Accepted Manuscript

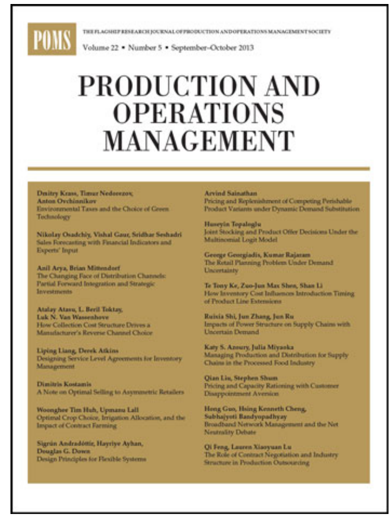
Title: Storing Fresh Produce For Fast Retrieval In An Automated Compact Cross-Dock System

Authors: Nima Zaerpour, Yugang Yu, René B.M Koster

DOI: http://dx.doi.org/doi:10.1111/poms.12321

Reference: POMS 12321

To appear in: Production and Operations Management



Please cite this article as: Zaerpour Nima., et al., Storing Fresh Produce For Fast Retrieval In An Automated Compact Cross-Dock System. *Production and Operations Management* (2014), http://dx.doi.org/doi:10.1111/poms.12321

STORING FRESH PRODUCE FOR FAST RETRIEVAL IN AN AUTOMATED COMPACT CROSS-DOCK SYSTEM

Nima Zaerpour[a], Yugang Yu[b], and René B.M de Koster[c]

[a]*Faculty of Economics & Business Administration, VU University Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, Netherlands, n.zaerpour@vu.nl, +31 20 5986403*

[b]*School of Management, University of Science and Technology of China, Hefei 230026, China, ygyu@ustc.edu.cn, +86 551 63606187*

[c]*Rotterdam School of Management, Erasmus University, PO Box 1738, 3000 DR Rotterdam, Netherlands, rkoster@rsm.nl, +31 10 4081992*

Abstract

We study temporary storage of fresh produce in a cross-dock center. In order to minimize cooling cost, compact storage systems are used. A major disadvantage of these systems is that additional retrieval time is needed, caused by necessary reshuffles due to the improper storage sequence of unit loads. In practice therefore, a dedicated storage policy is used in which every storage lane in the system accommodates only one product. However, this policy does not use the planned arrival time information of the outbound trucks. To exploit this information, this paper proposes a mathematical model for a shared storage policy that minimizes total retrieval time. The policy allows different products to share the same lane. In order to solve real-sized problems, an effective and efficient heuristic is proposed, based on a greedy construction and an improvement part, which provides near optimal solutions. The gaps between the results of the heuristic and the lower bound are mostly less than 1%. The resulting shared storage policy is generally robust against disturbances in arrival or departure times. We compare our shared storage heuristic with dedicated storage to determine which policy performs best under which circumstances. For most practical cases, shared storage appears to outperform dedicated storage, with a shorter response time and better storage lane utilization.

Keywords: Warehousing, Cross-docking, Compact storage system, Shared storage, Robust assignment

Received: March 2011; accepted: October 2014 by Eric Johnson after four revisions.

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/poms.12321

1. Introduction

Many companies use cross-dock processes to decouple outbound from inbound transport. In a cross-dock system, the arriving supply has already been allocated to demand. Compared to warehousing the products, cross-docking leads to lower handling and inventory costs, while it shares the benefit of having high truck load factors.

This paper studies temporary storage in cross-dock facilities, using a compact automated storage system. Our study is inspired by a cross-dock center storing fresh produce (vegetables and fruits) which is harvested on demand partly from greenhouses. Products are temporarily stored in a refrigerated warehouse and, at the planned truck arrival times, retrieved to fill customer orders. The company has several main performance measures:

- serving the outbound trucks within their planned arrival and departure times,
- preventing quality loss by minimizing the time during which the unit loads to be shipped are outside the conditioned (i.e. refrigerated, humidity and air-controlled) storage lanes,
- increasing the throughput and system utilization.

Minimizing total retrieval time is a good proxy for all measures. When an outbound truck arrives at the dock door, fast retrieval of unit loads of customer orders reduces the truck's waiting time. Therefore, the truck can more likely be served within its planned arrival and departure times. The risk of quality loss is also reduced as the products spend the minimum time outside the conditioned area. Reducing the waiting time of each truck potentially provides the opportunity to serve more trucks within a shift, leading to higher throughput and system utilization. Cross-docking has been implemented widely in many supply chain networks by companies such as Wal-Mart (United States), Albert Heijn (Netherlands), Tesco (UK) and Carrefour (France). The cross-dock operates in a shift-based mode: products are harvested in the evening, arrive during the night, are stored temporarily, and are retrieved and shipped the next morning. Retrieval requests are often more critical than storage jobs, as they are directly linked to customer orders (Bartholdi and Hackman, 2011). Short lead times are possible due to the use of automated storage and handling systems.

Such automated storage/retrieval systems (AS/RS) offer high throughput, short response time and labor saving (Park, 2001). An AS/RS is capable of automatically handling unit loads (e.g. pallets, totes) and contains storage racks, storage/retrieval (S/R) machines, aisles and I/O-points as its main components (Roodbergen and Vis, 2009). S/R machines can autonomously move, pick up, and drop off unit loads. They operate in aisles between the racks. Every machine has an input/output point (I/O point or depot) to drop off retrieved unit loads or to pick up incoming unit loads for storage. In a conventional AS/RS (known as 2-dimensional or 2D, AS/RS), unit loads are stored in single-deep racks (i.e. with a depth of only one unit load). The aisles between the racks occupy a significant percentage of the floor space (Yu and De Koster, 2009). In such a facility, the building with fixed installations may account for up to 60% to 70% of total operational costs (rent/interest, write-offs, maintenance, energy; De Koster 1996). A very important design criterion is therefore reducing space consumption as much as possible.

A solution to this space problem is given by multi-deep AS/RS, also called compact storage systems. Compact storage systems have become increasingly popular for storing products (Hu et al., 2005). They are particularly used for refrigerated storage, as cooling cost is directly related to storage space. In compact storage systems, unit loads are stored in multiple-deep racks, requiring less space than 2D systems. The most common type of compact storage system uses an S/R machine with a satellite machine to handle the depth movement. Figure 1 illustrates such a satellite-based compact storage system. The satellite, linked to the S/R machine, moves within a storage lane to store or retrieve the unit loads. The S/R machine waits in front of the lane until the satellite machine returns.

The main difficulty with multi-deep systems is that the unit loads need to be reshuffled. This is the process of removing unit loads stored in front of a unit load that needs storage or retrieval. In a system which operates automatically, in a shift-based mode, outbound reshuffles (the reshuffles required during the outbound process) are more critical than inbound reshuffles. Unit loads of inbound trucks can be assigned temporarily to the conditioned storage lanes while minimizing the total storage time and so the time that products spend in the unconditioned area. While waiting inside the conditioned area, the products can be reshuffled automatically and relocated to their proper storage locations without quality loss. There is ample time for such reshuffles during the night shift. Therefore, storage allocation decisions, taken during the inbound process, can be decoupled from outbound sequencing decisions to minimize the total retrieval time. Hence, unlike the outbound process, the inbound process is not a bottleneck in our cross-dock operation. In addition, since the truck loading process is much faster than the retrieval process, the bottleneck of the outbound process is the retrieval process. Minimizing the total retrieval time during the outbound process is therefore the company’s prime objective.

The current solution to avoid outbound reshuffles, incrementing the total retrieval time in compact storage systems, is to use the dedicated storage policy in which every lane is dedicated to only one product, originating from one production batch. Hence, if a unit load of a product is requested, the load located at the rack's front face can be retrieved without any reshuffles. However, this policy does not use the known order information, and results in low space utilization ('honeycombing') due to partially filled lanes (Tompkins et al., 2010). As a consequence, additional lanes have to be used for the same number of unit loads to be stored, leading to longer S/R machine travel time and potentially longer lead time. Our sample company is no exception. It uses different storage lanes in the racks for each batch of a product, leading to partially filled lanes. However, the major advantage is that no reshuffles are required when trucks have to be loaded.

In this study, we propose a shared storage policy which allows unit loads of different products to share the same storage lane. We make use of customer order information (e.g. arrival time of outbound trucks) to avoid reshuffles during the outbound process, and minimize the response time simultaneously. A time window is used to realistically capture the arrival time instant of each outbound truck. Variations in actual arrival time instants, which can be observed from historical data, can be used to determine robust time windows. However, outbound trucks may still arrive outside their dedicated time windows. Uncertainty in truck arrival will increase the risk of reshuffling with shared storage.

In this study we focus on a compact cross-dock system for fresh produce. It turns out shared storage often outperforms dedicated storage. However, depending on system design and operational parameters, dedicated storage might be preferred. Examples of such parameters are rack depth, or the number of products that have to be stored. If the rack is not very deep, dedicated storage might be preferred, as a good rack utilization might be achieved by storing a single product with a small number of unit loads per storage lane. By using shared storage, little improvement in space utilization can be obtained, while the risk of extra reshuffling due to improperly sequenced stored loads increases. In a similar fashion, a large number of products to be stored favors shared storage, as dedicated storage requires a separate storage lane for every product, potentially leading to low rack utilization and larger travel distances.

We formulate a shared storage policy that potentially avoids reshuffling, as a mixed-integer model. The objective is to minimize total retrieval time by determining where to store unit loads in the compact storage system. The mathematical model is proven to be strongly NP-complete and so we propose an efficient heuristic to solve the problem. In addition, we test the robustness of the shared-storage policy to violations in outbound truck time windows. Shared storage appears to lead to shorter retrieval times and better storage lane utilization in most instances we tested. Dedicated storage yields shorter total retrieval times only in few scenarios, depending on rack design, order and storage profiles. Therefore, in most cases warehouse managers should preferably use shared, rather than dedicated, storage in a fast-turnover automated compact cross-dock system in order to serve the customer faster, to avoid quality loss, and to increase throughput and storage space utilization.

In the next section, we review previous studies on different storage policies in compact storage systems and assignment policies in cross-docks. In section 3, we give an overview of the configuration of the system under study. Furthermore, we develop a mathematical model to optimize the storage operation by minimizing the total retrieval time. A solution algorithm is proposed in section 4. In section 5, we introduce an approach to obtain a lower bound to measure the performance of the solution algorithm. Section 6 illustrates the numerical results for our reference company for problems of different sizes. We compare our proposed shared storage heuristic with dedicated storage by considering uncertainty in section 7. Finally, section 8 concludes the paper and proposes further research.

