HAL
open science

Branch and price approach for the container relocation problem

Elisabeth Zehendner, Dominique Feillet

To cite this version:

Elisabeth Zehendner, Dominique Feillet. Branch and price approach for the container relocation problem. 14ème congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision, Feb 2013, Troyes, France. emse-00805062

HAL Id: emse-00805062

https://hal-emse.ccsd.cnrs.fr/emse-00805062v1

Submitted on 26 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Branch and price approach for the container relocation problem

Elisabeth Zehendner, Dominique Feillet

Ecole des Mines de Saint-Etienne, CMP Georges Charpak, 13541 Gardanne, France

{zehendner, feillet}@emse.fr

Keywords : *container terminal, container relocation problem, column generation, branch and price*

1 Introduction

Container terminals offer transfer facilities to move containers from vessels to trucks, trains and barges and vice versa. Incoming containers are not immediately loaded on an outgoing vehicle, but stored in the yard for up to several days. The yard is divided into blocks with several bays consisting of stacks and tiers. Figure 1 illustrates these terms. Terminals stack containers to use their scare land efficiently. Only the topmost container of each stack can be accessed directly. If another container has to be retrieved, containers above have to be relocated. These unproductive moves cannot be avoided completely as little information about future retrievals is known when a container is stored. But, poor yard management increases the number of relocations and the time needed to retrieve containers. Thereby, it decreases the overall productivity of the terminal.

The container (or block) relocation problem consists in minimizing the number of relocations during the retrieval process. N containers have to be retrieved from a bay with W stacks and H tiers in ascending order (from 1 to N). At each period t ($t \in \{1, \ldots, N\}$), container $n = t$ has to be retrieved from a position (i, j) with $i \in \{1, \ldots, W\}$ and $j \in \{1, \ldots, H\}$. The retrieval order is imposed by the stowage plan of vessels or the service order of trucks and does not match the storage layout. Containers are relocated within the same bay as relocations between bays are very time consuming. We and many other studies limit relocations to containers above the target container.

The container relocation problem is shown to be NP-hard ([2]). Several heuristic solution approaches (e.g., [1], [3], [4], [5]) and few exact solution approaches exist (e.g., [2], [5], [6], [7]). We present a branch and price procedure to solve the problem exactly.

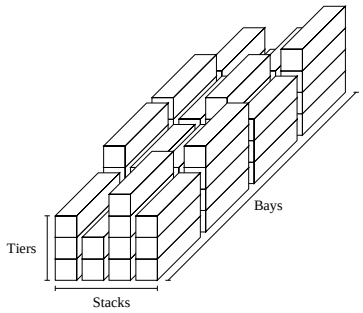

FIG. 1: A single block divided into bays, stacks and tiers

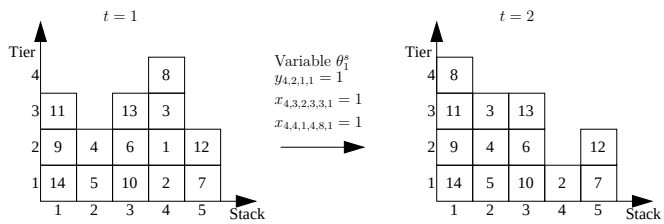

FIG. 2: Transformation of the bay by applying variable θ_1^s

2 Column generation

2.1 Master problem

Our master problem for the container relocation problem is a reformulation of the binary model presented in [2]. They introduced the following binary variables: b_{ijnt} to indicate if container n is at position (i, j) in period t, y_{ijnt} to indicate if container n is retrieved from position (i, j) in period t, x_{ijklnt} to indicate if container n is relocated from position (i, j) to (k, l) in period t. We introduce variables θ_t^s to indicate if a particular sequence of movements is applied to retrieve container $n = t$ and to relocate any containers stacked above. Binary parameters y_{ijnt}^s and x_{ijklnt}^s indicate which retrieval and which relocations have to be executed for variable θ_t^s. S_t counts the number of columns added to the master problem for period t. Figure 2 illustrates the way θ_1^s represents retrievals and relocations and how it transforms the bay if applied.

