Short term storage of goods in cross-docking operations

Research Memorandum 2004-19
Short term storage of goods
in cross-docking operations

Research Memorandum 2004-19

Iris F.A. Vis
Kees Jan Roodbergen

vrije Universiteit

amsterdam
Short term storage of goods in cross-docking operations

Iris F.A. Vis
School of Economics and Business Administration, Vrije Universiteit
Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands,
ivis@feweb.vu.nl

Kees Jan Roodbergen
Rotterdam School of Management, Erasmus University Rotterdam, P.O. Box
1738, 3000 DR Rotterdam, The Netherlands, K.Roodbergen@fbk.eur.nl

Abstract
Cross docking is one of the options to reduce lead times and inventories and to
improve customer response time in supply chains. Cross-docking centres are dynamic
environments where products arrive, are regrouped, and leave the same day. In this
paper we focus on the process of short-term storage of unit-loads in a cross-docking
environment. The goal is to determine temporary storage locations for incoming unit
loads such that the travel distances of the forklift trucks with these unit loads are
minimised. We model this problem as a novel application of the minimum cost flow
problem and show the applicability of the model for different types of layouts and
priorities.

Introduction
Warehouses are under a continuous pressure to decrease cost and to reduce the
duration of stay of the products. In some cases it is possible to almost eliminate
storage by using cross docking. Cross docking basically means that products are
unloaded from a truck and subsequently loaded into other trucks. Loads are divided
over the loading trucks based on their destination. Unit loads (e.g. a pallet with
products) can be stored on the floor between unloading and loading for a few hours to
wait for a truck to arrive. However, products are not stored for longer periods. This
way, cross docking enables very short lead times for products.
For a successful implementation of cross docking, attention could be paid at least to
the following factors (see Moore and Roy\(^1\), and Schaffer\(^2\)). Firstly, the supply must be
very reliable with short lead times, because there is no stock available to buffer
between supply and demand. Products should be delivered at the right time, in the
right quantity and of the right quality. Otherwise, trucks will travel too late or half-
empty to customers. Secondly, information should be available through the entire
supply chain. Arrival times, departure times and destinations have to be available in
time such that the planning of trucks can be done in advance. Thirdly, to ensure that
products arrive and depart on time, trucks should arrive on time. Transport should be
reliable to prevent delays in delivery to customers. Furthermore, products with high
demands and products with highly predictable demands are more suitable (see
Richardson\(^3\)). Though it must be noted that transport volume itself can be increased
by consolidating multiple orders (see Klincewicz and Rosenwein\(^4\)).
Except for the external factors mentioned, cross docking can only be successful if the
design of the cross-docking centre and the operating polices for all processes are
executed efficiently. Potential advantages of cross docking are, for example, reduction
of inventories and associated costs, shorter lead times, improvement of service to
customers, improvement of relations with suppliers and the possibility to take real-
time decisions. Schaffer\textsuperscript{5} discusses in detail how cross docking can improve
efficiency.

In a typical situation, the cross docking centre consists of a number of dock doors
where trucks can load or unload. These dock doors can be located along any of the
walls. Trucks arrive according to a schedule to load or unload. First trucks arrive that
need to be unloaded. Each arriving unit load, which is not immediately transferred to
another truck, has to be assigned to a storage location. General-purpose forklift trucks
generally perform transport between trucks and storage locations. When (almost) all
incoming loads have been received, trucks start arriving for loading. If a truck arrives
for loading, all pallets for this destination are moved from their storage locations to
this truck. Figure 1 gives an illustration of a possible layout of a cross-docking centre.

\section*{INSERT FIGURE 1}

Cross docking is an interesting subject, both from a research perspective as well as
from a practical point of view. Many organisations are gradually turning to the
concept of cross docking and are searching for efficient methods to regulate the
 process. Most cross-docking operations currently are run using simple heuristics.
Only little research has been performed on the subject of cross docking. Even though
the number of different operations in a cross-docking centre is small compared to a
full-service warehouse, the planning issues that do exist, are of a complex nature.
For example, an important planning problem in cross-docking centres concerns the
donk door assignment. The issue is to assign incoming and departing trucks to dock
doors such that the operations inside the facility can be performed as efficiently as
possible. Tsui and Chang\textsuperscript{6} present a heuristic to assign trucks to dock doors such that
the total travel distance in the facility is minimised. A branch-and-bound procedure
for this problem is described in Tsui and Chang\textsuperscript{7}.

