Inventory Control for Multi-location Rental Systems
van der Heide, Gerlach

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:
2016

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

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).

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.

Download date: 15-09-2019
Chapter 4

Repositioning of Stock in Rental Systems with a Support Depot

Abstract. This chapter considers the rental system with a support depot, applied to the setting of libraries. The support depot serves as a station for carrying out shipments to libraries in case of stock-outs, and for storing low-demand rental stock at low costs. Since shipments deplete the inventory at the support depot, part of the rental stock from the libraries must be taken back to the support depot in order to deal with future shipment requests. These shipment and take-back operations are carried out periodically. We focus on optimizing the shipment and take-back decisions by modeling the system as a Markov decision process and investigating its optimal policy for various problem instances. For the take-back decision, we distinguish between so-called threshold, reactive, and preventive take-backs. We use the insights from the MDP to develop a three-phase take-back heuristic. In experiments our heuristic performs within 1% on average from the optimal solution. For settings with a large number of libraries, it is shown that acceptable performance can be achieved by setting a base-stock level at the support depot and taking back stock accordingly.

4.1 Introduction

As discussed in Chapter 1.2.4, public libraries use support depots for dealing with shipment requests. An important reason for choosing this system over the system with lateral transshipments from Chapter 2 is that lateral transshipments can be expensive and logistically complex due to a combination of low volumes and long distances. With a support depot, the combined demand of libraries can be shipped using a single transportation device. The operational costs for carrying out shipments from the support depot are typically lower than for libraries, since the depot’s design can be process-driven instead of sales-driven, tasks can be automated, and the depot itself need not be located in expensive public areas.

Various operational decisions have to be made periodically in this library system. When items are requested at a library which are not available locally, shipments from the support depot have to be carried out. In contrast to sales-driven companies, where stock is bought and sold, stock in library systems is often fixed and (close to) 100% of the rented items are returned by the client. The support depot will thus have to be resupplied by carrying out a take-back operation of items at the individual libraries. The main difficulty lies in deciding how many items to take back in total and from which libraries. Since due to budget cuts the government funding for public libraries has significantly decreased in the last several years, it is important to carry out these operations efficiently.

An often encountered practical problem for public libraries is that a large portion of the rental stock consists of low-demand items. Muckstadt & Thomas (1980) conclude that two-echelon systems are important for low-demand items. Hence, storage of rental items in a low-cost support depot may also be an effective strategy to reduce holding costs and free up space at the libraries for other items. An important problem is deciding which low-demand items should be stored in the support depot.

In this chapter we simultaneously consider the decisions for storing items at the support depot and the operational decisions for (re)distributing stock, because these decisions are closely related. We consider a periodic review model where demands and returns at the libraries occur between reviews. At the review, stock is observed and there is an option to carry out shipments and take-backs. By first solving an MDP for a problem with a single library and single support depot, we obtain the main insights for storing low-demand items at the support depot. Subsequently, we solve MDPs for
problems with multiple libraries. By analyzing the optimal policy of several example configurations, we obtain insights that help to formulate a near-optimal heuristic for larger problem instances. This heuristic is compared with the optimal policy and several other simple heuristics in various experiments.

The research on rental systems has focused mainly on multi-location systems without a support depot. Most common are rental systems for vehicles such as cars (Oliveira et al., 2014) and bikes (Ting & Liao, 2013). In vehicle rentals the preferred option for dealing with stock-outs is to provide substitute vehicles, so shipments from a support depot or another location are typically not considered. As rental stock may transfer through the system because customers return vehicles at another location after the rental period, Köchel et al. (2003) focus on repositioning the rental fleet over the locations. For rental items such as books, shipments can be a practical option because the items are easily shipped and clients are typically willing to wait for a shipment. Despite their practical relevance, to the best of our knowledge it seems that rental systems with shipments from a support depot have not yet been considered in literature.

