Stochastic Container Relocation Problem with Constrained Pre-Processing Moves

Bernard Zweers, Rob van der Mei
Centrum Wiskunde & Informatica (CWI), The Netherlands, b.g.zweers@cwi.nl, r.d.van.der.mei@cwi.nl
Sandjai Bhulai
Vrije Universiteit Amsterdam, The Netherlands, s.bhulai@vu.nl

1. Introduction

Containers are the main way to ship goods around the world, and container terminals are an important link in this logistic operation. At container terminals, containers can be easily transshipped between different modes of transportation. During this transshipment, containers are stored at the terminal and to save space they are stacked on top of each other. Consequently, when a container that needs to be retrieved, the so-called target container, has other containers on top of it, then these so-called blocking containers need to be relocated to other stacks. These moves are unproductive and should be avoided as much as possible. In the literature, two ways to tackle this problem are discussed, namely the (Stochastic) Container Relocation Problem ((S)CRP) (Caserta, Schwarze, and Voß 2012, Ku and Arthanhari 2016, Galle et al. 2018) and the Container Pre-Marshalling Problem (CPMP)(Tierney, Pacino, and Voß 2017, Tanaka and Tierney 2018). In the SCRP, the goal is to relocate the blocking containers in such a way that the number of future relocation moves is minimized. In the CPMP, the containers are moved, using as few moves as possible, before any container is retrieved to obtain a layout in which no relocation moves are needed.

In this paper, we present a new problem that can be seen as a combination of these two existing problems. We call this problem the Stochastic Container Relocation Problem with Constrained Pre-Processing moves (SCRPCPP). The SCRPCPP is inspired by a problem faced by a container terminal in the port of Amsterdam. At this terminal, the crane operator is sometimes idle. During this time he or she could move some containers. We call these moves pre-processing moves. However, the idle time is not sufficient to perform all moves needed from the CPMP. As a result, some relocation might still be needed after the pre-processing moves to retrieve all containers. The goal of the SCRPCPP is to determine which pre-processing moves to execute such that the remaining
relocation moves are minimized. Our contribution is that we develop both a branch-and-bound algorithm and a heuristic for the SCRPCPP. Furthermore, we present an estimation method for the number of relocation moves in the stochastic setting.

2. Problem Formulation

In Figure 1(a), a container yard with a Rail Mounted Gantry Crane (RMGC) is seen from above. The RMGC is positioned above a single row of containers, called a bay. Attached to the RMGC is a trolley that can be used to pick up a container and place it in another stack or on a vehicle, as is shown in Figure 1(b). Compared with only moving the trolley, moving the entire RMGC is time-consuming. Hence, if a container is moved, then it is only moved to stacks in the same bay because then the RMGC does not need to move. Picking up and placing down a container is much more time-consuming than moving the trolley. Therefore, it is reasonable to simplify the problem and only consider the number of times a container is moved and not to which stack it is moved. Another assumption we make is that all containers in a bay will leave the bay before any new containers arrive. These assumptions are also made in the SCRP and the CPMP.

In the SCRPCPP, containers can be moved in two different phases: the pre-processing and the relocation phase. In the pre-processing phase, the number of moves is restricted by given a number. The relocation phase starts when the pre-processing phase has ended. To reduce the problem's complexity, only containers that are blocking the target container may be moved in the relocation phase. For every container, it is known in which time interval it will be retrieved. Nevertheless, multiple containers can be picked up in the same interval, and the retrieval order inside a time interval is a random uniform permutation. This situation corresponds to a terminal in which all truck drivers make an appointment to pick up containers in a specific time window, but there is no information about the order in which they will arrive in that interval. In the deterministic CRP, every interval consists only of a single container. Since the exact retrieval order is unknown, the number of relocation moves for a bay is a stochastic variable. Hence, the objective of the SCRPCPP is to minimize the expected number of relocation moves for the bay after the pre-processing moves.
3. Solution Methods

The CRP is NP-hard (Caserta, Schwarze, and Voß 2012), and similarly, one can prove that the SCRPCPP is also NP-hard. Therefore, we develop not only a branch-and-bound algorithm but also a multiheuristic to solve the SCRCPP. We focus on the pre-processing phase because both an optimal algorithm and a good heuristic for the relocation phase are given in (Galle et al. 2018). The idea of our multiheuristic is to investigate, for every combination of two stacks, the consequence of moving the top container of one stack in a correct position in the other stack. For every resulting bay, we estimate the number of remaining relocation moves and perform the pre-processing moves that result in the bay with the largest improvement. At three points in this heuristic, one has to choose between two options. Hence, the heuristic has eight different variants, which we all run for every instance. The final solution is the variant that gives the lowest number of estimated relocation moves.

Having a fast and accurate estimate for the number of relocation moves in a bay is crucial in our heuristic. To get such an estimation, we divide the containers into four different groups using simple rules. Containers in group 1 will never be relocated, for group 2 the expected number of relocation moves is less than one. We expect that containers in group 3 are only relocated once and in the last group are the containers that might be relocated more than once. Numerical experiments on the SCRP instances of (Ku and Arthanhari 2016) has shown that if we give the containers in group 1, 2, 3 and 4 a weight of, respectively, 0, 0.5, 1 and 1.4, the resulting sum is close to the mean number of relocation moves using simulations. The mean absolute percentage error between the estimation method and the simulation is about 5.2%. Moreover, this estimation method runs extremely fast and gives deterministic estimates.

We use the outcome of the heuristic as upper bound for the branch-and-bound algorithm. Furthermore, we develop a new lower bound for the SCRPCPP combining insights from lower bounds for the SCRP (Galle et al. 2018) and the CPMP (Tanaka and Tierney 2018). From the initial bay, we perform every possible pre-processing move in a depth-first tree and use the upper and lower bound to check if branching is needed for that bay.

4. Numerical Results

We solve the SCRPCPP for the instances of Ku and Arthanhari (2016) using both the heuristic and the branch-and-bound algorithm for the pre-processing phase and our estimation method for the relocation phase. We have set the maximum running time for the branch-and-bound algorithm to one hour and for the maximum number of pre-processing moves, we have chosen 25%, 50%, and 75% of the total number of containers in the bay. We investigate bays with five and ten stacks (S) with a maximum height between three and six containers (H) and with a fill rate of 50 and...
In Table 1, it is shown how many of the 30 instances could be solved by the branch-and-bound algorithm and the difference between the heuristic solution and the solution produced by the branch-and-bound algorithm. We see in Table 1 that for bays with high stacks and a fill rate of 67%, the branch-and-bound algorithm often does not terminate within one hour, whereas the heuristic runs in a couple of seconds. Moreover, the solutions of the heuristic are close to the solutions of the branch-and-bound algorithm.

### 5. Conclusion

With the SCRPCPP, two existing frameworks for stacking containers have been extended to solve a problem in which the idle time of a crane operator is utilized to do a limited number of pre-processing moves. We have developed a branch-and-bound algorithm for this problem but also a fast heuristic. A future extension of this work is to develop a method that can deal with a situation in which the total number of pre-processing moves is given per yard and that also has to be decided how many moves to allocate to each bay.

### References