Gue\textsuperscript{8} constructed an LP-model to determine the material flow in a facility depending on a parameter that determines the influence of the supervisor on the
assignment of incoming trailers to dock doors. A near-optimal assignment of
destinations to unloading dock doors is determined with a local search method, that
uses 2-interchange to alter trailer assignments to dock doors, and calculating material
handling requirements in each step using the LP-model. Bartholdi and Gue\textsuperscript{9} use a
simulated annealing approach to interchange designations of dock doors (i.e. which
trailers load at which doors). The objective is to minimise workers travel time and
waiting time due to congestion.

In warehouses, storage of goods is common. Within warehouses products are
stored, contrary to cross-docking centres, for a longer period of time. There exist
numerous ways to assign products to storage locations in warehouses. For example,
the random storage policy assigns every incoming load to a location that is selected
randomly from all eligible empty locations with equal probability. Furthermore,
several storage policies have been developed that take product turnover into account.
This means that products with the highest sales rates are located at the easiest accessible locations. For more detailed information on storage assignment in warehouses, refer to Van den Berg\textsuperscript{10}. The dynamics in cross docking are, however, very different from warehousing. There is, for example, no need to distinguish products by demand frequency, because each unit load arrives and leaves the same day, without any further processing. If we assume that for any unit load the unloading dock door and the loading dock door is known, which is not unusual, then we can determine the best storage location based on the full distance travelled in the entire facility. This in contrast with the retrieval processes in warehouses, where often only the distance within one specified area is minimised (see e.g. Ratliff and Rosenthal\textsuperscript{11}).

In this paper we seek to determine short-term storage locations such that the total travel distances in the entire facility are minimised. In the next section we describe the problem and assumptions and give the “row-based storage assignment algorithm” which can be used to solve the problem in polynomial time. Other approaches of the problem and related solution approaches are discussed thereafter. Finally, an example of the “row based storage assignment algorithm” and conclusions are given.

Model and Algorithm

At the cross-docking centre we observe loads arriving at \( i \) dock doors at one side of the centre and leaving at \( j \) dock doors at the opposite side (some of our initial assumptions will be relaxed in the next section). Trucks bring and take away loads from the centre. The storage area is divided into several parallel rows. Via these rows vehicles or workers can travel to transport loads from unloading doors to loading doors. Unit loads are stored on the ground or in racks where they can be accessed directly. Furthermore, we make the following assumptions:

1. Each storage location can only be used once in each unloading/loading cycle. This restriction will be satisfied automatically if arrival of unloading and loading trucks is separated in time. As an example, consider an end-of-line terminal where freight is picked up in the neighbourhood during the day, and shipped to other terminals or destinations in the evening. Freight arriving from other terminals is moved early in the morning (see e.g. Bartholdi and Gue\textsuperscript{3}). Flows are separated in time here and each location can only be used once in each cycle, because loads are only shipped once all loads have been received.

2. From the dock door assignment it is known for each truck at which door it arrives. Several authors have addressed this issue; see for example Tsui and Chang\textsuperscript{6,7}, Gue\textsuperscript{12} and Bartholdi and Gue\textsuperscript{3}.

3. Available capacity of each storage location and as a result the capacity of each row is known. This information can be easily obtained from the information system.

4. At each unloading dock door the exact aggregated quantity of unit loads designated for each of the loading dock doors is known. This is a direct consequence of assumption 2 combined with the requirement in cross docking that a good information system – preferably with EDI – is present (see for example Schäffer\textsuperscript{3}).
5. Total capacity of the facility is sufficient to store all freight that is to be transshipped in one cycle.

The objective of this section is to develop a model and polynomial time “row based storage assignment” algorithm to solve the problem of determining storage locations for arriving unit loads at a cross-docking centre such that the total travel distance within the facility is minimised. For each unit load a certain distance has to be travelled from the unloading dock door, via the storage location, to the loading dock door. For each combination of an unloading and a loading dock door, there exists a minimum distance path between the two doors. To transship a unit load from its unloading to its loading door at least this minimum distance has to be travelled.

During the transport through row(s) located on the shortest path numerous storage locations will be passed. If the unit load would be stored at such a location, then no extra distance would have to be travelled with the load to store it as compared to direct transfer. If none of the locations in row(s) on the shortest path is free, then a storage location in another row needs to be chosen. Consequently, an extra distance has to be travelled to store the unit load. We interpret this extra distance as costs to be made to store the unit load. Clearly, these costs are considered zero in the case that a storage location on the shortest path between the two dock doors is used.

