ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

T. Bacci, S. Mattia, P. Ventura

A NEW ALGORITHM FOR COMPUTING A LOWER BOUND FOR THE RESTRICTED BLOCK RELOCATION PROBLEM

R. 2, 2018

Tiziano Bacci – IASI - CNR — Via dei Taurini 19, 00185 Roma, Italia (`tiziano.bacci@iasi.cnr.it`).

Sara Mattia – IASI - CNR — Via dei Taurini 19, 00185 Roma, Italia (`sara.mattia@iasi.cnr.it`).

Paolo Ventura – IASI - CNR — Via dei Taurini 19, 00185 Roma, Italia (`paolo.ventura@iasi.cnr.it`).

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

via dei Taurini 19, 00185 ROMA, Italy

tel. ++39-06-49937101/02
fax ++39-06-49937106
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

In this paper we deal with the restricted Block (or Container) Relocation Problem. We present a polynomial time algorithm to calculate a new lower bound for the problem.

Key words: Block Relocation, Container Relocation, Lower Bound.

1. Introduction

A container terminal is a facility where containers are transferred between different transportation vehicles. Containers that wait for a truck, or a train, or a cargo ship, are stored in an area called *yard*. Since the yard is limited in space, the containers are piled into stacks. The height of each stack, i.e., the maximum number of containers that can be piled one above the other, is constrained by the height of the yard cranes used to move the containers. Typically, a container yard stores at the same time thousands of containers grouped into hundreds of stacks with a storage capacity which may be up to 10 containers [6]. Since a stack is accessible only from the top, when a container has to be retrieved from the storage area, any container located above it has to be moved into another stack with a *reshuffle operation*. Reshuffle operations are costly and time-consuming. Then, given a retrieval order of the containers, it is crucial to find a way to reallocate all the reshuffled containers so as to minimize the total number of reshuffle operations. This problem is known as the Block Relocation Problem (BRP). Here, we consider the restricted BRP, where only containers above the next one that has to be retrieved can be reshuffled.

Figure 1 gives an example of the BRP with an initial yard consisting of 3 stacks, 4 available slots for each stack, and 7 stored containers. Starting from the initial yard, the sequence of movements of an optimal solution is reported. At each step, the next container to be moved with a reshuffle or a retrieval operation is highlighted in gray. The minimum number of reshuffles required is six: containers 7, 3, 4 are reshuffled in order to retrieve container 1; container 4 is reshuffled to retrieve container 2; then, containers 7 and 6 are reshuffled to retrieve container 4.

Recent surveys on optimization problems arising in the management of container terminals can be found in [3] and [12]. In this context, the BRP problem is known to be NP-hard [4], as it generalizes the Mutual Exclusion Scheduling [2, 7]. Being the BRP problem both theoretically and computationally hard to solve, many heuristic approaches have also been introduced (see [5, 8, 10, 11, 14, 17] among the others). On the other hand, few contributions in the literature deal with lower bounds for the problem, although they can be used to certify the quality of heuristic approaches or to reduce the search space in exact algorithms [13, 17]. In this paper we present a new lower bound for the BRP problem. Experimental results show the effectiveness of our procedure. Throughout the paper, the words *container* and *block* will be used interchangeably.

In Section 2, we give a survey on the different lower bounds presented in the literature. In Section 3, we introduce the new lower bound. In Section 4, we compare our method with the lower bounds existing in the literature. In Section 5, we give the conclusions.

2. Lower bounds for the restricted BRP

All the lower bounds for the BRP problem existing in the literature can be calculated by iteratively solving relaxations of the *Generalized Minimum Blocking Items Problem* (GMBIP), whose definition is given in the following.

Let $M \in \mathbb{Z}^{w \times h}$ be a yard with w stacks of height h, and let $M(j,k)$ denote the block allocated in the k-th position of stack j ($M(j,k) = 0$ if the slot is empty). Now, let B be a set of n blocks

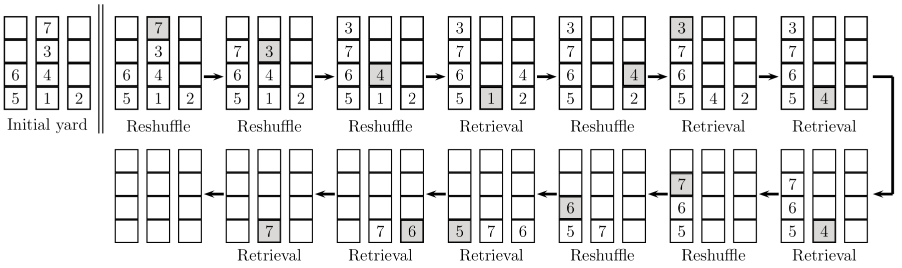

Figure 1: A representation of an optimal solution for the Block Relocation Problem.

