Jena Research Papers in Business and Economics

Scheduling Inbound and Outbound Trucks at Cross Docking Terminals

Nils Boysen, Malte Fliedner, Armin Scholl

17/2007

Jenaer Schriften zur Wirtschaftswissenschaft

Working and Discussion Paper Series
School of Economics and Business Administration
Friedrich-Schiller-University Jena

ISSN 1864-3108

Publisher:

Wirtschaftswissenschaftliche Fakultät
Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, D-07743 Jena

www.jbe.uni-jena.de

Editor:

Prof. Dr. Hans-Walter Lorenz
h.w.lorenz@wiwi.uni-jena.de

Prof. Dr. Armin Scholl
armin.scholl@wiwi.uni-jena.de

www.jbe.uni-jena.de

Scheduling Inbound and Outbound Trucks at Cross Docking Terminals

Nils Boysen[a,*], Malte Fliedner[a], Armin Scholl[b]

[a] *Universität Hamburg, Institut für Industrielles Management, Von-Melle-Park 5, D-20146 Hamburg,* {boysen,fliedner}@econ.uni-hamburg.de

[*] *Corresponding author, phone +49 40 42838-4640.*

[b] *Friedrich-Schiller-Universität Jena, Lehrstuhl für Betriebswirtschaftliche Entscheidungsanalyse, Carl-Zeiß-Straße 3, D-07743 Jena,* a.scholl@wiwi.uni-jena.de

Abstract

At cross docking terminals, shipments from inbound trucks are unloaded, sorted and moved to dispatch points where they are directly loaded onto outbound trucks for an immediate delivery elsewhere in the distribution system. This warehouse management concept aims at realizing economies in transportation cost by consolidating divergent shipments to full truckloads without requiring excessive inventory at the cross dock. The efficient operation of such a system requires an appropriate coordination of inbound and outbound trucks, e.g. by computerized scheduling procedures. This work introduces a base model for scheduling trucks at cross docking terminals, which relies on a set of simplifying assumptions in order to derive fundamental insights into the underlying problem's structure, i.e. its complexity, and to develop a building block solution procedure, which might be employed to solve more complex real-world truck scheduling problems.

Keywords: Logistics; Cross docking; Truck Scheduling; (Bounded) Dynamic Programming

1 Introduction

In contrast to traditional warehouses, a cross docking terminal is a distribution center carrying no (or at least a considerably reduced amount of) stock. Incoming shipments delivered by inbound trucks are unloaded, sorted and loaded onto outbound trucks waiting at the dock, which forward the shipments to the respective locations within the distribution system. Compared to traditional warehousing, a cost intensive storage and retrieval of goods is eliminated by a synchronization of inbound and outbound flows. An additional major advantage of cross docking is that economies in transportation cost can be realized by consolidating divergent shipments to full truckloads without depending on (enlarged) inventories at the cross dock (Apte and Viswanathan, 2000).

These advantages make cross docking an important logistic strategy receiving increased attention in today's globalized competition with its ever increasing volume of transported goods. Success stories about considerable competitive advantages realized due to the use of cross docking terminals are reported for many industries with high proportions of distribution cost like retail chains (Wal Mart; Stalk et al., 1992), mailing companies (UPS; Forger, 1995), automobile producers (Toyota; Witt, 1998) and less-than-truckload logistics providers (Gue, 1999).

A schematic representation of the material handling operations carried out at a cross docking terminal is depicted in Figure 1. Incoming trucks are either directly assigned to a receiving door upon arrival, or have to wait in a queue on the yard until they are assigned. Once docked, the products, i.e. pallets, packages or boxes, of an inbound trailer are unloaded and scanned to identify their respective destinations. Then, products are taken over by some means of conveyance. This might be a worker running a fork lift, e.g. in retail industries (Gue, 1999), or some kind of automated conveyor belt system, e.g. in mail distribution centers (McWilliams et al., 2005). The goods are forwarded to the designated shipping door, discharged in front of the outbound trailer and then loaded onto it. Once an outbound (inbound) trailer has been completely loaded (unloaded), it is removed from the dock, replaced by another trailer and the course of action repeats.

The efficiency of such a system depends critically on the appropriate coordination of inbound and outbound flows, e.g. by computerized scheduling procedures. The paper on hand deals with the truck scheduling problem, which generally comprises the assignment of each inbound and outbound truck to a door of the dock and of determining the schedule of all trucks assigned per door. Thus, the respective dispatcher faces two interrelated decisions: *where* and *when* the trucks should be processed at the terminal. This scheduling task has to consider additional constraints, because an outbound truck can be processed not until all required products have been delivered by their respective inbound trucks. Thus, to reduce the delay of shipments at the cross dock, a synchronization of inbound and outbound product flows becomes crucial. This general claim for synchronization can be operationalized by minimizing the total completion time of operations, which is also referred to as makespan in scheduling literature.

As the problem of truck scheduling is of considerable complexity, we reduce the problem to an "one inbound door serves one outbound door" setting in order to derive a base model. This will provide fundamental insights to the underlying structure of the

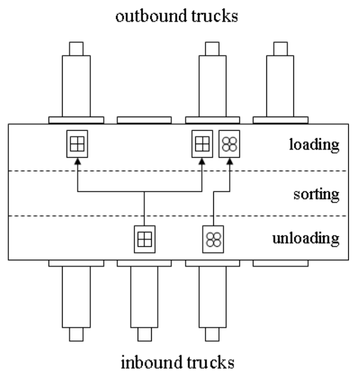

Figure 1: Material handling in a cross docking terminal

problem, which can hence be used to develop building block solution methods for more complex settings. The paper is organized as follows. After a brief literature review in Section 2, a base model of truck scheduling is introduced in Section 3 for which lower bounds are derived in Section 4. To solve the truck scheduling problem, we develop a decomposition approach in Section 5, which is based on a partial fixation of the problem. For a given inbound (outbound) sequence of trucks the (nearly) optimal outbound (inbound) sequence is determined by a suited Dynamic Programming approach as well as some heuristic procedures. Section 6 provides the results of a computational study and, finally, Section 7 discusses the application of the results to real-world cross docks, e.g., consisting of multiple inbound and outbound doors.

2 Literature review

Previous research on cross docking mainly investigates one of the following decision problems: (i) location of cross docks and other kinds of intermediate warehouses (e.g. see the summaries about hub location problems provided by Campbell, 1994; Klose and Drexl, 2005), (ii) layout of the dock (see Bartholdi and Gue, 2004), (iii) mid-term assignment of outbound destinations to dock doors (Tsui and Chang, 1990, 1992; Gue, 1999; Bartholdi and Gue, 2000; Bermudez and Cole, 2001; Oh et al., 2006).

Only very few research papers deal with the short-term scheduling problems arising during the daily operations of cross docking terminals. Li et al. (2004) consider material handling inside the terminal for a given truck schedule. Once a set of inbound and outbound trucks is docked, jobs consisting of products to be handled have to be assigned to recourses, i.e. workers and means of conveyance like fork lifts, in such a way that efficient unloading, sorting and loading operations render possible. Li et al. (2004) model this task as a machine scheduling problem and present a meta heuristic suited for its solution.

Thus far, only McWilliams et al. (2005) and Yu and Egbelu (2006) treat the truck scheduling problem. However, both previous research papers investigate detailed truck scheduling problems dedicated to quite special cross dock settings, so that their models cannot serve as a basic reference model, which might be generalizable to other cross dock settings. McWilliams et al. (2005) cover a specific truck scheduling problem at a parcel hub, which is solved by a simulation-based optimization approach. Yu and Egbelu (2006) treat a special kind of cross docking terminal with a conveyor belt system, where a single inbound door serves a single outbound door. The transportation of goods within the dock is modeled as a detailed scheduling problem, for which they present a priority rule based start heuristic.

Like them, we treat a stylized "one inbound door serves one outbound door" setting in order to generate fundamental insights to the underlying real-world problem structure. In contrast to Yu and Egbelu (2006), who model truck scheduling as a detailed scheduling problem, we, however, choose a more aggregate view as detailed handling times of products are in general hard to obtain. Exact handling times for inbound trailers, for instance, depend on the exact packing of goods and the sequence in which they can be obtained, whereas those for outbound trailers have to account for load stability and the sequence in which customers are served. Furthermore, the determination of transportation times between doors results to a complex optimization problem in itself (at least when the resources which handle products are scarce). Thus, handling times used in a detailed truck scheduling model are merely estimated average times and often bound to heavy inaccuracies. Under such prerequisites, detailed models may lead to more misleading or even infeasible plans when compared to aggregate models. Hence, we merge individual handling times for products to service slots to which inbound and outbound trucks are assigned. A slot comprises the time required for completely unloading an inbound truck and completely loading an outbound truck, respectively. Handling times in between dock doors are considered by a delay (measured in number of slots) which covers the time span until incoming products are available at an outbound door. This aggregate view seems to be sufficient to model the truck scheduling problem, especially in a medium-term horizon, when arrival and departure times of trucks are to be negotiated with logistics companies. However, we finally indicate that the same procedures can also be applied when detailed handling times are to be considered (see Section 7).