A related problem is the control of vehicles in a hub-and-spoke system. In library systems items are typically returned to the library they were originally rented from, whereas in hub-and-spoke system vehicles rented at the hub return at the spoke and vice versa. These essentially different dynamics demand different strategies for repositioning stock. Song & Carter (2008) derive the optimal repositioning policy for empty cars in a system with a single spoke and hub. This decomposition is used to formulate an effective heuristic. Köchel (2007) have a system in which orders at spokes are placed in batches, and transported from the hub using a limited number of trucks. Based on the analysis for a system with a single spoke, they formulate a heuristic for optimizing the number of trucks and the assignment of these trucks to the spokes. In our library setting, we also use insights from a single library setting to formulate a heuristic for a multiple library setting.

As mentioned in the introduction of Chapter 3, emergency shipments from a support depot are common in spare parts inventory control (Kranenburg & Van Houtum 2009; Axsäter et al., 2013; Van Wijk et al., 2013). Stock in these spare parts systems is shipped from the support depot to local stock points, but no attention is paid to the take-back process of stock from local stock points to the support depot. In Chapter 3 the process of taking back items from the support depot is static, since all stock
shipped from the support depot is assumed to return to the support depot after the rental period. In this chapter we instead consider a dynamic take-back process, with decisions that are allowed to vary depending on the current availability of stock in the system.

The outline of the chapter is as follows. §4.2 introduces the model for the library system with a support depot. In §4.3 MDPs are solved for one, two, and three library problems to gain insight into the relevant trade-offs. In §4.4 and 4.5 shipment and take-back heuristics are developed, which are compared with the optimal policy and with each other in §4.6. §4.7 concludes.

4.2 Model for Shipping and Repositioning Stock

In this section we formulate the problem of shipping and repositioning stock for a library system with \( n \) libraries and one support depot. The system is depicted schematically in Figure 4.1. A downstream movement of stock from the support depot to the libraries is called a shipment. An upstream movement from a library to the support depot is called a take-back. We consider the inventory control for a single item type, e.g., a specific book title. There is a fixed number \( K \) of these items available in total in the system.

At time \( t \), the support depot has stock \( x_{0t} \geq 0 \). Each library \( i \) has stock \( x_{it} \), which may be negative in case of backorders, and \( y_{it} \geq 0 \) items rented by clients. The state of the system at time \( t \) is represented by the tuple \( S_t = (x_{0t}, x_t, y_t) \), with \( x_t = (x_{1t}, \ldots, x_{nt}) \) and \( y_t = (y_{1t}, \ldots, y_{nt}) \). Libraries in practice typically allow a limited number of backorders (or: reservations) per library in order to reduce administrative inconvenience and excessive waiting times. Therefore, we introduce a maximum \( \beta > 0 \) to the number of backorders per library, which implies \( -\beta \leq x_{it} \). Any additional demand is lost.

Figure 4.2 summarizes the time line of events in this system. The library system employs a periodic review policy. The length of the period depends on the specific application; in most library systems these reviews are executed on a daily, biweekly, or weekly basis. A period starts with clients demanding \( D_t \) new items and returning \( R_t \) previously rented items at the libraries. At the end of each period, the inventory levels at the libraries and the support depot are reviewed. Accordingly, shipment and take-back decisions are carried out. After transferring the stock, a new period starts.
We label the state variable after demands/returns, shipments, and take-backs with zero, one, and two primes, respectively. For example, the on-hand inventory levels after these respective phases are given by \( x_t, x'_t \) and \( x''_t \).

The shipment and take-back decisions facilitate pooling the rental items in the system. Shipments are carried out to deal with unfulfilled demand at the libraries, while take-backs are carried out to store items at the support depot. Trucks simultaneously execute the shipments and take-backs for a library, since after shipping it is efficient to fill the empty space in a truck with take-back stock. It is assumed that take-back stock cannot be used immediately for shipments to other libraries in the current period, since it is typically impractical and too time consuming to sort items en route. Hence, lateral transshipments (Paterson et al., 2011) between libraries are prohibited, but it is possible to ship an item from one library to another via the support depot with a delay of one period. Inexpensive lateral transshipments between nearby libraries can be included in the formulation by clustering nearby libraries to-
gether and regarding them as one library. In this way, libraries in the cluster receive
products from nearby libraries before requesting shipment from the support depot.
The support depot then forms an alternative for expensive lateral transshipments
between distant libraries.

