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
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
that have to be located in the available slots of the stacks of $M$, according to a given order $\phi$ (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 $\phi$, 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 $r'$ and that $r$ is blocking if it is $r'$-blocking for some $r'$ in $\overline{M}$. Given an input instance defined by $(M, B, \phi)$, 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 $i = 1, \ldots, n$, let $B^i$ be the set of $i$-blocking blocks of $M^{i-1}$ taken in the order $\phi^i$ 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 $j \in \{1, \ldots, w\}$ and $k \in \{1, \ldots, h\}$;
- 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

$$\sum_{i=1}^{n} (|B^i| + G^*(\tilde{M}^i, B^i, \phi^i)).$$ (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 $G^*(\tilde{M}, B^i, \phi^i)$. In other cases, such a lower bound on $G^*(\tilde{M}, B^i, \phi^i)$ 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 $\phi$ 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 an $O(2^n)$ algorithm to solve the GMBIP variant (here denoted by GM-BIP)$^T$ 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 $\phi$.

3. A new lower bound for the restricted BRP

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
 & $M_1$ & $\overline{M}_1^T$ & $\overline{M}_1^B$ \\
\hline
$B_1$ = \{2, 3\} & 5 & 3 & 3 \\
$\phi_1$ = [3, 2] & 4 & 1 & 2 \\
\hline
\end{tabular}
\caption{An instance $(M_1, B_1, \phi_1)$ with $G^T(M_1, B_1, \phi_1) = 0 < G^B(M_1, B_1, \phi_1) = 1$.}
\end{figure}

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
 & $M_2$ & $\overline{M}_2^T$ & $\overline{M}_2^B$ \\
\hline
$B_2$ = \{2, 3\} & 1 & 3 & 2 \\
$\phi_2$ = [2, 3] & 5 & 1 & 3 \\
\hline
\end{tabular}
\caption{An instance $(M_2, B_2, \phi_2)$ with $G^B(M_2, B_2, \phi_2) = 0 < G^T(M_2, B_2, \phi_2) = 1$.}
\end{figure}

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 $\phi$. 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

$$LB_K \leq LB_Z \leq LB_T, LB_B$$

(2)

Note that, as the GMBIP$^T$ and GMBIP$^B$ are defined on two different relaxations of GMBIP, there does not exist 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 \((M_1, B_1, \phi_1)\) and \((M_2, B_2, \phi_2)\) with \(G^T(M_1, B_1, \phi_1) < G^B(M_1, B_1, \phi_1)\) (Fig. 2) and \(G^T(M_2, B_2, \phi_2) > G^B(M_2, B_2, \phi_2)\) (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 \((M_i, B_i, \phi_i)\), 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 \(\phi_2\).

Now, let \(\delta_j\) be the residual capacity of each stack \(j\) of \(M\) and let \(\sigma_j\) 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 \(\delta_j\). 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 \(G^B = 0\).

**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 \(i = 1, \ldots, n\) 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 \(G^B = G^B + 1\), \(\delta[j] = \delta[j] - 1\);

