Measures of congestion in container terminals

Matteo Salani – TRANSP-OR EPFL
Ilaria Vacca – TRANSP-OR EPFL
Arnaud Vandaele – Univ. of Louvain

AIRO 2008 - Ischia
10 September 2008
Outline

• Introduction and motivation
• Literature
• Modeling
• Congestion measures
• Optimization
• Preliminary results
• Future work
Container Terminals (CT)

Zone in a port to import/export/transship containers
Container utilisation is the **easiest way** to transport goods
Different areas in a terminal: berths, yard, gates
Different types of vehicles to travel between the yard and the berth

Optimization in CT:
- Berth allocation
- Quay Cranes scheduling
- Yard operations
Motivation

Along the quay, several boats with **thousand** containers are loaded/unloaded. These containers movement lead to an high traffic in the yard zone, where the most of slowdowns occur.

(Tactical) Terminal planners often optimize the distance (time) traveled by container carriers because it impacts directly on the performance indicators, disregarding:

- Congestion issues (operations slowdowns because of bottlenecks)
- Alternative solutions (symmetries)

**considering deterministic data**

**Aim of this study:**

- Model the terminal and develop measures of congestion
- Evaluate the impacts of the optimization of such measures on the terminal
Assumptions

In the berth/yard allocation plans, we define a path as an OD pair:
- An origin (berth or block)
- A destination (berth or block)
- number of containers

We consider flows of containers over a working shift.

Decisions could be taken on:
- The berth allocation plan (boats and berths)
- The yard allocation plan (destination blocks)
- Demand splitting (for berth to block flows only)

Given a set of $p$ paths, determine the destination blocks
Literature

Layout:

Congestion:


Modeling

Basic element
Modeling

(m x n) basic elements for a yard

Coordinates system to indicate origin and destination of containers

Only Berth-to-Yard and Yard-to-Berth paths:

\((x_o, y_o) \rightarrow (x_d, y_d)\)
Routing rules:
- Horizontal lanes are one way
- Vertical lanes are two way

Toward the block, closest left vertical lane, turn right.

Toward the quay, turn right at the first vertical lane.

Back to origin berth position.

Distance travelled, closed formula (Manhattan)
Symmetries

Minimize distance:
In a 2x2 yard with 2 paths, no capacity on blocks

Number of solutions with equal distance
Measures

**Aim:** Estimate the state/congestion of a yard when implementing a plan *(using simple closed formulas)* in order to identify secondary objectives.

We considered:

- Interference among blocks sharing the same lane
- Lane congestion
- Path interference
Block congestion

\( b_{ij} = \begin{cases} 
1 & \text{path } j \text{ arrive in area } i \\
0 & \text{otherwise} 
\end{cases} \)

\# of containers in each area and best case:

\[
N_i = \sum_{j=1}^{p} b_{ij} c_j \\
N^x = \frac{\sum_{j=1}^{p} c_j}{r}
\]

Considered norm-1 and norm-2 with respect to the best over the worst case.

\( r = \min(s, p) \)

\[
C_b = \frac{D}{D_{\max}} = \frac{\sqrt{\sum_{j=1}^{p} (N_i - N^x)^2}}{\sqrt{(r - 1) \frac{\sum_{j=1}^{p} c_j}{r}}}
\]

we have \( s = 2n + n(m-1) \) areas
Example with 2 norm

Possible solutions = 1728
It simply measures the average traffic over an edge in the best traffic situation when flows are spread over the network.

$$
\theta = \max_k f_k > 0 \quad \mu = \min_k f_k > 0
$$

$$
C_e = \frac{\theta - \mu}{\sum_{j=1}^{n} c_j}
$$

Improved:

$$
C_e = \frac{n(\theta - \sum_{j=1}^{p} c_j)}{(n-1) \sum_{j=1}^{p} c_j}
$$
Path congestion

Measure disturbances among paths.

**Proximity matrix** $P$ (2p X 2p), symmetric, 0 on the diagonal, influenced by routing rules.

Worst case: all 1 matrix (except diagonal)

$$C_p = \frac{1^T P\tilde{c}}{(2p-1)\sum_{j=1}^{2p} c_j}$$
Example

Objective function: $z = C_b + C_e + C_p$
**Example**

Objective function: \( z = C_b + C_e + C_p \)

