HAL
open science

The truck scheduling problem at cross-docking terminals

Lotte Berghman, Cyril Briand, Roel Leus, Pierre Lopez

To cite this version:

Lotte Berghman, Cyril Briand, Roel Leus, Pierre Lopez. The truck scheduling problem at cross-docking terminals. International Conference on Project Management and Scheduling (PMS 2012), Apr 2012, Louvain, Belgium. 4p. hal-00661318

HAL Id: hal-00661318

https://hal.science/hal-00661318v1

Submitted on 19 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The truck scheduling problem at cross-docking terminals

L. Berghman[1], C. Briand[2,3], R. Leus[1] and P. Lopez[2,3]

[1] ORSTAT; K.U.Leuven; Naamsestraat 69, 3000 Leuven, Belgium
{Lotte.Berghman, Roel.Leus}@econ.kuleuven.be
[2] CNRS; LAAS; 7 Avenue du colonel Roche, 31077 Toulouse, France
{Cyril.Briand, Pierre.Lopez}@laas.fr
[3] Université de Toulouse; UPS, INSA, INP, ISAE; UT1, UTM, LAAS, 31077 Toulouse, France

Keywords: cross docking, truck scheduling, parallel machine scheduling, integer programming, branch-and-bound.

1 Introduction

Cross docking is a warehouse management concept in which items delivered to a warehouse by inbound trucks are immediately sorted out, reorganized based on customer demands and loaded into outbound trucks for delivery to customers, without requiring excessive inventory at the warehouse. Compared to traditional warehousing, the storage as well as the length of the stay of a product in the warehouse is limited, which requires an appropriate coordination of inbound and outbound trucks. The truck scheduling problem, which decides on the succession of truck processing at the dock doors, is especially important to ensure a rapid turnover and on-time deliveries. The problem studied in this paper concerns the operational level: 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 also do not model the resources that may be needed to load or unload the trucks, which implies the assumption that these resources are available in sufficient quantities to ensure the correct execution of an arbitrary docking schedule.

The concept of cross docking has received a lot of attention in recent literature: cases with one receiving and one shipping door are most frequently studied. A comprehensive overview of different variations and the existing literature can be found in Boysen and Fliedner (2010). Carlo and Bozer (2011) state that in a typical cross-dock application, each dock serves exclusively either as an outbound dock or as an inbound dock throughout the schedule's execution; this is called the *exclusive mode*. Their experiments show that clustering the inbound docks together and the outbound docks together is generally not a good configuration when minimizing the travel distance of the forklifts inside the warehouse. For this reason, we study the more general case in which each dock can be used both for loading and unloading; this is referred to as the *mixed mode*.

In the following section, a detailed problem statement is given. A time-indexed (linear programming) formulation is presented in Section 3. Section 4 describes a branch-and-bound algorithm, and Section 5 summarizes our contributions.

2 Problem statement

We examine a cross-docking warehouse where incoming trucks $i \in I$ need to be unloaded and outgoing trucks $o \in O$ need to be loaded (where I is the set containing all inbound trucks while O is the set containing all outbound trucks). The warehouse features n docks that can be used both for loading and unloading (this is called *mixed mode*). The processing time of truck $j \in I \cup O$ equals p_j. This processing time includes the loading or unloading

but also the transportation of goods inside the cross dock and other handling operations between dock doors. It is assumed that there is sufficient workforce to load/unload all docked trucks at the same time. The products on the trucks are packed on unit-size pallets, which move collectively as a unit. Each pallet on an inbound truck i needs to be loaded on an outbound truck o, which gives rise to a precedence relation $(i, o) \in P \subset I \times O$. Each truck j has a release time r_j (planned arrival time) and a deadline $\tilde{d}_j$ (its latest allowed departure time). Products can be transshipped directly from an inbound to an outbound truck, if the outbound truck is placed at a dock. Otherwise, the products are temporarily stored and will be loaded later on. Each couple $(i, o) \in P$ has a weight w_{io}, representing the number of pallets that go from inbound truck i to outbound truck o. The objective is to minimize the weighted duration of the pallets stocked in the warehouse. According to N. Boysen and M. Fliedner (2010), this is a valuable objective because the cross-docking concept relies on a rapid turnover of shipments. Also the danger of late shipments is reduced in this way: the number of products in the storage area can only be decreased by loading them on outbound trucks to leave the terminal as early as possible. Moreover, a lower stock size also reduces the material handling effort inside the terminal. Our problem can be modeled as a parallel machine scheduling problem with release dates, deadlines and precedence constraints. As $1|r_i|L_{\max}$ is NP-hard, even finding a feasible solution for the problem considered is difficult.