2. Literature review

Different decision problems of cross-dock systems have been studied in the literature. For a comprehensive review on cross-dock systems we refer to a paper by Boysen and Fliedner (2010). The problem of locating single or multiple cross-dock systems in a distribution network is studied in the literature on hub location problems, e.g., by Klose and Drexl (2005) and Chen (2007). Bartholdi and Gue (2004) investigate the layout and the shape of a cross-dock system. Many papers study operational-level decision problems in cross-dock systems. A commonly studied problem is the assignment of destinations (mid-term horizon decision) and trucks (short-term horizon decision) to dock doors (e.g. Tsui and Chang, 1992, Gue, 1999, Bartholdi and Gue, 2000, Lim et al., 2005, Lee et al., 2006, Bozer and Carlo, 2008, and Miao et al., 2009). However, few papers focus on minimizing the total retrieval time of a block of retrievals stored in a cross-dock system. Stadtler (1996) introduces a five-module storage and retrieval assignment concept for a deep lane storage system where a lift takes care of vertical movement. He assumes a cross-dock process where the expected retrieval instant is known at the time of storage. He also assumes that retrieval as well as storage requests consist of a number of unit loads of the same product and that they are therefore stored together in a storage lane of the rack.

Several papers have analyzed the performance of different storage systems and policies. De Koster et al. (2008) and Roodbergen and Vis (2009) provide a comprehensive literature review on AS/RSs. We here only review papers on storage policies and compact storage systems.

Four main classes of storage policies can be distinguished (Francis et al. 1991): random, dedicated, class-based, and shared storage policy based on the duration of stay (DOS). With a *random storage policy*, each unit load has an equal chance of being stored in any of the storage locations. The random storage policy is studied broadly in the literature (Hausman et al., 1976, Bozer and White, 1984, Lee and Elsayed, 2005 and De Koster et al., 2008). Random storage requires the fewest data since no product information is used in determining storage assignment (Goetschalckx and Ratliff, 1990). On the other hand, the optimal dedicated storage and class-based storage policies require historical information of the turnover rates for each product. A *dedicated storage policy* in a 2D system involves the assignment of specific storage locations to each product. In a compact storage system, this means each lane accommodates unit loads of only one product. The dedicated storage policy in 2D systems has been studied extensively in the literature (e.g. Montulet et al., 1998, Malmborg, 1996). An early study of a dedicated storage policy is on the cube-per-order index (COI) rule by Heskett (1963). As a compromise between dedicated and random storage, a *class-based* dedicated storage policy is frequently used in which the unit loads are partitioned into a small number of classes based on their turnover rates. The class of unit loads with the highest turnover rate is assigned to locations closest to the I/O point. Storage is random within a class. Using product turnover frequency, class-based and dedicated storage policies may reduce S/R machine travel time substantially compared to a random storage policy (e.g. Hausman et al., 1976, Graves et al., 1977, Rosenblatt and Eynan, 1989, Eynan and Rosenblatt, 1994, Kouvelis and Papanicolaou 1995, Kulturel et al., 1999, Van den Berg and Gademann, 2000, Foley et al., 2004). Park and Webster (1989b) present a class-based storage policy in a compact system where total travel time is minimized.

With a *shared storage policy* based on DOS, different products can share a common storage location over time if their DOS does not overlap, which allows more flexible use of space than allowed by a dedicated storage policy. This provides the opportunity to reduce the maximum effective storage area and to better utilize the more desirable storage locations (Chen et al., 2010). Goetschalckx and Ratliff (1990) show that, for single-command storage and retrieval, a DOS-based shared storage policy potentially decreases S/R machine travel time compared to dedicated storage, random storage, and class-based storage policies. Chen et al. (2010) address the combined location assignment and interleaving problem in an AS/RS with a shared storage policy based on DOS. Their objective is to minimize the travel time of an S/R machine considering storage assignment and interleaving at the same time. The authors provide an integer-programming model to solve the model optimally.

Compact storage systems have not yet been widely studied in the academic literature. Although Park and Webster (1989a, 1989b) appear to be the first to study compact storage systems, they actually study 2D pallet storage systems with multiple aisles. Gue (2006) develops models for very high-density storage systems, in which interfering unit loads have to be moved to gain access to desired unit loads. Gue and Kim (2007) study a “puzzle-based” compact storage system where a unit load can move in *x* and *y*-directions as long as an empty slot is available next to it. Sari et al. (2005) study a flow rack compact storage system where the pallets are stored and retrieved at different rack sides by two S/R machines responsible for storage and retrieval respectively. De Koster et al. (2008) and Yu and De Koster (2009, 2012) study a compact storage system with built-in multi-deep circular conveyors. The system is fully automated, and every pallet stored is accessible individually by rotating conveyors. Zaerpour et al. (2013) show that the optimal shape of a compact storage system, which minimizes the response time for single command cycles, is independent of the used storage policy. This means decisions on the shape of the storage system and storage policy can be decoupled simplifying system design. Zaerpour et al. (2014) study a live-cube compact storage system where shuttles carry the loads at each level in *x*- and *y*- directions while a lift takes care of vertical movement. They derive closed-form expressions for the optimal configuration and investigate the environmental and financial aspects of such systems.

The previous papers study cross-dock processes, compact storage systems and policies in isolation. However, as many cross-docks apply temporary (compact) storage in order to partly decouple inbound and outbound flows, it is necessary to integrate these systems and policies in system optimization decisions, which is the purpose of this paper. Different from Goetschalckx and Ratliff (1990), shared storage is now defined as a storage policy which simultaneously accommodates different products in a single multi-deep lane. We focus on overall performance, integrating the storage, retrieval, and shipping processes in such a cross-dock.

3. Problem description, assumptions, and general model

The compact storage system studied in this paper is sketched in Figure 1. The system consists of a compact storage rack, an I/O point (or depot) and an S/R machine (an automated crane) with an attached satellite.

The storage rack has R rows, C columns, and I depth tiers. $J = R \times C$ is the number of lanes in the rack. The S/R machine is capable of moving in both vertical and horizontal directions simultaneously. Therefore, its travel time is determined by the Chebyshev metric. While the S/R machine waits in front of the lane, the satellite attached to the S/R machine can move inside the lane to store or retrieve unit loads multi-deep. The S/R machine's pick-up or drop-off time of a unit load at the I/O point or a storage location is assumed to be constant. All storage and retrieval movements start with the S/R machine at the I/O point (located at the lower left-hand corner of the rack). Unit loads retrieved are dropped off by the S/R machine at the I/O point on an outgoing conveyor for further transport.

We are interested in cross-docking situations where orders are known at product receipt. The system operates in a shift-based mode: products received during the night shift have to be buffered and will be shipped during the next day shift according to appointment. Upon storage of a unit load, the exact arrival time of its departure truck is not yet known. Instead, a time window is used which captures all possible arrival time instants. The truck is allowed to arrive anywhere within this time window without imposing any reshuffling to the system. This makes our model robust to deviations from the truck's appointment time.

Therefore a time window is defined as follows:

Definition 1 (**time window**). The time window of an outbound truck refers to a time interval which captures all possible arrival time instants of the truck.

We start with the assumptions and then present the travel-time model for storage assignment. Most of our assumptions are rather standard in AS/RS literature. When the S/R machine is idle, it stops at the depot. The pick-up and deposit time of a unit load is constant; therefore, it is omitted from the minimization model. In addition, since the system operates in a shift-based mode, the S/R machine only uses single command (SC) cycles with only product storages during the night shift and product retrievals during the day shift. Moreover, we assume a cross-dock concept, i.e. all customer orders are known at product receipt. Outbound trucks are assigned to the dock doors in first come, first served (FCFS) sequence. In addition, the S/R machine serves the retrieval requests of an outbound truck according to the FCFS rule. Reshuffling is avoided.

The main notations are:

Objective

TRT The total retrieval time of a given block of retrieval requests. A block includes all the unit loads which have to be retrieved during the day shift.

Decision variables

x_{ijk} $x_{ijk} = 1$ if the unit load k is assigned to lane j and depth i during the night shift, otherwise $x_{ijk} = 0$.

Sets and indices

i Depth tier index of a storage lane in the rack $1 \le i \le I$; the smaller the index of a tier is, the closer it is to the rack's front face.

j Lane index $1 \le j \le J$; the smaller index of lane, the closer it is to the I/O point.

k Index of unit load to be retrieved, $1 \le k \le K$.

$S(k)$ Set of unit loads that cannot be stored in front of unit load k to avoid reshuffling.

Parameters

X_j , Y_j, Travel time of the S/R machine from the depot to lane j and then back to the depot in horizontal and vertical directions, respectively.

Z_i Travel time of the satellite machine to transport a unit load from the front face of the lane to depth tier i without interference and then back to the front face of the lane.

We obtain the total retrieval travel time given by Equation (1) and the model of the storage assignment problem as below (Model P):

Model P:

$$\min \quad TRT = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk}[Max\{X_j, Y_j\} + Z_i] \tag{1}$$

Subject to

$$\sum_{k=1}^{K} x_{ijk} \le 1 \qquad for \qquad i = 1,...,I,\ j = 1,...,J, \tag{2}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ijk} = 1 \qquad for \qquad k = 1,...,K, \tag{3}$$

$$x_{ijk} + \sum_{m \in S(k)} x_{ojm} \le 1 \qquad for \qquad i = 2,...,I,\ j = 1,...,J, k = 1,...,K, o = 1,...,i-1, \tag{4}$$

$$x_{ijk} \in \{0,1\} \qquad for \qquad i=1,...,I, j=1,...,J, k=1,...,K, \qquad (5)$$

where x_{ijk} represents the assignment of unit load k to lane j and depth i. Since the S/R machine can move in horizontal and vertical directions simultaneously, the time needed to go from the I/O point to the front face of lane j in Equation (1) is $Max\{X_j, Y_j\}$. Furthermore, the satellite machine cannot move in the depth dimension until the S/R machine is positioned at the lane's front face. Therefore, the satellite machine's travel time (Z_i) is added to the S/R machine's travel time. Constraints (2) ensure that every storage location can accommodate at most one unit load. Constraints (3) make sure that every unit load is assigned to exactly one storage location. Constraints (4) ensure that reshuffling is prevented by defining an appropriate set S for each unit load. In order to define set S for a unit load, we consider the outbound truck in which it is contained. Assume trucks A and B arrive at the outbound area with time windows $[i_A, j_A]$ and $[i_B, j_B]$, respectively. Then we might have two situations:

a) If $j_A < i_B$ or $j_B < i_A$: In this case, the unit loads a and b of the two trucks A and B, respectively, can be assigned to the same lane. If $j_A < i_B$, truck A arrives earlier than truck B. Therefore, the unit loads of truck A should be assigned to the positions in front of unit loads of truck B to avoid reshuffling, if products are not identical. Hence, $b \in S(a)$ and $a \notin S(b)$. If $j_B < i_A$, truck B arrives earlier than truck A. Therefore, unit loads of truck B should be stored in front of unit loads of truck A to avoid reshuffling, if the products are not identical. Hence, $a \in S(b)$ and $b \notin S(a)$. If unit loads a and b contain the same product, they can be assigned to any locations of the same lane without any reshuffles, i.e. $a \notin S(b)$ and $b \notin S(a)$.

b) If $i_A < i_B < j_A$ or $i_A < j_B < j_A$ or $i_B < i_A < j_B$ or $i_B < j_A < j_B$: In this situation, the time windows of trucks A and B overlap and so the arrival sequence of trucks A and B cannot be determined. Thus, reshuffling cannot be avoided if the unit loads (of different products) of trucks A and B are assigned to the same lane. Therefore, two unit loads of different products of trucks A and B cannot be stored in the same lane. If a and b are two unit loads of different products of trucks A and B respectively, then set S for unit loads a and b can be defined as $b \in S(a)$ and $a \in S(b)$. If no information of arrival times of outbound trucks is available, the model will automatically assign unit loads of the same product to each lane (dedicated-storage policy). Having more accurate information of truck arrival times results in more narrow time windows with smaller chance of overlap, and hence more unit loads of different products can share the same lane.

