The truck scheduling problem at cross-docking terminals

L. Berghman, C. Briand, R. Leus and P. Lopez; PMS 2012

ORSTAT, KU Leuven; CNRS; LAAS
Outline

• What is cross-docking?
• Exclusive versus mixed mode
• Problem statement and notations
• Time-indexed formulation
• Branch-and-bound
• Computational results
• Conclusions and future research
What is cross-docking?

- Items are immediately sorted out, reorganized, based on customer demands, and loaded into outbound trucks.
- The storage capacity and the length of the stay of a product in the warehouse are limited.
- Appropriate coordination of inbound and outbound trucks is needed.
The truck scheduling problem

- It decides on the succession of truck processing at the dock doors
- Trucks are allocated to the different docks so as to minimize the storage usage during the product transfer
- The internal organization of the warehouse is not explicitly taken into consideration
- We do not model the resources that may be needed to load or unload the trucks
Exclusive mode

Each dock serves exclusively either as an outbound dock or as an inbound dock throughout the schedules execution.

http://people.hofstra.edu/geotrans/eng/ch5en/conc5en/crossdocking.html
Mixed mode

Each dock can be used both for loading and unloading
Problem statement and notations

- A set of incoming trucks $i \in I$ need to be unloaded
- A set of outgoing trucks $o \in O$ need to be loaded
- The processing time of truck $j \in I \cup O$ equals $p_j$
- Every truck has its release time $r_j$ (planned arrival time) and its deadline $\tilde{d}_j$ (latest allowed departure time)
- There are precedence relations $(i, o) \in P \subset I \times O$: $w_{io}$ represents the number of pallets transshipped from $i$ to $o$
- $s_j$ is the starting time of the handling of truck $j$
- There are $n$ docks that can be used in mixed-mode
Conceptual problem statement

\[
\min \sum_{(i,o) \in P} w_{io}(s_o - s_i)
\]

subject to

\[
\begin{align*}
    s_j & \geq r_j & \forall j \in I \cup O \\
    s_j + p_j & \leq \tilde{d}_j & \forall j \in I \cup O \\
    s_o - s_i & \geq 0 & \forall (i,o) \in P \\
    |A_t| & \leq n & \forall t \in T
\end{align*}
\]

\[
A_t = \{j \in I \cup O | s_j \leq t < s_j + p_j\} \text{ the set containing all tasks being executed at time } t
\]

\[
T \text{ the set containing all time instants considered}
\]
Time-indexed (linear) formulation

For all inbound trucks $i \in I$ and all time periods $\tau \in T_i$,

$$x_{i\tau} = \begin{cases} 
1 & \text{if the unloading of inbound truck } i \text{ is started during time period } \tau, \\
0 & \text{otherwise},
\end{cases}$$

with $T_i = [r_i + 1, \tilde{d}_i - p_i + 1]$, the relevant time window for inbound truck $i$.

For all outbound trucks $o \in O$ and all time periods $t \in T_o$,

$$y_{o\tau} = \begin{cases} 
1 & \text{if the loading of outbound truck } o \text{ is started during time period } \tau, \\
0 & \text{otherwise},
\end{cases}$$

with $T_o = [r_o + 1, \tilde{d}_o - p_o + 1]$. 
Time-indexed formulation

\[
\begin{aligned}
\min \sum_{(i,o) \in P} \sum_{\tau \in T} w_{io\tau} (y_{o\tau} - x_{i\tau}) \\
\text{subject to}
\end{aligned}
\]

\[
\begin{aligned}
\sum_{\tau \in T_i} x_{i\tau} &= 1 \quad \forall i \in I \quad (1) \\
\sum_{\tau \in T_o} y_{o\tau} &= 1 \quad \forall o \in O \quad (2) \\
\sum_{\tau \in T} \tau (x_{i\tau} - y_{o\tau}) &\leq 0 \quad \forall (i, o) \in P \quad (3) \\
\sum_{i \in I \atop u=\tau-p_i+1} x_{iu} + \sum_{o \in O \atop u=\tau-p_o+1} y_{ou} &\leq n \quad \forall \tau \in T \quad (4) \\
\end{aligned}
\]

\[x_{i\tau}, y_{o\tau} \in \{0, 1\}\]
Two different precedence constraints

\[ \sum_{\tau \in T} \tau \left( x_{i\tau} - y_{o\tau} \right) \leq 0 \quad \forall (i, o) \in P \]

\[ \sum_{u=1}^{\tau} x_{iu} - \sum_{u=1}^{\tau} y_{ou} \leq 0 \quad \forall (i, o) \in P; \forall \tau \in T \]

- Aggregated versus disaggregated constraint
- Disaggregated is theoretically stronger
- The additional CPU time needed to solve the larger linear program does not always counterbalance the significant improvement of the bound
Branch-and-bound I

• At each node, an uncapacitated cross-docking problem is considered

\[
\min \sum_{(i,o) \in P} w_{io}(s_o - s_i)
\]
subject to

\[
s_o - s_i - \delta_{i,o} \geq 0 \quad \forall (i, o) \in P
\]
\[
\delta_{i,o} \in [\underline{\delta}_{i,o}, \overline{\delta}_{i,o}] \quad \forall (i, o) \in P
\]

