Yard traffic and congestion in container terminals

Ilaria Vacca
Transport and Mobility Laboratory, EPFL

joint work with
Michel Bierlaire, Matteo Salani & Arnaud Vandaele

6th International Conference on Computational Management Science
May 1st, 2009
Outline

- Introduction and motivation
- Modeling
- Congestion measures
- Optimization
- Computational results
- Future work
Container Terminals (CT)

- Zone in a port to import/export/transship containers
- Different areas in a terminal: berths, yard, gates
- Different types of vehicles to travel between the yard and the berth
Motivation

- Along the quay, containers are loaded/unloaded onto/from several boats
- Containers’ transfer lead to a high traffic in the yard zone

- The **berth&yard allocation plan** assigns ships to berths and containers to yard blocks
- Terminal planners usually minimize the total distance travelled by the carriers, disregarding:
  - Congestion issues (operations slowdowns because of bottlenecks)
  - Alternative solutions (symmetries)

**Aim of this study:**

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

- We take into account flows of containers from the quayside to the yard
- Given a berth&yard allocation plan, we define a path as an OD pair:
  - origin (berth)
  - destination (block)
  - number of containers
- We consider flows of containers over a working shift
- Decisions could be taken on:
  - the berth allocation plan (berths and ships)
  - the yard allocation plan (destination blocks)
  - demand splitting over blocks

⇒ In this study: given a set of $p$ paths, determine the destination blocks
Literature

- **Layout:**

- **Congestion:**
Modeling the terminal

Basic element
Modeling the terminal

- \((m \times n)\) basic elements of 2 blocks each compose the yard
- coordinates system for OD pairs \((x_o, y_o) - (x_d, y_d)\)
- only berth-to-yard and yard-to-berth paths are considered
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
Congestion measures

- **Aim of the study:**
  - estimate the state/congestion of a yard when implementing a plan
  - provide simple closed formulas, to be used as secondary objectives

- **Factors taken into account:**
  - interference between blocks sharing the same lane
  - lane congestion
  - interference between paths
1. Block congestion

- congestion among blocks sharing the same lane
- "area": blocks with the same entrance node
  - # of areas: $s = 2n + n(m-1)$
  - $c_j$: # of containers on path $j = 1...p$
  - $N_i$: # of containers allocated to area $i$
  - $N^*$: # of containers in each area in the optimal solution (even distribution among areas)

$$C_b = \frac{D}{D_{max}} = \sum_{i=1}^{s} \frac{|N_i - N^*|}{2(s-1)} \sum_{j=1}^{p} c_j$$

- 1-norm and 2-norm w.r.t. the best over the worst case
1. Block congestion

- 3 paths in a 2x3 yard (12 blocks) → possible solutions: $12^3 = 1728$
- Number of solutions with same block congestion (distribution of 2-norm $C_b$):
2. Edge congestion

- this indicator simply measures the average traffic over an edge

\[
\begin{align*}
\theta &= \max_k f_k \\
\mu &= \min_k f_k \\
\theta_{\text{max}} &= \sum_{j=1}^{p} c_j \\
\mu_{\text{min}} &= 0
\end{align*}
\]

\[
C_e = \frac{\theta - \mu}{\theta_{\text{max}} - \mu_{\text{min}}} = \frac{\theta - \mu}{\sum_{j=1}^{p} c_j}
\]

- the best traffic situation is when flows are spread over the network: \( \mu^* = \frac{\sum_{j=1}^{p} c_j}{n} \)

\[
C_e = \frac{\theta - \mu^*}{\theta_{\text{max}} - \mu^*} = \frac{\theta - \mu^*}{\sum_{j=1}^{p} c_j - \frac{\sum_{j=1}^{p} c_j}{n}} = \frac{(n) \theta - \sum_{j=1}^{p} c_j}{(n - 1) \sum_{j=1}^{p} c_j}
\]
2. Edge congestion

- 3 paths in a 2x3 yard (12 blocks) → possible solutions: $12^3 = 1728$
- number of solutions with same edge congestion (distribution of improved $C_e$):
3. Path congestion

- interference among “crossing” paths
  - proximity matrix $P$ (2$p$ X 2$p$)
  - $p$ berth-to-yard + $p$ yard-to-berth paths
  - $P$ is symmetric, 0 on the diagonal, 1 if two paths are “neighbours”
  - definition of $P$ is influenced by routing rules
  - worst case: all 1 matrix (except diagonal)

$$C_p = \frac{p}{N_{max}} = \frac{1^T P_c}{(2n - 1) \sum_{i=1}^{2n} c_i}$$
Example

- 3 paths in a 2x3 yard
- Distribution of the objective function \( z = \lambda_b C_b + \lambda_e C_e + \lambda_p C_p \)
Example

Objective function: \( z = \lambda_b C_b + \lambda_c C_c + \lambda_p C_p \)

<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>
Optimization algorithm: GRASP

- GRASP: Greedy Randomized Adaptive Search Procedure

- Objective: assign a destination to each path such that congestion is minimized

- The algorithm builds a solution iteratively:
  - at each step, the destination for one specific path is chosen
# Optimization algorithm: GRASP

<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><strong>(2x2) – 3 paths</strong></td>
<td>0.4764</td>
<td>0.2</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td><strong>(2x2) – 4 paths</strong></td>
<td>0.3473</td>
<td>1.4</td>
<td>0.2</td>
<td>10</td>
</tr>
<tr>
<td><strong>(2x2) – 5 paths</strong></td>
<td>0.5068</td>
<td>12.23</td>
<td>0.5</td>
<td>30</td>
</tr>
<tr>
<td><strong>(2x2) – 6 paths</strong></td>
<td>0.461</td>
<td>112.85</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td><strong>(2x3) – 3 paths</strong></td>
<td>0.4764</td>
<td>0.67</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td><strong>(2x3) – 4 paths</strong></td>
<td>0.3473</td>
<td>7.29</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td><strong>(2x3) – 5 paths</strong></td>
<td>0.13</td>
<td>121.65</td>
<td>0.5</td>
<td>25</td>
</tr>
<tr>
<td><strong>(2x3) – 6 paths</strong></td>
<td>0.1953</td>
<td>??</td>
<td>15</td>
<td>1000</td>
</tr>
</tbody>
</table>
### Computational tests

#### More realistic instances

<table>
<thead>
<tr>
<th>Instance</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>0.4764</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>0.3473</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>0.13</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>0.195</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>0.267</td>
</tr>
<tr>
<td>(3x10) – 8</td>
<td>0.26</td>
<td>0.1692</td>
<td>0.1646</td>
<td>0.1646</td>
<td>0.1646</td>
<td>0.1646</td>
</tr>
<tr>
<td>(3x10) – 9</td>
<td>0.304</td>
<td>0.2763</td>
<td>0.2763</td>
<td>0.2763</td>
<td>0.2763</td>
<td>0.2763</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>
Conclusions and Outlook

- simple closed formulas to evaluate congestion in container terminals
- useful to differentiate symmetric solutions with equal distance

Ongoing work:
- validation of our approach via a CT simulator
- multi-objective optimization problem (explore other than weighted sum)
- improve the algorithm: study an exact approach; relax the assumptions, i.e. extend the set of possible decisions (berth allocation, demand splitting)
Thanks for your attention!