The most difficult constraints are constraints (4). In Appendix A, we prove the problem is strongly NP-complete. Exact algorithms fail to solve the model optimally even for relatively small problem sizes. For real-life problems, the number of variables and constraints might be very large. Therefore, heuristic methods have to be developed to find solutions.

4. Solution construction and improvement (C&I) algorithm

Our C&I algorithm includes three main steps, which results in a shared storage policy. Step 1 defines an 'Ideal' storage area used to benchmark feasible solutions of Model P. Step 2 constructs a feasible solution that can be improved later to a near optimal solution. Step 3 consists of local improvements of the feasible solution and finds a near optimal solution.

To further explain the steps, we use illustrative example 1. Appendix B refers to the input parameters of example 1.

Step 1. Ideal boundary construction

We drop constraints (4) in Model P and then obtain Model R. Model R is a simple assignment problem that can be solved easily in polynomial time (Munkres, 1957). Its optimal solution provides a lower bound of the objective value of Model P.

Optimally solving Model R gives a solution for assigning K unit loads into a storage area in the rack. Obviously, the optimal solution of Model R results in the Ideal storage area and boundary defined below.

Definition 2 (the Ideal storage area). The Ideal storage area is the area consisting of the K closest storage locations to the I/O point.

Definition 3 (the Ideal boundary). The Ideal boundary is the boundary of the Ideal storage area.

By solving example 1, we obtain the Ideal storage area and boundary shown in Figure 2(a). Figure 2(a) illustrates the top view of a one-level rack (for every number d or $d(b)$ in each cell of Figures 2(a-e), d represents the distance from the cell location to the I/O point and b represents the truck index). The storage locations within the bold lines form the Ideal storage area and the border of the Ideal storage area forms the Ideal boundary. In case of a sufficiently large continuous rack, the top-viewed shape of the Ideal storage area equals a quarter circle.

If we can locate the K unit loads within the Ideal boundary such that constraints (4) are satisfied, then an optimal solution of Model P is obtained. However, in most cases unit loads will be shipped by different trucks with overlapping time windows, and relocations of unit loads cannot possibly satisfy constraints (4) without violating the Ideal boundary. Steps 2 and 3 below aim to find a storage area for feasibly assigning the K unit loads which violates the Ideal boundary as little as possible. As a result, these steps may provide a near optimal solution for Model P.

Step 2. Feasible solution construction

This step aims to find a feasible solution of Model P by using a graph representation approach. In Step 2.1, we represent the time windows of all the unit loads in a directed graph. Step 2.2 assigns unit loads to the storage rack without violating constraints (4).

Step 2.1. Graph representation

Because all the unit loads stored during a night shift will be shipped by trucks during the next day shift, the time window of each unit load is determined by the time window of its shipping truck. Hence, all the unit loads of one truck have the same time window. To represent the time windows of the unit loads, we group the K unit loads by truck and define the directed graph $G=(V,E)$. V is the set of vertices corresponding to the earliest and latest arrival times of all outbound trucks in the day shift. $V=\{0,1,2,...,\mathrm{T}\}$, where 0 and T represent the starting and ending times in the day shift. E is the set of directed arcs, where an arc represents the time window of all unit loads that will be shipped by the same truck. Each arc starts from a time node corresponding to the earliest arrival time of a truck (starting point of its time window) at the warehouse and ends at a time node corresponding to the latest arrival time of the truck (ending point of its time window). Each arc is weighted with the number of unit loads to be shipped by the truck which is called the length of the arc in the rest of paper.

In example 1, $V=\{0,1,2,3,4\}$ as shown in Figure 3. The arc $T1(2)$, for example, represents the time windows of 2 unit loads to be shipped by truck 1.

We aim to identify paths in this graph that connect unit loads of different trucks which can share the same storage lanes in the rack without reshuffling upon retrieval. We first define a partition set:

Definition 4 (a lane-sharing set). A lane-sharing set is a partition of all arcs into subsets with the following properties:

1) The time windows of any two arcs within each of the subsets do not overlap.

2) The subsets are 'maximal', i.e., any arc of a subset of a lane-sharing set cannot be added to another subset containing arcs with more unit loads, without violating property 1.

3) If an arc of a subset of a lane-sharing set is swapped with an arc of another subset containing arcs with more unit loads, it cannot increase the number of unit loads in the latter subset without violating property 1.

4) Given the subsets are maximal, they have minimal time window length; i.e. if two arcs of two different subsets of the partition have the same number of unit loads and they can be swapped without violating property 1, the arc belonging to the subset with more unit loads has a shorter time window.

Now, a lane sharing path can be defined as follows:

Definition 5 (a lane-sharing path). A lane-sharing path is a partition subset of a lane-sharing set.

Obviously, the unit loads in a path can be assigned to the same storage lanes without reshuffling if the unit loads of a truck arriving earlier are stored closer to the rack's front face than those arriving later. Any directed graph can be decomposed into a group of such paths.

Figure 4 shows a partition of the graph of Figure 3 into three lane-sharing paths which form a lane-sharing set with all four properties in Definition 4.

Step 2.2 addresses how to select lane-sharing paths and thereby to construct a feasible solution.

Step 2.2. Initial feasible solution construction

In this step, the C&I heuristic tries to obtain an initial feasible solution by selecting lane-sharing paths from a graph and assigning the unit loads based on the selected paths.

We first select a longest path in terms of the number of unit loads included in it. If there is a tie, we select the path with the shortest time window length (i.e. the time window between the earliest arrival time of the earliest truck in the path and the latest arrival of the latest truck in the path). To select a second path, we first remove the previous one from the graph and then select the longest one in the remaining part of the graph. The above process terminates when all the arcs have been removed. Eventually, an array of paths is obtained with non-increasing path lengths.

In Figure 3, the first path consists of arcs $T1(2)$ and $T2(2)$ with a length of 4 (=2+2) unit loads. By removing the longest path from Figure 3, we obtain Figure 5. The second path consisting of arcs $T3(1)$ and $T4(2)$ is selected from Figure 5. Figure 4 gives the result which consists of ordered paths 1, 2, and 3 for example 1.

The unit loads of the trucks, corresponding to the paths are now assigned to the storage locations within the Ideal boundary, in sequence. Note that a storage lane closer to the I/O point has a larger number of storage locations within the Ideal boundary (see Step 1). Hence, in order to find a good initial feasible solution for Model P, the unit loads in a longer path are assigned to storage lanes closer to the I/O point. Within a path, the unit loads of a truck with earlier arrival time are assigned to storage locations closer to the rack's front face in a lane, or to a lane closer to the I/O point if multiple lanes are used for the path. As unit loads of a truck all have the same time window, they can be assigned arbitrarily to storage locations in a rack lane. However, when the assignment process continues, some unit loads may not fit in the Ideal storage area. In this case, the locations closest to I/O point but outside the Ideal boundary are attached to the Ideal boundary.

As shown in Figure 2(b), unit loads in the first path are assigned to the storage locations of the first two lanes closest to the I/O point without violating the Ideal boundary and constraints (4). The unit loads of truck 1 arriving earlier than those of truck 2 are assigned to the storage locations closer to the rack's front face in the first lane. Unit loads of the second path are assigned to the 3rd and 4th lanes without violating the Ideal boundary and constraints (4). A unit load of truck 5 is stored in a location in the rightmost lane that is outside the Ideal area.

Theorem 1. *The output of Step 2 is a feasible solution of Model P.*

Proof. In Step 2, if a unit load is assigned to a location, no other unit load can be assigned to the same location, therefore constraints (2) are satisfied. If a unit load is assigned to a location, it will be removed from the graph and therefore it will not be assigned to any other location. The process repeats until all arcs are removed and therefore constraints (3) are also satisfied. Since unit loads from a truck with an earlier arrival time window are assigned to locations closer to the rack's front face, constraints (4) are satisfied and the output of Step 2 is a feasible solution of Model P. □

The results of the assignment produce an *actual assigned storage area*. If the area is the same as the Ideal storage area, the optimal solution of Model P has been found. Otherwise, we improve the solution in Step 3, in four sub-steps.

Step 3. Improvement of a feasible solution

If the actual assigned storage area is not the same as the Ideal storage area, the relocation of a unit load from outside the Ideal storage area into an empty location in the Ideal storage area will reduce total retrieval time (see Steps 3.1-3.3). However, this kind of relocation cannot always be achieved without violating constraints (4). Therefore, Step 3.4 consists of relocating unit loads to reduce the distance of the boundary of the actual storage area (see Step 3.4).

Step 3.1. Rearrange each path independently for future improvement

In the feasible solution of Step 2, unit loads in a path may be assigned to multiple lanes, so that some unit loads of a late arriving truck will occupy storage locations close to the front face of the rack in a lane. Empty locations behind these unit loads but within the Ideal boundary are difficult to use without violating constraints (4). Therefore, if these unit loads are swapped with unit loads of earlier arriving trucks in the same path without violating constraints (4), the empty locations behind can potentially accommodate more unit loads in next steps.

This step swaps a unit load assigned to a storage location with empty locations behind it with a unit load in the same path but of an earlier arriving truck and assigned to a deeper storage location without violating constraints (4). In example 1, path1 consists of arcs representing trucks 1 and 2 where truck 1 arrives earlier than truck 2. Figure 2(b) shows that there is an empty storage location within the Ideal boundary behind the unit load of truck 2 in the second lane. Therefore, this unit load will be swapped with the unit load of truck 1 located at the second depth tier of the first lane. Figure 2(c) illustrates the result of Step 3.1.

