Solving Berth Allocation Problem with Column Generation

Ceyda Oğuz · Özge Narin

Berth allocation problem is to find the best allocation of berths to the incoming ships in a container terminal. Container terminals are the regions that lay along the quayside where containers are transported from one place to another. Importance of the container terminals grows with the increasing need for transportation. The advantages of sea transportation can be realized better with efficiently operated container terminals. According to Imai et al. [1], influence of berth allocation has the highest impact on efficient operations of a container terminal. This is also supported by Vis and Koster [4], Steenken [2], and Murty [3] by showing that berths are the most important resources to operate a container terminal efficiently.

In Dynamic Berth Allocation Problem (DBAP) the ships will arrive during the planning horizon, which makes the problems much harder than the static berth allocation problem (SBAP). DBAP is solvable exactly up to 10 berths - 45 ships instance and there exists a good metaheuristic algorithm to handle larger instances [6]. We propose a column generation (CG) approach for solving DBAP and our motivation is to provide better solutions for large DBAP instances. Actually, this study is a preliminary study to assess the strengths and weaknesses of CG for DBAP. In the rest, we will briefly describe our CG approach and will present the preliminary results obtained for DBAP (Model used for DBAP is given in [5]).

CG is a good approach for those problems where the size grows immensely as a result of dealing with all the variables explicitly. CG generates only those variables which have the potential to improve the objective function. These variables, which have negative reduced costs, are determined by the pricing problem (subproblem). We have developed the CG based on the formulations given in [7] because of the similarity of that problem with DBAP.

Ceyda Oğuz
Köc Üniversitesi, Turkey
E-mail: coguz@ku.edu.tr

Özge Narin
Köc Üniversitesi, Turkey
E-mail: onarin@ku.edu.tr
We split the DBAP into two problems: the master problem and the subproblem (pricing problem). The master problem is the original problem with only a subset of variables being considered and the subproblem is a new problem created to identify a new variable. The objective function of the subproblem is the reduced cost of the new variable (column) with respect to the current dual variables, and the column should satisfy the constraints of DBAP.

For the master problem, all assumptions made in [5] for DBAP model remain the same and new assumptions are the following:

(a) All berths have a feasible column set $D_i$ indexed by 1, 2,..., $d_i$.
(b) Variable $z^d_i$ denotes $d^{th}$ column of berth $i$ and if this variable is 1, cost of $r_{id}$ is incurred for column $d=1, 2,...,D_i$ of berth $i=1,2,...,I$.
(c) Columns have $T$ number of components for the ships denoted by $z^d_i = \{z^d_{i1}, z^d_{i2},..., z^d_{IT}\}$.

We define $z^d_i$ as follows;

$$z^d_i = \begin{cases} 
1 & \text{if column } d \text{ is chosen for berth } i \\
0 & \text{otherwise} 
\end{cases}$$

If it is decided to schedule ship $t$ at berth $i$ in column $d$, $z^d_{it}$ component of column $d$ is 1, otherwise it is 0. So the master problem is;

Minimize $\sum_{i \in B} \sum_{d \in D_i} z^d_i r_{id}$ 

s. t.

$$\sum_{d \in D_i} z^d_i = 1 \quad \forall i \in B,$$  

$$\sum_{i \in B} \sum_{d \in D_i} z^d_i z^d_{it} = 1 \quad \forall t \in V,$$  

$$z^d_i \in \{0, 1\} \quad \forall i \in B, d \in D_i.$$  

The linear relaxation of the master problem is solved at each iteration with the existing columns and after this solution, dual variables of the constraint set (3) are passed to the pricing problem as $\lambda_j$ values. For the first iteration we need an initial solution to get dual variables to pass to the subproblem, which is found by applying a simple heuristic in a negligible time. During the iterations, once the stopping criteria is reached, the master problem is solved as a mixed integer programming model to obtain an integer solution.

In the subproblem we define;

$$a_{jk} = \begin{cases} 
1 & \text{if ship } j \text{ is scheduled at position } k \\
0 & \text{otherwise} 
\end{cases}$$

$w_{jk}$ = the length of the idle period before the arrival of ship $j$ that will be scheduled at position $k$. 