By a simultaneous scheduling of inbound and outbound trucks, incoming flows of products are harmonized with outbound flows, so that a Just-in-Time supply of products and, thus, a reduced turnover time is enabled.

3 Problem description and mathematical program

The basic truck scheduling problem considered within this paper determines service sequences for a set I of inbound trucks at the single receiving door and a set O of outbound trucks at the single shipping door of the cross docking terminal. Each inbound truck is loaded with units of different products $p \in P$. The number of units of type p contained in an inbound truck $i \in I$ is denoted by a_{ip}. All product units are completely unloaded within a service slot (period) t to which the respective inbound truck is assigned, so that all handling operations (e.g. docking, unloading, undocking) required to process the truck are executed within this time span, e.g., an hour or two. Moreover, all inbound trucks are assumed to be available for processing at the beginning of the planning horizon, so that a static problem with identical arrival dates of inbound trucks is considered.

The assumption of equidistant service slots can be seen as a reasonable approximation of reality, whenever vehicle capacities and the number of products per vehicle do not strongly differ. As trailers are typically of a standardized size and cross docking aims at moving only full truck loads, this premise is fulfilled whenever all processed products are of comparable size (e.g. mail distribution centers) or all truck loads resemble a representative average truck load (e.g. rotational deliveries of special promotional offers to all stores of a retail chain).

Once unloaded, the delivered products, typically, have to undergo a lot of operations in order to be available for being loaded onto the outbound trucks at the shipping door. These operations include recording of any product unit in the information system, examining the product correctness and quality, collecting, sorting, rearranging and packing to recombine products from different inbound trucks to form the load of a certain outbound truck. Finally, the products have to be transported (moved) to the shipping door, where they wait in an intermediate buffer of sufficient size until they are needed. This variety of tasks from recording to transporting usually takes considerably more time than loading/unloading a truck. Thus, we assume that this *movement time* takes a integer multiple m of the unit time span covered by a slot. Then, all products arriving in a slot t are available for loading at the shipping dock not before slot $t+m$ if the movement process can be started for any unloaded unit immediately, e.g. when applying a conveyor belt system. If the movement starts not before the complete inbound truck has been unloaded completely (e.g. a worker stacks all units behind the receiving door before moving them), the units are first available at slot $t+m+1$. However, the displacement m or $m+1$, respectively, can be ignored (set to zero) when modeling and solving the problem, because, after having determined a solution, an appropriate re-indexing of slots outbound trucks are assigned to (by adding the displacement) allows the exact determination of the outbound schedule.

Similarly to constant unloading times, it is assumed that the movement time m is independent of the inbound truck and the loaded products, because handling full truck loads, which may always consist of almost the same number of product units, should take very similar times. This assumption is realistic especially within an aggregated medium-term scheduling approach as proposed here.

At the shipping door, the set O of outbound trucks is to be loaded, each $o \in O$ with a predetermined number of units b_{op} of the different products $p \in P$. Again, it is assumed that all handling operations per truck are completed within a single slot. An outbound truck can be assigned to a slot t not before enough stock has accumulated in the intermediate buffer to serve all demanded product units of the truck. As only temporary stock is allowed (or desired) within a cross dock, it is assumed that temporary stock is empty before the first inbound truck arrives and is emptied out again after the last outbound truck was served. Hence, within our model the following premise holds: $\sum_{i \in I} a_{ip} = \sum_{o \in O} b_{op} \, \forall \, p \in P$.

In the following, the simplifying assumptions applied to our base model are summarized:

1. Inbound trucks are processed at a single receiving door of the terminal, which serves a single shipping door for outbound trucks. Both doors are distinct (segregated mode of service).
2. Equidistant service slots for processing trucks, i.e., each truck needs the same amount of time for (un-)loading.
3. No predefined restrictions on truck assignments to slots exist, e.g., due dates.
4. The input data is known in advance with certainty (static deterministic problem).
5. The movement time of products across the dock is a given constant and can, thus, be ignored.
6. The sum of units delivered by inbound trucks equals the sum of units consumed by outbound trucks for any product p (only intermediate stock).
7. Intermediate buffer for intermediate stock is not limited in size.

As a direct result of the simplifying assumptions the inbound and outbound schedule can be readily derived by the sequence of inbound and outbound trucks, so that the problem reduces to a truck sequencing problem (TRSP). The objective is to sequence the trucks in such a way that the makespan is minimized, which comprises the time span starting from the first slot to which an inbound truck is assigned and lasts until the final slot in which an outbound truck is processed. The notation used to formalize the TRSP is summarized in Table 1.

I	set of inbound trucks (index i)
O	set of outbound trucks (index o)
T	(maximal) number of time slots available for (un-)loading trucks (index t)
P	set of products (index p)
a_{ip}	quantity of product type p arriving in inbound truck i
b_{op}	quantity of product type p to be loaded onto outbound truck o
m	movement time of products across the dock (w.l.o.g., $m = 0$)
x_{it}	binary variable: 1, if inbound truck i is assigned to slot t; 0, otherwise
y_{ot}	binary variable: 1, if outbound truck o is assigned to slot t; 0, otherwise

Table 1: Notation

The problem consists of objective function (1) and constraints (2)-(8).

$$\text{(TRSP) Minimize } C(X,Y) = \max_{o \in O; t=1,\dots,T} \{y_{ot} \cdot t\} \tag{1}$$

subject to

$$\sum_{t=1}^{T} x_{it} = 1 \qquad \forall\, i \in I \tag{2}$$

$$\sum_{i \in I} x_{it} \leq 1 \qquad \forall\, t = 1, \ldots, T \tag{3}$$

$$\sum_{t=1}^{T} y_{ot} = 1 \qquad \forall\, o \in O \tag{4}$$

$$\sum_{o \in O} y_{ot} \leq 1 \qquad \forall\, t = 1, \ldots, T \tag{5}$$

$$\sum_{\tau=1}^{t} \sum_{i \in I} x_{i\tau} \cdot a_{ip} \geq \sum_{\tau=1}^{t} \sum_{o \in O} y_{o\tau} \cdot b_{op} \qquad \forall\, t = 1, \ldots, T;\; p \in P \tag{6}$$

$$x_{it} \in \{0, 1\} \qquad \forall\, i \in I;\; t = 1, \ldots, T \tag{7}$$

$$y_{ot} \in \{0, 1\} \qquad \forall\, o \in O;\; t = 1, \ldots, T \tag{8}$$

Objective function (1) minimizes the makespan, which is equal to the slot number to which the last outbound truck is assigned. Note that the objective function can easily be linearized by introducing an additional variable bounding all assigned slots from above. Equations (2) ensure that each inbound truck is processed in exactly one slot, whereas constraints (3) enforce that in each slot at most one inbound truck can be assigned. In analogy, these two conditions also hold for outbound trucks by constraints (4) and (5). Constraints (6) ensure that an outbound truck can only be assigned to a slot t, whenever all required products are available (delivered by preceding inbound trucks yet not consumed by preceding outbound trucks) to satisfy the demand for product units of each type p. Therefore, the available stock accumulated by all inbound trucks assigned to slots $\tau = 1, \ldots, t$ has to exceed the total demand for product units of outbound trucks scheduled up to the actual slot t (remember that this will actually be slot $t + m$ or even $t + m + 1$ when realizing the schedule).

As the makespan is to be minimized, the number of service slots actually required is unknown prior to the solution of the model. Thus, within the model the number of slots T is to be initialized with some upper bound $\overline{C}$ on the makespan: $T = \overline{C}$. A simple upper bound is given by equation (9):

$$\overline{C} = |I| + |O| - 1 \tag{9}$$

This bound is based on the consideration that in the worst case the first outbound truck scheduled requires a product loaded on the last inbound truck scheduled. Consequently, all outbound trucks have to wait until all inbound trucks are unloaded.

To tighten the model formulation, e.g. when using a generic MIP-solver, the following property of optimal inbound schedules can be utilized:

Left-shift property: It is sufficient to reduce the set of time slots considered for an assignment of inbound trucks to the first $|I|$ slots. This is obviously correct, because if there exists an optimal solution, where inbound trucks are not assigned to slots $t = 1, \ldots, |I|$ in direct succession, then trucks can be brought forward (without altering the sequence) and the objective value remains the same. Thus, there is always at least one optimal solution where inbound trucks are assigned to the first $|I|$ slots.

With this property on hand, the number of variables and constraints can be reduced. As alterations are truly straight forward we abstain from explicitly recording them. Furthermore, the following proposition with respect to the TRSP holds.

Proposition: The TRSP is NP-hard in the strong sense.

Proof: See Appendix A.