Step 3.2. Improve the solution by letting different paths share the same lane

After rearranging each path independently, this step tries to improve the solution by combining different paths of a graph. Unit loads of different paths can be assigned to empty locations of one of their lanes if their time windows do not overlap. To do so, select a longest path from the array of paths obtained in Step 2.2 (in our example, path 1 in Figure 4). If its storage lanes contain an empty location, check if any unit load of a shortest path in the array can be assigned to that empty location without violating constraints (4). Otherwise, select the second shortest path from the array. The process sequentially selects all shorter paths and then proceeds with the second longest path and repeats until there is no path to be selected.

In example 1, the storage location at the second depth tier of the second lane (Figure 2(c)), located within the Ideal boundary, is empty. Therefore, the objective value can be reduced by relocating the unit load located at the front face of the sixth lane, which is outside the Ideal boundary, to the empty storage location. This relocation reduces the objective value by eight units (the travel time is the time it takes to move from the I/O point to the point and then back to the I/O point; $2\times(12-8)=8$). The final output of Step 3.2 is shown in Figure 2(d). As the Ideal boundary and the actual storage assignment boundary coincide, the result of this step is optimal for the example.

Step 3.3. Improve the solution by considering products

Different unit loads of the same product can be swapped even if they have overlapping time windows. The first product to be selected is the one with the most unit loads. The graph is updated by adding a path representing unit loads of the selected product and adjusting the weights of the other arcs. This step then tries to reduce the objective value by grouping unit loads of the selected product and repeating the previous steps (Step 2.2 onward) for the updated graph. The new storage assignment is accepted if the objective value is reduced. Next, the product with the second most unit loads is selected. The process repeats until all products have been selected.

An updated graph for example 1 considering product “A” is shown in Figure 6. Each of the trucks 1 and 2 ships one unit load of “A”. Therefore, the weights of the arcs representing trucks 1 and 2 have been reduced by one unit load and the new path “A” has a length of 2 unit loads. The time windows of the unit loads in path “A” determine the starting point and ending point of path “A”.

The result of Step 3.3 is shown in Figure 2(e). The total retrieval time of the assignment in Figure 2(e) is still identical to the total retrieval time of the result of the previous step and so it is still optimal.

Step 3.4. Improve the solution by swapping two different lanes

In the final solution of Step 3.3, some lanes close to the I/O point may store a smaller number of unit loads than remote lanes. The solution may be improved by swapping lanes as follows: The more unit loads assigned to a lane, the closer the lane should be to the I/O point.

The final solution of the C&I heuristic is shown in Figure 7(a); comparing the result of the heuristic with its equivalent dedicated storage policy (Figure 7(b)) shows approximately 9% reduction in total retrieval time.

In order to test the quality of the C&I heuristic, we provide a tight lower bound in the next section.

Since Step 3.2 is the most complicated step, it determines the computational complexity of our C&I algorithm. In this step, unit loads of different paths can share the same lane if their trucks’ time windows do not overlap. In the worst case, all unit loads need to be compared pairwise. In addition, based on Step 3.3, the previous steps should be replicated based on the number of product types which equals the total number of unit loads in the worst case. Therefore, our C&I algorithm has a polynomial time complexity of $O(n^3)$ where n is total number of unit loads.

5. Lagrangean relaxation approach

This section provides a tight lower bound of the minimum objective value of Model P by using a Lagrangean relaxation approach. The lower bound can be used to evaluate the performance of the feasible solution found in Section 4; the feasible solution gives an upper bound of the minimum objective value of Model P. If the gap (i.e., ((the upper bound - the lower bound)/the lower bound)×100%) is small, the feasible solution is a good near optimal solution.

The constraints that complicate the solution of Model P are constraints (4), which ensure that reshuffling is avoided. We relax the constraints by introducing a set of non-negative Lagrange multipliers $\lambda = (\lambda_{ijko})_{(I-1)\times J\times K\times(i-1)}$ and then obtain a Lagrangean relaxed model as follows:

Model RP:

$$\min \quad TRT_{\lambda} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk}[Max\{X_j, Y_j\} + Z_i] + \sum_{i=2}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{o=1}^{i-1} \lambda_{ijko}(x_{ijk} + \sum_{m\in S(k)} x_{ojm} - 1)\,, \qquad (6)$$

subject to Constraints (2), (3) and (5).

Let $D(\lambda)$ be the minimum objective value of Model RP for any Lagrange multipliers $(\lambda_{ijko})_{(I-1)\times J\times K\times(i-1)} \geq 0$. The Lagrangean dual problem of Model RP, denoted by DP, can be formulated as:

Model DP:

$$\max_{\lambda\geq 0} D(\lambda)\,,$$

where λ is the vector of decision variables of Model DP.

According to the theory of Fisher (1981), for any given $(\lambda_{ijko})_{(I-1)\times J\times K\times(i-1)} \geq 0$,

$$D(\lambda) = TRT_{\lambda}^{*} \leq TRT^{*}\,,$$

where TRT_{λ}^{*} and TRT^{*} are the minimum objective values of Models RP and P. Therefore, the larger $D(\lambda)$ is, the better lower bound it provides for TRT^{*}. Model RP is a simple assignment problem that can be solved in polynomial time. The question is how to find good $(\lambda_{ijko})_{(I-1)\times J\times K\times(i-1)} \geq 0$ that can provide a larger $D(\lambda)$. The sub-gradient method is used here to update $(\lambda_{ijko})_{(I-1)\times J\times K\times(i-1)} \geq 0$. The procedure of the approach, based on the paper by Yu et al. (2008), is as follows:

Step 0: Initialize λ^t and θ^t at $t = 0$ where *t* is the iteration step ($\lambda^0 = 0, \theta^0 = 1$).

Step 1: Solve the relaxed Model RP to obtain TRT^t (i.e., $D(\lambda)$) that is the current dual value with a given λ^t (a lower bound).

Step 2: Set the step size s^t in iteration *t* by

$$s^t = \beta \frac{TRT^* - TRT^{[t]}}{\left\| g^t \right\|^2},$$

where β is a parameter with $0 < \beta < 1$, TRT^* of Model P is estimated by $\left(1 + \omega / \theta^\rho\right) TRT^{[t]}$. $TRT^{[t]}$ is the best dual value obtained prior to iteration *t,* parameters ω and ρ are random numbers within the their intervals where $\omega \in [0.1, 1.0]$, $\rho \in [1.1, 1.5]$. The value of θ in iteration t+1 is given by $\theta^{t+1} = \max(1, \theta^t - 1)$ if $TRT^t > TRT^{[t]}$, otherwise $\theta^{t+1} = \theta^t + 1$ and

$$\left\| g^t \right\|^2 = \sum_{i=2}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{o=1}^{i-1} \left(x_{ijk} + \sum_{m \in S(k)} x_{ojm} - 1 \right)^2.$$

Note that $s^t \geq 0$.

Step 3: Update the Lagrange multipliers in iteration $t + 1$

$$\lambda_{ijko}^{t+1} = \max \left\{ \lambda_{ijko}^t + s^t \left(x_{ijk} + \sum_{m \in S(k)} x_{ojm} - 1 \right), 0 \right\}.$$

Step 4: Check the stop criterion, which is given by either 1) The dual value TRT^t has not improved for a given number of iterations, or 2) A given maximum number of iterations has been reached. If the criterion is met, stop and output the results, otherwise set $t = t + 1$ and go to Step 1.

6. Numerical results

In this section, we test the quality of our C&I heuristic. In addition, we compare the performance of the C&I heuristic with a dedicated storage policy. Our base example is based on data from a company in the Netherlands which distributes fresh vegetables and fruits to retail customers in Europe in a shift-based cross-dock operation. The company has a refrigerated compact storage system consisting of five identical modules, each with an S/R machine and satellite using dedicated storage. Products with a larger inventory are stored in the lanes closer to the I/O point.

Input parameters describing orders placed by customers and parameters describing a single module of the storage system are shown in Table 1. The starting point of the time window of a truck arrival is randomly generated within the eight-hour day shift. The lengths of the time windows are varied. In the next step, the C&I heuristic is evaluated.

First, we present the results of the base example. Then, to evaluate the performance of our C&I heuristic for different sources of variation in the inputs of our model, a sensitivity analysis is performed. In each experiment, we vary one of the parameters of the base example over five different alternative values while the other parameters are fixed.

All test problems have been solved using MATLAB software on a portable computer Intel(R) Pentium(R) M1.86GHz with 512MB RAM and MS Windows XP professional. For each instance, including the base example, the results in the tables are the average of ten runs of simulated orders with corresponding time windows. The dedicated and shared storage policies are compared with the same data in every simulation run. Table 2 compares the solutions of the C&I heuristic and the dedicated storage policy for the base example. A one-tailed paired t-test ($H_1 : \mu_H < \mu_D$) shows the results of the C&I heuristic and the dedicated-storage policy differ significantly at a 5% level ($p < 0.0001$ for nearly all instances tested).

The results of the C&I heuristic for a varying number of orders and order size are shown in Tables 3 and 4, respectively. The results for a varying number of depth tiers are shown in Table 5. In order to make a fair comparison, the rack's capacity and the number of levels of the rack are assumed to be identical to those in the base example for all instances in Table 5.

The C&I heuristic performs very well, as the gaps between the results of the C&I heuristic and the lower bound are often zero and mostly less than 1%. For those solutions with zero gap, the best solution of the Lagrangean relaxation is obtained at $\lambda = 0$, i.e. it equals the optimal solution of Model R. Simultaneously, the C&I heuristic yields a solution within the Ideal boundary without any infeasibility. Therefore the gap becomes zero. However, the C&I heuristic cannot provide the optimal solution for every case (see Appendix C). The C&I algorithm finds near optimal solutions very fast; normally in less than one second. In addition, the relative improvement of the C&I heuristic compared to the dedicated storage policy ($(TRT_D - TRT_H)/TRT_D$) decreases with an increasing number of orders and order sizes (see Tables 3 and 4). As expected from the results in De Koster et al. (2008) the total retrieval time of the C&I heuristic (TRT_H) and dedicated storage policy (TRT_D) are convex functions of the number of depth tiers (see Table 5). Table 5 also shows that with an increasing number of depth tiers the relative improvement of the C&I heuristic compared to the dedicated storage policy increases.

In the next section we compare the performance of both the dedicated and shared storage policy for uncertainties in the arrival times.

7. Impact of uncertainty on storage policy selection

In our model formulation, the time window of an outbound truck is defined as a time interval which captures possible arrival time instants of the departing truck. However, in real life an outbound truck might arrive outside its allocated time window. In order to evaluate the performance of the shared-storage policy in the case time windows are violated, we expose our C&I heuristic to higher levels of uncertainty using a Monte Carlo simulation.

