

University of Groningen

Cross docking for libraries with a depot

van der Heide, G.; Roodbergen, K.J.; van Foreest, N.D.

Published in:
European Journal of Operational Research

DOI:
[10.1016/j.ejor.2020.08.034](https://doi.org/10.1016/j.ejor.2020.08.034)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2021

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

van der Heide, G., Roodbergen, K. J., & van Foreest, N. D. (2021). Cross docking for libraries with a depot. *European Journal of Operational Research*, 290(2), 749-765. <https://doi.org/10.1016/j.ejor.2020.08.034>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.



Innovative Applications of O.R.

Cross docking for libraries with a depot

G. Van der Heide*, K. J. Roodbergen, N. D. Van Foreest

University of Groningen, P.O. Box 800, Groningen 9700 AV, the Netherlands

ARTICLE INFO

Article history:

Received 8 March 2019

Accepted 18 August 2020

Available online 25 August 2020

Keywords:

Inventory control

Dynamic programming

Heuristics

Libraries

Rentals

ABSTRACT

Library organizations in the Netherlands show an increasing interest to employ depots for low-cost storage and demand fulfillment of item requests. Typically, all libraries in an organization have a shared catalog, and, on local unavailability, requests can be shipped from elsewhere in the organization. The depot can be used to consolidate shipment requests by making tours along all libraries, delivering requested items, but also picking up items that have to be stored at the depot, or that have to be shipped from one library to another. Cross docking and delayed shipments are two preferred methods for fulfilling requests that cannot be directly met using on-hand stock at the depot. In this paper, we compare these two methods from an inventory control perspective. We model the library system as a Markov Decision process. For one- and two-location systems, we derive analytical results for the average-cost optimal policy, showing that the decision to store items from the location at the depot satisfies a threshold structure depending on the number of rented items. For larger instances, an effective heuristic is proposed exploiting this threshold structure. In numerical experiments, important managerial insights are obtained by comparing cross docking and delayed shipments in different situations. Cross docking is shown to add most value in systems with low total stock, however, delayed shipments may achieve similar costs as cross docking when stock is high or when tours frequently visit all locations. Furthermore, effective decisions can be based on simple model formulations with memoryless rental time distributions.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

People read fewer books and, as a consequence, public library organizations face a declining membership and need to cut budgets due to lower revenues (Lammers, 2020). Libraries are simultaneously seeking new ways to contribute to society, for example, on social inclusion, equal access to information, and freedom of expression (Audunson et al., 2019). For this purpose, new activities are initiated and space in the library is reserved, for example, to host meetings. These trends induce a reduction in the number of books that are on display in libraries. In this societal context, we study the following setting. We consider a network of libraries cooperating in a system of interlibrary loans, a service where a customer of one library can borrow books from another library. Such system can serve to widen the variety of books that each library can make available to its customers. Furthermore, the libraries engage in jointly managing their collections. An extensive description of the essential underlying assumptions of our setting is given in Appendix A, where we also provide a number of examples from practice.

There is one depot, i.e., a central storage facility that supports the network of libraries. If a customer is looking for a particular book, there are several options. First, the customer may retrieve the book locally, i.e., from the shelves of the library where the customer is a member. Second, if there is no stock locally, the customer may request the book from the network. If the book is available at the depot, it can be transshipped from the depot to the local library. Finally, the book can be transshipped from another library to the local library. In all cases, the book is picked up and returned by the customer at the local library. This configuration is common in the Netherlands. It is also seen elsewhere, for example, in San Francisco where a central library supports 26 branch libraries located in residential neighborhoods (Apte & Mason, 2006), and in New York where a central processing facility exists that connects 150 involved libraries (Quandt, 2017).

In library networks with a depot, transportation usually takes place several times a week. The three types of transportation in the system are *shipments* (transport from the depot to meet requests at libraries), *take-backs* (transport from the libraries to store items at the depot), and *lateral transshipments* (transport between libraries to meet requests). Take-backs occur either with the intention of longer-term storage of books, or to pre-position books for upcoming demand in the network. Every transportation leg either originates at or is destined to the depot; books are not directly moved

* Corresponding author.

E-mail addresses: g.van.der.heide@rug.nl (G. Van der Heide), kj.roodbergen@rug.nl (K.J. Roodbergen), n.d.van.foreest@rug.nl (N.D. Van Foreest).

from one library to another library. Hence a lateral transshipment requires two transportation legs, first from the originating library to the depot, and second from the depot to the destination library. We distinguish two ways of handling lateral transshipments at the depot.

1. *Cross docking*. Two tours visit all libraries on the same day. All items that need to be laterally transshipped are exchanged at the depot in between the tours.
2. *Delayed shipments*. One tour visits all libraries. All items that need to be laterally transshipped are stored at the depot for one period and shipped at the next transport opportunity.

Numerical results and heuristics for delayed shipments can be found in [Van der Heide, Roodbergen, and Van Foreest \(2017\)](#). We introduce cross docking in this paper. We consider this system from an inventory perspective and optimize the operational decisions for shipments, cross docking, and take-backs. Our contributions are as follows. We derive the structure of the average cost optimal policy for a one-location problem and, under a mild restriction, for a two-location problem. The optimal policy for take-backs can be characterized by a series of thresholds on the on-hand inventories that depend on the number of loaned items at each location. Based on this threshold structure and other numerical insights, we develop a heuristic for a general number of locations. In numerical experiments, the heuristic is shown to be on average within 1% from the optimal costs.

Various managerial insights are provided by comparing situations with cross docking and delayed shipments. We find that, if sufficient stock is available in the system and the depot is frequently resupplied, both systems are equally effective. However, if stock in the system is quite low, then the cross-docking option is superior, which can be important considering the decreasing stock at libraries. The insights from this paper can assist library professionals to determine the best option for their situation. Other managerial insights concern the added value of the storage possibility at the depot, and the required information about the duration of borrowing for supporting effective decision making.

Depots are used similarly in other sectors. Construction companies, for example, share their expensive specialized equipment between their construction sites. Local hardware stores allow next-day delivery of rarely used tools from a nearby depot, while keeping frequently used tools in stock. An important factor that distinguishes libraries from other rental systems, is the higher willingness of customers to wait. In rental systems for products such as cars, clothing, and jewelry, customers may switch to a competitor on product unavailability, resulting in lost sales. Another difference is possible downtime for maintenance and cleaning of returned products. As also several similarities exist, and analytical results and insights in this paper may serve as inspiration for other rental systems, we deploy the common term “to rent” rather than “to borrow” or “to loan” in the remainder of this paper.

The outline of the article is as follows. Relevant literature is reviewed in [Section 2](#). Then, in [Section 3](#), we present the main assumptions and formulate the model as a Markov decision process. Analytical results for the optimal policy for the single-location and two-location problem are derived in [Sections 4](#) and [5](#). In [Section 6](#), a heuristic for the general problem is developed. Several experiments are carried out in [Section 7](#), in order to investigate the performance of the heuristic and to gain the discussed managerial insights. Finally, [Section 8](#) provides conclusions and directions for further research.

2. Literature

Since we consider a stochastic multilocation rental problem with a depot, we first review recent literature on stochastic rental

models. Afterward, we discuss closely related stochastic inventory control models.

Stochastic rental models can be grouped into single and multilocation models, with or without a depot. Single location rental models involve no depot and deal with particular issues such as setting the total rental stock for a finite time horizon ([Pasternack & Drezner, 1999](#); [Slaugh, Biller, & Tayur, 2016](#)), allocating demand to different customer classes ([Jain, Moinezadeh, & Dumrongsiri, 2015](#)), and analyzing usage of newly introduced products under a heterogeneous customer base ([Bassamboo, Kumar, & Randhawa, 2009](#)).

Multilocation rental models without a depot can be divided into two main streams. The first stream concerns optimizing rental fleets. For a finite horizon, [Baron, Hajizadeh, and Milner \(2011\)](#) optimize the allocation of rental products to several rental locations under various demand and return patterns. Transferring products between locations during the horizon is not considered. Models optimizing rental stock under a long-run average cost criterion are typically based on queueing theory. Given a policy for vehicle repositioning, [Papier and Thonemann \(2008\)](#) and [George and Xia \(2011\)](#) determine the long-run optimal fleet size. Long-run cost expressions are based on the availability of rental stock in loss models ([Papier & Thonemann, 2008](#)) and closed queueing networks ([George & Xia, 2011](#)).

The second stream concerns repositioning operations for rental stock. Such problems are commonly modeled as a Markov decision process (MDP) ([Puterman, 2009](#)). For a two location vehicle rental system, [Li and Tao \(2010\)](#) determine optimal fleet-sizes as well as the optimal policy for repositioning vehicles at the end of every period. [Brinkmann, Ulmer, and Mattfeld \(2019\)](#) and [Legros \(2019\)](#) apply MDPs for dynamic repositioning in bike-sharing systems. For a library setting, [Van der Heide and Roodbergen \(2013\)](#) optimize the trade-off between costs for lateral transshipment (in response to demand) and costs for repositioning (in anticipation of demand), without considering an option for low-cost storage.

Despite its practical relevance, only a limited number of authors have considered the use of a depot in multilocation rental models. [Van der Heide, Van Foreest, and Roodbergen \(2018\)](#) apply a queueing approach for the tactical problem of optimizing the inventory levels at the depot and each rental location under various types of backordering. However, it is not possible to dynamically reposition inventory based on the system state. Most related to our work is [Van der Heide et al. \(2017\)](#), who study a library system making use of delayed shipments. The authors numerically investigate optimal decisions for shipments and take-backs by solving MDPs. Numerical examples show that the optimal take-back policy has a state-dependent threshold structure, however, no analytical proofs are provided for the optimal policy structure. In this paper, we prove the policy structure in important special cases with one and two locations. In addition, we consider cross docking in our model and compare it to delayed shipments to generate new managerial insights.

Now we discuss related models in other application areas. In spare parts, a closely related concept is a quick-response warehouse that carries out shipments of spare parts to locations in response to stock-outs ([Axsäter, Howard, & Marklund, 2013](#)). [Howard, Marklund, Tan, and Reijnen \(2015\)](#) consider the use of pipeline information and optimize threshold policies for shipments. Demand is backordered if an order at a local warehouse will be delivered before a threshold time, otherwise it is met with a shipment from the quick-response warehouse. While we consider a less complex shipment policy, i.e., meet all demand, our rental system has several complicating factors not addressed in the spare parts system. Namely, there are stochastic and state-dependent returns, it is possible to cross dock, and the analytically convenient property that

inventory positions at each location are constant does not hold due to dynamic take-back decisions.

Another related problem is the repositioning of empty trucks in a hub-and-spoke system. Du and Hall (1997) determine effective heuristic threshold policies for sending empty trucks from the hub to the spokes and vice versa. Song and Carter (2008) provide the optimal control policy for a system with two spokes. By decomposing the system into several systems with a single spoke and a hub, they derive a heuristic policy that works well. We apply a similar approach in our heuristic by using structural results from the one location problem. In these papers, trucks are transferred between hubs and spokes, which in our rental terminology implies that products rented from a location are by definition returned to the depot and vice versa. The dynamics of such a hub-and-spoke system differ from the typical rental system where products are returned to the location they were originally rented from and thus require different policies.

In essence, the rental model can be seen as a lateral transshipment model with a specific transshipment structure; see Paterson, Kiesmüller, Teunter, and Glazebrook (2011) for an extensive review. The combination of lateral transshipment and returns is rarely considered. Ching, Yuen, and Loh (2003) and Tai and Ching (2014) consider lateral transshipment in combination with an exogenous return process, however, this does not appropriately capture the endogeneity of the return process in rental systems. Without returns, Wee and Dada (2005) show a threshold policy for a system with lateral transshipment between one location and one depot. We derive similar threshold policies, however, in our case the policy is state-dependent.

Inventory-routing problems also study the trade-off between transportation and inventory costs in systems with multiple stock points (Coelho, Cordeau, & Laporte, 2014). In some cases, lateral transshipment is also possible (Coelho, Cordeau, & Laporte, 2012). These problems are typically solved using a rolling-horizon approach, solving deterministic problems every period by substituting in forecasts of the stochastic demand. Recent literature on routing deals with dynamic dispatching of demanded products from a depot to delivery points, usually under capacity restrictions of vehicles or delivery points (Rivera & Mes, 2017; Ulmer & Streng, 2019; Van Heeswijk, Mes, & Schutten, 2017). Though not considered here, dynamic dispatching can be an interesting method for fulfilling online demand in rental systems with a depot.

3. Model

Underlying to our model are nine assumptions, which are all practically relevant. For brevity, we simply list these assumptions here, while an extended explanation and justification is presented in Appendix A. We assume the following for the setting in which libraries operate:

- Libraries work together and share an inventory.
- Libraries ship substantial amounts of books to other libraries every day.
- Libraries face significant inventory holding costs.
- There exists a (central) depot to store books.
- The depot must be used if a book is shipped from one library to another.
- Costs for shipping books increase linearly with the number of books shipped.
- Return times are more or less independent of rental times.
- Customers are willing to wait an infinite amount of time for a book.
- No preemptive supply of books is permitted, even if a library runs empty.