For modeling purposes, we assume that the lead time of shipments and take-backs
is negligible. This zero lead time assumption is reasonable especially in settings with
online ordering. Most libraries promise clients, provided the online order is placed
before a final ordering moment, that currently unavailable ordered items can be picked
up after a scheduled delivery moment. The time between the final ordering moment
and the promised delivery moment can then be used for shipments and take-backs.

In the remainder of this section we explain in detail the three main phases of the
model: demands and returns, the shipment decision, and the take-back decision. We
describe the dynamics of the system with several recursive equations and introduce
the cost structure. Finally, we formulate a Markov Decision Process (MDP) for
optimizing the shipment and take-back decisions.

4.2.1 Demands and Returns

Library $i$ faces demand $D_{it}$ and receives returns $R_{it}$ during period $t$. It is assumed
that $D_{it} \sim \text{Poisson}(\lambda_i)$. Library $i$ has $y_{it}$ items rented to clients, which are assumed
to return with a probability $p$ each, so that $R_{it} \sim \text{Binomial}(y_{it}, p)$. The probability $p$
is taken the same for each library, since an analysis of a data set with approximately 4
million library loans indicates that distribution of rental durations does not vary much
between libraries. The total rental time of each individual item is then Geometric($p$)
distributed. We have chosen geometrically distributed returns to provide a clear
and insightful analysis. This is equivalent to exponential service times in standard
queueing models. More advanced return distributions can be modeled by extending
the state variable with the current rental time of each item. However, this yields
complex policies since different decisions have to be specified for each possible vector
of rental times.

Suppose that the state is $S''_{t-1} = (x''_{0,t-1}, x''_{t-1}, y''_{t-1})$ after the review of previous
period $t - 1$. The support depot faces no demand, hence

$$x_{0t} = x''_{0,t-1}. \quad (4.1)$$
Let $D_t$ and $R_t$ be the vectors of demands and returns. At the libraries returning items are added to the stock and demanded items are taken by clients. Backorders that exceed $\beta$ are cut off. Hence, we take the pairwise maximum of the new stock after demands/returns and $-\beta$, i.e.,

$$x_t = \max\{x'_{t-1} + R_t - D_t, -\beta\}. \quad (4.2)$$

For the rented items, note that the difference in on-hand inventory before and after demands and returns is given by $(x''_{t-1})^+ - (x_t)^+$, with $(x)^+ = \max\{x, 0\}$. If the on-hand inventory decreases, clients must have rented these items. If the on-hand inventory increases, clients must have returned items. The rented items after demands and returns are therefore given by

$$y_t = y''_{t-1} + (x''_{t-1})^+ - (x_t)^+. \quad (4.3)$$

The new state after demands and returns is $S_t = (x_{0t}, x_t, y_t)$.

### 4.2.2 Shipment Decision

At the end of period $t$ the stock levels of state $S_t$ are reviewed and a decision has to be made regarding how many items to ship. Let the decision variable $a'_t \geq 0$ be the vector with numbers of items shipped from the support depot to the libraries. For example, if $n = 3$ and $a'_t = (1, 0, 0)$, one unit is shipped to the first library.

Shipments only take place when there are backorders, provided the support depot has on-hand inventory. We do not consider proactive shipments from the support depot to the libraries, since clients are assumed to be satisfied whenever the item is shipped in the period the demand occurred. In this situation it is best to first observe the demand and ship in response, provided lost demand is not too likely. The support depot is assumed to fulfill all backorders in the system to the extent possible. This assumption is reasonable when backorders are considered expensive relative to shipping costs, which is the case in most library systems.