We assume the actual arrival time of each outbound truck is normally distributed with known mean μ and standard deviation σ. However, the actual arrival time can fall outside the time window. Although in practice very early arriving trucks might be asked to wait until their scheduled arrival, in our experiments they are assumed to be served in a first come, first served sequence, potentially leading to more reshuffles. By increasing the level of uncertainty, the preset time window becomes less accurate. A given time window can capture $\mu \pm 3\sigma$, $\mu \pm 2.5\sigma$, …, $\mu \pm 0.5\sigma$ percentile of the actual arrival time instants depending on the level of uncertainty. For instance, an outbound truck with a time window that captures $\mu \pm 3\sigma$ of the actual arrival time distribution, arrives within its time window with a probability of 0.997.

If a truck arrives outside its time window, reshuffling might be required. For this purpose, buffer lanes are used to which interfering loads, blocking the required loads, are moved. Current practice is to use empty lanes closest to the ones that are occupied. Interfering loads are placed back in their original locations after the required load has been retrieved. For our experiments, we take the base example shown in Table 1. Each of the input parameters (number of orders, order size, number of depth tiers, time window length, number of products) is varied under different levels of uncertainty. The means of the arrival times of outbound trucks are uniformly distributed within the eight-hour shift period (from 6 a.m. to 2 p.m.). For each scenario, we randomly generate 10 instances (truck arrivals) and calculate the average number of required reshuffles and average reshuffling time for the C&I heuristic under different levels of uncertainty.

Table 6 gives both the total retrieval time of the C&I heuristic excluding reshuffling time (TRT_H) and the reshuffling time ($ResT$). The table shows that more frequent arrivals of outbound trucks outside their time windows add more reshuffling time to the shared-storage policy (see right-hand side columns of Table 6) and vice versa.

We next discuss the impact of different levels of uncertainty on each input parameter and managerial insights.

Figures 8(a-e) compare shared and dedicated storage for various scenarios and under different levels of uncertainty. TRT_D and TRT_S ($TRT_S = TRT_H + ResT$) represent the total retrieval time for dedicated storage and shared storage (including the reshuffling time). $(TRT_D - TRT_S)/TRT_D$ shows how shared storage performs in comparison with dedicated storage. In scenarios where $(TRT_D - TRT_S)/TRT_D > 0$, shared storage performs better and in scenarios where $(TRT_D - TRT_S)/TRT_D < 0$, dedicated storage performs better. We make the following observations.

- Shared storage is nearly always better; it is very robust to disturbances for different order profiles, rack configurations, and uncertainty levels.
- In Figures 8(a-e), all curves representing $\mu \pm 1.5\sigma$,…, $\mu \pm 3\sigma$ are very close together. This changes for higher uncertainty levels. The C&I shared storage heuristic is therefore virtually insensitive to time window violations as long as the probability that trucks arrive within their time windows is at least 85% (i.e. when the time window can at least capture $\mu \pm 1.5\sigma$ of the actual arrival time distribution).
- In case of a very high level of uncertainty (Figure 8(a)), and when the number of orders is large (e.g. during the peak seasons), dedicated outperforms shared storage. This is due to the fact that multiple orders share the same storage lane in combination with uncertainty in truck arrival times.
- In case of a very shallow rack (Figure 8(b)), the difference between shared storage and dedicated storage is negligible and in combination with a high level of uncertainty, dedicated storage performs better. In such a scenario, dedicated storage can often fully use the storage lanes closer to the I/O point. On the other hand, when the uncertainty level in arrival information of outbound trucks increases, shared storage cannot compensate the additional reshuffling time due to improper storage of products.

- The performance of shared storage declines with larger order size (Figure 8(c)), smaller number of products (Figure 8(d)), and wider time windows (Figure 8(e)). With larger order size, the performance of shared storage declines as the rack utilization and the number of necessary reshuffles increase. A smaller number of products favors dedicated storage as it allows to store more loads of the same product at the same lane. With wider time windows, unit loads of fewer trucks can share the same lane, jeopardizing the performance of shared storage.

Shared storage often outperforms dedicated storage for the practical instances tested and under different levels of uncertainty. However, situations exist where dedicated storage performs better even with reasonably reliable information. As an example, we consider a large number of small size orders from an assortment of only 3 products stored in very deep storage lanes (see Figure 8(f)). In this case, any time window violation imposes reshuffling to the system while dedicated storage can fully accommodate the storage lanes with single products (the variety of products is limited). In this case, dedicated storage outperforms C&I shared storage.

8. Conclusion

In fresh produce distribution centers, energy cost (mainly for controlling the temperature and humidity of storage area) takes up a large share of total operational costs. This cost is proportional to the volume of the storage system (Singh, 2008). As a solution, fully automated compact storage systems are used for temporarily storing refrigerated products. However, to retrieve a unit load stored multi-deep in such a system, the unit loads in front of it have to be removed first. This reshuffling increases the complexity of the storage and retrieval process and also increases the retrieval time of the unit loads. In practice, the dedicated storage policy is used to avoid reshuffling resulting in partially filled storage lanes. As a result, remote storage lanes have to be used increasing the total retrieval time. A lane-sharing storage policy is proposed to increase the lane utilization and reduce the total retrieval time. The main difficulty of shared storage is additional reshuffling time due to improper assignment of unit loads of different products to the same lane. However, in a cross-dock, the outbound truck arrival schedule is known in advance and can be used to avoid reshuffling. For each outbound truck a time window is defined such that it realistically captures the actual arrival time of the truck. Since an exact lane-sharing model is strongly NP-complete, we introduce a C&I heuristic based on a greedy construction and an improvement part. The gap between the C&I heuristic and

Lagrangean relaxation of the exact model appears to be very small indicating the C&I heuristic gives near optimal solutions, normally within a second computation time. We show that the C&I with a shared storage policy is robust to slight disturbances in outbound truck arrivals (i.e. when the probability of trucks arriving within their time windows is at least 85%). When this probability decreases further, the performance of shared storage starts declining. Our results show that the shared storage policy using the C&I heuristic outperforms the dedicated storage policy for most practical instances tested. The reason is that in a multi-deep compact storage system, the C&I heuristic fully occupies the storage lanes closer to the I/O point with unit loads of different products. The retrieval time saving achieved by using these storage lanes compensates additional reshuffles. As a result, shorter response times for customers can be achieved, a prime concern for managers, in particular for fresh produce. In addition, due to higher utilization of storage locations, more space will be available for serving new customers. This means, a true space "compaction" at both design and operational decision levels can be achieved. Using space better is a prime objective for warehouse managers, as it directly and indirectly lowers operational cost. However, if the storage system has shallow storage lanes (e.g. a three-deep storage system) a warehouse manager is still advised to use dedicated storage.

Several research questions regarding the shared storage policy in a compact storage system remain open. While we have studied satellite-based compact storage systems, results for other compact storage system configurations may also prove worthwhile investigating. In addition, some cross-docks have overlapping storage and retrieval processes. This would imply different storage and retrieval policies than studied here.

Acknowledgment

In this research, the first author was partly supported by Dinalog, in the project Cargo Driven Intermodal Transport. The second (corresponding) author was supported by the Thousand Young Scholar Program grants, the National Science Foundation of China (NSFC) for Distinguished Youth Scholars (No. 71225002), and NSFC grants 71110107024 and 71121061, and the Fundamental Research Funds for the Central Universities.

Appendix A: NP-completeness proof

In order to show our problem is strongly NP-complete we consider a special case of the problem. Our problem will be proven to be strongly NP-complete if our problem in this special case is strongly NP-complete.

To construct the special case we assume

$$Z_i << \max\left\{X_j, Y_j\right\} \qquad for\, i = 1...I, J = 1...J\ , \tag{7}$$

indicating the depth movement travel time is negligible compared to horizontal and vertical movement travel times. This holds when the satellite machine moves much faster than the S/R crane does. It implies that K unit loads should be stored in the smallest number of lanes closest to the I/O point without violating Constraints (4) in order to minimize (1).

Hence, with the assumption given by Equation (7), Model P is equivalent to the following model:

Model SP:

$$\min \quad TRT = \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ijk}[Max\{X_j, Y_j\}]\ , \tag{8}$$

subject to constraints (2), (3), (4) and (5).

As a solution of Model SP, we must know how to assign the K unit loads into the smallest number of lanes. Model SP is equivalent to minimizing the required number of storage lanes for storing K unit loads subject to constraints (2), (3), (4), and (5) (denoted by Model TP). Therefore, an optimal solution of Model TP is also an optimal solution of Model SP. If we can prove Model TP is strongly NP-complete, we then know Model SP, which is a special case of Model P, is strongly NP-complete.

In the following, we transform the problem defined by Model TP to be the Graph C-colorability problem.

Definition (Graph C-colorability (Chromatic number) problem): Define a graph where V and E are the sets of vertices and undirected edges. A positive integer C, $C \le |V|$, represents the total number of colors. Each vertex must be colored with only one color of $\left\{1, 2, ..., C\right\}$ and any two vertices on the same edge have to be colored differently. The question is: Is G C-colorable, i.e. is there a function $f : V \rightarrow \left\{1, 2, ..., C\right\}$ such that $f(u) \ne f(v)$ whenever $(u, v) \in E$?

Suppose each truck has one unit load and there are no two unit loads of the same product. Each vertex of the graph represents a shipping truck and if trucks u and v have overlapping time windows, vertices u and v are connected by an edge. A color can then correspond to a storage lane.

We need to make two steps:

1. Show that the problem belongs to NP.
2. Show that a strongly NP-complete problem reduces to our problem.

To demonstrate (1) it suffices to remark that a nondeterministic algorithm can check in polynomial time whether a given solution is feasible (i.e. $f(u) \neq f(v)$ whenever $(u,v) \in E$).

To demonstrate (2) we remark the formulation of Model TP exactly matches the Graph C-colorability (Chromatic number) problem which is proven to be strongly NP-complete by Garey and Johnson (1979). Hence, we can conclude that Model SP and then Model P are strongly NP-complete.

Appendix B: Illustrative example 1

Consider a single-level compact storage system with a rack configuration as given in Table 7. Figure 9 illustrates a top view of the rack. In each storage slot the travel distance to the I/O point is given. The time windows of unit loads are shown in Table 8.

Appendix C: The optimality of the C&I heuristic

In order to investigate the optimality of the C&I heuristic, we searched for a counterexample in small size problems. We obtained a counterexample that the C&I heuristic cannot provide the optimal solution and a gap exists between the optimal solution and the heuristic solution.

Consider a rack with 5 lanes and 2 depth tiers. Figure 10 shows a top view of the rack. The I/O point is located at lower left of the rack. The numbers in storage locations indicate the distance of the storage locations to the I/O point in time units.