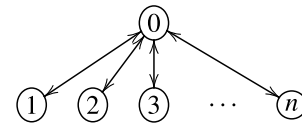


Fig. 1. A multilocation rental system with a depot and n rental locations.

In the remainder of this section, we start by describing the problem setting and by motivating the most relevant modeling choices. Afterward, we provide a mathematical formulation in terms of a Markov decision process by describing the state, transitions, actions, and costs.

3.1. Model description

We consider a rental system with n rental locations and a depot, depicted in Fig. 1, where periodically transport takes place between the depot (index 0) and the rental locations. We are interested in the policy minimizing the long-run average cost of the system, hence we consider a periodic review model with an infinite horizon. We restrict the analysis to a single product type of which a finite number of items are available. We can repeat the same analysis for other product types.

Every period, customers demand and return items at the rental locations. Demand at each location follows a distribution that is discrete, finite, nonnegative, stationary, and state-independent. Demand distributions may differ between locations. Customers either demand items online or by visiting the rental location in person. Regardless of its source, demand is met immediately if an item is on hand (either picked up by a customer or kept aside for an online request). If no items are on-hand, customers can request a delivery of the item at the next delivery moment, provided the request is placed before the order deadline. The time between the order deadline and the next delivery moment is used to carry out any necessary transport. We consider a subscription-based rental system, so customers do not switch to competitors on a stock-out and are aware that not every request can be delivered immediately. Customers receive a delivery notification when their requested items are bound to arrive and are assumed to pick up these soon after the delivery moment.

Rented items return at the same location where they have been demanded and can be rented to another customer in the same period. Customers rent items for a stochastic number of periods, motivated by public library transaction data for the Groningen province in the Netherlands. Due to different visiting frequencies of customers and the possibility to request deadline extensions, there is a considerable variation in rental times; in fact, we found that only 25% of all transactions are returned in the last week of the three-week deadline. The rental time distribution is equal for each location since rental time mostly depends on product rather than location. For ease of presenting the model and for analytical tractability, we assume the rental time distribution is memoryless, i.e., geometrically distributed. This means we only track the total number of rented items at each location, rather than the exact rental time of each individual rented item. In an experiment in Section 7.7, we show that this choice leads to reasonable decisions even if the true rental time distribution is not memoryless.

In order to maximize service and minimize waiting time for customers, we assume all delivery requests are met whenever possible. The delivery requests are handled as follows. First, any returns at a location are allocated directly to delivery requests at that same location (no transport takes place). Second, any remaining delivery requests are met by a shipment from the depot, using available stock at the depot. Third, if the depot has insufficient stock, then the request is met with on-hand stock from another

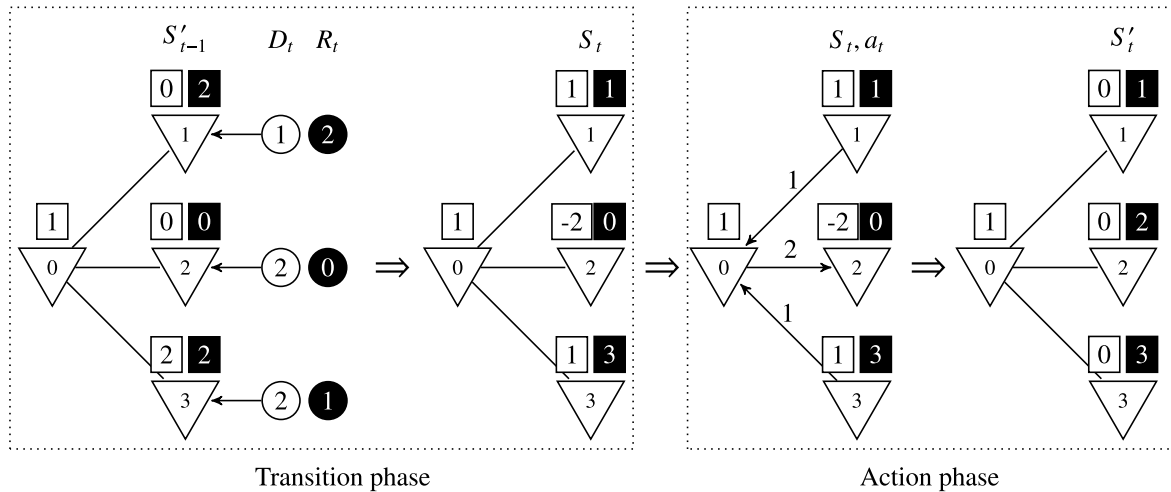


Fig. 2. Example of a state during the transition and action phase.

location. Besides meeting requests, it is possible to carry out take-backs to resupply the depot for future shipment requests and to deal with excess inventories at the rental locations. As indicated in §1, all transportation goes through the depot and cross docking is used to deal with lateral transshipment requests. We exclude the possibility to ship more items to a location than requested, because under our assumptions it is better to store items at the depot and ship them when they are needed.

The costs of the rental system are modeled as follows. Rental locations in our model are uncapacitated, so we model the (lack of) capacity in reality through a holding cost, which varies between locations and is higher for locations where capacity is tighter. The depot has the lowest holding cost per unit of inventory, since it has the largest storage space of all locations and it does not have to keep its assortment on display. Customer dissatisfaction from waiting is modeled by a backorder cost, incurred each period a requested item is not delivered. Because many different product types are transported together, typically every location has to be visited every period. Therefore, fixed costs for driving to a location and delivering crates with items cannot be avoided. The variable costs are due to handling, because every item needs to be picked manually and scanned on pick-up and delivery. Therefore, from the perspective of a single product type, it is reasonable to assume the transportation cost is linear. In addition, there is cross-docking handling cost, measuring the extra cost for exchanging items at the depot in between tours compared to a regular shipment. Usually, this exchange involves manual searching from crates, which is more difficult than systematically picking items from the shelves at the depot, hence the cross-docking handling cost is nonnegative. All cost parameters are linear, and for our long-run average cost analysis they are additionally assumed to be finite and stationary.

3.2. State variable

In period t , the state of the system is given by the tuple $S_t = (x_{0t}, x_t, y_t)$. Here, x_{0t} represents the stock level at the depot. The vectors $x_t = (x_{1t}, \dots, x_{nt})$ and $y_t = (y_{1t}, \dots, y_{nt})$ represent the stock levels x_{it} and number of rented items y_{it} at location i , $i = 1, \dots, n$. If $x_{it} < 0$, then location i has unmet requests. Define $(x)^+ = \max\{x, 0\}$ and $(x)^- = \max\{-x, 0\}$ as the positive and negative part function, taken element-wise for vectors. The total number of items in the rental system is denoted K , hence, for any state S_t , it must hold that $x_{0t} + \sum_{i=1}^n (x_{it})^+ + \sum_{i=1}^n y_{it} = K$. Note that due to this restriction one dimension can be dropped from the state variable, but

we will not do that here to keep the presentation as simple as possible.

Each period consists of a transition phase followed by an action phase. In order to distinguish between these two phases, we indicate the state variable after the action phase by a prime, i.e., the state after actions in period t is given by $S'_t = (x'_{0,t}, x'_t, y'_t)$.

3.3. Transition phase

In the transition phase, each location faces stochastic demands and returns by customers. The demands at each location during period t are denoted $D_t = (D_{1t}, \dots, D_{nt})$. Analogously, the returns are during period t are denoted $R_t = (R_{1t}, \dots, R_{nt})$. The success probability of the geometric rental time distribution is denoted p , which implies that R_{it} is Binomial(y_{it}, p) distributed if there are y_{it} rented items at location i .

The left block in Fig. 2 shows an example of the state before and after the transition phase. Rental locations and the depot are indicated by triangles. Stock levels are depicted as white squares and rented items as black squares. Similarly, demands are indicated as white circles and returns as black circles. For example, location 1 has 0 on-hand and 2 rented items before the transition phase. Because 1 item is demanded and 2 are returned during the transition phase, location 1 ends up with 1 on-hand and 1 rented item after the transition phase. In location 2, demand exceeds the available stock, hence the stock level after the transition phase becomes negative. The depot faces no demand and therefore its stock level does not change.

Expressed in mathematics, given a post-action state S'_{t-1} , the state variable S_t after the transition phase evolves according to

$$x_{0t} = x'_{0,t-1}, \tag{1}$$

$$x_t = x'_{t-1} + R_t - D_t, \tag{2}$$

$$y_t = y'_{t-1} + (x'_{t-1})^+ - (x_t)^+. \tag{3}$$

Eqs. (1) and (2) are trivial. To derive (3), note that on-hand items at location i increase by $(x'_{it})^+ - (x_{i,t-1})^+$ during the transition phase, and consequentially, rented items must decrease by the same number.

3.4. Action phase

In the action phase, decisions are made to carry out transportation actions. The action vector $a_t = (a_{1t}, \dots, a_{nt})$ specifies the num-

ber of items taken back from each location to the depot. If a_{it} is negative, this number is shipped from the depot to location i . If we let $a_t = (a_t)^+ - (a_t)^-$ then $(a_t)^+$ and $(a_t)^-$ can be interpreted as the number of items taken-back and shipped, respectively. We have the following constraints on the action:

$$(a_t)^+ \leq (x_t)^+, \tag{4}$$

$$(a_t)^- \leq (x_t)^-, \tag{5}$$

$$\sum_{i=1}^n (a_{it})^- \leq x_{0t} + \sum_{i=1}^n (a_{it})^+, \tag{6}$$

$$\sum_{i=1}^n (a_{it})^- = \min \left\{ \sum_{i=1}^n (x_{it})^-, x_{0t} + \sum_{i=1}^n (x_{it})^+ \right\}, \tag{7}$$

Constraint (4) ensures not taking back more from any location than is on hand, while constraint (5) prevents shipping more to a location than there are backorders. Constraint (6) prevents shipping more from the depot than is available after the take-back actions. Finally, constraint (7) ensures that all requests receive a shipment, unless the total on-hand inventory in the system is too small.

The right block in Fig. 2 shows an example for the action phase. First the action is shown, with the arrows indicating how many items are transported over an edge and in which direction. To the right, the resulting state after completing the action phase are shown. Here, the chosen action is $a_t = (1, -2, 1)$. Location 2 has a stock level of -2 , so one item is shipped from the depot and another item is cross docked from another location with on-hand stock. In addition, one item is taken back from the locations to re-supply the depot. All requests at location 2 are now met, hence the rented items there increase by 2. The stock at the depot remains as is, because 1 item was shipped, but 1 was also taken back from another location.

If S'_t is the state after transitions, the state after actions is given by

$$x'_{0t} = x_{0t} + \sum_{i=1}^n a_{it}, \tag{8}$$

$$x'_t = x_t - a_t, \tag{9}$$

$$y'_t = y_t + (a_t)^-. \tag{10}$$

In (8), inventory at the depot follows from the net difference between shipments and take-backs. In (9), the inventory position at the locations changes by the amounts transferred to or from that location. In (10), all shipped items are allocated to outstanding requests, so that the rented items increase by $(a_t)^-$. Any remaining unmet demand is backordered. After the action phase is completed, costs are incurred and a new period starts with a transition phase.

3.5. Costs and objective

The following notation is used for the cost parameters. For each unit of on-hand stock at the end of the period, the holding cost is h_0 at the depot and h_i at location i , with $h_0 < h_i$ for $i = 1, \dots, n$. The backorder cost at location i is $b_i > 0$ per backordered unit at the end of the period. The cost per unit shipped to and taken back from location i is $c_i > 0$. For cross docking, there is an additional handling cost $d \geq 0$ per item cross-docked. The total cost for cross docking from location i to j is thus $c_i + c_j + d$.

The costs in period t for state S_t and actions a_t are given by

$$C(S_t, a_t) = h_0 x'_{0t} + d \left(\sum_{i=1}^n (a_{it})^- - x_{0t} \right)^+ + \sum_{i=1}^n \left(h_i (x'_{it})^+ + b_i (x'_{it})^- + c_i ((a_{it})^+ + (a_{it})^-) \right). \tag{11}$$

Respectively, this gives the holding cost at the depot, the handling costs for cross docking, and the sum of the holding, backorder, and shipment costs of the locations. For example, the cost for the action in Fig. 2 is $h_0 + d + c_1 + 2c_2 + c_3$.

A stationary policy π specifies for each state S a corresponding action a . The goal of this paper is to determine a policy π that minimizes the average cost

$$\lim_{t \rightarrow \infty} \frac{1}{t} \mathbb{E}^\pi \left[\sum_{s=1}^t C(S_s, a_s) \right].$$

Although the cost structure is linear, the problem features various interesting trade-offs. The choice of actions depends, among others, on the different cost parameters, the demand rates, and the current state of the system. An action can impact the system state several periods into the future, since if an item is shipped to a certain location now, that location will have more inventory when the item returns. The optimal policy takes into account this trade-off between the direct and future impact of actions.