We define the following parameters and variables:

- \( N \) number of unloading dock doors.
- \( M \) number of loading dock doors.
- \( R \) number of storage rows in storage area.
- \( x_{ij} \) number of unit loads with origin at door \( i \) \((1 \leq i \leq N)\) and destination at door \( j \) \((1 \leq j \leq M)\).
- \( X \) total number of unit loads to be transshipped from the unloading dock doors to the loading dock doors.
- \( c_{kij} \) costs to store a unit load in row \( k \) \((1 \leq k \leq R)\) if the unit load is transshipped from door \( i \) \((1 \leq i \leq N)\) to door \( j \) \((1 \leq j \leq M)\). These costs are zero if row \( k \) is on the shortest path from door \( i \) to door \( j \).
- \( z_k \) storage capacity of row \( k \) \((1 \leq k \leq R)\).
- \( y_{v_iv_j} \) number of unit loads \((1 \leq y_{v_iv_j} \leq x_{ij})\) with origin at door \( i \) \((1 \leq i \leq N)\) and destination at door \( j \) \((1 \leq j \leq M)\) that are stored in row \( k \).

Using the above data we construct a directed network \( G \). Each flow of unit loads from an unloading dock door \( i \) \((1 \leq i \leq N)\) to a loading dock door \( j \) \((1 \leq j \leq M)\) is represented by a node \( v_{ij} \). For example, in the case that 3 unloading and 3 loading dock doors exist, 9 nodes represent the various flows. Namely, a node representing the flow of unit loads from unloading dock door 1 to loading dock door 1, a node representing the flow of unit loads from unloading dock door 1 to loading dock door 2 and so on.

For each node \( v_{ij} \) the total number of unit loads to be shipped from \( i \) to \( j \) is known and equals \( x_{ij} \). To include these quantities in the directed network, we introduce a source node \( S \). Directed arcs are added from source node \( S \) to the nodes \( v_{ij} \) for all \( i \) and \( j \). The lower bound and upper bound of the flow \( y_{Sv_{ij}} \) on each directed arc \((S, v_{ij})\) equal \( x_{ij} \).
Each storage row \( k \) (1 \( \leq \) \( k \) \( \leq \) \( R \)) is represented by a node \( w_k \). The storage capacity of row \( k \) is limited to \( z_k \). To incorporate these capacities into the directed network, we introduce a sink node \( T \). Directed arcs are added from each node \( w_k \) to the sink node \( T \). The number of unit loads stored in row \( k \) equals the flow \( y_{w_kT} \) on the arc \((w_k,T)\). The lower bound of the flow on each of these arcs equals zero. The upper bound of each directed arc \((w_k,T)\) equals \( z_k \). Consequently, the maximum number of unit loads assigned to a certain row equals the maximum capacity of the row.

The objective is to assign unit loads to storage locations such that travel distances with loaded vehicles are minimised. By connecting nodes \( v_j \) with nodes \( w_k \) we can derive an assignment of unit loads arriving at door \( i \) and leaving at door \( j \) to a storage row \( k \). Thus, we add directed arcs \((v_j,w_k)\) for all \( i \), \( j \), \( k \) to the network. As explained, extra travel distances may occur if unit loads with a known origin and destination are assigned to a storage row that are not on the shortest path from \( i \) to \( j \). These costs \( c_{kj} \) are assigned to the directed arcs \((v_j,w_k)\) for all \( i \), \( j \), \( k \). The costs of all other arcs equal zero.

Finally, we add a directed arc \((S,T)\) to ensure that the flow leaving from the source \( S \) equals the flow arriving in the sink node \( T \). The lower and upper bound of this flow \( y_{ST} \) equal the total number of unit loads \( X \) to be transhipped from the unloading dock doors to the loading dock doors. In all nodes the incoming flow equals the outgoing flow.

Thus, we have defined a directed network \( G = (V,A) \) with

\[
V = \{ v_i \mid 1 \leq i \leq N \text{ and } 1 \leq j \leq M \} \\
\cup \{ w_k \mid 1 \leq k \leq R \} \\
\cup \{ S,T \}
\]

\[
A = \{ (v_j, w_k) \mid v_j, w_k \in V \} \\
\cup \{ (S, v_j) \mid v_j \in V \} \\
\cup \{ (w_k, T) \mid w_k \in V \} \\
\cup \{ (S, T) \}
\]

With the lower and upper bounds

\[
x_j \leq y_{Sv_j} \leq x_j \\
0 \leq y_{w_kT} \leq z_k \\
X \leq y_{ST} \leq X
\]

The objective is to find flows \( y_{v_jw_k} \) from nodes \( v_j \) to nodes \( w_k \) in the directed network such that the total costs are minimised and the capacity constraints are met. We can use any minimum cost flow algorithm to obtain the values of \( y_{v_jw_k} \). A description of various minimum cost flow algorithms is given in Ahuja et al.\textsuperscript{13}. For example, the "enhanced capacity scaling algorithm" can be used, which has a time
complexity of $O((m \log n)(m + n \log n))$, where $n$ is the number of nodes and $m$ the number of arcs in the network.