that have to be located in the available slots of the stacks of M, according to a given order ϕ (denote by $\phi(i)$ the i-th block to be located). Finally, let $\overline{M}$ be the yard obtained from M once all the blocks of B have been allocated. $\overline{M}$ is said to be *feasible* if it is compatible with ϕ, i.e. for each couple $b, b' \in B$ with $\phi^{-1}(b) < \phi^{-1}(b')$, b is not located above b' in $\overline{M}$. Furthermore, we say that block $r = \overline{M}(j,k)$ is *r'-blocking* if it is located in some slot above $r' < r$ and that r is *blocking* if it is r'-blocking for some r' in $\overline{M}$. Given an input instance defined by (M, B, ϕ), the GMBIP problem is to find the feasible configuration $\overline{M}$ that minimizes the total number of blocking blocks of B. We denote by $G^*(M, B, \phi)$ such a minimum value.

The GMBIP problem is NP-hard in general, as it slightly generalizes the Minimum Blocking Items Problem (MBIP). In fact, in MBIP the initial yard is always empty. The computational complexity of MBIP has been analyzed in [1]. We now describe how to calculate a lower bound for the BRP by iteratively solving instances of GMBIP. Consider a BRP instance, with a set $\{1, ..., n\}$ of blocks located in a yard M. Recall that, at each time i, block i (located in stack say t^i) is retrieved from M and all the i-blocking blocks are reshuffled from t^i. Then, let $M^0 = M$ and, for each $i = 1, \dots, n$, let B^i be the set of i-blocking blocks of M^{i-1} taken in the order ϕ^i from the top to the bottom, and let M^i be obtained from M^{i-1} by removing the block i and all the blocks in B^i. Now observe that:

- $M^n(j,k) = 0$, for all $j \in \{1, ..., w\}$ and $k \in \{1, .., h\}$;
- block i could not be present in M^{i-1};
- all the blocks in B^i have to be reshuffled at time i in any solution of the input BRP instance defined by M.

Moreover, each reshuffled block of B^i can be reallocated in such a way that it becomes i'-blocking for some $i' > i$. In this case, it will have to be reshuffled again at time i'. It is not difficult to see that the minimum number of such blocks is exactly $G^*(\tilde{M}^i, B^i, \phi^i)$, where $\tilde{M}^i$ is obtained from M^i by removing stack t^i.

Therefore, a lower bound for the BRP instance defined by M is

$$\sum_{i=1}^{n} (|B^i| + G^*(\tilde{M}^i, B^i, \phi^i)). \tag{1}$$

As already mentioned, all the lower bounds for the BRP presented in the literature are derived from (1) substituting, at each iteration i, $G^*(\tilde{M}^i, B^i, \phi^i)$ with some lower bound.

In particular, the lower bound LB_K, introduced by Kim and Hong [9], uses 0 as a lower bound for $G^*(\tilde{M}^i, B^i, \phi^i)$. In other cases, such a lower bound on $G^*(\tilde{M}^i, B^i, \phi^i))$ is defined as the optimal value of some relaxed variant of GMBIP. Zhu et al. [17] defined LB_Z by solving a variant of GMBIP where both restrictions on the order ϕ of the incoming blocks as well as on the capacity h of the stacks are relaxed. The optimal value of such a relaxed problem, denoted here by $G^Z(M, B, \phi)$, is calculated in $O(n)$. Tanaka and Takii [13] proposed a $O(2^n)$ algorithm to solve the GMBIP variant (here denoted by GMBIP^T) obtained by relaxing the capacity restriction on the stacks of the current yard. We call $G^T(M, B, \phi)$ the optimal value of this problem and LB_T the corresponding lower bound for the BRP problem.

In the next section, we present a new lower bound for the BRP problem, obtained by solving to optimality (in $O(n\,log(n) + w\,log(w))$ time) the relaxation of GMBIP obtained by removing the restriction on the order ϕ.