Then the model is:

\[
\text{Minimize } \sum_{j \in V} \sum_{k \in O} (kc_{ij} + s_i - A_j)a_{jk} + \sum_{j \in V} \sum_{k \in O} kw_{jk} - \sum_{j \in V} \sum_{k \in O} \lambda ja_{jk} \tag{5}
\]

s. t.

\[
\sum_{k \in O} a_{jk} \leq 1 \quad \forall j \in V, \tag{6}
\]

\[
\sum_{l \in V} \sum_{m \in P_k} (c_{lm}a_{lm} + w_{lm}) + w_{jk} - (A_j - S_i)a_{jk} \geq 0 \quad \forall j \in W_i, k \in O, \tag{7}
\]

\[
a_{jk} \in {0, 1} \quad \forall j \in V, k \in O, \tag{8}
\]

\[
w_{jk} \geq 0 \quad \forall j \in V, k \in O. \tag{9}
\]

Subproblem is solved with the dual variables that come from the master problem. Objective function of the subproblem gives the reduced cost of the column generated for the berth at hand. Generated column is added to the master problem if reduced cost is negative. Since the generated column is for a specific berth, the subproblem is solved for each berth individually. Therefore the subproblem must be solved \( I \) (number of berth) times at each iteration.

We have tested CG approach on the data provided in [6] and compared its performance with the Variable Neighborhood Search (VNS) algorithm given in that study. Our solutions are obtained on an Intel(R) computer with Xeon(R) CPU 3.00 GHz, 4.00 GB of RAM. Ilog Concert Technology CPLEX 11.0 and Visual Studio 2005 are used for the solutions and the codes are written in C++. Our aim is to evaluate the CG approach in terms of the run time and solution quality.

The results are presented in Table 1, where error percentages and running times of VNS, CG, a heuristic CG (i.e. heuristic based subproblem) and, exact algorithms are given. Our results for DBAP show that for smaller instances, CG can find the optimal solution easily as VNS does. The superiority of CG becomes apparent for instances starting with 10 berths and 45 ships. For these instances CG is able to find the optimal solution whereas VNS cannot. We are able to solve instances of DBAP up to 10 berths and 50 ships.

CG and exact solution have the same solution quality but CPLEX is faster than CG for the instances that are solvable for it. When instances get larger and harder, CG becomes more important because the tree size of CPLEX solution cannot be coped due to the memory problem. Although we cannot find a solution with CPLEX for instances starting with 10 berths and 45 ships and larger, CG gives good solutions.

We are currently working on applying a heuristic method for the subproblem (pricing problem). Results received so far are given in the table at the end of this section in column Heuristic CG. This approach fairly decreased the run time of the CG algorithm without worsening the solution quality.
Table 1 DBAP instances

<table>
<thead>
<tr>
<th>Berths</th>
<th>Ships</th>
<th>$\delta_i^*$</th>
<th>Error (Percentage)</th>
<th>Running time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>VNS CG Heuristic CG Exact VNS CG Heuristic CG Exact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>1/2</td>
<td>0.08 0.00 0.00 (**)</td>
<td>3.54 16252.60 4661.11 (**)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3/5</td>
<td>0.03 0.00 0.00 0.00</td>
<td>2.74 7634.95 1553.47 3250.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5/8</td>
<td>0.04 0.00 0.00 0.00</td>
<td>2.29 8203.13 1787.41 1699.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7/8</td>
<td>0.00 0.00 0.00 0.00</td>
<td>0.36 2187.20 414.01 19.06</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>1/2</td>
<td>0.00 0.00 0.00 (**)</td>
<td>3.83 40669.23 17542.63 (**)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3/5</td>
<td>0.03 0.00 0.00 (**)</td>
<td>2.80 21255.73 5284.85 (**)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5/8</td>
<td>0.04 0.00 0.00 (**)</td>
<td>3.89 18301.43 6791.25 (**)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7/8</td>
<td>0.01 0.00 0.00 0.00</td>
<td>1.63 3752.8 886.91 24.29</td>
</tr>
</tbody>
</table>

(*): Availability times of the berths, $\delta_i^*$ = 1/2, 3/5, 5/8, 7/8, of the time interval between the arrivals of the first and last ships.

(**): Solution cannot be found because of “out of memory” error.

References