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

<table>
<thead>
<tr>
<th>Set</th>
<th>n</th>
<th>w</th>
<th>h</th>
<th>I</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>[9, 100]</td>
<td>[3, 10]</td>
<td>[5, 12]</td>
<td>40</td>
<td>840</td>
<td>18.37</td>
<td>0.00</td>
<td>22.04</td>
<td>0.00</td>
<td>22.87</td>
<td>0.00</td>
<td>22.05</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>100</td>
<td>102</td>
<td>5485.92</td>
<td>0.00</td>
<td>1117.68</td>
<td>0.00</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>[11]</td>
<td>[70, 720]</td>
<td>[16, 160]</td>
<td>[6, 8]</td>
<td>14</td>
<td>213.11</td>
<td>0.00</td>
<td>215.29</td>
<td>0.00</td>
<td>215.61</td>
<td>0.00</td>
<td>215.33</td>
<td>0.00</td>
</tr>
<tr>
<td>[14]</td>
<td>[3, 7]</td>
<td>[6, 36]</td>
<td>[4, 7]</td>
<td>8000</td>
<td>8.00</td>
<td>0.00</td>
<td>9.53</td>
<td>0.00</td>
<td>9.81</td>
<td>0.00</td>
<td>9.54</td>
<td>0.00</td>
</tr>
<tr>
<td>[15]</td>
<td>[3, 21]</td>
<td>6</td>
<td>[2, 5]</td>
<td>600</td>
<td>2.93</td>
<td>0.00</td>
<td>3.16</td>
<td>0.00</td>
<td>3.20</td>
<td>0.00</td>
<td>3.16</td>
<td>0.00</td>
</tr>
<tr>
<td>[16]</td>
<td>[7, 133]</td>
<td>[3, 12]</td>
<td>[3, 12]</td>
<td>4000</td>
<td>32.71</td>
<td>0.00</td>
<td>39.01</td>
<td>0.00</td>
<td>41.13</td>
<td>0.00</td>
<td>39.11</td>
<td>0.00</td>
</tr>
<tr>
<td>[17]</td>
<td>[15, 69]</td>
<td>[6, 10]</td>
<td>[3, 7]</td>
<td>12500</td>
<td>21.46</td>
<td>0.00</td>
<td>25.87</td>
<td>0.00</td>
<td>26.59</td>
<td>0.00</td>
<td>25.93</td>
<td>0.00</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>n</th>
<th>w</th>
<th>h</th>
<th>I</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
<th>Value</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>10</td>
<td>5</td>
<td>50</td>
<td>24.20</td>
<td>0.00</td>
<td>27.18</td>
<td>0.00</td>
<td>27.72</td>
<td>0.00</td>
<td>27.32</td>
<td>0.00</td>
</tr>
<tr>
<td>91</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td>63.36</td>
<td>0.00</td>
<td>79.16</td>
<td>0.00</td>
<td>82.46</td>
<td>0.00</td>
<td>79.30</td>
<td>0.00</td>
</tr>
<tr>
<td>451</td>
<td>10</td>
<td>50</td>
<td>50</td>
<td>406.54</td>
<td>0.00</td>
<td>592.18</td>
<td>0.00</td>
<td>*</td>
<td>*</td>
<td>592.18</td>
<td>0.00</td>
</tr>
<tr>
<td>496</td>
<td>100</td>
<td>5</td>
<td>50</td>
<td>268.84</td>
<td>0.00</td>
<td>272.80</td>
<td>0.00</td>
<td>273.46</td>
<td>0.00</td>
<td>272.92</td>
<td>0.00</td>
</tr>
<tr>
<td>991</td>
<td>100</td>
<td>10</td>
<td>50</td>
<td>697.40</td>
<td>0.00</td>
<td>715.70</td>
<td>0.00</td>
<td>723.32</td>
<td>0.01</td>
<td>716.28</td>
<td>0.00</td>
</tr>
<tr>
<td>4951</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>4502.16</td>
<td>0.00</td>
<td>5014.68</td>
<td>0.00</td>
<td>*</td>
<td>*</td>
<td>5014.76</td>
<td>0.00</td>
</tr>
<tr>
<td>4996</td>
<td>1000</td>
<td>5</td>
<td>50</td>
<td>2713.62</td>
<td>0.00</td>
<td>2718.08</td>
<td>0.00</td>
<td>2718.86</td>
<td>0.00</td>
<td>2718.08</td>
<td>0.00</td>
</tr>
<tr>
<td>9991</td>
<td>1000</td>
<td>10</td>
<td>50</td>
<td>7059.04</td>
<td>0.00</td>
<td>7081.30</td>
<td>0.00</td>
<td>7089.14</td>
<td>0.14</td>
<td>7081.44</td>
<td>0.00</td>
</tr>
<tr>
<td>49951</td>
<td>1000</td>
<td>50</td>
<td>50</td>
<td>4552.20</td>
<td>0.01</td>
<td>46134.12</td>
<td>0.01</td>
<td>*</td>
<td>*</td>
<td>46134.30</td>
<td>0.01</td>
</tr>
</tbody>
</table>

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