3. A new lower bound for the restricted BRP

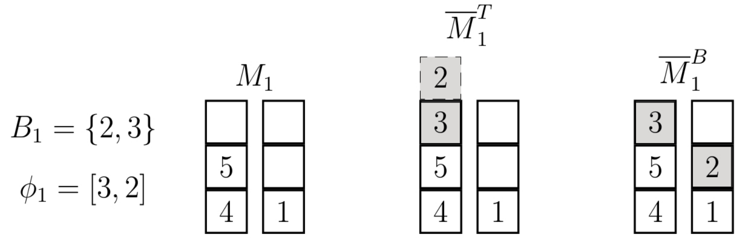

Figure 2: An instance (M_1, B_1, ϕ_1) with $\mathrm{G}^T(M_1, B_1, \phi_1) = 0 < \mathrm{G}^B(M_1, B_1, \phi_1) = 1$.

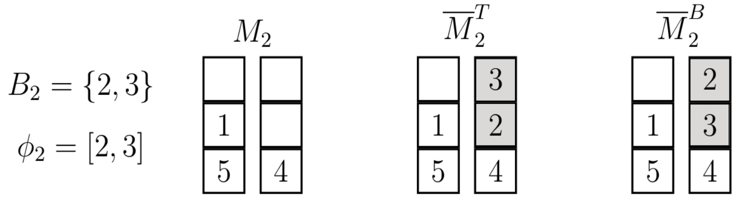

Figure 3: An instance (M_2, B_2, ϕ_2) with $\mathrm{G}^B(M_2, B_2, \phi_2) = 0 < \mathrm{G}^T(M_2, B_2, \phi_2) = 1$.

In this section, we introduce a new lower bound for the BRP problem. Let GMBIP^B be the GMBIP variant where the optimal configuration $\overline{M}$ does not need to be feasible with respect to the input order ϕ. We denote by $G^B(M, B, \phi)$ its optimal value and by LB_B the corresponding lower bound for BRP, obtained according to (1). As $0 \leq G^Z(M, B, \phi) \leq G^T(M, B, \phi), G^R(M, B, \phi)$, the following holds

$$LB_K \leq LB_Z \leq LB_T, LB_B \tag{2}$$

Note that, as the GMBIP^T and GMBIP^B are defined on two different relaxations of GMBIP, there does not exists a theoretical dominance relation between $G^T(M, B, \phi)$ and $G^B(M, B, \phi)$ (and therefore between LB_T and LB_B).

In Figure 2 and Figure 3, we present two input instances (M_1, B_1, ϕ_1) and (M_2, B_2, ϕ_2) with $\mathrm{G}^T(M_1, B_1, \phi_1) < \mathrm{G}^B(M_1, B_1, \phi_1)$ (Fig. 2) and $\mathrm{G}^T(M_2, B_2, \phi_2) > \mathrm{G}^B(M_2, B_2, \phi_2)$ (Fig. 3). In the figures, $\overline{M}_i^T$ and $\overline{M}_i^B$ represent the optimal solutions of GMBIP^T and GMBIP^B, respectively, obtained from the input instance (M_i, B_i, ϕ_i), for each $i = 1, 2$. Observe that, in Figure 2, the optimal configuration $\overline{M}_1^T$ (of value 0) does not satisfy the restriction on the capacity of the first stack. On the other hand, in Figure 3, in the optimal solution $\overline{M}_2^B$ (of value 0), block 2 is located above block 3, although 2 precedes 3 in ϕ_2.
Now, let δ_j be the residual capacity of each stack j of M and let σ_j be the smallest index of a block located in j. The GMBIP^B problem is then the minimum cost assignment problem where one wants to assign every block $b \in B$ in some available stack j of capacity δ_j. Here, each assignment (b, j) has cost 1, if $b \geq \sigma_j$, and 0, otherwise. In the following, we present an exact algorithm to solve GMBIP^B.

Algorithm 1 Algorithm for solving the GMBIP^B

```
STEP 0: Set $\mathrm{G}^B = 0$.
STEP 1: Construct the vector $\omega$, in which the blocks of $B$ are ordered non increasingly with respect to their index.
STEP 2: Construct the vector $\gamma$, in which the stacks of $M$ are ordered non increasingly with respect to the value of $\sigma$.
STEP 3:
for $i = 1, \dots, n$ do
    let $b = \omega[i]$;
    locate $b$ in the first stack $j$ of $\gamma$ with residual capacity $\delta[j] > 0$ and such that $\sigma[j] > b$, if any; in this case, set $\mathrm{G}^B = \mathrm{G}^B + 1$, $\delta[j] = \delta[j] - 1$;
    otherwise, locate $b$ in the last stack $j$ of $\gamma$ with a positive residual capacity; in this case, set $\delta[j] = \delta[j] - 1$.
```

Since, at every iteration of the loop in STEP 3, a suitable stack j can be found in constant time, the complexity of the overall procedure is $O(n\,log(n) + w\,log(w))$ time, being $n = |B|$ and w the number of stacks of M.