4. Single-location problem

In this section we analyze the average-cost optimal policy for the single-location problem (SLP). With a single location, cross docking is not possible and shipments from the depot are carried out as soon as the location has a stock-out. What remains to be optimized are the take-back actions, i.e., transporting items from the location to the depot. We prove that the optimal take-back action satisfies a threshold structure: it is optimal to take back all on-hand items above a threshold, and otherwise do nothing. This threshold is state-dependent: it decreases in the number of rented items.

The intuition behind the decreasing threshold follows by considering the last item at the location. Suppose we label one on-hand item at the location as the last item, to be rented to customers only if all other items at the location are rented. We can decide to take back the item to the depot and ship it back when it is requested, saving holding costs every period the item is at the depot. The holding cost savings exceed the transportation costs only if it takes long enough for the last item to be requested. All else being equal, the last item will be requested later if the number of on-hand items at the location increases. This implies that there must be some threshold inventory level above which we want to take back all items. All else being equal, the last item will also be requested later if the number of rented items increases, because returning rented items can be used to fulfill demand. Therefore, the threshold decreases in the number of rented items.

In the remainder of this section, we formally prove the threshold structure. We first explain the idea for the proof. After providing the proof, we derive analytical expressions for some threshold values, and we provide a fast iterative procedure to obtain all other threshold values.

Remark 1. Our analytical results from Sections 4 and 5 extend to various other settings with linear costs. The backordering assumption plays no role in the analysis, hence the same analysis applies to settings where demand is lost if not met at the first delivery moment. Furthermore, the two-location analysis extends to any setting where lateral transshipments are carried out when the

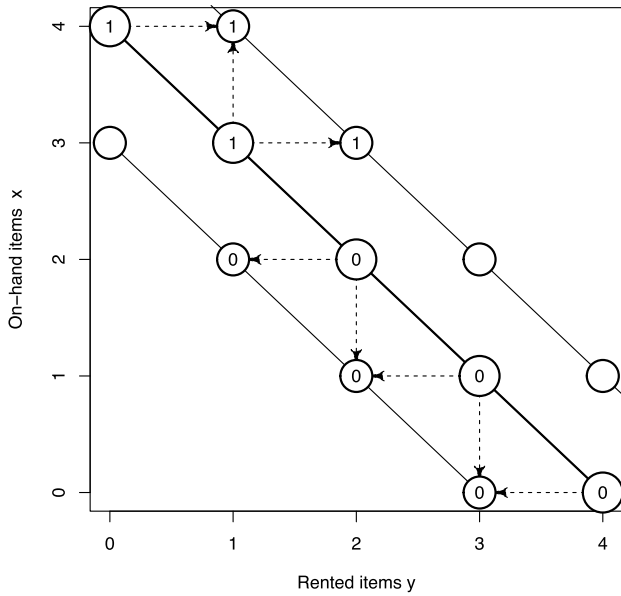


Fig. 3. Optimal take-back policy for the last item and its implications.

depot is out of stock, by replacing the cross-docking handling cost by an appropriate lateral transshipment cost.

Remark 2. While the system stock level K is an important parameter in our problem, our analytical results are valid for any K . Therefore, we do not study K explicitly until the numerical experiments in Section 7.

4.1. Idea for the proof

Our proof is based on studying the last item at the location. We formulate a finite MDP for the last item, and use its properties to prove the threshold structure. Fig. 3 illustrates the optimal take-back actions for some states, where 1 indicates that a take-back is optimal and 0 that it is not. The middle diagonal represents the state space of the last item. Every period, the state moves along the diagonal: up if returns exceed demand, and down if demand exceeds returns. The last item is requested when the on-hand inventory reaches 0. The first step of the proof is proving that the optimal take-back action decreases along the diagonal, i.e., first ones, then zeros. The next step is showing how optimal actions on a diagonal imply the optimal actions on adjacent diagonals, indicated by arrows in Fig. 3. We prove that if a take-back is optimal (not optimal) in a state, then it is also optimal (not optimal) in a state with one more (fewer) on-hand or rented item. By induction, it then follows that the threshold is non-increasing in the number of rented items.

4.2. Finite MDP for the last item

We now formulate a finite MDP to optimize the take-back actions of the last item until it is requested at the location. It is important to note that irrespective of the actions taken, the last item will be rented to a customer in the exact same period (either from on-hand stock, or by a shipment from the depot if it was taken back at some point). Our actions only impact the state of the system until the request occurs, therefore, it suffices to consider only this time frame. Without loss of generality, we subtract h_0 from each holding cost parameter, as this part of the holding cost cannot be influenced by our actions.

Dropping time and location indices for the remainder of this section, suppose that the location currently has $x > 0$ on-hand and

y rented items. We label the last item at the location as $m = x + y$. Item m is rented as soon as $y = m$. Therefore, the state space for item m consists of states $y = 0, \dots, m$, with $y = m$ the absorbing state.

Consider any state $y, y < m$. The action $\alpha_y^m \in \{0, 1\}$ indicates whether or not item m is taken back in state y . If we choose $\alpha_y^m = 1$, we pay the take-back cost $2c$. After the take-back, we can immediately enter the absorbing state because an item at the depot incurs no further costs. Otherwise, if $\alpha_y^m = 0$, we pay $h - h_0$ and make a transition to a new state. It follows that the costs for action α_y^m are given by

$$C(y, \alpha_y^m) = 2c\alpha_y^m + (h - h_0)(1 - \alpha_y^m). \tag{12}$$

The transition probabilities are as follows. If $\alpha_y^m = 1$, we move to the absorbing state m , hence

$$P_{ym}(\alpha_y^m = 1) = 1.$$

If $\alpha_y^m = 0$, we move to state z with probability

$$P_{yz}(\alpha_y^m = 0) = P(\min\{y + D - R, m\} = z) = \sum_{d=0}^{\infty} \sum_{r=0}^y \mathbf{1}\{\min\{y + d - r, m\} = z\} P(D = d) P(R(y) = r), \tag{13}$$

where $P(D = d)$ is the probability mass of the demand distribution and $P(R(y) = r)$ the probability mass of the Binomial(y, p) return distribution.

The value function satisfies the optimality equation

$$V^m(y) = \begin{cases} \min_{\alpha_y^m} \left\{ C(y, \alpha_y^m) + \sum_{z=0}^m P_{yz}(\alpha_y^m) V^m(z) \right\} & \text{if } y < m, \\ 0 & \text{if } y = m. \end{cases}$$

This optimality equation has a solution because the expected time until absorption is finite.

4.3. Optimal policy structure

We are now ready to provide structural results. Lemma 1 shows that the MDP from Section 4.2 has a monotone optimal policy. Furthermore, relations between decisions for MDPs with different values of m are proven. The proofs of all lemmas and propositions can be found in the appendix.

Lemma 1. Monotonicity properties of optimal take-back decisions for the SLP.

- (i) For fixed m , the optimal action α_y^m is monotone decreasing in y .
- (ii) If $\alpha_y^m = 1$ for some y and m , then $\alpha_{y+k}^{m+k} = 1$ for all $k = 0, \dots, K - m$. If $\alpha_y^m = 0$ for some y and m , then $\alpha_{y-k}^{m-k} = 0$ for $k = 0, \dots, y$.
- (iii) If $\alpha_y^m = 1$ for some y and m , then $\alpha_y^k = 1$ for all $k \geq m$. If $\alpha_y^m = 0$ for some y and m , then $\alpha_y^k = 0$ for all $0 \leq k \leq m$.

The interpretation of (i) is that it is relatively better to take back the last item if there are fewer rented items (or, equivalently, if the on-hand stock is higher). This also implies that, if many items are rented, it is sometimes better to postpone a take-back until more items return. For (ii), note that the on-hand stock $x = (m + k) - (y + k)$ is constant in k . Hence, with the same on-hand stock, it is better to carry out a take-back if there are more rented items. Similarly, (iii) shows that with the same number of rented items, it is better to carry out a take-back if there are more on-hand items.

The monotonicity properties in Lemma 1 immediately imply a threshold structure. We find that the threshold is decreasing in the number of rented items, with steps of at most 1.

Proposition 1. The optimal take-back policy for the SLP has the following structure.

- (i) The optimal take-back policy for the SLP is a threshold policy. There exists a threshold $x^*(y)$ that leads to take-back actions

$$a = \begin{cases} x - x^*(y) & \text{if } x > x^*(y), \\ 0 & \text{if } x \leq x^*(y). \end{cases} \quad (14)$$

- (ii) The threshold $x^*(y)$ is decreasing in y , with steps of at most one item, i.e.

$$0 \leq x^*(y) - x^*(y + 1) \leq 1.$$

4.4. Obtaining the threshold policy

In order to obtain the threshold policy, we analytically derive some of the threshold values and we explain an iterative procedure to obtain all remaining values. For the analytical expressions, we first need to characterize the time $\tau(x, y)$ until the last item is requested when starting with x on-hand and y rented items. We obtain

$$\tau(x, y) = \min \left\{ t : \sum_{s=1}^t D_s \geq x + \sum_{s=1}^t R_s(y_{s-1}) \right\}, \quad (15)$$

i.e., $\tau(x, y)$ is the first period in which the total demand equals (or exceeds) the initial inventory plus total returns. In (15), the returns $R_s(y_{s-1})$ depend on the rented items in the preceding period, with $y_0 \equiv y$. The expectation of $\tau(x, y)$ can be obtained using standard methods for finite Markov chains (Kemeny & Snell, 1976).

We can now state the threshold values for which we have analytical expressions.

Proposition 2. Threshold values for the SLP.

- (i) $x^*(0) = \min\{x \geq 0 : (h - h_0)E[\tau(x + 1, 0)] > 2c\}$,
- (ii) $x^*(y) = 0$ for $y \geq \bar{y}$, with $\bar{y} = \min\{y : P(D - R(y) > 0) \leq \frac{h-h_0}{2c}\}$, where $P(D - R(y) > 0)$ is the probability of positive net demand conditional on having y rented items.

The threshold at $y = 0$ is defined by a simple comparison between transportation costs and the expected holding costs until the location runs out of stock. Furthermore, if y increases, the threshold becomes zero at some point because the number of returning items $R(y)$ in the next period is almost always sufficient to cover demand. Note from (ii) that when transportation is very cheap, i.e., $2c < h - h_0$, then it is optimal to always store all items at the depot.

For intermediate values of y , we have no analytical expressions because we need to take into account that it is sometimes optimal to wait for rented items to return before carrying out a take-back. However, we can obtain the optimal threshold values using an iterative procedure. The procedure is based on finite Markov chains and therefore has significantly shorter computation times than solving a Markov decision process.

The idea for the procedure is as follows. Suppose $x^*(y)$ is known for some y and we want to determine the threshold for $y + 1$. Since the threshold decreases by steps of 1 (Proposition 2), the threshold is either $x^*(y)$ or $x^*(y) - 1$. We study the absorption time of a finite Markov chain to determine the correct threshold. The finite Markov chain has states $z = 0, \dots, m$, with $m = x^*(y) + 1 + y$. In all states $z \leq y$ a take-back of item m is necessary (the on-hand inventory exceeds $x^*(y)$). Therefore, all states $z \leq y$ are absorbing with value $V^m(z) = 2c$ and state m is absorbing with value $V^m(m) = 0$. Starting from transient state $y + 1$, we calculate the cost $V^m(y + 1)$ until absorption, incurring a holding cost $h - h_0$ each period the chain is not absorbed. If $V^m(y + 1) > 2c$, then it is cheaper to carry out a take-back in state $y + 1$ than to wait until absorption, so we set $x^*(y + 1) = x^*(y) - 1$. Otherwise, we set $x^*(y + 1) = x^*(y)$. The complete threshold can be obtained by starting at $x^*(0)$ and iteratively applying this procedure until $x^*(y) = 0$.

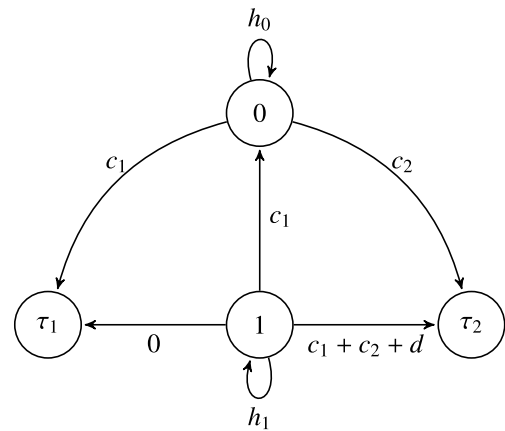


Fig. 4. Scenarios and costs for the m th item in the two-location problem.