<table>
<thead>
<tr>
<th></th>
<th>Nb solutions</th>
<th>Nb different values</th>
<th>MIN</th>
<th>Nb MIN</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2x2) - 3 paths</td>
<td>512</td>
<td>46</td>
<td>0.4764</td>
<td>10</td>
<td>0.2</td>
</tr>
<tr>
<td>(2x2) - 4 paths</td>
<td>4096</td>
<td>282</td>
<td>0.3473</td>
<td>30</td>
<td>1.4</td>
</tr>
<tr>
<td>(2x2) - 5 paths</td>
<td>32768</td>
<td>1831</td>
<td>0.5068</td>
<td>21</td>
<td>12.23</td>
</tr>
<tr>
<td>(2x2) - 6 paths</td>
<td>262144</td>
<td>7354</td>
<td>0.461</td>
<td>12</td>
<td>112.85</td>
</tr>
<tr>
<td>(2x3) - 3 paths</td>
<td>1728</td>
<td>52</td>
<td>0.4764</td>
<td>116</td>
<td>0.67</td>
</tr>
<tr>
<td>(2x3) - 4 paths</td>
<td>20736</td>
<td>470</td>
<td>0.3473</td>
<td>350</td>
<td>7.29</td>
</tr>
<tr>
<td>(2x3) - 5 paths</td>
<td>248832</td>
<td>4271</td>
<td>0.13</td>
<td>108</td>
<td>121.65</td>
</tr>
</tbody>
</table>
## Algorithm: GRASP

Greedy randomized adaptive search procedure

<table>
<thead>
<tr>
<th></th>
<th>MIN</th>
<th>CPU (s) (enumeration)</th>
<th>CPU (s) (algorithm)</th>
<th>Nb iteration to reach optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2x2) – 3 paths</td>
<td>0,4764</td>
<td>0,2</td>
<td>0,1</td>
<td>5</td>
</tr>
<tr>
<td>(2x2) – 4 paths</td>
<td>0,3473</td>
<td>1,4</td>
<td>0,2</td>
<td>10</td>
</tr>
<tr>
<td>(2x2) – 5 paths</td>
<td>0,5068</td>
<td>12,23</td>
<td>0,5</td>
<td>30</td>
</tr>
<tr>
<td>(2x2) – 6 paths</td>
<td>0,461</td>
<td>112,85</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>(2x3) – 3 paths</td>
<td>0,4764</td>
<td>0,67</td>
<td>0,1</td>
<td>5</td>
</tr>
<tr>
<td>(2x3) – 4 paths</td>
<td>0,3473</td>
<td>7,29</td>
<td>0,1</td>
<td>5</td>
</tr>
<tr>
<td>(2x3) – 5 paths</td>
<td>0,13</td>
<td>121,65</td>
<td>0,5</td>
<td>25</td>
</tr>
<tr>
<td>(2x3) – 6 paths</td>
<td>0,1953</td>
<td>??</td>
<td>15</td>
<td>1000</td>
</tr>
</tbody>
</table>
### More Tests

More realistic instances

<table>
<thead>
<tr>
<th></th>
<th>in 0.1s</th>
<th>in 1s</th>
<th>in 5s</th>
<th>in 10s</th>
<th>in 20s</th>
<th>in 60s</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3x10) – 3</td>
<td>0.4764</td>
<td>0.4764</td>
<td>0.4764</td>
<td>0.4764</td>
<td>0.4764</td>
<td></td>
</tr>
<tr>
<td>(3x10) – 4</td>
<td>0.3473</td>
<td>0.3473</td>
<td>0.3473</td>
<td>0.3473</td>
<td>0.3473</td>
<td></td>
</tr>
<tr>
<td>(3x10) – 5</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>(3x10) – 6</td>
<td>0.389</td>
<td>0.195</td>
<td>0.195</td>
<td>0.195</td>
<td>0.195</td>
<td></td>
</tr>
<tr>
<td>(3x10) – 7</td>
<td>0.343</td>
<td>0.267</td>
<td>0.267</td>
<td>0.267</td>
<td>0.267</td>
<td></td>
</tr>
<tr>
<td>(3x10) – 8</td>
<td>0.26</td>
<td>0.1692</td>
<td>0.1646</td>
<td>0.1646</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3x10) – 9</td>
<td>0.304</td>
<td>0.2763</td>
<td>0.2763</td>
<td>0.2763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3x10) – 15</td>
<td>0.2446</td>
<td>0.1931</td>
<td>0.1705</td>
<td>0.1582</td>
<td>0.1817</td>
<td>0.1602</td>
</tr>
<tr>
<td>(3x10) – 20</td>
<td>0.3275</td>
<td>0.2276</td>
<td>0.1663</td>
<td>0.1624</td>
<td>0.1609</td>
<td>0.1389</td>
</tr>
</tbody>
</table>
more Tests

Other ongoing tests (3x10), 3 to 5 ships, up to 20 paths:

- Balanced/unbalanced repartition of loads among paths
- Balanced/unbalanced repartition of loads among ships
- Number of paths per ship (2 to 5)

Optimizing measures does not degrade too much the main objectives while helping in differentiate symmetric solutions
Conclusions and Outlook

Simple closed formulas to evaluate congestion in container terminals
Useful to differentiate symmetric solutions with equal distance (or expected completion time)

TODO:
- Additional tests
- Multi-objective optimization problem, explore other than weighted sum
- Evaluate the effects with a CT simulator (queueing model)
- Improve the algorithm: study an exact approach, relax the assumptions, i.e. extend the set of possible decisions (berth allocation, demand splitting, working shifts)