4. Computational results

In this section, we present some computational results that show the effectiveness of our algorithm.

For the experiments, we used two datasets. The first one includes six sets of instances already known in the literature [5, 11, 14, 15, 16, 17] and provided by the authors. The corresponding results are reported in Table 1. The second dataset contains instances that we generated according to two parameters, the number of stacks and the height of each stack. Following [16], for each pair (w, h), we randomly generated 50 instances with $w \times h - (h - 1)$ blocks. The corresponding results are reported in Table 2.

Each row of the tables is related to a group of instances. In Table 1, each group corresponds to one of the datasets in the literature (indicated in the first column), with the exception of the dataset presented in [5], where we distinguish among small-medium and large size instances. In

					LB_K		LB_Z		LB_T		LB_B	
SET	n	w	h	I	Value	Time	Value	Time	Value	Time	Value	Time
[5]	$[9, 100]$	$[3, 10]$	$[5, 12]$	840	18.37	0.00	22.04	0.00	22.87	0.00	22.05	0.00
	10000	100	102	40	9485.92	0.00	11177.08	0.00	$*$	$*$	11177.52	0.00
[11]	$[70, 720]$	$[16, 160]$	$[6, 8]$	14	213.11	0.00	215.29	0.00	215.61	0.00	215.33	0.00
[14]	$[3, 7]$	$[6, 36]$	$[4, 7]$	8000	8.00	0.00	9.53	0.00	9.81	0.00	9.54	0.00
[15]	$[3, 21]$	6	$[2, 5]$	600	2.93	0.00	3.16	0.00	3.20	0.00	3.16	0.00
[16]	$[7, 133]$	$[3, 12]$	$[3, 12]$	4000	32.71	0.00	39.01	0.00	41.13	0.00	39.11	0.00
[17]	$[15, 69]$	$[6, 10]$	$[3, 7]$	12500	21.46	0.00	25.87	0.00	26.59	0.00	25.93	0.00

Table 1: Comparative analysis on different lower bounds obtained on six datasets taken from the literature

				LB_K		LB_Z		LB_T		LB_B	
n	w	h	I	Value	Time	Value	Time	Value	Time	Value	Time
46	10	5	50	24.20	0.00	27.18	0.00	27.72	0.00	27.32	0.00
91	10	10	50	63.36	0.00	79.16	0.00	82.46	0.00	79.30	0.00
451	10	50	50	406.54	0.00	592.18	0.00	$*$	$*$	592.18	0.00
496	100	5	50	268.84	0.00	272.80	0.00	273.46	0.00	272.92	0.00
991	100	10	50	697.40	0.00	715.70	0.00	723.32	0.01	716.28	0.00
4951	100	50	50	4502.16	0.00	5014.68	0.00	$*$	$*$	5014.76	0.00
4996	1000	5	50	2713.62	0.00	2718.08	0.00	2718.86	0.00	2718.08	0.00
9991	1000	10	50	7059.04	0.00	7081.30	0.00	7089.14	0.14	7081.44	0.00
49951	1000	50	50	4552.20	0.01	46134.12	0.01	$*$	$*$	46134.30	0.01

Table 2: Comparative analysis on different lower bounds on randomly generated instances

Table 2 the instances are grouped according to their size.

For each row, n, w and h indicate the number (or range) of blocks, stacks, and stack height, respectively, while I is the number of instances in the group. Then, columns LB_K, LB_Z, LB_T, and LB_B correspond to the lower bounds presented in the previous sections. For each of these columns, *Value* is the average value and *Time* is the average computing time (in seconds) on the instances of the group. The symbol $*$ is used to indicated that some instance of the corresponding group could not be solved within the time limit of one hour.