5. Two-location problem

We now analyze the average-cost optimal policy for a two-location rental system with a depot. The main difference with the single-location problem is cross docking: local items are cross-docked when the depot has insufficient stock to meet all requests at the other location. Under a mild restriction, we prove that the optimal take-back action in the two-location problem follows a state-dependent threshold policy. The threshold at a location is now two-dimensional, depending on the number of rented items at both locations. The threshold decreases in the number of rented items at the location itself, however, it increases in the number of rented items at the other location. The latter is the case because the probability of having to cross dock in the next period increases when the other location has more rented items. By carrying out a take-back, we can avoid possible cross-docking handling costs.

5.1. Approach

We apply the same approach as in Section 4, studying the costs of the last on-hand item at a location. Without loss of generality, we study costs of the last item of location 1, denoted $m = x_1 + y_1$; by symmetry, the same argument can be repeated for location 2. We derive optimal take-back actions for item m by considering the expected costs of all possible scenarios for the item until it is requested at either of the two locations.

Fig. 4 can be used to calculate costs for different scenarios for item m . Let τ_1 and τ_2 be the period in which item m is requested at location 1 and 2, respectively. Every period before a request occurs, we can decide to either take back item m or leave it at location 1. For example, when item m is kept at location 1, we pay h_1 each period, and depending on whether τ_1 or τ_2 occurs first, we pay 0 or $c_1 + c_2 + d$. When item m is taken back to the depot (node 0), we pay c_1 once, h_0 each period, and c_1 or c_2 when τ_1 or τ_2 occurs.

For the single-location problem, we used the analytically convenient property that take-back actions only impact the state of the system until item m is requested. In the two-location problem, there exists one event where this property does not necessarily hold. This event occurs when $\tau_1 = \tau_2$, i.e., item m is requested at both locations at the same time. If the item has not been taken back to the depot, it will be rented with certainty at location 1. However, if it has been taken back, we may decide to ship it to location 2 instead of location 1. In order to make an optimal decision, we would have to know the expected difference in future costs of the item ending up at location 1 and 2. We avoid this by imposing the restriction that we always ship to location

1 if $\tau_1 = \tau_2$. We believe that this restriction does not lead to a significantly different optimal policy for the following reasons. The event $\tau_1 = \tau_2$ is most likely when item m is in high demand at both locations. In this case the choice for a shipment location is typically not required under an optimal policy: since the demand is expected to occur soon, it seems suboptimal to incur an extra cost for a take-back and shipment. The choice seems more relevant when item m is in low demand at one or both locations, however, then the event $\tau_1 = \tau_2$ is not likely.

Remark 3. It is challenging to extend the last item approach to systems with more than two locations. With two locations, the only possible candidate for cross docking is the other location, however, with multiple locations, there can be multiple candidates. It is impossible to select the correct candidate by considering only the last item of location 1.

5.2. A finite MDP for two locations

As before, we model this as a finite Markov decision process, which can now be absorbed in multiple states. Since location 2 uses stock from itself and the depot before demanding item m , we can set $x_0 = 0$ and $x_2 = K - m - y_2$ without loss of generality. Since $x_1 = m - y_1$ and $x_2 = K - m - y_2$, we can represent the state of this MDP by $S = (y_1, y_2)$. The MDP is absorbed when $y_1 = m$ (corresponding to time τ_1 or a take-back) or when $y_2 = K - m + 1$ (corresponding to τ_2).

The binary take-back decision for this problem is denoted by α_{y_1, y_2}^m . The two-dimensional transition probabilities can be computed analogous to the SLP. The costs for the two location problem are slightly different due to the cross-docking action. The transportation cost $c_1 + c_2$ when item m is demanded at location 2 is unavoidable, so we only incur extra costs for a take-back if $\tau_1 \leq \tau_2$, hence

$$C(y_1, y_2, \alpha_y^m) = 2c_1 P(\tau_1 \leq \tau_2) \alpha_{y_1, y_2}^m + (h_1 - h_0)(1 - \alpha_{y_1, y_2}^m),$$

for $y_1 < m, y_2 < K - m + 1$. (16)

The probability $P(\tau_1 \leq \tau_2)$ can be computed by solving the corresponding finite Markov chain with starting state y_1, y_2 where we set $\alpha_{y_1, y_2}^m = 0$ in all states. Finally, if we are absorbed in $y_2 = K - m + 1$ before carrying out a take-back, we pay the additional cross-docking cost d .

$$V^m(y_1, K - m + 1) = d \quad \text{for } y_1 < m.$$

5.3. Threshold policy

Lemma 2 shows monotonicity of the optimal decisions in y_1 and y_2 . Take-back becomes less interesting as y_1 increases and more interesting as y_2 increases. The relation with y_1 has the same interpretation as in the SLP. Take-back becomes more interesting as y_2 increases because handling costs can be avoided when item m is demanded soon at location 2.

Lemma 2. Monotonicity in the finite MDP for two locations.

- (i) For fixed y_2 , α_{y_1, y_2}^m is monotonic non-increasing in y_1 .
- (ii) For fixed y_1 , α_{y_1, y_2}^m is monotonic non-decreasing in y_2 .

The monotonicity can be exploited in a similar way as for the SLP to show a threshold structure. We obtain the following.

Proposition 3. Threshold policy of the two-location problem.

- (i) The optimal take-back action for location 1 in the two-location problem is a threshold policy. There exists a threshold $x_1^*(y_1, y_2)$ that leads to take-back actions

$$a_1 = \begin{cases} x_1 - x_1^*(y_1, y_2) & \text{if } x_1 > x_1^*(y_1, y_2), \\ 0 & \text{if } x_1 \leq x_1^*(y_1, y_2). \end{cases} \quad (17)$$

- (ii) For fixed y_2 , $x_1^*(y_1, y_2)$ decreases in y_1 and for fixed y_1 , $x_1^*(y_1, y_2)$ increases in y_2 .
- (iii) For fixed y_1 , $x_1^*(y_1, y_2) \leq x_1^*(y_1)$ for all y_2 .

Hence, we now have a threshold policy that depends on both y_1 and y_2 . Interestingly, in the two-location setting we always take back at least as much in the single-location setting. The reason for this is the extra cross-docking cost that we could prevent by taking back in time, especially when the on-hand stock at location 2 is low. The holding cost trade-off from the single-location setting thus also applies to the two-location setting, with cross-docking costs as an additional incentive for take-backs.

For the special case $m = 1$, and $K = 1$, there is a simple expression for the take-back decision when $x_1 = 1$.

Lemma 3. For $m = 1$ and $K = 1$ in the two-location problem, a take-back of item m is optimal if

$$P(D_1 > 0) \leq \frac{h_1 - h_0 + d}{2c_1 + d}. \quad (18)$$

Hence, we carry out a take-back only if the demand rate at location 1 is small enough, and the higher d becomes, the more likely we are to take back.

5.4. Constructing the take-back policy

Similar to Section 4.4, we now state a procedure for obtaining the state-dependent take-back policy. The idea is to repeatedly compare the holding cost during the expected time until demand with possible transportation costs and cross-docking costs, taking into account that it may be optimal to postpone a take-back until we reach another state.

Let \mathcal{A} be the set of states in which it is optimal to carry out a take-back and $\tau_{\mathcal{A}}$ be the first moment we enter a state in set \mathcal{A} . The holding costs until absorption coincide with the moment of absorption, $\min(\tau_1, \tau_2, \tau_{\mathcal{A}})$. Cross-docking handling costs will only have to be paid if the first demand for item m happens at location 2 before the postponed take-back is carried out, i.e. if $\tau_1 > \tau_2$ and $\tau_2 \leq \tau_{\mathcal{A}}$. The extra transportation cost $2c$ for carrying out a take-back now can only be prevented if the demand at location 1 occurs before the postponed take-back, i.e. if $\tau_1 \leq \tau_2$ and $\tau_1 \leq \tau_{\mathcal{A}}$. Therefore, we base the decision on the condition

$$(h_1 - h_0)E[\min(\tau_1, \tau_2, \tau_{\mathcal{A}})] + dP(\tau_1 > \tau_2, \tau_2 \leq \tau_{\mathcal{A}}) \geq 2c_1 P(\tau_1 \leq \tau_2, \tau_1 \leq \tau_{\mathcal{A}}). \quad (19)$$

In words, we take back item m if the expected holding costs and cross-docking handling cost of not taking back right now exceed the expected take-back costs.

We can use Eq. (19) to iteratively generate optimal take-back actions by running diagonally over all possible states. We start with $\mathcal{A} = \emptyset$ and the most extreme state $y_1 = 0, y_2 = K - m$. If the condition holds, add this state to \mathcal{A} . Now iteratively check the condition with the updated \mathcal{A} , keeping y_1 constant and decreasing y_2 by 1. If the condition holds, we add the state to \mathcal{A} , if it does not, we stop checking for $z \leq y_2$. Then we increase y_1 by 1, set $y_2 = K - m$ and repeat the above procedure. We continue this until for some y_1 and $y_2 = K - m$ the condition does not hold, or until $y_1 = m$. Repeating the same procedure for all possible m yields the complete take-back policy.

6. Heuristic

For a general number of locations, it is challenging to obtain analytical results and to solve the MDP to optimality. Therefore, we propose a heuristic in order to take effective decisions in reasonable time. Any heuristic for our problem must include the following elements:

- A rule to select receiving locations in case of shipments and cross docking.
- A rule to select sending locations in case of cross docking.
- A rule to determine how much stock to take back from each location to the depot.

Van der Heide et al. (2017) propose various effective rules for shipments and take-backs, which we adapt here to tackle the situation with cross docking. The rules for selecting locations in Van der Heide et al. (2017) are based on an extensive numerical study of the optimal solution of the MDP in small instances. For the rules for take-backs, there is now a solid theoretical basis, drawing from the analytical results in Sections 4 and 5.

The heuristic consists of three phases. The first phase is the shipments/cross-docking phase, which is concerned with dealing with all unmet requests in the system. The second phase is the threshold take-back phase, which deals with taking back all on-hand items above the single location thresholds. The third phase is the preventive take-back phase, and is concerned with resupplying the depot in order to prevent future cross-docking handling costs. In what follows, rules used in these respective phases are described in more detail. Pseudo-code for the heuristic is shown in Algorithm 1.

Algorithm 1 Heuristic.

```

Z = min{∑_{i=1}^n x_{it}^-, x_{0t} + ∑_{i=1}^n x_{it}^+}    ▷ Shipments/cross-docking
for z = 1, ..., Z do
  if z ≤ x_0 then
    i* = 0
  else
    i* = arg min_{i ∈ {j: x_j > 0}} c_i - h_i + b_i q_i
    j* = arg max_{i ∈ {j: x_j < 0}} b_i - c_i + b_i g_i(y_i)
    x_{i*} := x_{i*} - 1
    x_{j*} := x_{j*} + 1, y_{j*} := y_{j*} + 1
  for i = 1, ..., n do    ▷ Threshold take-backs
    z = (x_i - x_i^*(y_i))^+
    x_i := x_i - z
    x_0 := x_0 + z
  i* = arg min_{i ∈ {j: x_j > 0}} (2c_i - h_i + h_0)q_i    ▷ Preventive take-backs
  while ∑_{i=1}^n x_i^+ > 0 and (2c_{i*} - h_{i*} + h_0)q_{i*} ≤ (d + h_{i*} - h_0)(1 - q_{i*})S_{i*} do
    x_{i*} := x_{i*} - 1
    x_0 := x_0 + 1
    i* = arg min_{i ∈ {j: x_j > 0}} (2c_i - h_i + h_0)q_i
    
```

In the shipments/cross-docking phase, the total number of items to be sent is known, and denoted by Z . We repeatedly select sending and receiving locations and update their stock levels until all Z items are sent. As long as the depot has stock, the depot is the sending location. Otherwise, cross docking is necessary, and the sending location is a location with stock that minimizes $c_i - h_i + b_i q_i$, where

$$q_i = P(D_i - R_i(y_i) > x_i - 1)$$

is the stock-out probability at location i after removing one item. This minimizes a combination of the direct costs in the current period and the expected backorder costs in the next period due to having lower inventory. The receiving location is a location with backorders that maximizes $b_i - c_i + b_i g_i(y_i)$, where

$$g_i(y_i) = \begin{cases} \frac{1}{P(R(y_i) > 0)} & \text{if } y_i > 0, \\ M & \text{if } y_i = 0, \end{cases}$$

is the expected number of periods until the first return when y_i items are rented, with M some large number. This maximizes a

combination of directly prevented backorder costs and future backorder costs when the location needs to wait for an item to return. Under this rule, items are typically sent to locations with low y_i , which should be prioritized because they are least likely to have returning items in the next period that can meet the backorder.

In the threshold take-backs phase, all items exceeding the single location thresholds $x_i^*(y_i)$ are taken back from the locations to the depot. This choice is motivated by the result in Proposition 3 (iii), showing that the two-location threshold is at least as low as the single-location threshold. Since the cost trade-offs yielding this result also exist in the general problem, we choose to apply this same property to the general problem. It is important to note that this step approximates the optimal solution when K is large. When K is large, the depot almost always has sufficient on-hand stock for shipments, hence each location can carry out its optimal single location policy without being affected by what happens at other locations.