4 Lower bounds

The first simple lower bound $\underline{C}^1$ reverses the logic of our upper bound. In the best case, each outbound truck has a direct counterpart among the inbound trucks, so that inbound and outbound trucks can be scheduled successively without any delay:

$$\underline{C}^1 = \max\{|I|; |O|\} \tag{10}$$

For the computation of another lower bound $\underline{C}^2$, the overall problem is decomposed in $|P|$ subproblems by cutting off the truck coherency of products. For each product the minimum makespan is deduced by separately scheduling inbound and outbound trucks. Thus, it is relaxed that for each product the same truck sequence has to be maintained.

The optimal solution for each subproblem can be determined by considering the following simple rules, which share some similarities with those of the famous Johnson algorithm for two-machine flow shop scheduling (Johnson, 1954):

- Sort the set I of inbound trucks with respect to descending loads a_{ip} of the product p actually considered. This leads to a sequence vector π^p with elements π_i^p ($i = 1, \ldots, |I|$). The schedule for this sequence vector is readily available because of the left-shift property: Inbound trucks are scheduled according to sequencing vector π^p in direct succession starting with slot $t = 1$.
- Sort the set O of outbound trucks with respect to ascending loads b_{op} of the actual product p. This sequence is stored in the vector μ^p with elements μ_o^p ($o = 1, \ldots, |O|$). The resulting slots s_{po} can be computed by assigning, in each case, the first feasible slot number t according to:

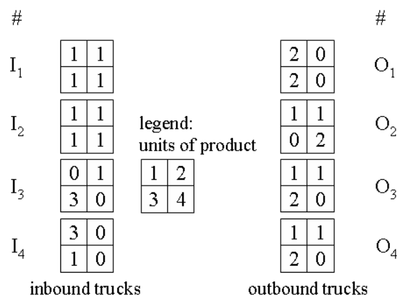

Figure 2: Example data

$$s_{op} = \min \left\{ t = s_{o-1p} + 1, \ldots, T \;\middle|\; \sum_{\tau=1}^{\min\{|I|;t\}} a_{\pi_\tau^p p} \geq \sum_{\tau=1}^{o} b_{\mu_\tau^p p} \right\} \quad \forall\, o \in O;\, p \in P \tag{11}$$

To initialize the recursive formulae (11), a slot μ_0^p has to be initialized with zero stock and slot number 0.

The lower bound $\underline{C}^2$ then simply amounts to the maximum makespan over all products

$$\underline{C}^2 = \max_{p \in P} \left\{ s_{|O|p} \right\} \tag{12}$$

Example: The computation of both bounds is to be clarified by an example with the data displayed in Figure 2. Lower bound $\underline{C}^1$ amounts to: $\underline{C}^1 = \max\{4; 4\} = 4$. The computation of bound $\underline{C}^2$ is shown in Table 2 and amounts to $\underline{C}^2 = \max\{4; 4; 4; 4\} = 4$.

	t	1	2	3	4	o	1	2	3	4	
$p = 1$	π^1	I_4	I_1	I_2	I_3	π^1	O_2	O_3	O_4	O_1	
	$\sum_{\tau=1}^{t} a_{\pi_\tau^1 1}$	3	4	5	5	$\sum_{\tau=1}^{o} b_{\mu_\tau^1 1}$	1	2	3	5	$\Rightarrow s_{\lvert O\rvert 1} = 4$
$p = 2$	π^2	I_1	I_2	I_3	I_4	μ^2	O_1	O_2	O_3	O_4	
	$\sum_{\tau=1}^{t} a_{\pi_\tau^2 2}$	1	2	3	3	$\sum_{\tau=1}^{o} b_{\mu_\tau^2 2}$	0	1	2	3	$\Rightarrow s_{\lvert O\rvert 2} = 4$
$p = 3$	π^3	I_3	I_1	I_2	I_4	π^3	O_2	O_1	O_3	O_4	
	$\sum_{\tau=1}^{t} a_{\pi_\tau^3 3}$	3	4	5	6	$\sum_{\tau=1}^{o} b_{\mu_\tau^3 3}$	0	2	4	6	$\Rightarrow s_{\lvert O\rvert 3} = 4$
$p = 4$	π^4	I_1	I_2	I_3	I_4	π^4	O_1	O_3	O_4	O_2	
	$\sum_{\tau=1}^{t} a_{\pi_\tau^4 4}$	1	2	2	2	$\sum_{\tau=1}^{o} b_{\mu_\tau^4 4}$	0	0	0	2	$\Rightarrow s_{\lvert O\rvert 4} = 4$

Table 2: Computation of bound $\underline{C}^2$ for the example

Lower bound $\underline{C}^2$ has a runtime complexity of $O(n \log n)$, where $n = \max\{|I|; |O|\}$, due to the sorting operations, which is considerably higher than that of $\underline{C}^1$ ($O(1)$). However, it can be shown that both bounds in any case lead to the same result. Thus, the following proposition with respect to the bounds presented holds:

Proposition: $\overline{C} \geq C^{min} \geq \underline{C}^2 = \underline{C}^1$.

Proof: See Appendix B.

However, if either one of the truck sequences is fixed, as is the case in the presented decomposition approach, lower bound $\underline{C}^2$ can very well result to a tighter bound compared to $\underline{C}^1$. This can be easily shown by an example and results directly from the fact that within the fixed truck sequence trucks are not necessarily sorted according to product loads.

5 A decomposition approach for the TRSP

In the following, we present algorithms for the TRSP, which are based on a decomposition of the overall problem. It is divided into subproblems by fixing a particular inbound (outbound) sequence and then finding the optimal outbound (inbound) sequence, respectively. In the following section we will formalize the considered subproblems, thereby identifying a strong structural relationship, which can be exploited in the solution procedure. Building on that Section 5.2 introduces exact and heuristic solution procedures for both subproblems, while Section 5.3 discusses control procedures, which can be employed to guide the repeated generation and solution of subproblems, in order to solve instances of TRSP.

5.1 Formalization of Subproblems

In the first problem let us assume that there is a fixed sequence π of inbound trucks given, so that the inbound schedule can immediately be deduced by assigning the trucks in the repective order to the first $|I|$ slots (see left-shift property). Thus, the number A_{tp} of product units available for outbound trucks is known in advance for each slot t: $A_{tp} = \sum_{\tau=1}^{\min\{|I|;t\}} a_{\pi_\tau p} \, \forall \, p \in P;\, t = 1, \ldots, T$. With the available stock (cumulated input) on hand, the problem OUTBOUND-TRSP reduces to objective function (1) subject to constraints (4),(5),(8) and (13):

$$A_{tp} \geq \sum_{\tau=1}^{t} \sum_{o \in O} y_{o\tau} \cdot b_{op} \quad \forall \, t = 1, \ldots, T;\, p \in P \tag{13}$$

Note that already this subproblem is NP-hard in the strong sense (see Appendix A). To solve OUTBOUND-TRSP an exact Dynamic Programming approach (Section 5.2.1) and a priority rule based heuristic start procedure (Section 5.2.2) are proposed.

Conversely, we can also fix the sequence μ of outbound trucks starting from period T to earlier ones and determine the optimal sequence of inbound trucks, respectively. This, however, necessitates additional modifications to the mathematical model. Note that in the original model, objective function (1) was defined in such a way, that it minimizes the index number of the service slot to which the last outbound truck in the sequence is assigned. As the outbound sequence is now not variable anymore, the objective function needs some adjustment. Makespan minimization can be readily expressed in terms of the inbound sequence by maximizing the first service slot to which any inbound truck is assigned. Let this slot in an optimal solution be denoted by t^*, then the first t^*-1 slots, to which no inbound trucks are assigned, can be discarded and the minimum makespan equals $T-t^*+1$. It can be easily verified that both objectives lead to the same optimal inbound and outbound sequences.