For all trucks $j \in I \cup O$, let

$$s_j = \text{the starting time of the handling of truck } j.$$

A conceptual problem statement with these variables is the following:

$$\min \sum_{(i,o)\in P} w_{io}(s_o - s_i) \tag{1}$$

subject to

$$s_j \geq r_j \qquad \forall j \in I \cup O \tag{2}$$

$$s_j + p_j \leq \tilde{d}_j \qquad \forall j \in I \cup O \tag{3}$$

$$s_o - s_i \geq 0 \qquad \forall (i, o) \in P \tag{4}$$

$$|A_t| \leq n \qquad \forall t \in T \tag{5}$$

with $A_t = \{j \in I \cup O | s_j \leq t < s_j + p_j\}$ the set containing all tasks being executed at time t and T the set containing all time instants considered. The objective function (1) minimizes the total weighted usage of the storage area. Constraints (2) and (3) impose the time windows for all trucks. Constraints (4) ensure that, if there exists a precedence relationship between inbound truck i and outbound truck o, then o cannot be processed before i. Finally, constraints (5) enforce the capacity of the docks.

3 Time-indexed formulation

A time-indexed formulation discretizes the continuous time space into periods τ of a fixed length (with $\mathcal{T}$ the set containing all time periods τ). Let period τ be the interval $[t-1, t[$. It is well known that time-indexed formulations perform well for scheduling problems because the linear programming relaxations provide strong lower bounds. For this reason, we will test the integer programming formulation below, which is called F1 in the sequel.

For all inbound trucks $i \in I$ and for all time periods $\tau \in \mathcal{T}_i$, we have

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

with $\mathcal{T}_i = [r_i + 1, \tilde{d}_i - p_i + 1]$, the relevant time window for inbound truck i. Additionally, for all outbound trucks $o \in O$ and for all time periods $t \in \mathcal{T}_o$, we have

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

with $\mathcal{T}_o = [r_o + 1, \tilde{d}_o - p_o + 1]$, the relevant time window for outbound truck o.

A time-indexed formulation for the considered truck scheduling problem is the following:

$$\min \sum_{(i,o)\in P} \sum_{\tau\in\mathcal{T}} w_{io}\tau \left(y_{o\tau} - x_{i\tau}\right) \tag{6}$$

subject to

$$\sum_{\tau\in\mathcal{T}_i} x_{i\tau} = 1 \qquad \forall i \in I \tag{7}$$

$$\sum_{\tau\in\mathcal{T}_o} y_{o\tau} = 1 \qquad \forall o \in O \tag{8}$$

$$\sum_{\tau\in\mathcal{T}} \tau \left(x_{i\tau} - y_{o\tau}\right) \leq 0 \qquad \forall (i,o) \in P \tag{9}$$

$$\sum_{j\in I\cup O} \sum_{u=\tau-p_j+1}^{\tau} x_{ju} \leq n \qquad \forall \tau \in \mathcal{T} \tag{10}$$

$$x_{i\tau} \in \{0,1\} \qquad \forall i \in I; \forall \tau \in \mathcal{T}_i \tag{11}$$

$$y_{o\tau} \in \{0,1\} \qquad \forall o \in O; \forall \tau \in \mathcal{T}_o \tag{12}$$

The objective function (6) minimizes the total weighted usage of the storage area. Constraints (7) and (8) demand each truck to be assigned to at least one gate. Constraints (9) ensure that if there exists a precedence relationship between inbound truck i and outbound truck o, then o cannot be processed before i. Constraints (10) enforce the capacity of the docks.

An alternative precedence constraint is the following:

$$\sum_{u=1}^{\tau} x_{iu} - y_{o\tau} \leq 0 \qquad \forall (i,o) \in P; \forall \tau \in \mathcal{T} \tag{13}$$

Christofides *et. al.* (1987) call this constraint *disaggregated*. The formulation obtained by replacing constraint (9) in formulation F1 by (13) will be referred to as F2. F2 is theoretically stronger since each feasible solution for formulation F2 is also a feasible solution to formulation F1. Artigues *et al.* (2008) observe, however, that the additional CPU time needed to solve the larger linear program can counterbalance the significant improvement of the bound. Both formulations will be tested empirically.