The backordered demand in the system at time $t$ is given by $\sum_{i=1}^n (x_{it})^-$ with $(x_{it})^- = -\min\{x_{it}, 0\}$. If the backordered demand exceeds the stock at the support depot, whatever stock available at the support depot is shipped downstream. Otherwise, all backordered demand is shipped. Hence, the number of shipped items
Chapter 4

The following restrictions on $a_t'$ ensure that the $A_t'$ items in total are shipped only to libraries with backorders:

\[
0 \leq a_t' \leq (x_t)^- \tag{4.5}
\]

\[
\sum_{i=1}^{n} a_{it}' = A_t'. \tag{4.6}
\]

In case $A_t' = \sum_{i=1}^{n} (x_{it})^-$, it is easy to see that the only feasible choice is $a_t' = (x_t)^-$, which means all backorders are dealt with. In case $A_t' < \sum_{i=1}^{n} (x_{it})^-$, some libraries cannot receive shipments, requiring a choice between libraries. This choice is not trivial, since a shipment influences not only backorders in the current period, but also on-hand stock at the library several periods later due to the future return of the item. We later solve an MDP to find the optimal decision.

After shipments at time $t$, the support depot’s inventory has decreased by $A_t'$:

\[
x_0t' = x_0t - A_t'. \tag{4.7}
\]

Since libraries with backorders receive $a_t'$ shipments, their number of backorders decreases by $a_t'$:

\[
x_t' = x_t + a_t'. \tag{4.8}
\]

Finally, all shipped items are given to waiting clients. Hence, the new number of rented items is

\[
y_t' = y_t + a_t'. \tag{4.9}
\]

The updated state after the shipment decisions is given by $S_t' = (x_0t', x_t', y_t')$.

### 4.2.3 Take-back Decision

The second part of the review at time $t$ concerns take-backs of stock from the libraries to the support depot. The state after shipments, $S_t'$, is first observed. Then a decision is made regarding which items, if any, are taken back from the libraries. These items are stored in the support depot and can be used for shipments. Let the decision
variable $a''_t \geq 0$ be the vector with the numbers of items taken back from libraries to the support depot. Clearly, there is the following restriction on $a''_t$:

$$0 \leq a''_t \leq (x_t)^+, \quad (4.10)$$

i.e., only combinations of on-hand inventory at the libraries are feasible take-back decisions. Note that (4.5) and (4.10) imply that the sets of libraries which are eligible for shipments and for take-backs are mutually exclusive, since by definition $(x_{it})^+ \cdot (x_{it})^- = 0$.

The state after take-backs updates straightforwardly. The support depot receives $\sum_{i=1}^n a''_{it}$ units, hence

$$x''_{0t} = x'_{0t} + \sum_{i=1}^n a''_{it}. \quad (4.11)$$

The local stock levels reduce by the number of items taken back

$$x''_t = x'_t - a''_t. \quad (4.12)$$

The rented items remain at the clients:

$$y''_t = y'_t. \quad (4.13)$$

The new state is then $S''_t = (x''_{0t}, x''_t, y''_t)$. Any remaining unmet demand carries over as a backorder into the next period.

### 4.2.4 Cost Structure

Libraries face several types of costs, which are assumed to be nonnegative and identical for all libraries in the system since they often have similar characteristics. There are holding costs $h_0 > 0$ at the support depot and $h > 0$ at the libraries per unit of on-hand stock per period. It is assumed that $h_0 < h$, since the support depot is dedicated to storage whereas libraries are dedicated to demand fulfillment. Moreover, there are unit costs $b > 0$ per backorder per time unit and $\ell > 0$ per lost demand. For shipments and take-backs existing infrastructure is used, with trucks driving a fixed route along the libraries every period. The main costs for transporting an additional item are therefore the handling costs for retrieving items from the shelf. Hence, there are transportation costs $c$ for each unit shipped and each unit taken back. We assume
$2c > h - h_0$ in order to rule out the trivial situation in which it is cheapest to always store all stock at the support depot regardless of the circumstances.