Instead of defining INBOUND-TRSP as a maximization problem, we, however, take a slightly different approach which reveals an interesting relationship between INBOUND- and OUTBOUND-TRSP. Recall that in the original model, index t denotes the index number of a service slot in ascending order, so that a lower number indicates that the service slot is processed prior to a slot with a higher number, which is an intuitive representation of time. For INBOUND-TRSP we will change the point of reference and introduce a new time index $j=1,\ldots,T$, where $j=1$ refers to the service slot to which the last outbound truck is assigned and an increase in j denotes a movement backwards in time until service slot T, which now constitutes the earliest point in time to which any inbound or outbound truck could be assigned. INBOUND-TRSP can now be stated as follows:

$$\text{(INBOUND-TRSP) Minimize } C(X)=\max_{o\in O;j=1,\ldots,T}\{x_{oj}\cdot j\} \tag{14}$$

subject to

$$\sum_{j=1}^{T} x_{ij}=1 \qquad \forall\, i\in I \tag{15}$$

$$\sum_{i\in I} x_{ij}\le 1 \qquad \forall\, j=1,\ldots,T \tag{16}$$

$$M_p-\left(\sum_{\tau=1}^{j}\sum_{i\in I} x_{i\tau}\cdot a_{ip}\right)\ge M_p-B_{jp} \qquad \forall\, j=1,\ldots,T;\, p\in P \tag{17}$$

$$x_{ij}\in\{0,1\} \qquad \forall\, i\in I;\, j=1,\ldots,T \tag{18}$$

Objective function (14) minimizes the makespan in terms of the inbound sequence by minimizing the number of service slots between the last outbound and first inbound truck assigned. Constraint (17) ensures that inbound trucks deliver product units in the required quantities, where $B_{jp}=\sum_{\tau=1}^{\min\{|O|;j\}} b_{\mu_{T-\tau+1}p}\,\forall\, p\in P;\, j=1,\ldots,T$ denotes the total number of units of product p demanded by the last j outbound trucks in the fixed sequence μ and $M_p=\sum_{i\in I} a_{ip}=\sum_{o\in O} b_{op}$ is the total number of parts delivered/demanded. Constraints (15), (16) and (18) are simply modified according to the new time index j. Note that constraint (17) can be rewritten as follows:

$$B_{jp}\ge\sum_{\tau=1}^{j}\sum_{i\in I} x_{i\tau}\cdot a_{ip} \qquad \forall\, j=1,\ldots,T;\, p\in P \tag{19}$$

A comparison of OUTBOUND-TRSP and INBOUND-TRSP as formulated above, now reveals that their mathematical structure is exactly identical. As a consequence any algorithm for OUTBOUND-TRSP can be used to solve INBOUND-TRSP and vice versa. In fact INBOUND-TRSP can be seen as a reverted OUTBOUND-TRSP, in the sense that the solution of an instance of INBOUND-TRSP with an algorithm designed for OUTBOUND-TRSP requires the following steps:

(1) Revert the given outbound sequence μ and set it as the modified inbound sequence π. Change the set of inbound trucks I to be scheduled to the modified set of outbound vehicles O.

(2) Solve OUTBOUND-TRSP with the modified input data.

(3) The reverted optimal outbound sequence constitutes the optimal inbound sequence for the original INBOUND-TRSP instance.

5.2 Algorithms for subproblems

In the following we will introduce an exact Dynamic Programming approach (Section 5.2.1) and a priority rule based heuristic start procedure (Section 5.2.2) to solve the identified subproblems. The algorithmic descriptions are limited to OUTBOUND-TRSP, as they are directly transferable to INBOUND-TRSP as explained above.

5.2.1 Dynamic Programming approach

The Dynamic Programming (DP) approach to solve OUTBOUND-TRSP is based on an acyclic digraph $G = (V, E, r)$ with a node set V divided into $|O| + 1$ *stages*, a set E of arcs connecting nodes of adjacent stages and a node weighting function $r : V \to \mathbb{N}$ (see Bautista et al., 1996; Boysen et al., 2007, for related approaches to scheduling mixed-model assembly lines). Each sequence position σ is represented by a stage which contains a subset $V_\sigma \subset V$ of nodes representing *states* of the partial outbound schedule up to sequence position σ. Additionally, a start level 0 is introduced. Each index $j \in V_\sigma$ identifies a state (σ, j) defined by the vector $\mathbf{Y}_{\sigma j}$ of binary indicators $Y_{\sigma j o}$ of all outbound trucks $o \in O$ already scheduled up to sequence position σ. It is sufficient to store the set of scheduled trucks instead of their exact partial sequence, because actual stock of products available to schedule an outbound truck at sequence position $\sigma + 1$ only depends on the trucks scheduled up to position σ irrespective of their order and the truck scheduled in $\sigma + 1$.

The following conditions define all *feasible* states to be represented as nodes of the graph:

$$\sum_{o \in O} Y_{\sigma j o} = \sigma \qquad \forall\, \sigma = 0, \ldots, |O|;\ j \in V_\sigma \tag{20}$$

$$Y_{\sigma j o} \in \{0, 1\} \qquad \forall\, o \in O;\ \sigma = 0, \ldots, |O|;\ j \in V_\sigma \tag{21}$$

Obviously, the node set V_0 contains only a single node (initial state $(0,1)$) corresponding to the vector $\mathbf{Y}_{01} = [0,0,\ldots,0]$. Similarly, the node set $V_{|O|}$ contains a single node (final state $(|O|,1)$) with $\mathbf{Y}_{|O|1} = [1,1,\ldots,1]$. The remaining stages σ have a variable number of nodes depending on the number of different truck vectors $\mathbf{Y}_{\sigma j}$ possible.

Two nodes (σ,j) and $(\sigma+1,k)$ of two consecutive stages σ and $\sigma+1$ are connected by an arc if the associated vectors $\mathbf{Y}_{\sigma j}$ and $\mathbf{Y}_{\sigma+1k}$ differ only in one element, i.e., exactly one outbound truck is additionally scheduled in position $\sigma+1$. This is true if $Y_{\sigma jo} \leq Y_{\sigma+1ko}$ holds for all $o \in O$, because both states are feasible according to (20) and (21). The overall arc set is defined as follows:

$$E = \{((\sigma,j),(\sigma+1,k)) \mid \sigma = 0,\ldots,|O|-1; j \in V_\sigma; k \in V_{\sigma+1} \text{ and } Y_{\sigma jo} \leq Y_{\sigma+1ko}\ \forall o \in O\} \tag{22}$$

With these arcs on hand, the set $P_{\sigma j}$ of direct predecessor states (nodes) of each state (node) (σ,j) can be determined.

Finally, node weights $r_{\sigma j}$ assign the minimum makespan of the partial schedule presented by state (σ,j). For this purpose, cumulative demands $B_{\sigma jp}$ for all products p are to be compared with cumulative product units delivered by inbound trucks. Cumulative demands $B_{\sigma jp}$ induced by a state (σ,j) for all products p are derived as follows:

$$B_{\sigma jp} = \sum_{o \in O} Y_{\sigma jo} \cdot b_{op} \qquad \forall\, p \in P \tag{23}$$

With the help of these cumulative demands, all node weights $r_{\sigma j}$ (except for the source node's weight, which is to be set to $r_{01} = 0$) can be determined as follows:

$$r_{\sigma j} = \max\left\{\min_{(\tau,k)\in P_{\sigma j}}\{r_{\tau,k}\} + 1;\ \min\{\tau = 1,\ldots,|I|\,|A_{\tau p} \geq B_{\sigma jp}\,\forall p \in P\}\right\} \quad \forall\, \sigma = 1,\ldots,|O|;\ j \in V_\sigma \tag{24}$$

The slot for state (σ,j) can be no smaller than the earliest scheduling time of any predecessor state (τ,k) plus one slot for the additional truck (first term of the maximum function). Furthermore, it has to be ensured that enough product units have accumulated over time by inbound trucks (second term of the maximum function).

With this graph on hand, the minimum makespan C^{min} corresponds to the node weight of the sink node, i.e., $C^{min} = r_{|O|1}$, whereas the determination of the optimal outbound sequence for OUTBOUND-TRSP reduces to finding the shortest path from the unique source node at level 0 to the unique sink node at level $|O|$. The optimal sequence μ of outbound trucks can be deduced by considering each arc $((\sigma,j),(\sigma+1,k))$ with $\sigma = 0,...,|O|-1$ on a shortest path SP. The outbound truck to be assigned at sequence position $\sigma+1$ is the only $o \in O$ for which $Y_{\sigma+1ko} - Y_{\sigma jo} = 1$ holds.

Example (cont.): The given inbound sequence π is assumed to be $\pi = \{I_1, I_2, I_3, I_4\}$. The resulting graph along with a (bold-faced) shortest path for our example is depicted in Figure 3. This path corresponds to the optimal outbound truck sequence

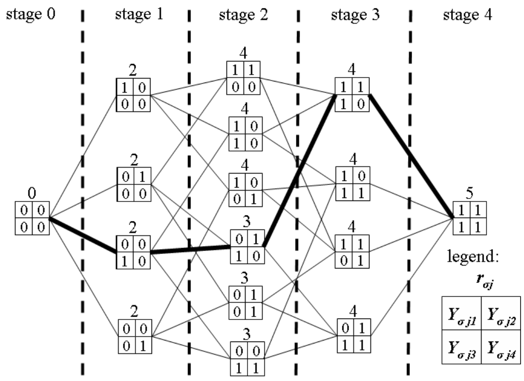

Figure 3: Example graph of DP for OUTBOUND-TRSP

$\mu = \{O_3, O_2, O_1, O_4\}$ with a minimum makespan of $C^{min} = r_{|O|1} = 5$.

Instead of constructing the complete graph before computing the shortest path, the more efficient DP approach consists of determining the shortest path from the initial state to each node stage-by-stage ($\sigma = 0, ..., |O|-1$). In order to do so, only two complete stages of the graph have to be stored simultaneously, because the shortest path to a node $(\sigma+1, k)$ in stage $\sigma + 1$ is composed of a shortest path to a node (σ, j) in stage σ (already determined and stored) and the connecting arc $((\sigma, j), (\sigma+1, k))$. Among all predecessor states of $(\sigma + 1, k)$ one with minimal node weight is to be selected and stored as predecessor in the shortest path. After reaching the final state $(|O|, 1)$ in stage $|O|$, the optimal path can be retrieved in backward direction stage-by-stage using the stored predecessor nodes.