$$\min \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{W} \sum_{l=1}^{H} \sum_{t=1}^{N-1} \sum_{n=t+1}^{N} \sum_{s=1}^{S_t} x_{ijklnt}^s \cdot \theta_t^s$$

$$\sum_{n=t}^{N} b_{ijnt} \leq 1 \qquad \forall i = 1, \ldots, W, j = 1, \ldots, H, t = 1, \ldots, N-1 \tag{1}$$

$$\sum_{n=t}^{N} b_{ijnt} - \sum_{n=t}^{N} b_{ij+1nt} \geq 0 \qquad \forall i = 1, \ldots, W, j = 1, \ldots, H-1, t = 1, \ldots, N-1 \tag{2}$$

$$b_{ijnt+1} = b_{ijnt} + \sum_{s=1}^{S_t} \sum_{k=1}^{W} \sum_{l=1}^{H} \theta_t^s \cdot x_{klijnt}^s - \sum_{s=1}^{S_t} \sum_{k=1}^{W} \sum_{l=1}^{H} \theta_t^s \cdot x_{ijklnt}^s \tag{3}$$
$$\forall i = 1, \ldots, W, j = 1, \ldots, H, n = t+1, \ldots, N, t = 1, \ldots, N-2$$

$$b_{ijtt} - \sum_{s=1}^{S_t} \theta_t^s \cdot y_{ijtt}^s = 0 \qquad \forall i = 1, \ldots, W, j = 1, \ldots, H, t = 1, \ldots, N-1 \tag{4}$$

$$\sum_{s=1}^{S_t} \theta_t^s = 1 \qquad \forall t = 1, \ldots, N-1 \tag{5}$$

$$\sum_{k=1}^{j} b_{iktt} + \sum_{n=t+1}^{N} b_{ijnt+1} \leq 1 \qquad \forall i = 1, \ldots, W, j = 1, \ldots, H, t = 1, \ldots, N-2 \tag{6}$$

The objective function minimizes the total number of relocations. Constraint (1) makes sure that each position (i, j) is occupied by at most one container. Constraint (2) prevents gaps within stacks. Constraint (3) links the bay layout at period t with the layout at period $t+1$ via the executed relocations. Constraint (4) makes sure that container t is retrieved from the bay in period t. Constraint (5) imposes that one move sequence is chosen per period. All θ_t^s and b_{ijnt} are linear variables except for b_{ijn1} with $t = 1$ which indicate the initial bay configuration. Constraint (6) is a cut to enhance the quality of the model. It imposes that position (i, j) has to be empty at period $t + 1$ if container t is retrieved from a position (i, j') with $j' \leq j$.

2.2 Pricing subproblem

The pricing subproblem determines variables that could reduce the value of the objective function of the restricted master problem. Let γ_{ijnt} be the dual variables of Constraint (3), δ_{ijt} of Constraint (4) and μ_t of Constraint (5). We are looking for variables θ_t^s with negative reduced costs; thus satisfying Constraint (7).

$$\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{W}\sum_{l=1}^{H}\sum_{n=t+1}^{N} x_{ijklnt}^s \cdot (1+\gamma_{ijnt}-\gamma_{klnt}) + \sum_{i=1}^{W}\sum_{j=1}^{H} y_{ijtt}^s \cdot \delta_{ijt} - \mu_t < 0 \tag{7}$$

We use enumeration to create all feasible variables for each period t. To be feasible, a variable has to meet several criteria. In period t, only container $n = t$ may be retrieved and only containers stacked above may be relocated. Containers cannot be relocated to the stack from which they are picked up. If two or more containers have to be relocated, the LIFO order has to be respected: if container n is stacked below container n' before the relocation, n cannot be stacked below container n' after the relocation. The initial bay layout limits the positions where containers can be found in subsequent periods. We use this information to reduce the number of enumerated variables. We enumerate only variables where containers are picked up from positions that they may occupy, put into positions that may be free and not suspended (not above an empty position).

2.3 Bounding mechanisms

Let $R_{\max}^t$ denote the maximum number of relocations at period t. We present two upper bounds on $R_{\max}^t$ to sift out variables that cannot be found in an optimal solution. These bounds do not only reduce the number of variables to enumerate at period t, but also the attainable layouts and the variables to enumerate in subsequent periods. We use column generation within an iterative approach to speed up the subproblem. Let z^t describe the minimum number of relocations needed to retrieve containers 1 to t. Knowing z^t for all $t = 1, \ldots, N-1$ makes it possible to compute a tight $R_{\max}^t$ and to introduce an optimality criterion.