4 Branch and bound

We propose a branch-and-bound algorithm where at each node u of the search tree, an uncapacitated cross-docking problem is considered. We impose release times r^u and deadlines $\tilde{d}^u$ for the tasks. At the root 0 of the search tree, r_j^0 and $\tilde{d}_j^0$ are set to their initial values r_j and $\tilde{d}_j$, respectively.

In a first step, the problem defined at each node is solved. It corresponds to the linear program of minimizing (1) subject to (2)-(4). With minor modifications, the dual of this

problem is a max-cost flow problem, which can be solved in polynomial time. If no feasible solution is found, then the node can be fathomed. Otherwise, an integer solution is returned that gives us the local optimal objective value F^u as well as the starting times s_i^u and s_o^u.

In a second step, the returned solution is analyzed to identify a minimal time period $\Omega^u = [\underline{t}^u, \overline{t}^u]$, with its corresponding set of critical tasks σ^u, for which the gate capacity n is exceeded. If no such period is found then the LP solution is locally optimal for the capacitated problem and so, a feasible solution is found. We use *best* to represent the tightest upper bound found so far. For a feasible solution, the couple of tasks $(y, z) \in P$ with minimal w_{yz} is selected such that either y or z belongs to σ^u. If $y \in \sigma^u$, then two new nodes v and w are expanded from u by setting $r_y^v = \overline{t}_u - p_y + 1$ and $\tilde{d}_y^v = \tilde{d}_y^u$ ($r_y^w = r_y^u$ and $\tilde{d}_y^w = \overline{t}^u$), respectively. Similarly, if $z \in \sigma^u$, two new nodes v and w are also generated from u by setting $r_z^v = \underline{t}^u$ and $\tilde{d}_z^v = \tilde{d}_z^u$ ($r_z^w = r_z^u$ and $\tilde{d}_z^w = \underline{t}^u + p_z - 1$), respectively. Each new node is inserted on top of a stack for depth-first search. A node u can be fathomed either if $F^u \geq best$ or if one interval $[r_j^u, \tilde{d}_j^u]$ becomes smaller than p_j. Note also that, at each node, computing mandatory amounts of resource consumption per period can be easily performed (for more details, see Artigues *et al.* (2011)), so that infeasible nodes can be cut earlier in the search tree.

5 Conclusions and future work

We have presented a time-indexed (integer programming) formulation and a branch-and-bound method for the truck scheduling problem at cross-docking terminals. This work is still in progress, and we will compare both approaches in preliminary experimental results. Moreover, we will compare experimentally the mixed mode strategy with the exclusive one.

For future research, it may be interesting to investigate the special case of the problem with $p_i = p$. The complexity of $Pm|r_i, \tilde{d}_i, p_i = p| \sum w_i C_i$ is open (Kravchenko and Werner 2011) and this problem is a special case of our problem with $p_i = p$, so the complexity of our problem with $p_i = p$ is open as well. Another interesting problem is an extension in which trailers are allowed to remain at the gate longer as strictly needed for loading or unloading. In this way, the number of direct transfers from inbound to outbound trailers can be augmented and as such, the usage of the storage area can be decreased.

References

C. Artigues, S. Demassey and E. Néron, 2008, "Resource-Constrained Project Scheduling", *Control Systems, Robotics and Manufacturing Series*, ISTE Ltd.

C. Artigues, P. Lopez and A. Haït, 2011, "The energy scheduling problem: industrial case study and constraint propagation techniques", *International Journal of Production Economics*, in press.

N. Boysen and M. Fliedner, 2010, "Cross dock scheduling: Classification, literature review and research agenda", *Omega*, Vol. 38, pp. 413-422.

H. J. Carlo and Y. A. Bozer, 2011, "Analysis of optimum shape and door assignment problems in rectangular unit-load crossdocks", *International Journal of Logistics: Research and Applications*, Vol. 14 (3), pp. 149-163.

N. Christofides and R. Alvarez-Valdés and J. M. Tamarit, 2003, "Project scheduling with resource constraints: A branch-and-bound approach", *European Journal of Operational Research*, Vol. 29 (3), pp. 262-273.

S. A. Kravchenko and F. Werner, 2011, "Parallel machine problems with equal processing times: a survey", *Journal of Scheduling*, in press.