We assume that costs are incurred at the end of the period, immediately after carrying out shipments and take-backs. Let $a_t = (a'_t, a''_t)$ be the decisions in period $t$. Excluding lost demand costs, the costs in period $t$ with starting state $S_t$ and decisions $a_t$ are given by

$$ C_t(S_t, a_t) = h_0 x''_0 t + \sum_{i=1}^{n} \left( h(x''_{it})^+ + b(x''_{it})^- + c a'_{it} + c a''_{it} \right). \tag{4.14} $$

Here, the first part gives holding cost at the support depot after the periodic review. The summation includes the holding, backorder, shipment, and take-back costs for the libraries.

We deal with lost demand costs separately because these costs require explicit knowledge about the number of items cut off in (4.2). This information is not included in the state variable $S_t$ and keeping track of it would come at a significant computational cost. However, given knowledge of the state $S''_t$ after decisions we can calculate the expected lost demand costs for the demands/returns phase of the next period. The expected lost demand in the transition after period $t$ is given by

$$ L_t(S_t, a_t) = \ell \sum_{i=1}^{n} E[(x''_{it} + R_{i,t+1} - D_{i,t+1} + \beta)^-], \tag{4.15} $$

which is the unit lost demand cost multiplied by the total expected number of lost demands. The expression depends explicitly on $a_t$ to show that the decision influences the lost demand costs of the next period. The total transition cost for decision $a_t$ in state $S_t$ is then $C_t(S_t, a_t) + L_t(S_t, a_t)$.

### 4.2.5 Markov Decision Process

Now we can model a Markov Decision Process with an average cost criterion (Tijms, 2003). Let $S$ the state space and let $V_0(S) = 0$ be the terminal costs for all $S \in S$. The value function is then

$$ V_t(S) = \min_{a_t} \{ C_t(S, a_t) + L_t(S, a_t) + E[V_{t-1}(S)] \}. \tag{4.16} $$
This is simply the sum of the direct costs for decision $a_t$, the expected lost demand costs made during the transition after decision $a_t$, and the expected value of the states reached after demands and returns.

An optimal policy for this average cost MDP can be found using value iteration. For each state the policy specifies which decision to take. Letting $M_t = \max_{S \in S} \{V_t(S) - V_{t-1}(S)\}$ and $m_t = \min_{S \in S} \{V_t(S) - V_{t-1}(S)\}$, the following convergence criterion for the average cost MDP is applied

$$\frac{M_t - m_t}{m_t} < \epsilon.$$  \hfill (4.17)

For the optimal average cost $g$ it is known that $m_t < g < M_t$ \cite{Tijms2003}. Since we consider average cost optimal policies, we drop the time index $t$ in the remainder of the chapter.

### 4.3 Examples of Optimal Shipment and Take-back Decisions

In this section we analyze optimal shipment and take-back decisions by solving the MDP from §4.2.5 for several example scenarios. We have calculated the optimal policy for multitudes of scenarios, and we consider the examples shown here to be representative for all scenarios. We start with the take-back decisions for the case with one support depot and one library, i.e. $n = 1$, where there is clearly no interaction between libraries. For the $n = 2$ case we then see how the inclusion of a second library influences these take-back decisions. If in the $n = 2$ case a take-back is required due to a backorder, then there is only one other library that can supply the item. Therefore we also consider the $n = 3$ case to see how to choose between potential supplying libraries. Finally, we pay attention to the shipment decisions by showing an example for the $n = 3$ case. In this way, we gradually build intuition in the various relevant aspects of the optimal policy.

#### 4.3.1 Take-backs in the Single Library Case

In the $n = 1$ case there is a single library with a support depot. It is important to observe that all shipment decisions are fixed here, since the support depot ships
its available stock whenever the library has a backorder. We can thus restrict the analysis to the take-back policy.

Figure 4.3 graphically shows the take-back policy for two example configurations that differ only in the number of items $K$. For each combination of on-hand stock $x_1$ and rented stock $y_1$, the graphs show the optimal stock levels after the take-back. The solid line can be regarded as a threshold on the on-hand inventory. If, for a given value of $y_1$, $x_1$ exceeds the solid line, then all items above the line are taken back to the support depot. If, for a given value of $y_1$, $x_1$ is below the solid line, then the optimal decision is to do nothing. The right graph shows the line for all values of $y_i$, obtained by setting $K = 9$. The line decreases in $y_i$ and ultimately becomes 0.

