3	**0.08 (0)**
20	0.03 (1.03)	14	0.14 (0.62)	8	0.15 (0.22)	2	0.04 (0)
19	0.06 (0.98)	13	0.14 (0.55)	7	**0.18 (0.13)**	1	0 (0)
18	0.08 (0.91)	12	0.16 (0.47)	6	**0.24 (0.22)**		
17	0.09 (0.85)	11	0.14 (0.42)	5	**0.18 (0.13)**		
16	0.10 (0.78)	10	0.13 (0.37)	4	**0.13 (0)**		

Table 2: Exepted difference between L and η_3 at all stages averaged over 100,000 permutations (at stage n, $Gap = \sum_{s\in\mathcal{S}_n} R(s)\xi_\theta(s, s^{bal}(N)) - \big(R(s^{bal2}(n))u(s^{bal}(N)) + R(s^{bal}(n))(1 - u(s^{bal}(N)))\big)$) and in parenthesis the value of $R(s^{bal2}(n))u(s^{bal}(N)) + R(s^{bal}(n))(1-u(s^{bal}(N)))$. The difference is nearly constant at all stages. So the best relative improvement could be when the bay gets emptier. Indeed, as n decreases and and reaches $C-1$, the lower bound only considers $s^{bal}(n)$ that has one container per column at most and hence has an expected number of relocations that is 0. Therefore a potential improvement could be to consider even more unbalanced case (more than 1 empty column) but there is no trivial extension of η_3.

3 Dynamic Online Container Relocation Problem

We extend this setting to the dynamic case where new containers arrive to the bay and have to be stacked. We refer to this problem as the Dynamic Stochastic Online CRP (DOSCRP). First note that any policy for the OSCRP can be extended for stacking as those problems are equivalent (stacking is relocating a container without the constraint of choosing among $C-1$ columns but all of them).

Consider that the configuration has n containers at a given time, then we suppose that a stacking occurs with probability p_n. Hence there is a retrieval with probability $1-p_n$ and in that case, each container is the one to be retrieved with probability $\frac{1}{n}$. We suppose that if $n=0$ and a retrieval occurs, nothing happens. Similarly if $n = N_{max} = HC - (H-1)$ and stacking occurs, nothing happens (no container is stacked as we want to insure feasibility of the configuration at all time). From this point we suppose that $p_0 < 1$ or $p_{N_{max}} > 0$.

Given a restricted Markovian policy θ, this process define a Markov chain on the states of class of configurations. Let us consider policies which makes this Markov chain irreducible (then it will also be aperiodic since $p_0 < 1$ or $p_{N_{max}} > 0$). We denote Θ_d the set of those policies. In that case, there exists a stationary distribution $\pi_\theta(p)$ on this Markov chain that depends on the policy θ and the regime p.

We use similar notations than the previous part for $X(s,c)$ and $R(s)$. Note that Properties 2.1 and 2.1 still hold. We consider $g(\theta,p) = \mathbb{E}_{s\in\pi_\theta(p)}\left[R(s)\right]$ which represents the number of relocations done if a container had to be retrieved under the stationary distribution $\pi_\theta(p)$. This function g represents the performance of the policy θ under steady-state. The goal is to find the optimal policy θ^*_d to the following problem:

$$\theta^*_d = \underset{\theta\in\Theta_d}{Argmin}\left(g(\theta,p)\right) \text{ , for all regime } p \tag{6}$$

Again, it is intuitive that the L policy should be optimal for this problem. Using the same ideas as for the static case, we define two lower bounds and show experimentally that these lower bounds are 'relatively close' to $g(\theta_L)$. Note again that this is no proof of the optimality of L policy but it gives an upper limit on the potential improvement of any policy over L.

3.1 First lower bound

First we define the notion of aggregate states: for all $n \in \{0,\dots,N_{max}\}$, we consider a unique state called the aggregate state for all configurations s such as $n = \sum_{i=1}^{H} is_i$ and we denote $\gamma_n(p)$ their stationary probabilities. Note that this distribution does not depend on the policy but only on the p_n's. Using the same idea that η_1, we have

$$g(\theta,p) = \mathbb{E}_{s\in\pi_\theta(p)}[R(s)] \geq \sum_{n=1}^{N_{max}} \min_{\left\{s:\sum_{i=1}^{H} is_i=n\right\}} (R(s))\gamma_n(p) = \sum_{n=1}^{N_{max}} R(s^{bal}(n))\gamma_n(p)$$
$$= \sum_{n=1}^{N_{max}} k_0(n)\left(1-\frac{C(k_0(n)+1)}{2n}\right)\gamma_n(p) = \lambda_1(p) \tag{7}$$

where $k_0(n) = \lfloor \frac{n}{C} \rfloor$.