The computational results show that, as expected, $LB_B \geq LB_K$ and $LB_B \geq LB_Z$ (see (2)). Moreover, they also show that, in practice, $LB_T > LB_B$. However, the computational times required to solve GMBIP^T are, in particular on the large instances, much higher than the ones needed for solving GMBIP^B. Indeed, recall that the algorithm proposed by Tanaka and Takii is exponential in the number of reshuffled blocks. Therefore, even if it produces better bounds, LB_T cannot be used in practice within an algorithm that solves BRP real-size instances. Hence, LB_B seems to present a good compromise between quality of the solution and computational time, as it always outperforms the values provided by LB_K and LB_Z in, essentially, the same computing time.

5. Conclusions

We introduced a new lower bound (LB_B) for the restricted Block Relocation Problem, that is widely studied in the context of logistics of containers in container terminals. We also presented an algorithm to calculate LB_B in polynomial time. Computational results showed that our lower bound is very effective and it is able to produce good values even on large instances. This suggests that it could be successfully integrated into heuristic algorithms for solving real-size instances, both to certify the quality of the solutions as well as to limit the search space.

Acknowledgements

The authors have been partially supported by Ministry of Instruction University and Research (MIUR) with the program PRIN 2015, project "SPORT - Smart PORt Terminals", code 2015XAPRKF; project "Nonlinear and Combinatorial Aspects of Complex Networks", code 2015B5F27W; project "Scheduling cuts: new optimization models and algorithms for cutting, packing and nesting in manufacturing processes", code 20153TXRX9.

References

[1] T. Bacci, S. Mattia, and P. Ventura, "Some complexity results for the minimum blocking items problem," *Springer Proceedings in Mathematics and Statistics*, vol. 217, pp. 475–483, 2017.

[2] F. Bonomo, S. Mattia, and G. Oriolo, "Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6261 – 6268, 2011.

[3] H. Carlo, I. Vis, and K. Roodbergen, "Storage yard operations in container terminals: Literature overview, trends, and research directions," *European Journal of Operational Research*, vol. 235, no. 2, pp. 412 – 430, 2014.

[4] M. Caserta, S. Schwarze, and S. Voß, "A mathematical formulation and complexity considerations for the blocks relocation problem," *European Journal of Operational Research*, vol. 219, pp. 96–104, 2012.

[5] M. Caserta, S. Voß, and M. Sniedovich, "Applying the corridor method to a blocks relocation problem," *OR Spectrum*, vol. 33, no. 4, pp. 915–929, 2011.

[6] H. O. Günter and K. H. Kim, *Container Terminals and Automated Transport Systems.* Springer Berlin Heidelberg, 2005.

[7] K. Jansen, "The mutual exclusion scheduling problem for permutation and comparability graphs," *Information and Computation*, vol. 180, pp. 71–81, 2003.

[8] R. Jovanovic and S. Voß, "A chain heuristic for the Blocks Relocation Problem," *Computers & Industrial Engineering*, vol. 75, pp. 79 – 86, 2014.

[9] K. H. Kim and G. P. Hong, "A heuristic rule for relocating blocks," *Computers & Operations Research*, vol. 33, pp. 940–954, 2006.

[10] D. Ku and T. Arthanari, "On the abstraction method for the container relocation problem," *Computers & Operations Research*, vol. 68, no. Supplement C, pp. 110 – 122, 2016.

[11] Y. Lee and Y. L. Lee, "A heuristic for retrieving containers from a yard," *Computers & Operations Research*, vol. 47, pp. 1139–1147, 2010.

[12] J. Lehnfeld and S. Knust, “Loading, unloading and premarshalling of stacks in storage areas: Survey and classification,” *European Journal of Operational Research*, vol. 239, pp. 297–312, 2014.

[13] S. Tanaka and K. Takii, “A faster branch-and-bound algorithm for the block relocation problem,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 181–190, 2016.

[14] T. Unluyurt and C. Aydin, “Improved rehandling strategies for the container retrieval process,” *Journal of Advanced Transportation*, vol. 46, no. 4, pp. 378–393, 2012.

[15] Y. Wan, J. Liu, and P.-C. Tsai, “The Assignment of Storage Locations to Containers for a Container Stack,” *Naval Research Logistics*, vol. 56, no. 8, pp. 699–713, 2009.

[16] K. C. Wu and C. J. Ting, “A beam search algorithm for minimizing reshuffle operations at container yards,” *International Conference on Logistics and Maritime Systems, Busan, Korea, September 1517, 2010*, 2010.

[17] W. Zhu, H. Qin, A. Lim, and H. Zhang, “Iterative Deepening A* Algorithms for the Container Relocation Problem,” *IEEE Transactions on Automation Science and Engineering*, vol. 9, pp. 710–722, 2012.