As we observed from the thresholds in the two-location problem in Section 5, in the preventive take-backs phase the depot is resupplied for future shipments so that lower cross-docking handling costs are incurred. Clearly, this is necessary only if there is a high enough probability of having to cross dock in the next period. We model this by a regret based rule, comparing the expected regret of a take-back with the expected regret of no take-back. After carrying out a take-back, with probability q_i the item is demanded again at location i in the next period at extra cost $2c_i - h_i + h_0$, so the expected regret of a take-back is $(2c_i - h_i + h_0)q_i$. Let the stock-out probability of the depot, excluding demand at location i , be given by

$$s_i = P\left(\sum_{j \neq i} (D_j - R_j(y_j) - x_j)^+ > x_0\right). \tag{20}$$

When no take-back is carried out, with probability $(1 - q_i)s_i$ the item needs to be cross docked to another location at extra cost $d + h_i - h_0$, so then the expected regret is $(d + h_i - h_0)(1 - q_i)s_i$. We carry out preventive take-backs as long as the regret of a take-back exceeds that of no take-back.

Remark 4. The probability in Eq. (20) is a convolution of $2(n - 1)$ random variables, so we may need to approximate it for large n . In case D_i is Poisson for $i = 1, \dots, n$, we can use the approximation by Van der Heide et al. (2017). They approximate the distribution of $(D_i - R_i(y_i) - x_i)^+$ by a Poisson($-\log(P(D_i - R_i(y_i) \leq x_i))$) distribution, so that $\sum_{j \neq i} (D_j - R_j(y_j) - x_j)$ is also Poisson distributed.

7. Numerical experiments

In this section, numerical experiments are carried out with several aims. First, we test the quality of the heuristic by comparing it to the optimal solution. Second, we compare cross docking and delayed shipments, to find out which type of transportation to use in which circumstances. Third, we determine the value of storage at the depot and we evaluate the benefits of using complete rental return time distribution information over using aggregated information only. Finally, we want to see whether there is an inventory pooling effect in instances with high demand rates and a high number of locations. Before carrying out the experiments, we discuss the set of instances used in most experiments and the numerical implementation of the MDP.

7.1. Instances

We create a set of 50 instances with different parameter configurations that we use in most of our experiments. The included parameters and their values are shown in Table 1. The parameter

Table 1
Possible parameter values in the instances.

Parameter	Symbol	Level 1	Level 2	Level 3	Level 4	Level 5
System stock level	K	high	low			
Visit frequency	f	1	2	3	4	5
Depot holding cost	h_0	0.4	0.5	0.6	0.7	0.8
Backorder cost	b	8	11	14	17	20
Shipment cost	c	2	4	6	8	10
Cross-docking handling cost	d	0	1	2	3	4
Demand rate	λ	0.05	0.1	0.15	0.2	0.25

values are inspired by public libraries in the Netherlands, and some are based directly on library transaction data from the Groningen province. We use an orthogonal design, which allows for testing a wide range of parameter values with a limited number of experiments (Taguchi, 1986). Specifically, we use the L50 array (see, e.g., NIST, 2017), which has 1 two-level factor and up to 11 five-level factors. We used the first 7 factors of the L50 array, in the same order as in Table 1. We repeat all 50 instances for different numbers of locations, and for a fair comparison between instances with a different n , all rental locations in an instance have identical cost parameters and demand/return distributions.

The two-level factor is the system stock level K . Although we have been unable to formally establish convexity, we observe numerically that the optimal average cost is convex in K . Therefore, we determine a value for K by calculating the optimal average cost for $K = 1, 2, \dots$ until it no longer decreases, which we label as ‘high K ’ since it turns out to be rather high. Note that this choice for K does not involve purchasing costs. In practice, fewer items may be purchased due to budget cuts, so we check the impact of having ‘low K ’ by setting

$$K = 1.5 \frac{\sum_{i=1}^n \lambda_i}{p}, \quad (21)$$

rounded to the nearest positive integer.

Without loss of generality, the weekly holding cost is $h_i = 1$ for $i = 1, \dots, n$ in all experiments; all other cost parameters can be scaled accordingly. We do vary the holding cost at the depot and the other cost parameters. As customers arrive randomly over the week, we assume the weekly demand follows a Poisson distribution with rate λ . The values are based on our data set, where weekly demand rates for almost all items are in the range $[0, 0.25]$. Furthermore, the weekly return probability in the same data set is $p = 0.3$.

The visit frequency f measures the number of times per week transportation takes place between the depot and the locations. Where appropriate, we rescale the other parameters to match their correct weekly rates, e.g., the holding cost per period should be h_i/f if the visit frequency is f . In order to have a weekly return probability of p , the return probability per period for a given f is $1 - (1 - p)^{1/f}$.

7.2. Implementation details

The MDP and the heuristic have been implemented in Python. All experiments are run on a computer with a Core i7-4770 CPU (3.4 gigahertz) and 16 gigabytes memory. In order to have finite demand, we cut off the Poisson demand distributions at their 99.99% quantile. Moreover, in order to have a finite state space, we introduce a maximum number of backorders $B = 2$ at each location after demands/returns, with lost demands penalized at cost $\ell = 2b$. The number of possible states in the MDP with n locations,

Table 2
Summary statistics for the percentage optimality gap of the heuristic.

n	Average	Min.	1st quartile	Median	3rd quartile	Max.
2	0.40	0.00	0.00	0.00	0.04	6.25
3	0.59	0.00	0.03	0.15	0.50	4.52
4	0.63	0.00	0.05	0.14	0.74	4.29

K items, and B maximum backorders per location is

$$|S| = \sum_{i=0}^n \binom{2n+K-i}{K} \binom{n}{i} B^i.$$

Using value iteration, instances with $n = 4$, $K = 8$, and $B = 2$ (185,526 states) are solvable within 2 minutes. Instances with $n = 5$, $K = 9$, and $B = 2$ (2,930,642 states) take several days, which is why we limit the instance size for the MDP to at most 4 locations.

7.3. Performance of the heuristic

In this first experiment, we evaluate the performance of the heuristic. For each value of n , we run the 50 instances (150 in total), comparing the average cost of the heuristic with the average cost of the optimal policy from the MDP. Summary statistics for the optimality gap of the heuristic are shown in Table 2.

The average optimality gaps are similar for each value of n and are well within 1%. In 126 out of 150 instances, the optimality gap is within 1%, and in the remaining instances, the gap is at most 6.25%. The heuristic has the largest optimality gaps in instances with a combination of high c , high λ , and high K , because it can slightly overestimate the take-back amounts to the depot.

7.4. Cross docking vs. delayed shipments

We now want to obtain managerial insights into the different transportation methods for dealing with lateral transshipment requests. To that end, we compare the difference in costs between situations with cross docking (as presented in this paper) and delayed shipments (Van der Heide et al., 2017). Figs. 5 and 6 show the resulting average costs for both policies for high and low values of the total system stock. Separate graphs are shown for each parameter, and the point at each parameter value gives the average for all configurations with that same value.

From Fig. 5, we observe that the difference between cross docking and delayed shipments is small when K is high. The system stock level is high enough for the depot to have sufficient stock to meet all shipment requests. This implies that cross-docking actions and delayed shipments are not used much, and therefore, both policies have practically the same cost in most cases. The same reasoning also explains why the average cost is almost constant in the graphs for b , d , and f . The average costs do increase strongly in n , c , and λ , for obvious reasons. Interestingly, delayed shipments sometimes have lower costs than cross docking due to the delay of one period. Items returning during the delay can be used to meet backorders, avoiding the need to ship and saving shipment costs in the process. This is most relevant when b is very low or when f is large, because then the backorders costs incurred during the delay are limited. Hence, costs can sometimes be saved by not cross docking, which we further investigate in Section 7.5. Overall, delayed shipments seem preferred to cross docking in cases with high stock, because resulting costs are similar without the need to visit rental locations a second time.

For our practical case, we see that library organizations may purchase fewer books due to budget cuts. The clear gap between cross docking and delayed shipments for low K in Fig. 6 indicates that cross docking can be quite important in such situations. The

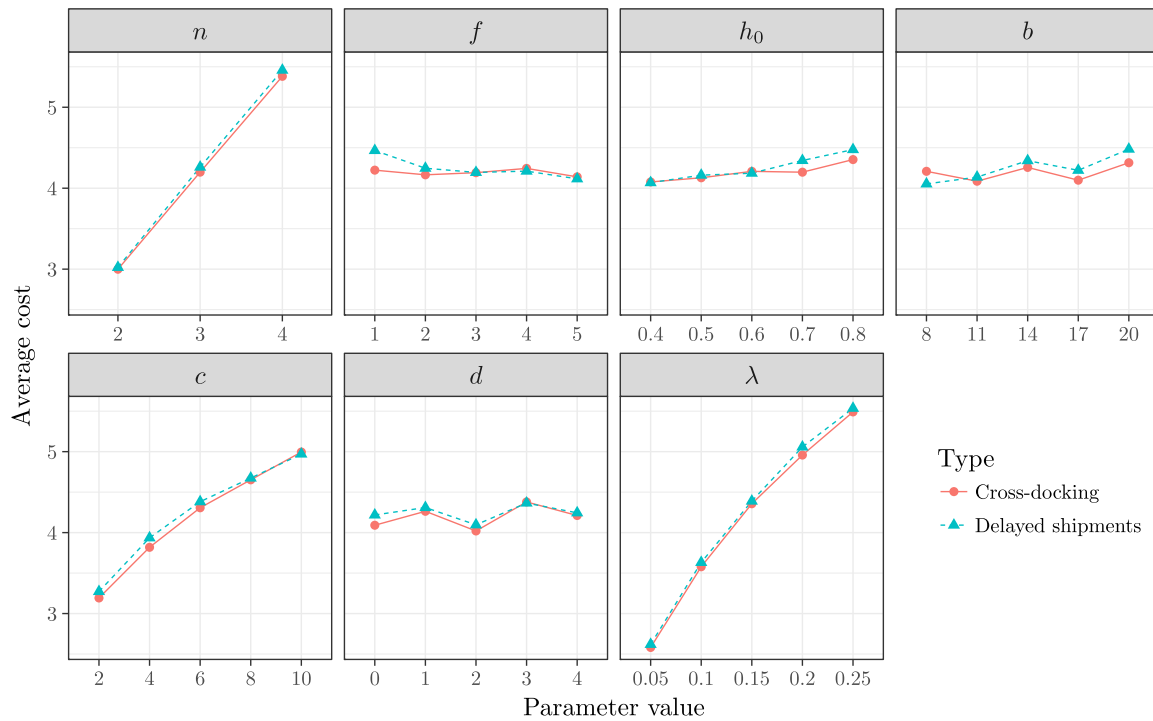


Fig. 5. Cost of delayed shipments and cross docking when K is high.

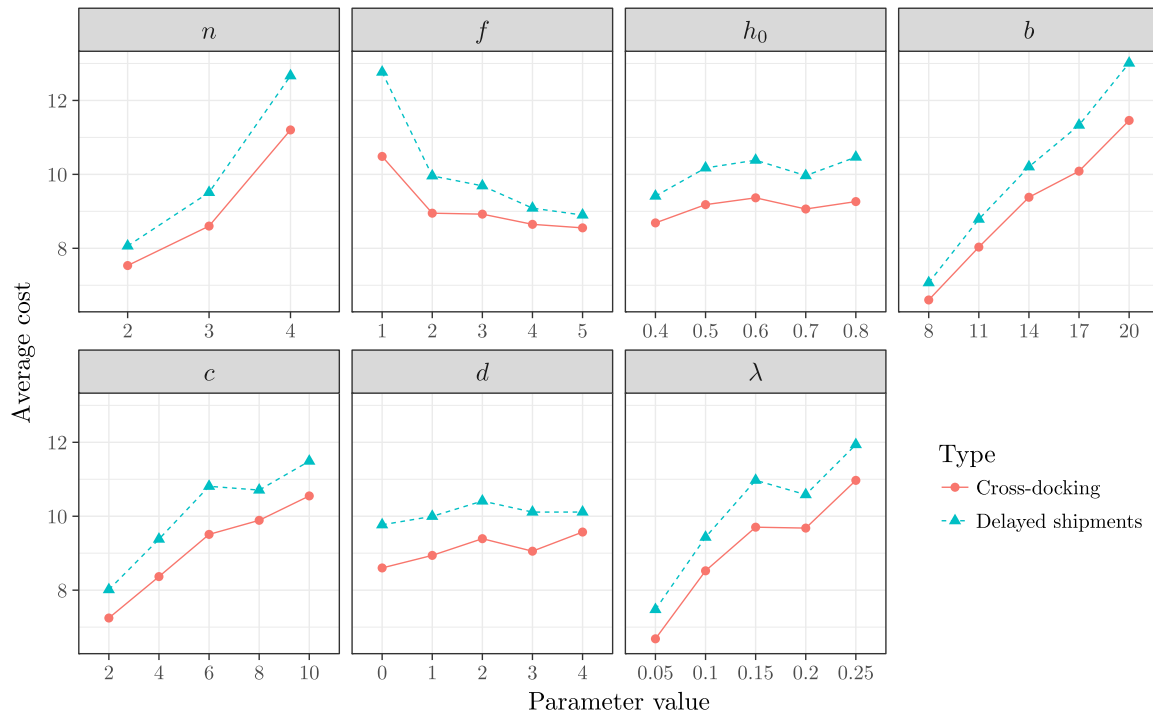


Fig. 6. Cost of delayed shipments and cross docking when K is low.