Since $\gamma_n(p)$ can be analytically computed for certain regimes of p, this lower bound has the great asset of being a closed formula so no need to simulate any Markov chain to estimate it.

3.2 Second lower bound

The idea here is the same than for η_3. Due to the inner structure of the problem, there is a positive probability to be more unbalanced, i.e. having at least one empty column. Let denote $\delta_n(p)$ a lower bound on the probability of having at least one empty column in the n^{th} aggregate state. So $\gamma_n(p)-\delta_n(p)$ is an upper bound on the probability of not having a empty column (so potentially being in the balanced state) which gives

$$g(\theta,p) = \lambda_1(p) + \sum_{n=C}^{N_{max}} (R(s^{bal2}(n)) - R(s^{bal}(n)))\delta_n(p) = \lambda_2(p) \tag{8}$$

Proposition 3.1. *Let θ^* be the optimal solution of equation 6, $\lambda_1(p)$ in equation 7 and $\lambda_2(p)$ in equation 8, then we have*

$$g(\theta^*,p) \geq \lambda_2(p) \geq \lambda_1(p)$$

This property holds since $R(s^{bal2}(n)) - R(s^{bal}(n)) \geq 0$ and $\delta_n(p) \geq 0$. However $\delta_n(p)$ might not be as easily expressible as $\gamma_n(p)$. Therefore we need to use simulation to estimate these probabilities. Among all restricted policy, L is one which minimizes $\delta_n(p)$ because if there is an empty column, L policy will allocate a container to this column as soon as it is possible, hence minimizing the number of empty columns at all time so minimizing $\delta_n(p)$. Therefore we will use values of $\delta_n(p)$ obtained by simulating the Markov chain for the L policy.

3.3 Regimes of p_n

We consider three main regimes. The first one is referred to as *'uniform'*. We consider that for all $0 \leq n \leq N_{max}, p_n = \frac{1}{2}$. This models a system where at every aggregate state, a retrieval and a stack are equally likely. In that case, we have $\gamma_n(p) = \frac{1}{N_{max}+1}$.

The second main regime is *'proportionnal'*. It models a similar behavior as if the arrivals and departures were poisson processes and given $0 < \overline{N} < N_{max}$, we consider a distribution that is centered around this average value $\overline{N}$. In that case we take $p_n = \frac{1}{1+n/\overline{N}}$,

The last main regime we consider is called *'centered'* regime. If $n \leq \overline{N}$, then $p_n = 1 - \left(\frac{1}{2}\right)^{\overline{N}-n+1}$ and if $n > \overline{N}$ then $p_n = \left(\frac{1}{2}\right)^{n-\overline{N}+1}$. This regime is the closest to what happens in real life since for most ports the yard occupancy fluctuates around an average occupancy and does not deviate too much from it.

For both 'proportionnal' and 'centered' cases, $\gamma_n(p)$ can also be computed analytically (formula are not shown here for clarity purposes).

3.4 Computational Results

In this paragraph we show numerically how the two lower bounds ensure guarantees on the L policy under all three regimes for regular sized instances ($H = 4$, $C = 7, 10$ and $\overline{N} = \lfloor 0.9 * N_{max} \rfloor$). We present our results in Table 3. We simulate the Markov chain for the L policy with 10^6 steps and we get a simulated estimation of $\pi_{\theta_L}(p)$ and $\delta_n(p)$. Using these estimators we can compute $g(\theta_L, p)$ as well as $\lambda_1(p)$ and $\lambda_2(p)$.

Regime	Centered		Proportionnal		Uniform	
Number of Columns	7	10	7	10	7	10
$g(\theta_L, p)$	1.275	1.305	1.122	1.179	0.566	0.589
$\frac{g(\theta_L,p)-\lambda_1(p)}{\lambda_1(p)}$	9.1%	5.8%	9.8%	6.7%	13.6%	9.8%
$\frac{g(\theta_L,p)-\lambda_2(p)}{\lambda_2(p)}$	**5.3%**	**3.4%**	**5.5%**	**4.0%**	**6.6%**	**5.2%**

Table 3: Guaranteed performance of the L policy for DOSCRP: both lower bounds are relatively close to $g(\theta_L, p)$ for all three regimes even though $\lambda_2(p)$ almost halves the guaranteed gap of L with optimality compared to $\lambda_1(p)$. In addition the guaranteed gap decreases with C. This suggests that these simple lower bounds tend to get closer to the optimal solution as the problem gets larger. Finally, the guaranteed gap is lower for the 'centered' regime. This is due to the fact that the chosen $\overline{N}$ is quite high and our lower bounds are closer to optimality when there are more containers (see Part 2).