To further speed-up the procedure the idea of *Bounded Dynamic Programming* (BDP) (e.g. Morin and Marsten, 1976; Marsten and Morin, 1978; Carraway and Schmidt, 1991; Bautista et al., 1996; Boysen et al., 2007) is employed, which reduces the number of states to be constructed. BDP extends the DP approach explained above by additionally computing a lower bound $\underline{C}(\sigma, j)$ on the makespan of scheduling remaining outbound trucks. Furthermore, a global upper bound $\overline{C}$ is determined upfront by some heuristic procedure(s). Whenever $r_{\sigma j} + \underline{C}(\sigma, j) \geq \overline{C}$, the node (σ, j) can be fathomed as it can not be part of a solution with a better objective value than the incumbent solution.

To derive a lower bound both approaches presented in Section 4 can be applied: $\underline{C} = \max\{\underline{C}^1, \underline{C}^2\}$. Lower bound $\underline{C}^1$ amounts to the maximum number of remaining inbound or outbound trucks: $\underline{C}^1(\sigma, j) = \max\{|I| - r_{\sigma j}; |O| - \sigma\}$. In order to determine lower bound $\underline{C}^2$ the remaining product demands b_{op} for each type p are considered individually and sorted in ascending order, respectively. Then, starting with the actual makespan $r_{\sigma j}$ a part-wise scheduling according to the given inbound sequence μ yields lower bound $\underline{C}^2(\sigma, j)$.

Example (cont.): Assume that an upper bound of $\overline{C} = 6$ has been calculated by some heuristic procedure prior to generating the graph. In this case, node (2,1) can be

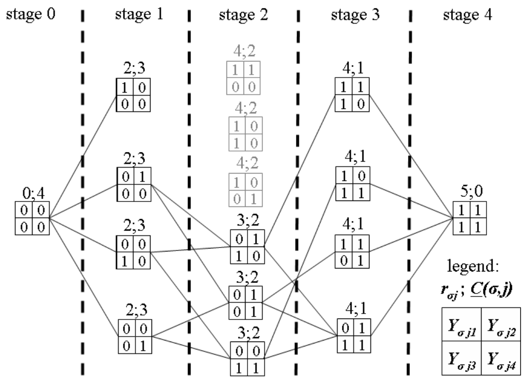

Figure 4: Example graph of the BDP for OUTBOUND-TRSP

fathomed, because the sum of node weight $r_{21} = 4$ and the lower bound $\underline{C}^1(2,1) = \max\{4-4; 4-2\} = 2$ is not lower than the upper bound. The overall reduction in the number of nodes resulting from the incorporation of lower bound $\underline{C}^1$ is depicted in Figure 4. Fathomed nodes are colored light grey.

5.2.2 Heuristic start procedure

As an alternative for the exact BDP approach a heuristic start procedure (HSP) can be applied to solve OUTBOUND-TRSP or, at least, to derive an initial upper bound. Such a method simply fills the solution vector μ of outbound trucks from left to right by fixing an unscheduled outbound truck o at the actual decision point τ. The actual choice is guided by some priority function $f(o)$. The procedure takes the following steps:

(1) Initialize data: $O^{out} := O$; $t := 1$; $\tau := 1$

(2) Determine the set POS_t of possible outbound trucks for actual slot t:

$$POS_t = \left\{ o \in O^{out} \middle| A_{tp} \geq \sum_{k=1}^{t-1} b_{\pi_k p} + b_{op} \, \forall \, p \in P \right\} \tag{25}$$

Thus, an outbound truck o belongs to set POS_t of possible trucks at the actual slot t, if it is not yet scheduled at preceding sequence positions and enough product units are actually available to fulfill the truck's product demand.

(3) Choose an outbound truck out of set POS_t to be fixed at actual sequence position τ with regard to some priority value $f(o)$:

$$\mu_\tau = \text{argmax}_{o \in POS_t} \{f(o)\} \tag{26}$$

(4) If $\tau = |O|$ (sequence of outbound trucks is completed), then terminate the procedure and return the makespan $C = t$.

(5) Decide on the advancement of the procedure:

- If POS_t was empty, so that no truck could be chosen for actual sequence position τ within step (3), then set $t := t + 1$ and proceed with step (2).
- If an outbound truck was chosen for the actual sequence position, then set $O^{out} := O^{out} \setminus \{\mu_\tau\}$; $\tau := \tau + 1$; $t := t + 1$ and proceed with step (2).

In the DP-graph, HSP equals a follow-up of that respective arc which leads to a connected node with the least node weight r_{tj} in each stage. The priority rule is used to break ties if more than one arc lead to the same minimum node weight.

Different kinds of priority functions $f(o)$ are possible to guide the search process, which can be further distinguished into static, i.e. independent of choices at preceding sequence positions, and dynamic ones, i.e. priority values may vary from decision point to decision point as they are dependent of preceding choices. A selection of reasonable priority rules is listed in Table 3. To simplify the representation of dynamic priority rules, we denote the number of stocked units of type p in dependency of the actual slot t considered with $R_p(\cdot)$, when outbound truck o is assigned at the actual decision point τ of sequence π: $R_p(\cdot) = A_{tp} - \sum_{t'=1}^{\tau} b_{\pi_{t'}p} - b_{op}$.

name	description	formula	static vs. dynamic rule
LPU	Least number of product units	$f(o) = \frac{1}{\sum_{p \in P} b_{op}}$	static
LFV	Least fraction of total product volume	$f(o) = \frac{1}{\sum_{p \in P} \frac{b_{op}}{\sum_{\sigma \in O} b_{\sigma p}}}$	static
LMAX	Least maximum product volume	$f(o) = \frac{1}{\max_{p \in P}\{b_{op}\}}$	static
MRS	Maximum remaining stock	$f(o) = \sum_{p \in P} R_p(\cdot)$	dynamic
MMRS	Maximum minimum remaining stock	$f(o) = \min_{p \in P}\{R_p(\cdot)\}$	dynamic

Table 3: Priority rules for OUTBOUND-TRSP

Example (cont.): When applying the LFV rule to our example, the priority values of outbound trucks amount to: $f(1) = 1.364, f(2) = 0.652, f(3) = f(4) = 1.154$. To break ties between trucks 3 and 4 the least truck number is chosen, so that HSP leads to the outbound truck sequence $\mu = \{O_1, O_3, O_4, O_2\}$ and a resulting makespan of $C = 6$.

5.3 Decomposition approaches to TRSP

Now, the unique solution procedure for the subproblems (OUTBOUND- and INBOUND-TRSP) is to be applied to solve the original TRSP, where both inbound and outbound sequences are to be determined simultaneously. In the following we present a decomposition approach where both sub problems are solved by our exact BDP-approach or the heuristic HSP in an alternating manner.

To do so, merely an initial inbound or outbound truck sequence and an appropriate stop criterion is to be determined. As the initial truck sequence we choose a randomly generated inbound sequence. Although other alternatives are possible, only random sequences are evaluated, because preliminary tests revealed the initial sequence be of minor importance for the solution performance.

To systematically test the convergence properties of our decomposition approach, three different stopping criteria are applied:

SC-1: Immediately stop after completing the solution for TRSP, which means that only a single OUTBOUND-TRSP is solved for the initial inbound sequence.

SC-2: Stop when the solution value is not improved anymore by the latest sequence computed.

SC-3: Stop when the identical solution of the previous cycle is rebuilt.

With these choices on hand the decomposition approach can be applied either by utilizing the BDP-approach or the HSP to alternately solve the subproblems.

6 Computational study

In the following, the performance of the proposed solution procedures is evaluated by a computational study. As no established test bed is available, we will start out with explaining the systematic generation of test instances.

6.1 Instance generation

In the study we distinguish between two classes of test instances: small and large sized instances. The small instances are designed such that optimal solutions for the overall TRSP can be obtained. These optimal solutions are determined by coupling a complete enumeration of all possible inbound sequences (first stage) with the BDP-approach for OUTBOUND-TRSP (second stage), which determines the minimal makespan for all of these inbound sequences. This way, all possible inbound sequences are evaluated, so that the least makespan found for these sequences is the minimal makespan of the overall TRSP. With these optimal solutions on hand the performance of the decomposition approach for the TRSP in its different configurations can be evaluated. Large sized instances are applied to evaluate the procedures for the subproblems (OUTBOUND-TRSP and INBOUND-TRSP). The parameters listed in Table 4 are systematically varied to generate the sets of inbound and outbound trucks along with their load coefficients a_{ip} and b_{op}.