average gap between the two is 9.56%. With low system stock, cross docking and delayed shipments are more often necessary and the average costs are much higher than for high K . It is no longer possible to avoid situations with backorders, as can be seen from the increase of the average cost in b . Part of the gap is caused by the direct backorder cost incurred for delayed shipments. Besides that, delayed shipment essentially prolongs the rental time by one

period, resulting in higher utilization of the stock in the system and extra backorders. Delayed shipments save cross-docking handling costs at the expense of backorder costs, therefore the gap decreases in b and increases in d . The visit frequency also has significant impact. If the visit frequency is once per week, the average gap is 20.51%. However, if the visit frequency is higher, the gap between cross docking and delayed shipments becomes smaller

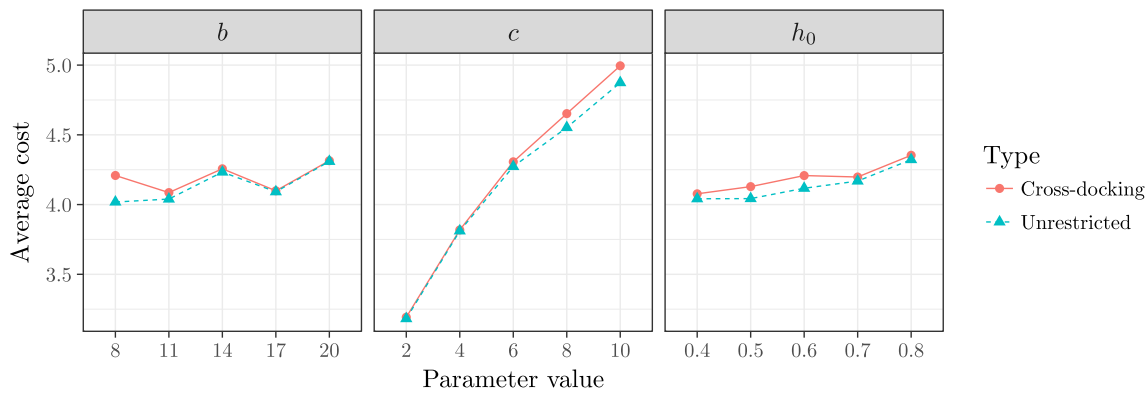


Fig. 7. Cost of cross docking and unrestricted actions when K is high.

quickly. The delay becomes shorter, therefore less backorder costs are incurred during the delay. This interplay between visit frequency and transport method is important to consider when organizing transportation in a rental system with a depot.

7.5. Unrestricted actions

An important observation from Section 7.4 is that sometimes cost savings are possible by choosing not to ship/cross dock. Therefore, we compare our cross-docking model to a model with unrestricted actions, which is free to decide whether to ship, cross dock, or do nothing when there is a backorder. Note that the model with unrestricted actions has cross docking and delayed shipments as special cases.

On average, the cost reduction from unrestricted actions is 0.14% for low K and 1.08% for high K . Waiting for a returning item rather than shipping/cross docking immediately is most interesting in situations where a location has many rented items, therefore the potential gains are larger for high K . Fig. 7 displays the impact of the three most influential parameters for high K . Evidently, unrestricted actions are most effective when backorder costs are low or shipment costs are high. Furthermore, if the holding cost at the depot is quite low, we may prefer to keep items at the depot rather than shipping them. Overall, the gains from unrestricted actions are limited in most cases, so we would advise to always meet demand to maximize customer satisfaction.

7.6. Value of storage

One of the main reasons to use a depot is to reduce holding costs at the locations. To gain more insight into the extent of the possible gains, we study a variant of the model where take-back actions are prohibited, so that the depot cannot hold stock and shipment requests must always be met by cross docking.

The average gap between ‘no take-backs’ and cross docking is 1.37% for low K and 7.68% for high K . Under high K , some items are not used much, so it is useful to store them at the depot. In contrast, under low K , most stock is in demand, so we cannot achieve a similar benefit. Fig. 8 shows the gaps for some parameters under high K . The most clear effects are visible in the graphs for the depot holding cost and the shipment cost. Clearly, if the holding cost at the depot is lower, the gains from storing at the depot will be larger. Similarly, if the transport is cheap, we can afford to store more items at the depot. The absolute gap remains more or less constant in the demand rate, so the relative benefit of the depot is largest for low demand rates. We also tested this for extremely low demand items with $\lambda < 0.01$; then the gap can be over 50% in some instances.

7.7. Value of rental information

The aim of this experiment is to provide insights into the required information about rented items for making good decisions. This helps to determine when it is appropriate to use simple information structures (tracking only the number of rental items at each location) over more advanced information structures (tracking the rental time for each rented item). To that end we compare policies under geometric and time-dependent return distributions with each other. We solve MDPs with time-dependent distributions to optimality, and then evaluate the optimality gap by using the optimal actions from the MDP with geometric returns in the same state of information.

We generate 1000 random configurations. The time-dependent return distribution has probability density function $f(j)$ for $j = 1, \dots, T$, where T is the due date. Each $f(j)$ is randomly generated from a Uniform(0, 1) distribution, rescaled such that the total sums to 1. The corresponding parameter of the geometric distribution is $p = 1 / \sum_{j=1}^T j f(j)$. We sample n , K , and T uniformly from the sets $\{2, 3\}$, $\{2, 3, 4\}$, and $\{2, 3, 4\}$. The cost parameters are uniformly distributed between ranges $0.3 < h_0 < 0.9$, $10 < b < 20$, $1 < c < 10$, and $0 < d < 5$. The demand parameters λ_i are Uniform($v_1, v_1 + v_2$) distributed with $v_1 \sim \text{Uniform}(0.05, 0.15)$ and $v_2 \sim \text{Uniform}(0, 0.2)$. The remaining parameters are $f = 1$, $B = 2$, and $\ell = 2b$. Fig. 9 shows a histogram of the deviations in all configurations.

The average deviation is 0.1%, indicating that the geometric optimal policy gives good results even if the returns are time-dependent. Of the 1000 configurations, there are 92 for which the deviation exceeds 0.3%, with the maximum being 0.89%. In those configurations typically the average return time is long and the backorder cost is small compared to the sum of the cross-docking and shipment cost. Overall the assumption of geometric returns appears to yield good results in this experiment, in line with the findings of Alfredsson and Verrijdt (1999) for exponential lead times in a base-stock system.

7.8. Pooling effect in larger systems

In this final experiment, we study the impact of a larger number of locations and higher demand rates on the costs of operating the rental system. We do this by simulating the average cost of the heuristic for a base case from our experimental design. The cost of every instance is based on 1000 simulation runs of 1000 periods, excluding a 100 period warmup that starts with all stock at the depot. The parameters of the base case are $f = 1$, $b = 14$, $c = 6$, $d = 2$, $h_0 = 0.6$. We vary the number of locations n between 1 and 50, and the demand rate λ between 0.1 and 5. We optimize

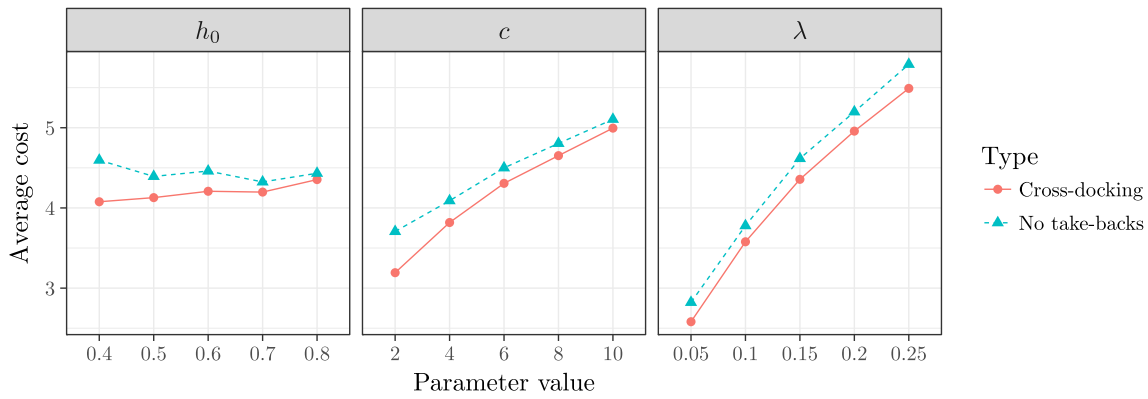


Fig. 8. Percentage cost increase of not using the depot.

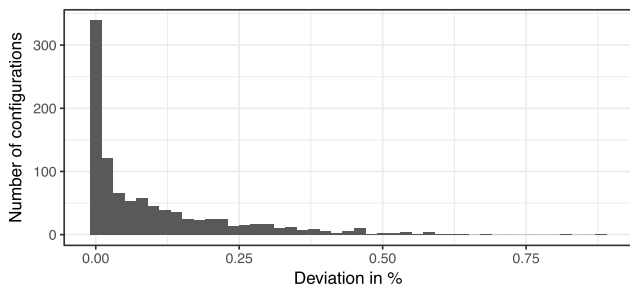


Fig. 9. Histogram of the deviations in the 1000 configurations when using geometric policies in case of time-dependent returns.

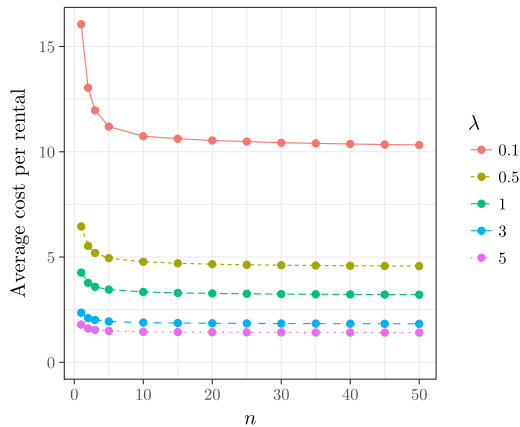


Fig. 10. Average costs per rental.

K in the range $[K_{\min}, 5K_{\min}]$ by a bisection search on the simulated costs, where $K_{\min} = \lceil n\lambda/p \rceil$ is the smallest value of K needed for stability.

Fig. 10 shows the average cost per rental for different values of n and λ . The average cost per rental decreases in n and also in λ . The decrease in λ is due to demand becoming more predictable (the coefficient of variation $1/\sqrt{\lambda}$ decreases in λ). The decrease in n is due to the inventory pooling effect visible in Fig. 11, depicting several time-averages for $\lambda = 1$. The average on-hand inventory per location, both locally and centrally, decreases quickly at first and becomes constant at some point. The on-hand stock at the depot is high enough to meet all shipment requests on average. Clearly, the average number of rented items must be exactly $\lambda/p \approx 3.33$. All in all, standard insights about inventory pooling also seem to hold in our rental system.

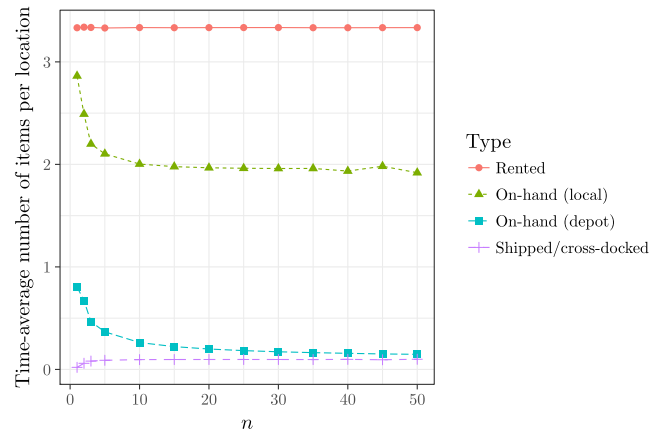


Fig. 11. Time-average items per location for $\lambda = 1$.

8. Conclusion

This paper studies a library system with a low-cost depot that can be used for storage, shipments, and cross docking. We provide theoretical results for the structure of the optimal policy in settings with one and two library locations. In both cases, optimal policies can be characterized by a series of state-dependent thresholds in the number of rented items. For one location the thresholds are based on a trade-off between holding and transportation costs, while for two locations there is a richer trade-off also involving cross-docking handling costs. Based on these structural results, we develop an effective heuristic for the multi-location case.