Table 9 shows the orders that have to be shipped to customers in the next shift.

Figure 11(a) shows the results of the C&I heuristic and Figure 11(b) shows the optimal solution for the proposed example. The 8% gap between the total retrieval times of the solution provided by the C&I heuristic and the optimal solution proves that the C&I heuristic cannot guarantee the optimal solution for every instance.

References

Bartholdi, J.J., K.R. Gue. 2000. Reducing labor costs in an LTL cross-docking terminal. *Operations Research* 48 823-832.

Bartholdi, J.J., K.R. Gue. 2004. The best shape for a cross-dock. *Transportation Science* 38 235-244.

Bartholdi, J. J., S. T. Hackman. 2011. Warehouse and Distribution Science: Release 0.95. http://www.warehouse-science.com. Accessed 21 December 2012.

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

Bozer, Y.A., J.A. White. 1984. Travel-time models for automated storage/retrieval systems. *IIE Transactions* 16(4) 329-338.

Bozer, Y.A., H. Carlo. 2008. Optimizing inbound and outbound door assignments in less-than-truckload cross-docks. *IIE Transactions* 40 1007-1018.

Chen, J.F. 2007. A hybrid heuristic for the uncapacited single allocation hub location problem. *Omega* 35 211-220.

Chen, L., A. Langevin, D. Riopel. 2010. The storage location assignment and interleaving problem in an automated storage/retrieval system with shared storage. *International Journal of Production Research* 48(4) 991-1011.

De Koster, M.B.M. 1996. Ontwerp van een eenvoudig magazijn. In: J.P. Duijker et al. (Eds.), *Praktijkboek Magazijnen/Distributiecentra* 3.10A.01-3.10A.20 Deventer: Kluwer.

De Koster, M.B.M., T. Le-Duc, Y. Yu. 2008. Optimal storage rack design for a 3-dimensional compact AS/RS. *International Journal of Production Research* 46(6) 1495-1514.

Eynan, A., M.J. Rosenblatt. 1994. Establishing zones in single-command class-based rectangular AS/RS. *IIE Transactions* 26 (1) 38-46.

Fisher, M.L. 1981. The Lagrangian relaxation method for solving integer programming problems. Management Science 27(1) 1-18.

Foley, R.D., S.T. Hackman, B.C. Park. 2004. Back-of-the-envelope miniload throughput bounds and approximations. *IIE Transactions* 36(3) 279-285.

Francis, R.L., F. McGinnis, J.A. White. 1991. *Facility Layout and Location: An Analytical Approach,* 2nd ed., Prentice Hall.

Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman.

Goetschalckx, M., H.D. Ratliff. 1990. Shared storage policies based on the duration stay of unit loads. *Management Science* 36(9) 1120-1132.

Graves, S.C., W.H. Hausman, L.B. Schwarz. 1977. Storage-retrieval interleaving in automatic warehousing systems. *Management Science* 23(9) 935-945.

Gue, K.R. 1999. The effect of trailer scheduling on the layout of freight terminals. *Transportation Science* 33 419-428.

Gue, K.R. 2006. Very high density storage systems. *IIE Transactions* 38(1) 79-90.

Gue, K.R., B.S. Kim. 2007. Puzzle-based storage systems. *Naval Research Logistics* 54(5), 556-567.

Hausman, W.H., L.B. Schwarz, S.C. Graves. 1976. Optimal storage assignment in automatic warehousing systems. *Management Science* 22(6) 629-638.

Heskett, J.L. 1963. Cube-per-order index – A key to warehouse stock location. *Transport and Distribution Management* 3 27-31.

Hu, Y.H., S.Y. Huang, C. Chen, W.J. Hsu, A.C. Toh, C.K. Loh, T. Song. 2005. Travel time analysis of a new automated storage and retrieval system. *Computers & Operations Research* 32(6) 1515-1544.

Klose, A., A. Drexl. 2005. Facility location models for distribution system design. *European Journal of Operational Research* 162 4-29.

Kouvelis, P., V. Papanicolaou. 1995. Expected travel time and optimal boundary formulas for a two-class-based automated storage/retrieval system. *International Journal of Production Research* 33(10) 2889-2905.

Kulturel, S., N.E. Ozdemirel, C. Sepil, Z. Bozkurt. 1999. Experimental investigation of shared storage assignment policies in automated storage/ retrieval systems. *IIE Transactions* 31 739-749.

Lee, M.K., E.A. Elsayed. 2005. Optimization of warehouse storage capacity under a dedicated storage policy. *International Journal of Production Research*. 43(9) 1785-1805.

Lee, Y.H., J.W. Jung, K.M. Lee. 2006. Vehicle routing scheduling for cross-docking in supply chain. *Computers & Industrial Engineering* 51 247-256.

Lim, A., Z. Miao, B. Rodrigues, Z. Xu. 2005. Transshipment through cross-docks with inventory and time windows. *Naval Research Logistics* 52 724-733.

Malmborg, C.J. 1996. Storage assignment policy tradeoffs. *International Journal of Production Research* 34(2) 363-378.

Miao, Z., A. Lim, H. Ma. 2009. Truck dock assignment with operational time constraint within cross-docks. *European Journal of Operational Research* 192 105-115.

Montulet, P., A. Langevin, D. Riopel. 1998. Minimising the peak load: an alternate objective for dedicated storage policies. *International Journal of Production Research* 36(5) 1369-1385.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics* 5(1) 32-38.

Park, B.C. 2001. An optimal dwell point policy for automated storage/ retrieval systems with uniformly distributed, rectangular racks. *International Journal of Production Research* 39(7) 1469-1480.

Park, Y.H., D.B. Webster. 1989a. Modeling of three-dimensional warehouse systems. *International Journal of Production Research* 27(6) 985-1003.

Park, Y.H., D.B. Webster. 1989b. Design of class-based storage racks for minimizing travel time in a three-dimensional storage system. *International Journal of Production Research* 27(9) 1589-1601.

Roodbergen, K.J., I.F.A. Vis. 2009. A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research* 194 343-362.

Rosenblatt, M.J., A. Eynan. 1989. Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Science* 35 (12) 1519-1524.

Sari, Z., C. Saygin, N. Ghouali. 2005. Travel-time models for flow-rack automated storage and retrieval systems. *International Journal of Advanced Manufacturing Technology* 25 979-987.

Singh, R.P. 2008. Benchmarking study of the refrigerated warehousing industry sector in California. *Public interest energy research program contract and research project reports*. California energy commission.

Stadtler, H. 1996. An operational planning concept for deep lane storage systems. *Production and Operations Management* 5 266-282.

Tompkins, J.A., J.A. White, Y.A. Bozer, J.M.A. Tanchoco. 2010. *Facilities Planning* 4th ed., John Wiley & Sons.

Tsui, L.Y., C.H. Chang. 1992. An optimal solution to dock door assignment problem. *Computers and Industrial Engineering* 23 283-286.

Van den Berg, J.P., A.J.R.M. Gademann. 2000. Simulation study of an automated storage/retrieval system. *International Journal of Production Research* 38 (6) 1339-1356.

Yu, Y., H. Chen, F. Chu. 2008. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research* 189(3) 1022-1040.

Yu, Y., M.B.M. de Koster. 2009. Designing an optimal turnover-based storage rack for a 3D compact automated storage and retrieval system. *International Journal of Production Research* 47(6) 1551-1571.

Yu, Y., M.B.M. de Koster. 2012. Sequencing heuristics for storing and retrieving unit loads in 3D compact storage and retrieval systems. *IIE Transactions* 44(2) 69-87.

Zaerpour, N., M.B.M. de Koster, Y. Yu. 2013. Storage policies and optimal shape of a storage system. *International Journal of Production Research* 51(23-24) 6891-6899.

Zaerpour, N., Y. Yu, M.B.M. de Koster. 2015. Small is beautiful: a framework for evaluating and optimizing live-cube compact storage systems. *Transportation Science.* forthcoming.

List of Tables

Table 1. Parameters related to orders and storage system

Parameters	Base example	Range for scenarios	Fixed parameters of storage system
Time window length (hours)	1	[0.001, 8]	
Number of orders	36	[1, 60]	
Order size (number of unit loads)	10	[1, 15]	
Number of products (PT)	10	[1, 40]	
Number of depth tiers	5	[1, 11]	
Duration of shift (hours)			8
Size of pallet (centimeters)			width=120, height=260, depth=80
Size of storage location (centimeters)			width=150, height=300, depth=100
Size of rack (storage slots)			level=3, depth tier=5, lanes/level=24
Size of rack (meters)			width=36, height=6.4, depth=5
S/R machine horizontal speed (v_x) (m/s)			2
S/R machine vertical speed (v_y) (m/s)			0.5
Satellite speed (v_z) (m/s)			0.25
Nominal capacity of S/R machine (No. of unit loads/hour)			60

Table 2. Comparison of solutions of C&I heuristic and dedicated storage policy for the base example

No. of orders	Order size	PT	TRT_D	TRT_H	Imp (%)	P-value	CT_H	TRT_{LR}	Gap (%)	# runs Gap=0	CT_{LR}
36	U[1,10]	10	6373.9	5513.1	13.6%	<0.0001	0.9	5490.1	0.4%	4	73.9

Note: (1) *PT*, *Gap* and *#runs Gap*=0 are the total number of products, gap between C&I heuristic and LR (Lagrangean relaxation), and the number of runs out of 10 runs with gap=0, respectively.

(2) TRT_x is the total average retrieval time for method x (in seconds), where x = D (dedicated storage), H (C&I heuristic), LR (Lagrangean relaxation).

(3) CT_x is the average computational time per instance for method x (in seconds), where x = H (C&I heuristic), LR (Lagrangean relaxation).

(4) $\text{Imp} = (\frac{TRT_D - TRT_H}{TRT_D}) \times 100\%$.