Within each test case, given parameters are combined in a full-factorial design and for each parameter constellation instance generation is repeated 10 times, so that $6 \cdot 6 \cdot 3 \cdot 10 = 1080$ different TRSP instances are obtained per class. On the basis of a given set of parameters each single instance is generated as follows.

For each product $p \in P$ repeat the following steps to generate inbound and outbound trucks:

symbol	description	values small	values large
$\|I\|$	number of inbound trucks	3, 4,..., 8	13, 14,..., 18
$\|O\|$	number of outbound trucks	3, 4,..., 8	13, 14,..., 18
$\|P\|$	number of products	3, 5, 7	
SPU	sum of product units in all inbound and in all outbound trucks	1000	

Table 4: Parameters for instance generation

- Randomly choose an ordered set V_p of inbound (outbound) trucks containing product p, whose number of elements is $|V_p|$, where $|V_p|$ is determined by an equally distributed integer random number out of the interval $[1; |I|]$ or $[1; |O|]$, respectively.
- Randomly draw an equally distributed real number rnd_p out of the interval $[0; 1]$ for each truck in set V_p. Proportionally to these random numbers, the number SPU of units per product type p is partitioned within truck set V_p. To avoid rounding errors we distinguish between the ordered set V_p, which contains all randomly chosen trucks, and V_p^-, which is a copy of V_p missing its last element:

$$a_{ip} = \begin{cases} \left\lfloor \frac{rnd_p \cdot SPU}{\sum_{p' \in V_p} rnd_{p'}} \right\rfloor, & \forall \, p \in V_p^- \\ SPU - \sum_{p' \in V_p^-} a_{ip}, & p \in V_p \setminus V_p^- \end{cases} \tag{27}$$

  When outbound trucks are generated instead of inbound trucks, load coefficients a_{ip} have to be replaced by b_{op} in Equation (27).

If this procedure leads to empty inbound ($\sum_{p \in P} a_{ip} = 0$) or outbound ($\sum_{p \in P} b_{op} = 0$) trucks, the whole problem instance is discarded and instance generation for the respective parameter constellation is repeated until an instance with desired properties is obtained. All generated instances can be downloaded from the internet (www.ibl-unihh.de\tea_boy.htm).

6.2 Performance of solution procedures

Our computational study evaluates the performance of the algorithms for the subproblems (OUTBOUND-TRSP and INBOUND-TRSP) and the decomposition approach for the overall TRSP. The evaluation criteria applied to compare the performance of algorithms are listed in Table 5. All methods have been implemented in Visual Basic.NET (Visual Studio 2003) and run on a Pentium IV, 1800 MHz PC, with 512 MB of memory.

measure	description
# inst.	Number of benchmark instances evaluated
# opt.	Number of optimal solutions (or objective values)
⊘ rel. dev.	Average relative deviation (from optimum or best solution available) in %
max. rel. dev.	Maximum relative deviation (from optimum or best solution available) in %
⊘ abs. dev.	Average absolute deviation (from optimum or best solution available) in %
max. abs. dev.	Maximum absolute deviation (from optimum or best solution available) in %
⊘ sub.	Average number of solved subproblems
max. sub.	Maximum number of solved subproblems
⊘ cpu-sec.	Average computation time in seconds

Table 5: Evaluation criteria

	BDP	$\underline{C}^1$	$\underline{C}^2$	LPU	LFV	LMAX	MRS	MMRS
# inst.	1080	1070	1070	1070	1070	1070	1070	1070
# opt.	1070	233	585	948	948	933	947	949
⊘ rel. dev.	0.0	10.8	3.2	0.7	0.7	0.8	0.7	0.7
max. rel. dev.	0.0	46.4	20.0	13.0	13.0	13.0	13.0	23.5
⊘ abs. dev.	0.0	2.26	0.59	0.12	0.12	0.14	0.12	0.13
max. abs. dev.	0.0	14	4	3	3	3	3	4
⊘ cpu-sec.	6.3	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

Table 6: Results for OUTBOUND-TRSP

6.2.1 Performance of algorithms for the subproblems

First, algorithms dedicated to solve OUTBOUND-TRSP, where outbound sequences are sought for given inbound sequences, are examined. The instances evaluated are obtained from the large sized test instances by fixing the inbound sequence according to the respective truck numbers (in ascending order). Average results over all medium sized instances are summarized by Table 6.

As can be seen, the BDP approach solves 1070 out of 1080 instances to optimality within a given time frame of 300 CPU-seconds per instance (with an average of only 6.3 CPU-seconds). As all of the 10 unsolved instances occur when $|O| = 18$ this can be interpreted as an upper limit up to which the BDP-approach can be reasonably applied.

To examine the heuristic approaches on a solid basis, we only consider the 1070 instances for which optimal solutions are known through BDP. The HSP approach performs surprisingly well independent of the priority rule applied. However, the priority rule LMAX seems slightly inferior in comparison to the others. The other priority rules (LPU, LFV, MRS, MMRS) solve up to 949 instances (89%) to optimality, which leads to an average relative deviation (measured by $\frac{C(HSP)-C(BDP)}{C(BDP)}$) of just 0.7%. Note that LPU and LFV in our test always result to the same solution, as the total product vol-

	BDP	$\underline{C}^1$	$\underline{C}^2$	LPU	LFV	LMAX	MRS	MMRS
# inst.	1080	1068	1068	1068	1068	1068	1068	1068
# opt.	1068	219	577	945	945	911	945	940
⊘ rel. dev.	0.0	11.1	3.1	0.7	0.7	1.0	0.7	0.7
max. rel. dev.	0.0	45.2	18.8	11.8	11.8	22.2	11.8	13.6
⊘ abs. dev.	0.0	2.29	0.57	0.13	0.13	0.18	0.13	0.13
max. abs. dev.	0.0	14	4	2	2	4	2	3
⊘ cpu-sec.	6.6	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

Table 7: Results for INBOUND-TRSP

ume applied by the LFV rule, which additionally weights the number of product units in contrast to LPU, is constant for any product within our test instances. A comparison of lower bounds $\underline{C}^1$ and $\underline{C}^2$ reveals that $\underline{C}^2$ yields considerably tighter bounds with respect to all evaluation criteria. Note that relative deviation per instance is measured by $\frac{C(BDP)-\underline{C}}{C(BDP)}$. Although computation of $\underline{C}^2$ is more time-consuming than that of $\underline{C}^1$ (see Section 3), computational time is negligible. Within our BDP approach lower bound $\underline{C}^1$ is calculated first. If the node can not be fathomed based on $\underline{C}^1$, $\underline{C}^2$ is additionally calculated. An initial upper bound $\overline{C}$ on the objective value is determined by the LPU priority rule.

The results for INBOUND-TRSP are listed in Table 7. Again, the instances are derived from the large sized test bed by fixing the outbound sequences according to ascending outbound truck numbers. In light of the shown equivalence of OUTBOUND-TRSP and INBOUND-TRSP (see Section 4.1) it is not astounding that the results are very similar. Even the increased computational effort for reverting the given outbound sequence is negligible at this size of test instances.

6.2.2 Performance of the decomposition approach for the TRSP

To compare the performance of the decomposition approach for the overall TRSP the small instances are solved to optimality by a complete enumeration of all inbound truck sequences (first step) each of which is evaluated by the BDP-approach for OUTBOUND-TRSP (second step). All 1080 instances could be solved to optimality within a given time frame of 300 CPU-seconds. The average computation time is 8.1 seconds.

Table 8 reports the solution performance of the decomposition approach for stopping criteria SC-1 to SC-3 when either the BDP (left three columns) or the HSP (right three columns) is applied to solve the subproblems, where HSP is based on the LPU priority rule and BDP does not make use of a priority rule in order to retrieve an upper bound.

First, the solution performance of the BDP based decomposition approach is investigated. With respect to the convergence properties of the algorithms the results of Table 8 allow for two conclusions:

- The approach does not converge to the optimum, which is clarified by only 835 optimal solutions out of 1080 gathered with the stopping criterion SC-3 where break-off

		BDP			HSP	
	SC-1	SC-2	SC-3	SC-1	SC-2	SC-3
# opt.	394	711	835	388	926	1004
⊘ rel. dev.	15.1	6.2	4.0	15.3	2.3	1.1
max. rel. dev.	87.5	66.7	66.7	87.5	40.0	25.0
⊘ abs. dev.	1.0	0.4	0.2	1.0	0.2	0.1
max. abs. dev.	7	4	4	7	2	2
⊘ sub.	1	2.3	3.4	1	2.7	4.2
max. sub.	1	4	8	1	4	16
⊘ cpu-sec.	$<$0.01	$<$0.01	0.44	$<$0.01	$<$0.01	0.02

Table 8: Results for TRSP

is caused by identical sequences of adjacent solutions of the same subproblem.

- Furthermore, the procedure does not necessarily converge to a unique solution. If stopping criterion SC-3 is applied the BDP based decomposition approach circles within 31 instances (stopped after 10000 solved subproblems) between different solutions with identical solution value.