Bound 1 based on lower and upper bounds on the number of relocations The heuristic presented in [2] provides an upper bound UB on the number of relocations needed to retrieve all N containers. The lower bound LB_{t+} defines the minimum number of relocations that occur in periods $t+1$ to $N-1$. LB_{t+} depends on the number of containers placed above the target container and on the number of additional relocations caused by relocating containers to stacks containing containers with higher priorities. $R_{\max,1}^t$ is defined by Constraint (8).

$$R_{\max,1}^t = UB - \left\lceil z^{t-1} \right\rceil - LB_{t+} - 1 \qquad \forall t = 1, \ldots, N-1 \tag{8}$$

Bound 2 based on values of dual variables of the restricted master problem The potential gain of a variable has to outweigh its cost (the number of relocations) since only variables with negative reduced costs are added to the master problem. Constraint (9) limits $R_{\max,2}^t$ with regard to the maximum gain that may be reached at period t. We use the information about attainable layouts to improve the quality of the bound. Dual variables that correspond to movements of containers from or to positions that are not attainable are ignored. But, the bound does not consider interactions among different containers like LIFO and might therefore overestimate $R_{\max,2}^t$.

$$R_{\max,2}^t = \left\lceil \sum_{n=t+1}^{N} \left(-\min_{ij} \gamma_{ijnt} + \max_{kl} \gamma_{klnt} \right) - \min_{ij} \delta_{ijt} + \mu_t \right\rceil - 1 \qquad \forall t = 1, \ldots, N-1 \tag{9}$$

Optimality criterion Constraint (10) checks if the heuristic solution UB is optimal. This is the case if the upper bound equals the lower bound.

$$UB = \lceil z^t \rceil + LB_{t+} \qquad \exists t = 1, \ldots, N-1 \tag{10}$$

3 Branch and price

Column generation may obtain non-integer solutions where containers are split and occupy several positions within the bay. Two different situations may occur: a container is put into different stacks or a container is put into different tiers of the same stack. We We defined branching strategies for both cases: 1) If the container is split between two stacks i and i', we create two nodes that forbid relocations to i and i', respectively. One node prevents relocations to stack i and to the first $\lceil W/2 \rceil$ stacks without stacks i and i'. The other one prevents relocations to stack i' and the remaining stacks. 2) If the container is split between two tiers j and j' of the same stack i, we create two nodes that forbid relocations to (i, j) and (i, j'), respectively. The first nodes prevents relocations to tiers $1, ..., \min(j, j')$ of stack i. The other one prevents relocations to tiers $\min(j, j') + 1, ..., J$ of stack i. We chose to branch on the fractional b_{ijnt} at the earliest period t and closest to 0.5.

4 Computational results

We applied our column generation approach on commonly used instances from [3]. To evaluate the performance of branch and price, we also solve a modified version of the binary programming model presented in [2] with Cplex. Results show that column generation provides tighter lower bounds on the minimum number of relocations than the linear relaxation of the binary model (avg. gap of 1% vs. avg. gap of 10%). Despite the used bounding mechanisms, the enumeration subproblem is impractical for bigger instances as too many columns have to be generated. Our objective is to implement a more performing subproblem to improve the method.

References

[1] Marco Caserta, Silvia Schwarze, and Stefan Voß. A new binary description of the blocks relocation problem and benefits in a look ahead heuristic. In Carlos Cotta and Peter Cowling, editors, *Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science*, volume 5482, pages 37–48. Springer, 2009.

[2] Marco Caserta, Silvia Schwarze, and Stefan Voß. A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research*, 219:96–104, 2012.

[3] Marco Caserta, Stefan Voß, and Moshe Sniedovich. Applying the corridor method to a blocks relocation problem. *OR Spectrum*, 33:915–929, 2011.

[4] Florian Forster and Andreas Bortfeldt. A tree search procedure for the container relocation problem. *Computers & Operations Research*, 39:299–309, 2012.

[5] Kap Hwan Kim and Gyu-Pyo Hong. A heuristic rule for relocating blocks. *Computers & Operations Research*, 33:940–954, 2006.

[6] Y Lee and N Hsu. An optimization model for the container pre-marshalling problem. *Computers & Operations Research*, 34:3295–3313, 2007.

[7] Wenbin Zhu, Hu Qin, Andrew Lim, and Huidong Zhang. Iterative deepening A* algorithms for the container relocation problem. *IEEE Transactions on Automation Science and Engineering*, 9:710–722, 2012.