Table 3. Sensitivity analysis for a varying number of orders

No. of orders	TRT_D	TRT_H	Imp (%)	P-value	CT_H	TRT_{LR}	Gap (%)	# runs Gap=0	CT_{LR}
1	36.2	30.8	16.9%	.0125	0.5	30.8	0.0%	10	3.4
10	1130.0	816.2	28.0%	<.0001	0.1	806.3	1.2%	3	276.1
20	3015.6	2422.2	19.9%	<.0001	0.4	2410.9	0.5%	5	78.2
50	9479.9	8816.5	7.1%	<.0001	1.1	8792.7	0.3%	5	95.3
60	10532.9	9960.8	5.4%	<.0001	1.4	9944.2	0.2%	5	84.9

Table 4. Sensitivity analysis for varying order sizes

Order size	TRT_D	TRT_H	Imp (%)	P-value	CT_H	TRT_{LR}	Gap (%)	# runs Gap=0	CT_{LR}
U[1, 1]	685.2	469.5	31.4%	<.0001	0.2	462.4	1.5%	3	251.2
U[1, 3]	1717.8	1315.5	23.4%	<.0001	0.5	1305.8	0.8%	5	94.0
U[1, 5]	2955.8	2369.0	20.0%	<.0001	0.5	2354.2	0.6%	5	67.2
U[1, 8]	4930.4	4119.5	16.6%	<.0001	0.8	4097.6	0.5%	4	75.7
U[1, 15]	9282.1	8588.0	7.6%	<.0001	1.0	8567.2	0.2%	6	83.9

Table 5. Sensitivity analysis for a varying number of depth tiers

Depth tiers	TRT_D	TRT_H	Imp (%)	P-value	CT_H	TRT_{LR}	Gap (%)	# runs Gap=0	CT_{LR}
1	11384.0	11384.0	0.0%	_	0.2	11384.0	0.0%	10	44.1
3	5529.0	5355.9	3.1%	<.0001	0.5	5341.9	0.3%	3	66.7
7	7178.2	5636.0	21.6%	<.0001	0.9	5614.8	0.4%	6	72.2
9	7847.2	5997.5	23.7%	<.0001	0.8	5969.8	0.4%	8	100.1
11	9445.6	7097.6	25.1%	<.0001	1.1	7057.4	0.6%	6	126.0

Table 6. Comparison of the C&I heuristic and dedicated storage in real application

Variable	Range	TRT_D (1)	TRT_H (1)	$\mu \pm 3\sigma$ NRes (2)	$\mu \pm 3\sigma$ ResT (3)	$\mu \pm 2.5\sigma$ NRes	$\mu \pm 2.5\sigma$ ResT	$\mu \pm 2\sigma$ NRes	$\mu \pm 2\sigma$ ResT	$\mu \pm 1.5\sigma$ NRes	$\mu \pm 1.5\sigma$ ResT	$\mu \pm \sigma$ NRes	$\mu \pm \sigma$ ResT	$\mu \pm 0.5\sigma$ NRes	$\mu \pm 0.5\sigma$ ResT
# orders	20	2968.2	2379.8	0.1	6.7	0.2	13.6	0.2	13.6	0.2	13.6	0.3	21.3	1.2	79.1
	50	9325.8	8614.8	0.5	19.1	1.4	63.7	1.4	70.0	1.6	71.4	6.7	378.7	11.4	610.1
	60	10520.0	9792.0	1.7	72.5	3.5	215.5	3.6	223.4	3.8	234.8	9.2	566.5	***25.2***	***1588.0*** (4)
Order size	[1, 3]	1714.8	1288.0	0.0	0.0	0.2	11.9	0.3	18.3	0.3	18.3	0.3	21.8	1.6	115.0
	[1, 8]	4758.4	3952.8	0.2	14.4	0.2	14.8	0.2	14.8	0.3	21.2	0.7	34.3	4.6	304.8
	[1, 15]	10138.0	9525.4	1.3	74.3	2.3	115.2	2.2	101.9	3.1	187.1	4.6	233.5	9.5	501.2
# depth tiers	3	5627.8	5456.4	0.3	30	0.4	37.8	0.5	40.4	0.5	44.5	1.7	165.8	***2.5***	***243.8***
	9	8273.6	6438.4	3.2	135.6	3.6	152.6	3.6	153	4.9	206.2	7.4	303.2	18.2	749.9
	11	10047.0	7704.2	3.8	210.2	4.5	221.9	5.1	242.2	5.8	304.4	13.6	681.8	27.4	1231.2
Time window length (hour)	0.5	6205.6	5306.1	0.0	0.0	0.0	0.0	0.2	8.0	0.5	17.0	0.6	28.5	3.0	194.3
	1	6200.5	5345.3	0.3	13.9	0.5	30.9	0.7	41.0	1.2	63.7	2.0	129.6	4.5	272.0
	1.5	6195.3	5388.9	0.5	17.1	1.5	78.8	2.0	121.6	2.5	148.2	3.8	219.2	7.2	430.1
# products	5	5918.6	5455.4	0.5	20.4	0.7	36.2	0.9	45.1	0.9	43.6	1.9	99.6	3.5	186.6
	20	6254.1	5532.1	0.4	23.0	0.6	39.2	0.8	41.4	0.9	57.6	1.6	117.5	3.7	225.5
	30	6424.1	5662.1	0.7	28.5	0.8	32.4	1.0	46.1	1.3	77.2	2.4	124.2	5.7	338.9

(1) TRT_x is the total average retrieval time for method x (in seconds), where x = D (dedicated storage), H (C&I heuristic), excluding reshuffling time.

(2) *NRes* is the average number of reshuffles for C&I heuristic.

(3) *ResT* is the average reshuffling time for C&I heuristic (in seconds).

(4) Bold and italic numbers represent situations where $TRT_D < TRT_H + ResT$

Table 7. Rack configuration and system parameters

Parameter	value
Number of levels	1
Number of lanes	7
Number of depth tiers	3

Table 8. Earliest and latest arrival times of trucks

Truck index	Earliest arrival time	Latest arrival time	Latest - Earliest(hours)	Number of unit loads	Indices of unit loads	Products
1	0	1	1	2	{1,2}	{A, C}
2	1	3	2	2	{3,4}	{A, D}
3	0	2	2	1	{5}	{B}
4	2	3	1	2	{6,7}	{E, F}
5	2	4	2	2	{8,9}	{B, G}

Table 9. Orders placed by five different customers in a shift

Order	Order size	Product	Time window ($a<b<c$)
O_1	2	A,B	[a, b]
O_2	1	C	[b, c]
O_3	1	D	[a, c]
O_4	1	D	[a, c]

List of Figures

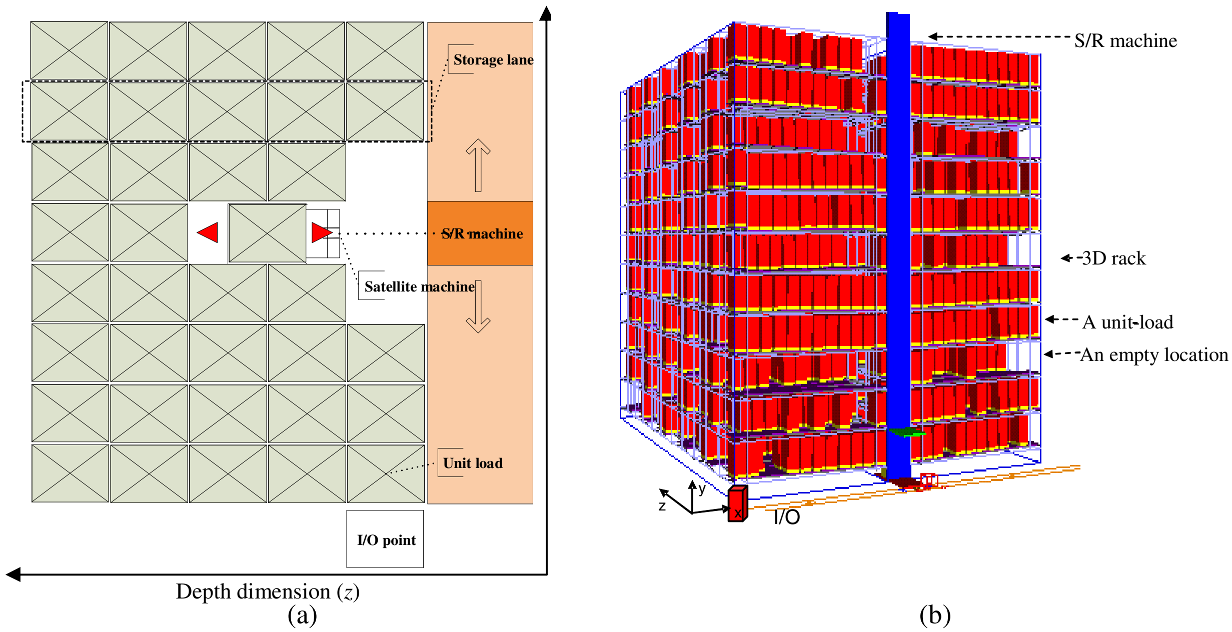
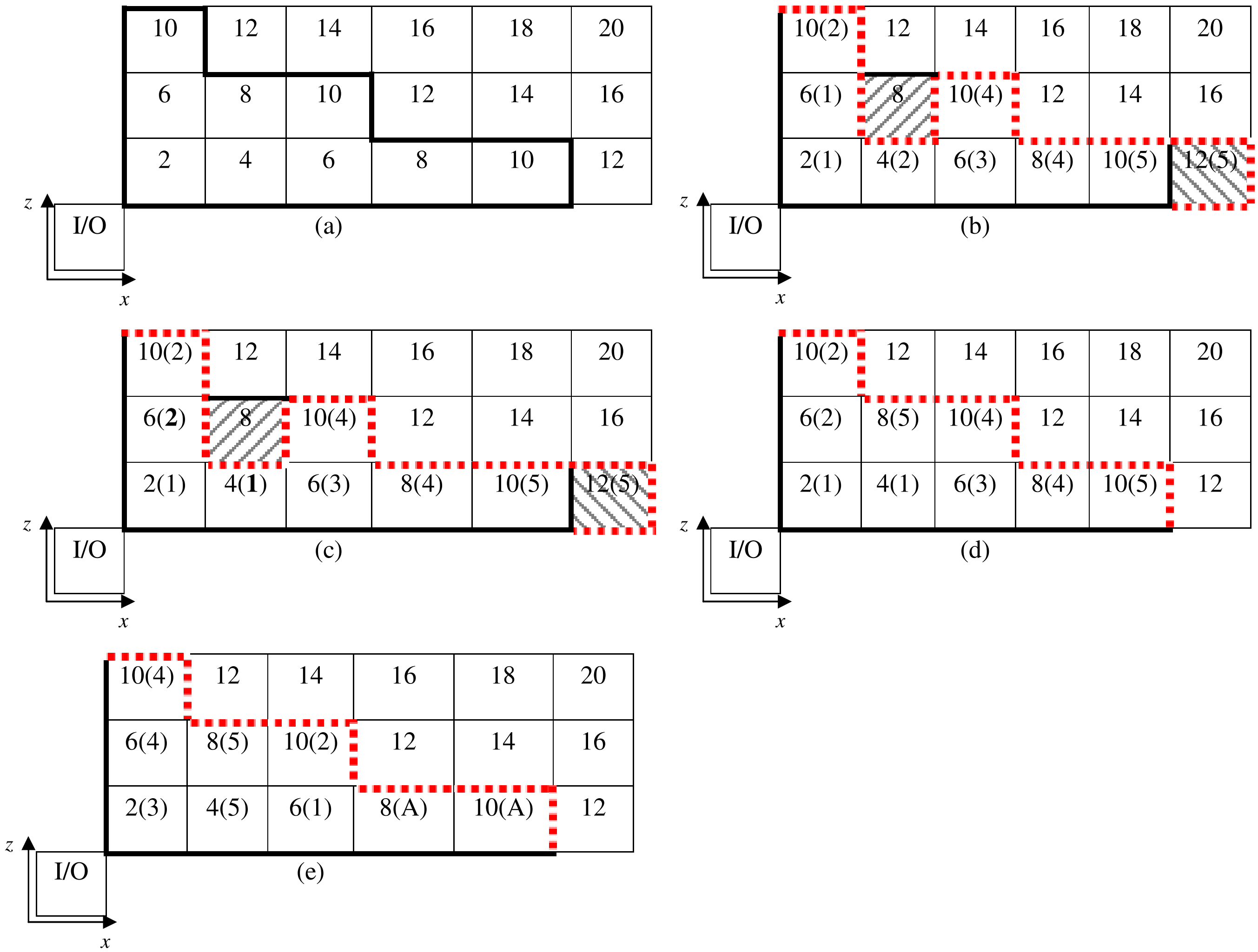


Figure 1. (a)Top view (b) 3D view of a compact storage system



Notes. (1) The bold line forms the Ideal boundary.