A future work could include tighter lower bounds using smarter decomposition of the Markov chain similar to the one done for $\lambda_2(p)$. This would help showing that the potential gain of introducing other policies is even smaller than the ones presented in Table 3.

4 Conclusion

This paper introduces the OSCRP and DOSCRP two new versions of the CRP. Intuitively, the 'Leveling' policy is the optimal policy for both problems. We derive several lower bounds for both problems and compute an optimality gap for the L policy. Using computational experiments we reach an optimality gap of 3.4% for the dynamic case which models most realistically the operations in port yards with no appointment systems.

Future work in this direction could refine lower bounds specially in the case of configurations with few containers, or prove optimality of L using dynamic programming or Markov Chains coupling techniques. Another intersting problem is the online problem with an adversarial component where L should still be optimal while the opponent would take the container on the bottom of highest column at each stage. Finally, a more generalized problem could be to assume a non-uniform probability distribution about the retreival orders.

References

[1] Akyüz, M. H. and Lee, C.-Y. (2014). A mathematical formulation and efficient heuristics for the dynamic container relocation problem. *Naval Research Logistics (NRL)*, 61(2):101–118.

[2] Borgman, B., van Asperen, E., and Dekker, R. (2010). Online rules for container stacking. *OR Spectrum*, 32:687–716.

[3] Borjian, S., Galle, V., Manshadi, V., Barnhart, C., and Jaillet, P. (2015a). Container relocation problem: Approximation, asymptotic and incomplete information. *ArXiv: 1505.04229*.

[4] Borjian, S., Manshadi, V. H., Barnhart, C., and Jaillet, P. (2015b). Managing relocation and delay in container terminals with flexible service policies. *ArXiv: 1503.01535*.

[5] Caserta, M., Schwarze, S., and Voß, S. (2009). A new binary description of the blocks relocation problem and benefits in a look ahead heuristic. *Evolutionary Computation in Combinatorial Optimization*, 5482:37–48.

[6] Caserta, M., Schwarze, S., and Voß, S. (2012). A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research*, 219(1):96–104.

[7] Dekker, R., Voogd, P., and van Asperen, E. (2006). Advanced methods for container stacking. *OR Spectrum*, 28:563–586.

[8] Jang, D. W., Kim, S. W., and Kim, K. (2013). The optimization of mixed block stacking requiring relocations. *International Journal of Production Economics*, 143:256–262.

[9] Kim, K. (1997). Evaluation of the number of rehandles in container yards. *Computers & Industial Engineering*, 32(4):701–711.

[10] Kim, K. H. and Hong, G.-P. (2006). A heuristic rule for relocating blocks. *Computers & Operations Research*, 33(4):940–954.

[11] Ku, D. (2014). Rehandling problem of pickup containers under truck appointment system. In *Computational Collective Intelligence. Technologies and Applications*, volume 8733 of *Lecture Notes in Computer Science*, pages 272–281. Springer International Publishing.

[12] Lee, Y. and Lee, Y.-J. (2010). A heuristic for retrieving containers from a yard. *Computers & Operations Research*, 37(6):1139–1147.

[13] Park, T., Choe, R., Kim, Y. H., and Ryu, K. R. (2011). Dynamic adjustment of container stacking policy in an automated container terminal. *International Journal of Production Economics*, 133:385–392.

[14] Petering, M. E. and 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.

[15] Tanaka, S. and Takii, K. (2014). A faster branch-and-bound algorithm for the block relocation problem. *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, 14:7–12.

[16] Zehendner, E., Feillet, D., and Jaillet, P. (2015). An algorithm with performance guarantee for the online container relocation problem. *working paper*.

[17] Zhang, H., Guo, S., Zhu, W., Lim, A., and Cheang, B. (2010). An investigation of IDA* algorithms for the container relocation problem. *Proceedings of the 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, 31-40.

[18] Zhao, W. and Goodchild, A. V. (2010). The impact of truck arrival information on container terminal rehandling. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):327–343.

[19] Zhu, W., Qin, H., Lim, A., and Zhang, H. (2012). Iterative deepening A* algorithms for the container relocation problem. *IEEE Transactions on Automation Science and Engineering*, 9(4):710–722.