![Optimal take-back decisions with K=4](image1)

![Optimal take-back decisions with K=9](image2)

Figure 4.3: Optimal stock levels after the take-back in a single library problem with $K = 4$ and $K = 9$. For both graphs, the parameters are $\beta = 2, \lambda = 0.3, p = 0.3, h_0 = 0.7, h = 1, c = 5, b = 10, \ell = 20$.

The result that the optimal policy resembles a threshold structure can be explained by considering the last on-hand item on shelf at the library, which will be on-hand the longest of all items. We can decide to take-back the last on-hand item to the support depot at cost $c$, gain a reduction of costs $h - h_0$ each period it is stored there, and ship it at cost $c$ as soon as it is demanded at the library. Clearly, the expected number of periods with reduced holding costs should be sufficient to cover the costs $2c$. It seems intuitive that the threshold decreases in $y_1$. The expected number of returning items $R_1$ increases in $y_1$, so the net demand $D_1 - R_1$ decreases in $y_1$. Since the net demand decreases, fewer on-hand items are required to meet demand and therefore the threshold decreases in $y_i$. From now on we refer to the items taken back with the purpose of saving holding costs as threshold take-backs.
4.3.2 Take-backs in the Two Library Case

Now consider the take-back policy for the \( n = 2 \) case. We will again graphically show the take-back decisions in order to observe the trade-offs in the optimal policy. Denote the total number of items at library 1 and 2 by \( K_1 \) and \( K_2 \), with \( K_i = (x_i)^+ + y_i, i = 1, 2 \). Figure 4.4 depicts the optimal take-backs at library 1 for a scenario with the same parameters as in Figure 4.3, with the addition of a second library with demand rate \( \lambda_2 = 0.2 \). Four graphs are shown with \( K_1 = 4, 3, 2, 1 \) items dedicated to library 1 and \( K_2 = 0, 1, 2, 3 \) to library 2. The \( K_1 = 0 \) case is not shown since take-backs from library 1 are not possible.

The figure depicts only the decisions for states with \( x_0 = 0 \) in order to reduce the dimensions of the graph. For each value of \( y_1 \), the height of the bars represent the total available stock \( x_1 \) at library 1 before the take-back. The threshold lines represent the stock after the take-back, and these thresholds may vary with the state of \( x_2, y_2 \) at library 2. For example, for \( K_1 = 4, K_2 = 0 \) with \( x_1 = 3, y_1 = 1 \), we take-back 1 item if the state \( x_2, y_2 \) at library 2 is \((0,0)\), but 2 items if the state is \((-1,0)\) or \((-2,0)\).

We now elaborate on various important observations from Figure 4.4. It is interesting to compare the take-back policy of the \( K_1 = 4, K_2 = 0 \) case with the single library take-back policy for \( K = 4 \) in Figure 4.3. For any \( y_1 \), the thresholds in the two library case are lower than or equal to the single library threshold. The intuition is that, in addition to a favorable trade-off between transportation and holding costs, the take-back of the last on-hand item prevents backorder costs at the other library. Therefore, it seems preferable to take-back at least as much in the two library case as in the single library case.

As in the single library case, we observe that the thresholds decrease in \( y_1 \). The height of the threshold can vary quite heavily with the state at library 2. Interestingly, in the \( K_1 = 3, K_2 = 1 \) case there are even three different thresholds for the four possible states at library 2. The thresholds seem to decrease in the on-hand stock at library 2. Hence, if stock at library 2 is relatively scarce, more of the stock from library 1 is taken back to the support depot. Especially with backorders, i.e., \( x_2 < 0 \), the number of take-backs from location 1 is high.