Summarising, our "row based storage assignment algorithm", which can be used to determine storage locations for arriving unit loads at a cross-docking centre such that total travel distances are minimised, consists of two steps. Namely, first we construct a network according to the above mentioned steps and secondly we apply any minimum cost flow algorithm.

In the section "illustrative example" we will show how the algorithm can be used. In the next section, we will describe some variations on our model and algorithm.

Simplification and extensions of the algorithm

Simplification for symmetric layouts

For cross-docking centres with a symmetric layout we can speed up the "row based storage assignment algorithm". In a symmetric layout the number of loading dock doors equals the number of unloading dock doors. Furthermore, the rows used for temporarily storage are positioned exactly between the loading and unloading dock doors. As a consequence, the total number of storage rows $R$ equals the number of unloading dock doors and equals the number of loading dock doors. For an illustration of a symmetric layout, refer to Figure 2. In this specific situation the storage costs for the flow of unit loads from door $i$ to door $j$ equal the storage costs for the flow of unit loads from door $j$ to door $i$. As a result, it is possible to use one node $v_{ij}$ for flows between door $i$ to door $j$ in both directions. Instead of using $R^2$ nodes to represent the flow of unit loads from one dock door to another we now just need $\frac{1}{2} R(R+1)$ nodes.

The lower and upper capacity of the flow on the arcs $(s, v_{ij})$ will be equal to $x_{ij} + x_{ji}$. The rest of the solution procedure remains the same. The example in the next section illustrates this reduction of the number of nodes in the network.

Zone-based storage

In the "row-based storage assignment algorithm" the costs for each storage location in a row are equal. Time pressure is, however, often higher for loading trucks than for unloading. Therefore, it might be useful to try and locate the unit loads as close to the loading doors as possible. To this end, we divide each row into $\ell$ zones. We replace the nodes, which represent rows by a series of nodes representing the zones. Since we are now only interested in the distance to the loading dock, we define the costs $c_{kj}$ as the distance from zone $k$ ($1 \leq k \leq \ell R$) to loading dock door $j$. Furthermore, we need to redefine $x_{kj}$ to be the storage capacity of zone $k$ ($1 \leq k \leq \ell R$).

Location-based storage

We can take the adaptation we made for zone-based storage one step further by taking into account the individual storage locations. This means that $\ell$ will now represent the number of locations in a row. The advantage is that we can more precisely determine the most appropriate storage location for each load. The downside is that the number of nodes in the network increases, which will increase computation times.
Other layouts of the cross-docking centre

Until so far we have studied cross-docking centres with a so-called I-layout (see Figure 1), where loading and unloading occurs on opposite sides of the building. Other layouts may also be used in practice. For example, a U-layout with all dock doors on the same side or a layout where loading and unloading doors are intermingled. In the original algorithm we defined the costs c_{kj} as the excess travel time compared to the shortest path. If the loading and unloading doors are on the same wall, however, the shortest path does not pass any storage locations. Therefore, the excess travel time cannot be easily defined. Furthermore, if loading and unloading can occur at the same side of the building, we will need to know the exact storage location within the row, because each location of a row might result in a different travel times. To solve this problem, we just use the adapted version for location-based storage, which already identifies individual storage locations. If loading has priority over unloading, then no changes have to be made. If loading and unloading are seen as equally important, the costs c_{kj} can be redefined as the actual distance to go from dock door i to dock door j via storage location k.

Illustrative example

To illustrate the application of our algorithm for a symmetric layout, we discuss the following illustrative example. We consider a cross-docking centre with a symmetric layout. There are three unloading dock doors ($N = 3$) and three loading dock doors ($M = 3$). Between each pair of opposite doors a row is positioned for temporary storage of unit loads. Consequently, the value of $R$ equals three. This layout with distances (in meters) is illustrated in Figure 2.