The results from this paper provide several managerial insights regarding the use of depots in library systems. The cross-docking option at the depot is shown to be better than delayed shipments under low stock levels, but delayed shipments are almost as effective under high stock levels or when locations are visited often by a vehicle from the depot. The value of storing stock at the depot is highest for low demand rates and relatively high stock levels. For various types of settings it can thus be worthwhile to employ a depot.

Theoretical insights from the optimal policy structure, for example the single-location problem, can be used in practice for dividing items between a front and back area of a library location, or for deciding which items to remove from locations in order to make room for new items. We also identified in which cases it can be important to consider information about rental durations rather than only the total number of rented items. This is most important when there is a small difference between backordering

and transportation costs and rental times are long. However, in most cases it is sufficient to base decisions on the total number of rented items, leading to formulations that are easier to solve and heuristics that are easier to understand.

Several interesting model extensions can be considered in future research. An important assumption in the model is that rented items are returned to their original location. The model can be extended to the situation where items return to other locations, for example in car rentals, by introducing routing probabilities for each returning item. Shipment costs can be generalized, for example, to a fixed cost plus a variable cost per item shipped. While such non-linear costs are trivial to implement in the MDP, the last item approach from our analysis will no longer work and a different approach must be used. For settings with seasonal demand, it is interesting to consider non-stationary demand. A possible way to solve this is by applying a rolling-horizon approach, using forecasts of the parameters of the demand distribution for each period in a finite horizon. Another extension is introducing reservations, meaning that customers wait for a return at their location rather than having items shipped from the depot. Finally, it is interesting to study a multi-item setting, where a fraction of customers substitutes to another product on a local stock-out instead of asking for a shipment from the depot. A starting point could be to extend our single-location problem analysis to two items. All in all, we believe there are ample future research opportunities.

Acknowledgment

This research has been funded by Dinalog, the Dutch Institute for Advanced Logistics. We thank three anonymous reviewers for their valuable comments that helped improve the paper. One reviewer provided the structure for [Appendix A](#).

Appendix A. Description of the practical context

In this appendix, we present the main underlying assumptions of our setting, explain their rationale, and provide examples from practice. We study a network of cooperating libraries. There is a central depot that is used for centralized storage of books, and all transportation of books is routed via this depot.

A.1. Libraries work together and share an inventory

Libraries in the Netherlands are by law required to participate in the national system of interlibrary loans. This implies that books from all libraries must be accessible to members of all other libraries. National and provincial coordination enable preservation of books with low demand (the long-tail collection) in one or a few locations ([Lammers, 2020](#)). Similar cooperative constructs for libraries exist in numerous places, and have been around for decades ([Bartlett, 2014](#)). Many university libraries are organized as cooperative systems, for example at Linköping University in Sweden consisting of 5 libraries ([Burman & Brage, 2016](#)). Also public libraries are organized as such, for example, the library system of the Brooklyn Public Library and the New York Public Library ([Quandt, 2017](#)).

A.2. Libraries ship substantial amounts of books to other libraries every day

The following statistics for the public libraries in the Netherlands are presented in [Lammers \(2020\)](#) for the year 2018. A total of 66.5 million books are borrowed from libraries. Requests for locally unavailable books can be fulfilled from other libraries in the same province, which occurred for 1.39 million books. Books that are not available from any library in the province, are obtained

from public libraries elsewhere in the country or from university libraries, which occurred 40,209 times. Similarly, the Brooklyn and New York public libraries move 7.4 million books annually ([Quandt, 2017](#)), and the San Francisco Public Library 1.76 million ([Apte & Mason, 2006](#)).

A.3. Libraries face significant inventory holding costs

[May and Black \(2010\)](#) note three roles of libraries that require space: provider of books and information, provider of access to technology and provider of a social space where members of the public are welcome. Taking only the first role into account, classic inventory costs are likely to be moderate. A large cost component in the Netherlands arises from the prescribed annual replacement rate of 10% ([Lammers, 2020](#)). However, the other two roles add considerably to the inventory costs in the form of opportunity costs. Space used for storing books cannot be used for other activities. For example, a library from Denmark transferred large amounts of books to a shared storage facility to make room for people and activities ([Petersen & Kooistra, 2020](#)). Assigning value to space allows for taking these current developments in the roles of libraries into account. In contrast, storage in depots is typically significantly cheaper. There are many factors that potentially contribute to this, for example, depots can be located where space is cheaper, theft rates (e.g., [Cromwell, Alexander, & Dotson, 2008](#)) are lower in areas not open to the public, and storage density (e.g., [Boysen, Briskorn, & Emde, 2017](#)) is higher in warehouses than in libraries.

A.4. There exists a (central) depot to store books

Already in 1890 the Victorian statesman W.E. Gladstone proposed to shelve books according to their ‘sociability’, moving less sociable items to mobile shelving or other maximum-density storage areas ([Scarre, 2017](#)). Depots for book storage continue to exist in various forms. For example, many countries, including Finland, Norway, Estonia and France, have a national repository from which books can be borrowed ([Vattulainen, 2004](#)). In the Netherlands, approximately 30 so-called Plus-libraries, take on a leading role at the national level and provide central storage for certain categories of books ([Lammers, 2020](#)). At the local level, libraries in the Netherlands are often organized with a central library and a number of small branch libraries, where the central library functions as a depot with which branch libraries exchange books. University libraries have a long history in using this approach, among others for the efficiency gains attained by central depots ([Walsh, 1969](#)).

A.5. The depot must be used if a book is shipped from one library to another

This configuration is often referred to as a hub-and-spoke network. [Apte and Mason \(2006\)](#) describe this system for the San Francisco Public Library. The books are placed without sorting in bins at the originating branch library. Subsequently the books are brought to the main library, where they are sorted and placed in separate bins per destination. Finally, bins are delivered to the appropriate destinations. The New York Public Library does not perform the sorting at the main library, but has a dedicated facility for this purpose, which has an automated sortation system with a capacity of 12,000 books per hour ([Quandt, 2017](#)). In the Netherlands, this is organized per province ([Lammers, 2020](#)); a commercial parcel carrier connect the provincial depots for the 3% of shipments that occur between provinces.

A.6. Costs for shipping books increase linearly with the number of books shipped

Since books are small and can be transported in bulk in bins, the transportation costs are quite low. For example, at the San Francisco Public Library total transport costs divided by the number of transported books provided a cost estimate of 12.8 cents per book (Apte & Mason, 2006). On the other hand, per item handling and processing costs inside the library are substantial. From an analysis at our own university library (De Boer, 2017), we found that fulfilling a request from another library takes 395 seconds on average, of which 150 seconds is for retrieving the book from storage. The remaining 245 seconds is for administrative processing, which is time consuming since it involves three separate IT systems. At an hourly wage of 25 euro, this amounts to a cost per book of 1.04 euro for retrieval and 1.70 euro for administration. Libraries in the same region often use the same IT system, which reduces administrative time. However, IT systems are often outdated and not user-friendly. Barton, Eighmy, Chao, Munson, and Varnum (2016) describe: “Staff processing of requests remains labor intensive. The ability to automatically route requests remains underdeveloped and stunted.”

A.7. Return times are more or less independent of rental times

Libraries indicate a deadline for returning a book, however, loans can easily be renewed for free via the library's website. Furthermore, if a book is returned late, often no fines are imposed. More than a third of libraries in the Netherlands do not charge fines, and this number is steadily increasing (Deckers, 2019). A similar situation is observed in the United States (ULC, 2020). Moreover, the American Library Association asks libraries to reconsider their practice of imposing fines for late returns (ALA, 2019), since fines discourage library usage especially for those who could most benefit from it. These facts result in a situation in which return times of books correlate only weakly with the return deadline, as is also visible from practical data (see Section 3.1). Furthermore, we provide numerical results (see Section 7.7) showing that even if such correlation exists, the assumption is not restrictive.

A.8. Customers are willing to wait an infinite amount of time for a book

Customers may renege in practice if the waiting is too long, however, the probability of this event occurring is small and therefore of little consequence for the outcomes of the model. First, customers are unlikely to renege as their willingness to wait is high, mostly because there is typically no viable alternative to obtain the book for free. In a survey, the acceptable waiting time was found to be twelve days (Ruigrok, 2017). Second, the probability that the requested book is available in the network is high. In the Netherlands, there are 24.3 million library books that are borrowed 66.5 million times (Lammers, 2020), so books on average are out-of-stock only three times per year. Finally, requested books typically remain available for pickup by the customer for two weeks. During this time, the book is unavailable for other customers, making the effect of a renegeing customer on system performance proportionate to a loan.

A.9. No preemptive supply of books is permitted, even if a library runs empty

Books are not shipped to a location, unless there is a request (and no stock) at that location. We explain this by contradiction. Suppose there is no current request for a book at a location, but nevertheless we transport it there. Then two situations may occur.

(a) The first request in the system does not occur at the location where the book was shipped to. Then unnecessary transport costs will have been made, as the book will need to be returned without having been used. (b) The first request in the system does occur at the location where the book was shipped to. Then no savings have been achieved in transport costs, but inventory costs could have been lowered by shipping the book at a later time. Hence, the only reason for moving a book in advance, would be to provide customers with an interesting portfolio of books to browse in the library, which is not the focus of this paper.

Appendix B. Proofs

B.1. Proof of Lemma 1

Proof. (i) It is easy to see that the transition probabilities in Eq. (13) are stochastically monotone in y , that is, $E[y_{t+1}|y_t] \leq E[y_{t+1}|y'_t]$ for $y_t \leq y'_t$ (Altman & Stidham, 1995). Now consider equivalent sample paths starting from state y and $y + 1$. In equivalent sample paths, the demands in each period are the same, as are the returns of the common y rented items, while the extra rented item in state $y + 1$ returns according to a Geometric(p) distribution. Let $\tau(x, y)$ be the time until item $m = x + y$ is demanded, defined in Eq. (15). For every equivalent sample path, it is evident that $\tau(m - y, y) \geq \tau(m - y - 1, y + 1)$. The holding cost benefit from a take-back, i.e., $(h - h_0)\tau(m - y, y)$, is thus non-increasing in y , while the transportation cost $2c$ is constant. Since the transition probabilities are stochastically monotone in y and the benefit from a take-back is non-increasing in y , it follows that α_y^m is monotonic non-increasing in y .

(ii) For the first part, assume $\alpha_y^m = 1$ is the optimal take-back decision for state y with m items. Now consider the action α_{y+1}^{m+1} in state $y + 1$ for the problem with $m + 1$ items. Both these states have on-hand stock $x = m - y$. Similar to (i), for each equivalent sample path we have $\tau(m - y, y + 1) \geq \tau(m - y, y)$. For the same x , the benefit from a take-back is non-decreasing in y , hence $\alpha_y^m = 1$ implies $\alpha_{y+1}^{m+1} = 1$. By induction, the claim follows.

For the second part, assume that $\alpha_y^m = 0$. Now suppose $\alpha_{y-1}^{m-1} = 1$. The first part implies that $\alpha_y^m = 1$, leading to a contradiction. Hence, if $\alpha_y^m = 0$, then $\alpha_{y-k}^{m-k} = 0$ for all $k \in 0, \dots, y$.

(iii) For the first part suppose $\alpha_y^m = 1$ for some y and m . Then by (ii), $\alpha_y^m = 1$ implies that $\alpha_{y+1}^{m+1} = 1$. By (i), $\alpha_{y+1}^{m+1} = 1$ implies $\alpha_y^{m+1} = 1$. Hence, if $\alpha_y^m = 1$ then $\alpha_y^{m+1} = 1$. By induction $\alpha_y^k = 1$ for all $k > m$. The second part can be proven analogously. \square

B.2. Proof of Proposition 1

Proof. (i) Immediate from Lemma 1 (iii).

(ii) Lemma 1 (ii) implies $x^*(y) \geq x^*(y + 1)$. We still need to prove $x^*(y) \leq x^*(y + 1) + 1$. To that end, suppose that $y + 1$ and threshold $x^*(y + 1)$ are given. Then it is optimal to take back on-hand item $x^*(y + 1) + 1$, or, equivalently, $\alpha_{y+1}^m = 1$ in the MDP for last on-hand item $m = x^*(y + 1) + y + 2$. By Lemma 1 (i), then also $\alpha_y^m = 1$, so the on-hand stock after take-backs for y is at most $m - y - 1 = x^*(y + 1) + 1$. Hence, $x^*(y) \leq x^*(y + 1) + 1$. \square

B.3. Proof of Proposition 2

Proof. (i) Suppose the threshold $x^*(0) = m$ is given. In the threshold, we require $a_0^m = 0$ and $a_0^{m+1} = 1$. By Lemma 1 (i), we have $a_y^m = 0$ for all y . The expected cost of $a_0^m = 0$ is therefore $hE[\tau(m,$

0)], while the expected cost of $a_0^m = 1$ is $h_0 E[\tau(m, 0)] + 2c$. So in the threshold state we need to have

$$(h - h_0)E[\tau(x^*(0), 0)] \leq 2c < (h - h_0)E[\tau(x^*(0) + 1, 0)].$$

As $E[\tau(x, y)]$ is increasing in x the result follows.