• The dual of this problem is a max-cost flow problem that can be solved efficiently, which gives a lower bound.
Branch-and-bound II

- Initially, \( [\delta_{i,o}, \bar{\delta}_{i,o}] = [0, \tilde{d}_o - r_i - p_o] \)
- At each node, the relaxed solution is analyzed in order to find a time \( \tau^* \) for which the gate capacity \( n \) is exceeded (if the solution respects the capacity, it is a local optimum);
- Then a pair \( (i^*, o^*) \in P \) such that either \( i^* \) or \( o^* \) is in progress at time \( \tau^* \) and \( w_{i^*,o^*} \) is minimal.
- Two child nodes are considered:
  - Child 1: \( [\delta_{i,o}, \bar{\delta}_{i,o}] \leftarrow \lceil (\delta_{i,o} - \bar{\delta}_{i,o}) / 2 \rceil, \bar{\delta}_{i,o} \)
  - Child 2: \( [\delta_{i,o}, \bar{\delta}_{i,o}] \leftarrow [\delta_{i,o}, \lfloor (\delta_{i,o} - \bar{\delta}_{i,o}) / 2 \rfloor] \)
Generation of instances

- \( n = 10 \) and \(|I| = 30\)
- \(|O| = \alpha \times |I|\) with \( \alpha = \{0.8, 1, 1.2\}\)
- \( p_i \in [\beta, 30] \) with \( \beta = \{10, 20, 30\}\)
- \( w_{io} \in \left[ \frac{0.8 \times p_i}{\gamma}, \frac{1.2 \times p_i}{\gamma} \right] \) with \( \gamma \in [1, p_i]\)
- \( r_i \in [1, \frac{\delta \times \sum p_i}{n}] \) with \( \delta = \{0.3, 0.6, 0.9\}\)
- \( \tilde{d}_o \in [1.5 \times d_o, 8 \times d_o] \) with \( d_o = \max_{(i,o) \in P} \{r_i + p_o\}\)
- \( r_o \in [\max_{(i,o) \in P} \{r_i\}, d_o - p_o] \)
- \( \tilde{d}_i \in [1.5 \times (r_i + p_i), \min_{(i,o) \in P} \{d_o\}] \)
Preliminary computational results I

- Solving with Cplex ($T_{cpu} \leq 5$ minutes)

| $|O|$ | $\beta$ | $\delta$ | # opt | # feas | # infeas | objective value |
|-----|--------|--------|------|-------|---------|-----------------|
| 24  | 10     | 0.3    | 1    | 5     | 4       | 18181           |
| 24  | 10     | 0.6    | 0    | 9     | 1       | 25842           |
| 24  | 10     | 0.9    | 1    | 6     | 3       | 40231           |
| 24  | 20     | 0.3    | 0    | 5     | 5       | 25209           |
| 24  | 20     | 0.6    | 0    | 9     | 1       | 38763           |
| 24  | 20     | 0.9    | 1    | 5     | 4       | 41819           |
| 30  | 10     | 0.3    | 0    | 6     | 4       | 17919           |
| 30  | 10     | 0.6    | 0    | 7     | 3       | 36644           |
| 30  | 10     | 0.9    | 0    | 7     | 3       | 33085           |
| 30  | 20     | 0.3    | 0    | 5     | 5       | 26200           |
| 30  | 20     | 0.6    | 1    | 8     | 1       | 42643           |
| 30  | 20     | 0.9    | 0    | 8     | 2       | 48581           |
| 30  | 30     | 0.3    | 1    | 6     | 3       | 39727           |
| 30  | 30     | 0.6    | 0    | 8     | 2       | 71194           |
| 30  | 30     | 0.9    | 0    | 9     | 1       | 71040           |
| 36  | 10     | 0.3    | 0    | 10    | 0       | 23012           |
| 36  | 10     | 0.6    | 0    | 7     | 3       | 28343           |
| 36  | 10     | 0.9    | 0    | 5     | 5       | 48968           |
| 36  | 20     | 0.3    | 0    | 8     | 2       | 25043           |
| 36  | 20     | 0.6    | 1    | 7     | 2       | 43823           |
| 36  | 20     | 0.9    | 0    | 8     | 2       | 49757           |
Preliminary computational results II

- Experimenting our branch-and-bound procedure
  - First results are encouraging but further improvements are needed ...
  - Solving the problem relaxation is very fast
  - But the procedure fails in finding feasible solutions for almost all the instances
  - Nodes are cut only when they become unfeasible
- Improvement ideas:
  - Modify the branching strategy to favor deadline satisfactions
  - Use a greedy algorithm for trying to find a feasible solution at each node (intensification)
  - Any other (good) idea is welcomed!
Conclusions and future research

Conclusions

- Truck scheduling problem at cross-docking terminals
- Time-indexed (integer programming) formulation
- Branch-and-bound

Future research

- Branch-and-bound improvements
- Comparison between the mixed mode strategy and the exclusive one
- Analysis of the special case $p_i = p$
Future research: staging