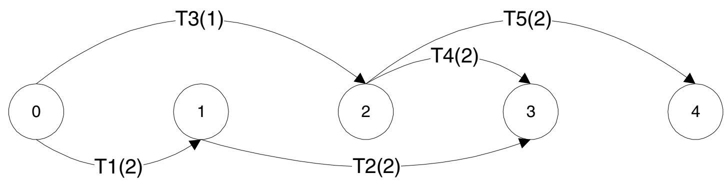
(2) The dotted line gives the boundary from an actual feasible unit load assignment.

(3) represents an empty storage location within the Ideal boundary.

(4) represents an occupied storage location outside the Ideal boundary.

(5) The highlighted numbers represent the swapped unit loads in Figure 6(c).

Figure 2. Improvement of storage assignment over different steps



Note. *Ta*(*b*) represents the arc for truck *a* that ships *b* unit loads.

Figure 3. Truck time window graph

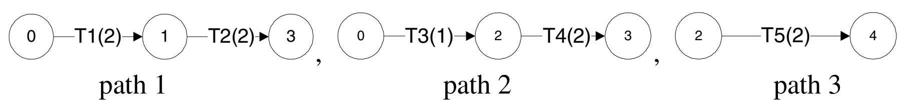


Figure 4. A set of lane-sharing paths for the above example

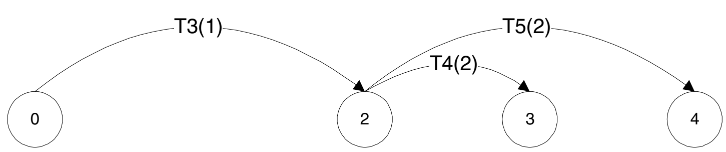


Figure 5. The graph obtained by removing the longest path in Figure 2

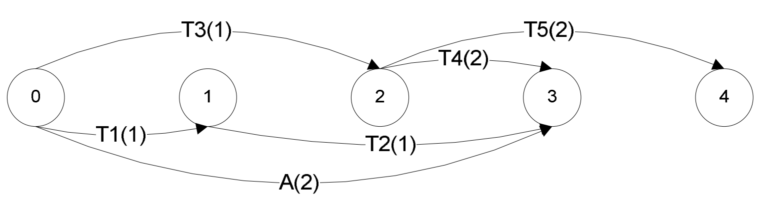


Figure 6. Updated graph considering product "A"

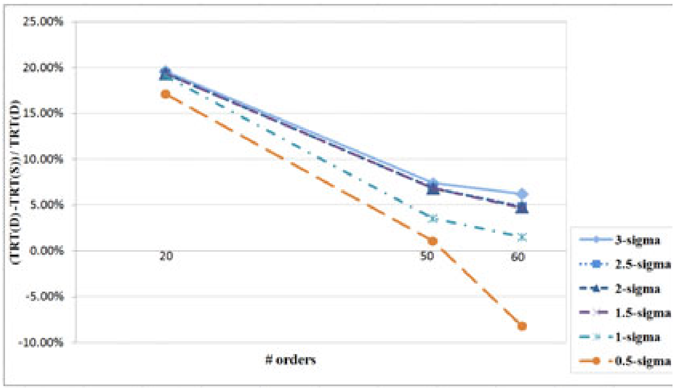
10(4)	12	14	16	18	20	22
6(4)	8(5)	10(2)	12	14	16	18
2(3)	4(5)	6(1)	8(A)	10(A)	12	14

10	12	14	16	18	20	22
6(A)	8(B)	10	12	14	16	18
2(A)	4(B)	6(C)	8(D)	10(E)	12(F)	14(G)

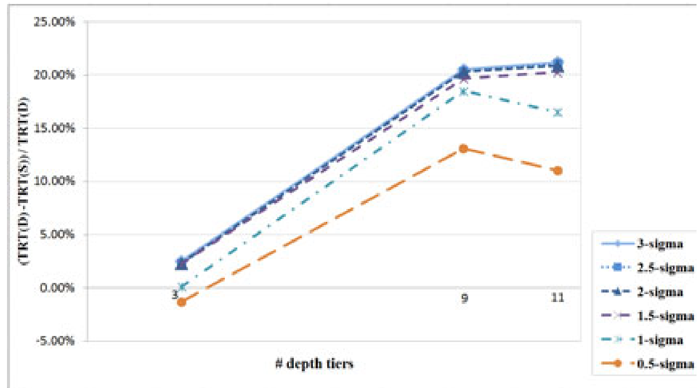
Note. Products are denoted by capital letters

(a) (b)

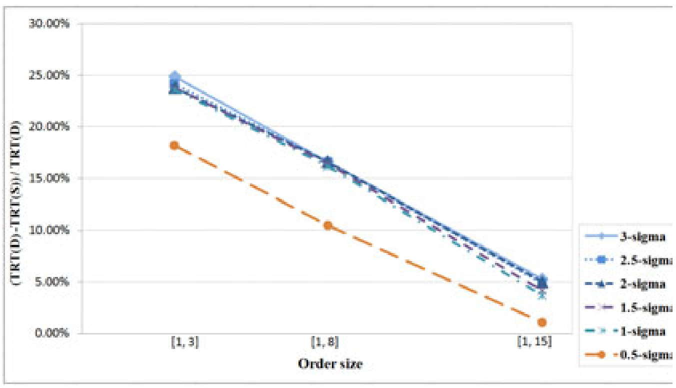
Figure 7. (a) Solution of the shared-storage heuristic, (b) Solution of the dedicated-storage policy



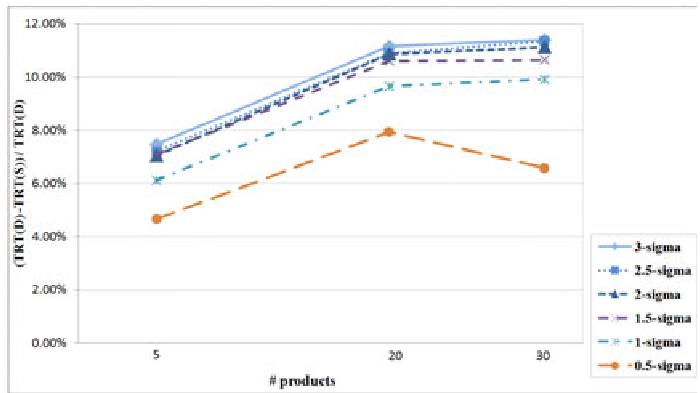
(a)



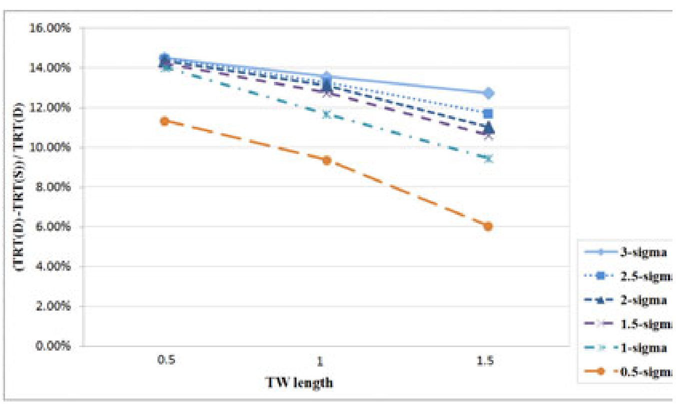
(b)



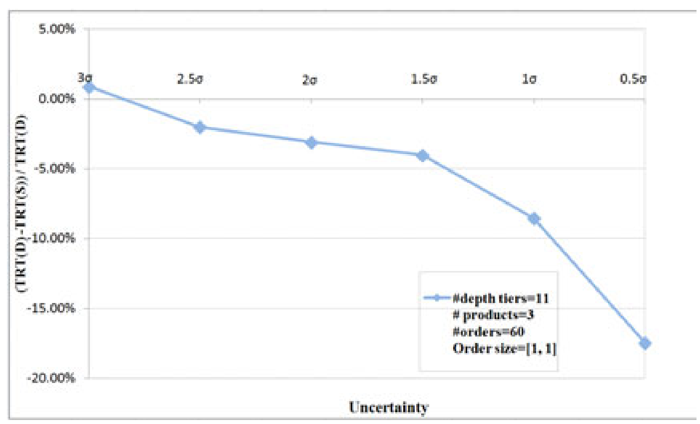
(c)



(d)



(e)



(f)

Note. *TRT*(*D*) and *TRT*(*S*) represent the total retrieval time for dedicated and shared storage (including reshuffling time), respectively.

Figure 8. Comparison of dedicated and shared storage for various scenarios and uncertainty levels

	10	12	14	16	18	20	22
	6	8	10	12	14	16	18
z	2	4	6	8	10	12	14
I/O							
x							

Figure 9. Travel distance of each storage location to the I/O point

	2.1	3.1	4.1	5.1	6.1
	1	2	3	4	5
I/O					

Figure 10. Travel time distance of storage locations from I/O point

	O_2				
	O_1	O_1	O_3	O_4	
I/O					

a) Heuristic solution (TRT=24.2 time units)

	O_2	O_4			
	O_1	O_3	O_1		
I/O					

b) Optimal solution (TRT= 22.4 time units)

Figure 11. (a) Solution of the heuristic, (b) optimal solution for the counterexample