(ii) Introduce a three state finite Markov chain for the last on-hand item $m = y + 1$. This chain starts in state y with 1 on-hand item and y rented items. With probability $P_{y,y} = P(D - R(y) = 0)$ we remain in state y and pay $h - h_0$. With probability $P_{y,y} = P(D - R(y) < 0)$ we enter the set of states $Y = \{0, \dots, y - 1\}$ with boundary condition $V^m(Y) = 2c$ (take back the item). With probability $P_{y,m} = P(D - R(y) > 0)$ we enter state m with boundary condition $V^m(m) = 0$ (the item is rented by a customer). The value $V^m(y)$ of state y can be determined from

$$V^m(y) = h - h_0 + P_{y,y}V^m(Y) + P_{y,y}V^m(y) + P_{y,m}V^m(m),$$

resulting in

$$V^m(y) = \frac{h - h_0 + 2cP(D - R(y) < 0)}{1 - P(D - R(y) = 0)}.$$

Whenever $V^m(y) \geq 2c$, carrying out a take-back saves $V^m(y) - 2c \geq 0$. The take-back of the last on-hand item is therefore profitable whenever

$$P(D - R(y) > 0) \leq \frac{h - h_0}{2c}.$$

As $P(D - R(y) > 0)$ is decreasing in y the result follows. \square

Proof of Lemma 2

Proof. (i) First observe that the transition probabilities of y_1 and y_2 are monotone and independent of each other. Therefore, τ_2 is constant in y_1 for fixed y_2 . Now, as in the proof of Lemma 1 (i), consider an equivalent sample path for the stock process at location 1 when starting at y_1 and $y_1 + 1$. For every possible equivalent sample path, we have $\tau_1(m - y_1, y_1) \geq \tau_1(m - y_1 - 1, y_1 + 1)$. Since τ_1 decreases in y_1 , the take-back cost $2c_1P(\tau_1 \leq \tau_2)$ increases in y_1 , while the expected reduction in holding costs of a take-back decreases in y_1 . Therefore, α_{y_1, y_2}^m is monotonic non-increasing in y_1 .

(ii) Analogous to (i), an equivalent sample path argument for location 2 can be used to show that the expected cross-docking handling costs increase in y_2 . \square

B.4. Proof of Proposition 3

Proof. (i) Using similar sample path arguments as in Lemmas 1 and 2, we can show that for fixed values of y_1 and y_2 , $\alpha_{y_1, y_2}^m = 1 \Rightarrow \alpha_{y_1, y_2}^{m+1} = 1$ and $\alpha_{y_1, y_2}^m = 0 \Rightarrow \alpha_{y_1, y_2}^{m-1} = 0$. By induction the threshold result then follows.

(ii) Analogous to Lemma 2.

(iii) For this consider the following sample path argument. Suppose the on-hand stock at location 1 exceeds $x_1^*(y_1)$, so that in the SLP we would take back item m . For all sample paths with $\tau_1 \leq \tau_2$, the expected cost-savings for a take-back of item m are the same as in the SLP. Now consider a sample path with $\tau_2 < \tau_1$. Then the cost-savings from carrying out a take-back are $\tau_2(h_1 - h_0) + d$. Since, with the presence of location 2, the take-back action leads to at least the same cost-savings as in the SLP, we will hold no more than $x_1^*(y_1)$ on-hand items at location 1. \square

B.5. Proof of Lemma 3

Proof. The first moments of demand for item m are given by $\tau_1 = \tau_1(1, 0)$ and $\tau_2 = \tau_2(1, 0)$. Clearly, $\tau_i, i = 1, 2$, is Geom(p_i) distributed with $p_i = P(D_i > 0)$. Hence, $\min(\tau_1, \tau_2)$ is Geom($1 - (1 -$

$p_1)(1 - p_2)$) distributed. We thus find

$$E[\min(\tau_1, \tau_2)] = \frac{1}{1 - (1 - p_1)(1 - p_2)},$$

and

$$P(\tau_1 \leq \tau_2) = \frac{p_1}{1 - (1 - p_1)(1 - p_2)}.$$

Substituting the expressions into (19) (leaving out τ_A since it is ∞) and removing the common denominator gives the result. \square

References

- ALA (2019). *Resolution on monetary library fines as a form of social inequity (2018–2019 ALA CD#38 Rev.1/27)*. American Library Association.
- Alfredsson, P., & Verrijdt, J. (1999). Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science*, 45(10), 1416–1431.
- Altman, E., & Stidham, S. (1995). Optimality of monotonic policies for two-action Markovian decision processes, with applications to control of queues with delayed information. *Queueing Systems*, 21(3–4), 267–291.
- Apte, U. M., & Mason, F. M. (2006). Analysis and improvement of delivery operations at the San Francisco public library. *Journal of Operations Management*, 41(4), 325–346.
- Audunson, R., Aabo, S., Blomgren, R., Evjen, S., Jochumsen, H., Larsen, H., ... Koizumi, M. (2019). Public libraries as an infrastructure for a sustainable public sphere a comprehensive review of research. *Journal of Documentation*, 75(4), 773–790.
- Axsäter, S., Howard, C., & Marklund, J. (2013). A distribution inventory model with transshipments from a support warehouse. *IIE Transactions*, 45(3), 309–322.
- Baron, O., Hajizadeh, I., & Milner, J. (2011). Now playing: DVD purchasing for a multitilation rental firm. *Manufacturing & Service Operations Management*, 13(2), 209–226.
- Bartlett, W. K. (2014). *Floating collections: A collection development model for long-term success*. Santa Barbara, CA: Libraries Unlimited.
- Barton, B., Eighmy, M., Chao, Z., Munson, K., & Varnum, K. (2016). A vision for next generation resource delivery. Available at: <https://www.btaa.org/docs/default-source/library/attachment-8-btaa2dactioncommitteereportdraft.pdf>. Accessed May 2020.
- Bassamboo, A., Kumar, S., & Randhawa, R. S. (2009). Dynamics of new product introduction in closed rental systems. *Operations Research*, 57(6), 1347–1359.
- Boysen, N., Briskorn, D., & Emde, S. (2017). Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research*, 259(1), 293–307.
- Brinkmann, J., Ulmer, M. W., & Mattfeld, D. C. (2019). Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. *Computers & Operations Research*, 106, 260–279.
- Burman, E., & Brage, C. (2016). Implementing floating collections: to float or to sink? A case study. *Qualitative and Quantitative Methods in Libraries*, 5, 201–209.
- Ching, W. K., Yuen, W. O., & Loh, A. W. . . (2003). An inventory model with returns and lateral transshipments. *Journal of the Operational Research Society*, 54(6), 636–641.
- Coelho, L. C., Cordeau, J. F., & Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11), 2537–2548.
- Coelho, L. C., Cordeau, J. F., & Laporte, G. (2014). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52, 55–67.
- Cromwell, P., Alexander, G., & Dotson, P. (2008). Crime and incivilities in libraries: Situational crime prevention strategies for thwarting biblio-bandits and problem patrons. *Security Journal*, 21, 147–158.
- De Boer, B. (2017). *Optimalisering werkprocessen frontoffice*. University of Groningen. Internal document
- Deckers, M. (2019). Boetebarmeter. Available at: <https://www.markdeckers.net/2019/06/boetebarmeter-ruim-45-van-nederland.html>. Accessed May 2020.
- Du, Y., & Hall, R. (1997). Fleet sizing and empty equipment redistribution for center-terminal transportation networks. *Management Science*, 43(2), 145–157.
- George, D. K., & Xia, C. H. (2011). Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, 211(1), 198–207.
- Howard, C., Marklund, J., Tan, T., & Reijnen, I. (2015). Inventory control in a spare parts distribution system with emergency stocks and pipeline information. *Manufacturing & Service Operations Management*, 17(2), 142–156.
- Jain, A., Moinedjeh, K., & Dumrongkiri, A. (2015). Priority allocation in a rental model with decreasing demand. *Manufacturing & Service Operations Management*, 17(2), 236–248.
- Kemeny, J. G., & Snell, J. L. (1976). *Finite Markov chains*. Berlin: Springer.
- Lammers, E. (2020). Gezamenlijk collectieplan. Beleidskader collectiebeleid van het netwerk van openbare bibliotheekvoorzieningen 2020–2024. Available at: https://www.kb.nl/sites/default/files/docs/gezamenlijk_collectieplan_2020_2024.pdf. Accessed May 2020.
- Legros, B. (2019). Dynamic repositioning strategy in a bike-sharing system; How to prioritize and how to rebalance a bike station. *European Journal of Operational Research*, 272(2), 740–753.
- Li, Z., & Tao, F. (2010). On determining optimal fleet size and vehicle transfer policy for a car rental company. *Computers & Operations Research*, 37(2), 341–350.
- May, F., & Black, F. (2010). The life of the space: Evidence from Nova Scotia public libraries. *Evidence Based Library and Information Practice*, 5(2), 5–34.

- NIST (2017). Dataplot: Tabulated designs. Available at: <https://www.itl.nist.gov/div898/software/dataplot/designs.htm>. Accessed Sept. 2019.
- Papier, F., & Thonemann, U. W. (2008). Queuing models for sizing and structuring rental fleets. *Transportation Science*, 42(3), 302–317.
- Pasternack, B. A., & Drezner, Z. (1999). The videotape rental model. *Journal of Applied Mathematics and Decision Sciences*, 3(2), 163–170.
- Paterson, C., Kiesmüller, G., Teunter, R., & Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2), 125–136.
- Petersen, J. H., & Kooistra, R. B. (2020). Rethinking libraries in the book business. *Public Library Quarterly*, 39(1), 64–88.
- Puterman, M. L. (2009). *Markov decision processes: Discrete stochastic dynamic programming*. New York: John Wiley & Sons.
- Quandt, K.R. (2017). The magic and hard work behind Brooklyn public library's millions of materials. Available at: <http://www.bkmag.com/2017/04/12/magic-hard-work-brooklyn-public-library-millions-of-materials/>. Accessed May 2020.
- Rivera, A. E. P., & Mes, M. R. K. (2017). Anticipatory freight selection in intermodal long-haul round-trips. *Transportation Research Part E: Logistics and Transportation Review*, 105, 176–194.
- Ruigrok (2017). Biebpanel: Customer journey reserveren & verlengen. Available at: <https://bibliotheek.ideacultuur.nl/content/11-actueel/biebpanel/309-biebpanel-22>. Accessed May 2020.
- Scarre, G. (2017). The compages, the bonds and rivets of the race: W. E. Gladstone on the keeping of books. *Library & Information History*, 33(3), 182–194.
- Slaugh, V. W., Biller, B., & Tayur, S. R. (2016). Managing rentals with usage-based loss. *Manufacturing & Service Operations Management*, 18(3), 429–444.
- Song, D. P., & Carter, J. (2008). Optimal empty vehicle redistribution for hub-and-spoke transportation systems. *Naval Research Logistics*, 55(2), 156–171.
- Taguchi, G. (1986). *Introduction to quality engineering: Designing quality into products and processes*. Tokyo: Asian Productivity Organization.
- Tai, A. H., & Ching, W. K. (2014). Optimal inventory policy for a Markovian two-echelon system with returns and lateral transshipment. *International Journal of Production Economics*, 151, 48–55.
- ULC (2020). Fine free libraries map. Available at: <https://www.urbanlibraries.org/member-resources/fine-free-map>. Accessed May 2020.
- Ulmer, M. W., & Streng, S. (2019). Same-day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*, 108, 1–19.
- Van der Heide, G., & Roodbergen, K. J. (2013). Transshipment and rebalancing policies for library books. *European Journal of Operational Research*, 228(2), 447–456.
- Van der Heide, G., Roodbergen, K. J., & Van Foreest, N. D. (2017). Redistributing stock in library systems with a depot. *Computers & Operations Research*, 83, 66–77.
- Van der Heide, G., Van Foreest, N. D., & Roodbergen, K. J. (2018). Optimizing stock levels for rental systems with a support warehouse and partial backordering. *European Journal of Operational Research*, 265(1), 107–118.
- Van Heeswijk, W. J. A., Mes, M. R. K., & Schutten, J. M. J. (2017). The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science*, 53(1), 203–221.
- Vattulainen, P. (2004). National repository initiatives in Europe. *Library Collections, Acquisitions, and Technical Services*, 28(1), 39–50.
- Walsh, R. R. (1969). Branch library planning in universities. *Library Trends*, 18(2), 210–222.
- Wee, K. E., & Dada, M. (2005). Optimal policies for transshipping inventory in a retail network. *Management Science*, 51(10), 1519–1533.