A comparison of the BDP and the HSP decomposition approach yields interesting results. For stopping criterion SC-1, the BDP-approach performs better than the heuristic. This is not surprising as only a single inbound sequence is evaluated in both cases with BDP finding the optimal solution for that sequence. For the other two criteria however, the results seem counter intuitive. Here, the HSP based approach performs better than the BDP-approach, although the latter determines optimal solutions for subproblems. The superiority of HSP can be explained by the following two observations:

- There often exist more than one optimal solutions for a subproblem. For instance, take the example for OUTBOUND-TRSP (see Figure 3) where 12 out of 24 possible sequences yield the same optimal solution value. The BDP-approach determines only a single shortest path and, thus, does not differentiate between optimal solutions. However, with regard to subsequent subproblems it would be advantageous to choose an optimal path, which postpones inbound (outbound) trucks carrying less (more) products. In contrast to that the HSP tends to generate (nearly) optimal sequences with this desired attribute, which can be exploited during succeeding iterations of the decomposition approach.

- The BDP-approach converges faster as compared to HSP. While the BDP based decomposition approach solves an average of 2.3 (3.4) subproblems for stopping criterion SC-2 (SC-3) is applied, HSP solves 2.7 (4.2) subproblems on average. With regard to the overall TRSP a higher number of iterations promises better solution values, so that the wider exploration of solution space by HSP turns out to be a second major advantage.

In the best configuration (HSP with SC-3) our decomposition approach solves 1004 instances (93%) to optimality with an average relative deviation from the optimum of merely 1.1% at negligible computational time. To emphasize the ability of our decomposition approach to deliver near optimal solutions, a further configuration is tested, where the decomposition approach is started multiple times. With 100 iterations of the decomposition approach (with in each case a new random initial sequence) within the best configuration (HSP with SC-3) the algorithms solves 1075 (99.5%) of the small instances to optimality with an average relative deviation from the optimum (⌀ rel. dev.) of only 0.06% at an average time of 1.6 CPU-seconds.

7 Discussion of results

The paper on hand presents a very basic truck sequencing problem. However, from a practical point of view, further operational characteristics might need to be integrated in the TRSP and the algorithms presented. This is discussed in the following:

Due dates: Frequently, shipments are bound to due dates negotiated with the customers. With given (or estimated) remaining transportation times, outbound truck specific due dates, which denote the latest point in time at which the respective truck has to leave the terminal to ensure on-time deliveries, can be determined. These due dates can be easily incorporated into the DP-approach for OUTBOUND-TRSP. As an arc captures the processing of a dedicated truck in a specific service slot, those arcs which lead to late deliveries just need to be excluded from the graph.

Arrival times: Inbound trucks do often not arrive simultaneously, but shifted in time depending on an inbound truck specific arrival time and may, thus, not be processed prior to their arrival. As arrival times are typically expressed as a delay measured from the start of the schedule (service slot $t = 1$), they cannot be directly considered as part of the modified INBOUND-TRSP of Section 5.2. The TRSP problem can nevertheless be solved to optimality by a repeated solution of OUTBOUND-TRSP.

Truck-specific processing times: If loading or unloading times of trucks differ considerably, e.g. as a result of diverging numbers or types of carried products, individual processing times need to be considered. In a single doors setting, this can be easily incorporated in the DP-approach. During the construction of inbound and outbound sequences, all predecessors of a truck are known, so that their processing times can simply be added up to derive the exact starting point. Instead of weighting each service slot with a single (equidistant) time unit, merely individual processing times of the trucks assigned to the respective service slot need to be considered.

Multiple doors: Typically, a cross dock consists of multiple inbound and multiple outbound doors. If the movement time of products remains constant, then within each service slot as many inbound and outbound trucks can be processed as there are inbound and outbound doors available. This can be readily considered in the node generation of the DP-approach. However, if multiple doors are available, then the distances to be covered by material handling devices inside the terminal will rather depend on the assignment of trucks to doors. In order to consider such door-dependent movement times, the DP-approach needs to be extended accordingly.

Mixed service mode: Organizational guidelines at cross docking terminals, typically aim at avoiding mixed assignments of inbound and outbound trucks per door. To ease material handling inside the terminal, one side of the dock is usually exclusively dedicated to inbound and the other to outbound operations (see Bartholdi and Gue, 2004). However, also a mixed service mode, where intermixed sequences of inbound and outbound trucks are processed at each single door, might exceptionally be employed. Then, a single schedule (in the very basic version for a single door otherwise for multiple doors) is to be determined. If this is the only alterations of our premises, then the schedule can be deduced without optimization, because a trivial solution is to schedule all inbound trucks first and then all outbound trucks in direct succession.

In real-world cases, typically several of the described extensions will need to be considered simultaneously. However, even in these cases the proposed solution concepts might be reasonably employed as adequate approximation techniques in order to quickly derive priority sequences for scheduling trucks in the respective real-world setting.

As truck scheduling often constitutes an online-problem optimized plans need nevertheless to be updated, whenever trucks arrive early at the yard or already planned vehicles fail to appear or arrive late. This requires a continuous replanning in a rolling planning horizon, so that very detailed models can be subject to multiple revisions. In light of this, the determination of robust plans, i.e. by integrating a minimum buffer time between any two trucks assigned to the same door, seems an important topic, which has not been addressed thus far.

Appendix

A. Proof of NP-Hardness for TRSP, INBOUND-TRSP and OUTBOUND-TRSP

We will prove NP-hardness for TRSP and INBOUND-TRSP from which NP-hardness of OUTBOUND-TRSP directly follows. For this purpose, we transform instances of 3-Partion, which is well known to be NP-Hard in the strong sense (see Garey and Johnson, 1979), to TRSP instances. 3-Partition is shortly restated in the following.

3-Partition Problem: Given $3q$ positive integers r_i $(i = 1, ..., 3q)$ and a positive integer B with $B/4 < r_i < B/2$ and $\sum_{i=1}^{3q} r_i = qB$ does there exist a partition of the set $\{1, 2, ..., 3q\}$ into q sets $\{A_1, A_2, ..., A_q\}$ such that $\sum_{j \in A_i} r_j = B \quad \forall i = 1, ..., q$?

Transformation of 3-Partition to TRSP: Consider $3q + q$ inbound trucks $I = \{1, 2, ..., 3q + q\}$ and the same number of outbound trucks $O = \{1, 2, ..., 3q + q\}$ and four products $P = \{1, 2, 3, 4\}$. For convenience, the outbound and inbound trucks are further subdivided into two sets each, such that $I_1 = \{1, 2, ..., 3q\}$, $I_2 = \{3q+1, ..., 3q+q\}$,

$O_1 = \{1, 2, ..., q\}$ and $O_2 = \{q+1, ..., 3q+q\}$. Inbound (outbound) trucks in the same set have identical structures of product demands. We shall refer to them as inbound (outbound) trucks of type 1 and 2, respectively. Let the q outbound trucks of type 1 demand products in the following quantities:

$$\begin{aligned} b_{o1} &= b_1 = B \\ b_{o2} &= b_2 = 3 \cdot C - B \\ b_{o3} &= 0 \\ b_{o4} &= 0 \end{aligned} \qquad \forall o \in O_1 \tag{28}$$

with C being an integer such that $C \geq \lceil B/2 \rceil$. The $3q$ outbound trucks of type 2 demand only products 3 and 4 in the same quantities of $b_{o3} = b_{o4} = 3 \quad \forall o \in O_2$.

The $3q$ inbound trucks of type 1 deliver products in quantities of:

$$\begin{aligned} a_{i1} &= r_i \\ a_{i2} &= C - r_i \\ a_{i3} &= 1 \\ a_{i4} &= 0 \end{aligned} \qquad \forall i \in I_1 \tag{29}$$

with r_i being $3q$ positive integers such that $B/4 < r_i < B/2 \quad \forall i = 1, ..., 3q$ and $\sum_{i=1,...,3q} r_i = qB$. As further $C \geq \lceil B/2 \rceil$, all a_{i2} are also positive integers. The q inbound trucks of type 2 deliver only product 4 in quantities of $a_{i4} = 3 \quad \forall i \in I_2$. As the number of input parameters of such a TRSP-instance is bounded in length by $3q$ it can be readily generated from any instance of 3-Partition in polynomial time.

Let us further assume that there is no movement time ($m = 0$). Note that an outbound truck of type 1 can at the earliest be scheduled to the third slot in the sequence as at least three inbound trucks of type 1 need to have delivered their products as $a_{i1} < b_1/2 \quad \forall i \in I_1$ and that an outbound truck of type 2 can only be scheduled after an additional inbound truck of type 2 has arrived. As $|I| = |O| = 4q$ it immediately follows that the lower bound $\underline{C}$ for this problem instance can be tightened to $\underline{C} = 4q+2$. We will now show that the decision problem of determining whether a solution with an objective value $C \leq \underline{C}$ actually exists, is as hard as the 3-Partition problem.