**INSERT FIGURE 2**

With the data presented in Figure 2 we construct the following matrix $D(i,j)$ containing the length of the shortest paths between each pair $(i,j)$ for $1 \leq i \leq 3$ and $1 \leq j \leq 3$:

$$D(i,j) = \begin{pmatrix}
50 & 60 & 70 \\
60 & 50 & 60 \\
70 & 60 & 50
\end{pmatrix}$$

In the case that unit loads are not stored in the row which can be traversed on the shortest route from door $i$ to door $j$ extra distances need to be travelled. These costs $c_{kj}$ are presented in the following matrix. The rows of the matrix indicate the rows ($k = 1, 2, 3$) in which the storage can occur. Each column represents an unloading door in combination with a loading door. Namely, unloading door 1 combined with loading door 1, unloading door 1 combined with loading door 2, ..., unloading door 3 combined with loading door 3. The elements in the matrix $E(k,ij)$ represent the extra costs in the case that unit loads are transhipped from door $i$ ($1 \leq i \leq 3$) to door $j$ ($1 \leq j \leq 3$) via temporarily storage in row $k$ ($1 \leq k \leq 3$).

$$E(k,ij) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 40 \\
20 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 20 & 0 & 20 & 0 & 0 & 0
\end{pmatrix}$$
Each row \( k (1 \leq k \leq 3) \) has a certain (remaining) storage capacity \( z_k \). These capacities are presented in the following matrix:

\[
\begin{pmatrix}
75 \\
80 \\
150
\end{pmatrix}
\]

For each unloading dock door \( i \) it is known which quantity of unit loads need to be transhipped to each of the loading dock doors \( j \). These quantities \( x_{ij} \) are given in the following matrix \( X(i,j) \):

\[
X(i,j) = \begin{pmatrix}
80 & 10 & 10 \\
20 & 70 & 10 \\
5 & 45 & 50
\end{pmatrix}
\]

From matrix \( X(i,j) \) it can be concluded that in total 300 unit loads need to be transhipped. Consequently, the value of \( X \) equals 300. With these data we construct the directed network \( G \). From the proposed simplification in the previous section, we know that we can use one node to represent the flow of unit loads from door \( i \) to door \( j \) and from door \( j \) to door \( i \). For example, we can use the node \( v_{12} \) to represent the flow from unloading door 1 to loading door 2 and for the flow from unloading door 2 to loading door 1. Clearly, we can decrease the total number of nodes in the directed network from 9 to 6.

The costs remain the same on the directed arcs. However, the quantities to be transhipped, which are indicated in \( X(i,j) \) need to be transformed. For example, \( X(1,2) \) becomes \( x_{12} + x_{21} = 10 + 20 = 30 \). The new matrix \( X'(i,j) \) is:

\[
X'(i,j) = \begin{pmatrix}
80 & 30 & 15 \\
-70 & 55 \\
- & - & 50
\end{pmatrix}
\]

Figure 3 presents the directed network \( G \) related to the data given.

**INSERT FIGURE 3**

The objective is to assign the unit loads to storage locations such that the total travel distance is minimised. Therefore, we determine a flow in the directed network such that the extra distances to be travelled are minimised. By applying a minimum cost flow algorithm the following solution is obtained (all other variables have value zero):

\[
\begin{align*}
y_{v_{11}w_1} &= 75 & y_{v_{11}w_2} &= 15 \\
y_{v_{12}v_1} &= 5 & y_{v_{12}v_2} &= 70 \\
y_{v_{11}v_2} &= 10 & y_{v_{11}v_3} &= 55 \\
y_{v_{12}v_3} &= 20 & y_{v_{12}v_4} &= 50
\end{align*}
\]

From these values it can be concluded that almost all unit loads can be stored at a storage location along the shortest path from their origin to their destination. However, some of the unit loads need to be stored in a row, which is not traversed on the shortest route from the origin to the destination of the unit load. As a result, an
extra distance of 600 meters needs to be travelled to tranship all unit loads from their origin to their destination via a storage location.

Conclusions
This paper describes a method that is capable of determining storage locations for loads in a cross-docking centre, such that the travel time is minimised for the vehicles that transport loads. The method consists of a network formulation that incorporates loading and unloading dock door locations, travel distances and available storage space in the facility. Solutions can be obtained in polynomial time. The method can also easily be adapted to accommodate other situations. We showed adaptations that enable the applicability of the model for situations with different layouts and for situations where loading incurs a higher time pressure than unloading. An example has been given to illustrate the effectiveness of the method.

References
FIGURES

Figure 1:

![Diagram of loading and unloading doors with storage locations labeled]

Figure 2:

![Diagram of loading and unloading doors with storage locations labeled]
CAPTIONS
Figure 1: Simplified layout of a cross-docking centre
Figure 2: Layout of the cross-docking centre used in the example
Figure 3: directed network G with (lower bound, upper bound) and costs