Let us first investigate the structure of the outbound sequence of such an optimal solution. The outbound sequence needs to begin with a truck of type 1 at service slot 3, as a type 2 truck could at the earliest be scheduled after 4 inbound trucks. The next two outbound trucks in such a sequence need to be of type 2, as a second type 1 truck could at the earliest be assigned to service slot 6. However, the assignment of an outbound truck of type 2 requires the prior assignment of at least one inbound truck of type 2. This in turn means that the second outbound truck of type 1 can at first be assigned to slot 7, so that three trucks of type 2 need to be scheduled successively. In fact, at slot 7 only an outbound truck of type 1 can be assigned as the fourth truck of type 2 requires the prior assignment of six inbound trucks of type 1 and two inbound trucks of type 2, respectively. The argument can be continued in the same fashion so that in any solution with an objective value of $4q + 2$ three outbound trucks of type 2 follow an outbound truck of type 1 alternately. This at the same time determines the structure of the inbound sequence as is displayed in Table 9, where the values 1 and 2 denote that an inbound (outbound) truck of type 1 or type 2 is to be assigned at the respective position.

service slot	1	2	3	4	5	6	7	8	9	10	...	4q-2	4q-1	4q	4q+1	4q+2
inbound seq.	1	1	1	2	1	1	1	2	1	1	...	1	1	2		
outbound seq.			1	2	2	2	1	2	2	...	2	2	1	2	2	2

Table 9: Optimal Solution Structure

Note that due to the delivery and demand structure it will always be possible to assign outbound trucks of type 2 at their positions. However, whether or not an outbound truck of type 1 can be assigned, depends on the quantities of products 1 and 2 delivered by the inbound trucks of type 1.

The solution of any YES-instance of 3-Partition can be polynomially transformed to a solution of TRSP by simply ordering the sets A_i arbitrarily and replacing any triplet of inbound trucks of type 1 by the respective elements of sets A_i. It can be easily verified that such an inbound sequence allows the assignment of the optimal outbound sequence without delay, so that the makespan in fact results to $4q+2$.

Further note that for any 3 inbound trucks of type 1, say, $l, m, n \in I_1$ the following relationship holds:

$$\begin{aligned} & a_{l1} + a_{m1} + a_{n1} = b_1 - \epsilon \\ \wedge \quad & a_{l2} + a_{m2} + a_{n2} = b_2 + \epsilon \end{aligned} \qquad \forall l, m, n \in I_1 \tag{30}$$

with ϵ being an integer value. This can be easily shown by substitution:

$$\begin{aligned} & a_{l2} + a_{m2} + a_{n2} && = b_2 + \epsilon \\ & \text{by using (28) and (29)} && \\ \rightarrow \quad & C - a_{l1} + C - a_{m1} + C - a_{n1} && = 3 \cdot C - B + \epsilon \\ \leftrightarrow \quad & a_{l1} + a_{m1} + a_{n1} && = B - \epsilon = b_1 - \epsilon \end{aligned}$$

It directly follows that whenever the sum of delivered units of product 1 for any three inbound trucks exceeds the demand of an outbound truck of type 1 for product 1 ($\epsilon > 0$), then the sum of delivered units of product 2 falls below the demand for product 2 and vice versa. Consequently, any outbound truck of type 1 can only be directly scheduled after three inbound trucks of type 1, if the sum of delivered units of product 1 exactly equals the demand b_1.

It follows that a solution with a makespan of $4q+2$ does exist if and only if the answer to the corresponding instance of 3-Partition is a YES-instance. NP-hardness in the strong sense for TRSP immediately follows. As any fixed outbound sequence in line with the structure of Table 9 would still require the solution of a 3-Partition problem, already INBOUND-TRSP is NP-hard in the strong sense. NP-hardness for OUTBOUND-TRSP can be proven in the same fashion by simply reverting the demand and delivery structures for inbound and outbound trucks and is thus omitted.

B. Proof of lower bound relationship

Proposition: $\underline{C}^2 = \underline{C}^1$.

Proof: It holds that $M_p = \sum_{i\in I} a_{ip} = \sum_{o\in O} b_{op}$, which is a basic assumption of the TRSP model (see Section 3). For a sequence π^p of inbound trucks ordered by decreasing a_{ip} the cumulated quantity delivered up to a slot t amounts at least to $S_{pt}^{min} = t \cdot \frac{M_p}{|I|}\, \forall p \in P; t = 1, \ldots, |I|$, otherwise trucks are not ordered by decreasing a_{ip}. In contrast to that the maximum possible demand level D_{pt}^{max} of a sequence μ^p of outbound trucks ordered by increasing b_{op} amounts to: $D_{pt}^{max} = t \cdot \frac{M_p}{|O|}\, \forall p \in P; t = 1, \ldots, |O|$. If $|I| \leq |O|$, then $S_{pt}^{min} \geq D_{pt}^{max}$ holds at any slot for any product. Thus, the stock level is always larger or equal than cumulative demand, so that inbound and outbound trucks can be scheduled in direct succession, which is exactly the underlying assumption of lower bound $\underline{C}^1$. If $|I| > |O|$, then O outbound trucks at the latest must be scheduled in slots $t = |I| - |O| + 1, \ldots, |I|$ to fulfill the basic assumption of lower bound $\underline{C}^1$. Thus, it must hold that $(|I| - t) \cdot \frac{M_p}{|I|} \geq (|O| - t) \cdot \frac{M_p}{|O|}\, \forall\, t = 0, \ldots, |O| - 1$, which is obviously the case if $|I| \geq |O|$. $\square$

References

[1] Apte, U.M., Viswanathan, S., 2000. Effective cross docking for improving distribution efficiencies, International Journal of Logistics: Research and Applications 3, 291–302.

[2] Bartholdi, J.J., Gue, K.R., 2000. Reducing labor costs in an LTL crossdocking terminal, Operations Research 48, 823–832.

[3] Bartholdi, J.J., Gue, K.R., 2004. The best shape for a crossdock, Transportation Science 38, 235–244.

[4] Bautista, J., Companys, R., Corominas, A., 1996. Heuristics and exact algorithms for solving the Monden problem, European Journal of Operational Research 88, 101–113.

[5] Bermudez, R., Cole, M.H., 2001. A genetic algorithm approach to door assignments in breakbulk terminals. Technical Report MBTC-1102, Mack Blackwell Transportation Center, University of Arkansas, Fayetteville, Arkansas.

[6] Boysen, N., Fliedner, M., Scholl, A., 2007. Sequencing mixed-model assembly lines to minimize part inventory cost, OR Spectrum (to appear), doi: 10.1007/s00291-007-0095-2.

[7] Campbell, J.F., 1994. A survey of network hub location, Studies in Locational Analysis 6, 31–49.

[8] Carraway, R.L., Schmidt, R.L., 1991. An improved discrete dynamic programming algorithm for allocating resources among interdependent projects, Management Science 37, 1195–1200.

[9] Forger, G., 1995. UPS starts world's premiere cross-docking operation, Modern Material Handling, 36–38.

[10] Garey, M.R., Johnson, D.S., 1979. A guide to the theory of NP-completeness, Freeman, New York.

[11] Gue, K.R., 1999. The effect of trailer scheduling on the layout of freight terminals, Transportation Science 33, 419–428.

[12] Johnson, S.M., 1954. Optimal two- and three-stage production schedules with setup times included, Naval Research Logistics Quarterly 1, 61–68.

[13] Klose, A., Drexl, A., 2005. Facility location models for distribution system design, European Journal of Operational Research 162, 4–29.

[14] Li, Y., Lim, A., Rodrigues, B., 2004. Crossdocking - JIT scheduling with time windows, Journal of the Operational Research Society 55, 1342–1351.

[15] Marsten, R.E., Morin, T.L., 1978. A hybrid approach to discrete mathematical programming, Mathematical Programming 14, 21–40.

[16] Morin, T.L., Marsten, R.E., 1976. Branch-and-bound strategies for dynamic programming, Operations Research 24, 611–627.

[17] Oh, Y., Hwang, H., Cha, C.N., Lee, S., 2006. A dock-door assignment problem for the Korean mail distribution center, Computers & Industrial Engineering 51, 288–296.

[18] Stalk, G., Evans, P., Shulman, L.E., 1992. Competing on capabilities: The new role of corporate strategy, Harvard Business Review 70(2), 57–69.

[19] Tsui, L.Y., Chang, C.-H., 1990. A microcomputer based decision support tool for assigning dock doors in freight yards, Computers & Industrial Engineering 19, 309–312.

[20] Tsui, L.Y., Chang, C.-H., 1992. An optimal solution to dock door assignment problem, Computers & Industrial Engineering 23, 283–286.

[21] Witt, C.E., 1998. Crossdocking: Concepts demand choice, Material Handling Engineering 53(7), 44–49.

[22] Yu, W., Egbelu, P.J., 2006. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage, European Journal of Operational Research (to appear), doi: 10.1016/j.ejor.2006.10.047.