Besides threshold take-backs, we can distinguish two other types of take-backs here. The first type is a reactive take-back, carried out in response to current backorders at library 2. There is a clear incentive for this because of the backorder cost...
Figure 4.4: Take-back policy at library 1 for a two library problem. Parameter settings: $K = 4, \beta = 2, \lambda_1 = 0.3, \lambda_2 = 0.2, p = 0.3, h_0 = 0.7, h = 1, c = 5, b = 10, \ell = 20$.

$b$ each time unit. We see for the cases $K_1 = 4, K_2 = 0$ and $K_1 = 3, K_2 = 1$ that the number of items taken back is always sufficient to meet all backorders at library 2. For the case $K_1 = 2, K_2 = 2$, only one item is taken-back in case $x_1 = 2$ and $x_2 = -2$. For the case $K_1 = 1, K_2 = 3$ take-backs from library 1 are never carried out. Since library 2 has several rented items, backorders are dealt with by waiting for a returning item. An additional reason not to take back the final item of library 1 is the potential future backorder cost at library 1 itself.

The second type is a preventive take-back, carried out to prevent costs of future backorders. In the graphs for $K_1 = 4, K_2 = 0$ and $K_1 = 3, K_2 = 1$ there are several states with $x_2 \geq 0$ for which the number of items taken back exceeds the number of threshold take-backs that we would expect from the single library case. The probability that stock-outs occur at library 2 is quite high in these states, motivating the decision to carry out preventive take-backs. For the states with $K_1 \leq 1$, no
preventive take-backs are carried out, because typically the stock-out probability of library 2 is low. Moreover, if the last item of library 1 is taken back, there is a significant probability that it has to be shipped back to library 1 immediately in the next period. Then a transportation costs $2c$ has been incurred for no tangible gain. In Figure [4.4] no states were shown with $x_0 > 0$, but even in these states preventive take-backs are sometimes carried out.

### 4.3.3 Take-backs in the Three Library Case

Here, we deal with the take-back policy for the $n = 3$ case. Most aspects of the optimal take-back policy have been covered in the discussion of the single and two library cases. However, the take-back policy does have an interesting extra aspect with $n > 2$ libraries. Namely, if a preventive or reactive take-back is required, it must be decided which library should supply the item. Table 4.1 below shows such take-back decisions for several states in an example configuration.

<table>
<thead>
<tr>
<th>#</th>
<th>$x_0$</th>
<th>$x$</th>
<th>$y$</th>
<th>$a''$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(2,1,1)</td>
<td>(0,0,0)</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(1,2,1)</td>
<td>(0,0,0)</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>(1,1,2)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>(1,1,1)</td>
<td>(0,1,0)</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>(1,1,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>(2,1,0)</td>
<td>(0,1,0)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>(2,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
</tr>
</tbody>
</table>

In states 1, 2, and 3 listed in Table [4.1] all stock is on-hand with no rented items. Here, in the state with $x = (2, 1, 1)$ no take-backs are carried out. The stock is located perfectly in accordance with the heights of the demand rates. However, if $x = (1, 2, 1)$ or $(1,1,2)$, then we will take-back the second item from the library with the highest stock level. These libraries have lower demand than library 1, so by a take-back we can prevent potential backorders at library 1. States 4 in 5 have in common that $x = (1,1,1)$. Here, no take-back is carried out if library 2 has a rented item, but a take-back is carried out if library 3 has a rented item. Since library 3 has a lower demand rate, we are willing to take away its final item. However, library 2 has a
reasonable chance that the on-hand item is needed in the next period, so if we carry out a take-back, then we may have to ship it back in the next period, leading to unnecessary shipment costs. In states 6 and 7 we have stock $x_1 = 2$ at library 1, and one of the other libraries has $x_i = 1$ and $y_i = 1$. If this is at library 2, then we will take back one item from library 1. However, if this is at library 3, then we will take back one item from library 3 since it has the lowest demand rate and hence a low stock-out probability.

4.3.4 Shipments in the Three Library Case

Now consider the shipment decisions. The shipment decision is often easy due to the assumption that items are shipped from the support depot whenever there is a backorder somewhere. If $x_0 \geq \sum_{i=1}^{n} (x_i)^-$, then the support depot is capable of meeting all backorders. Only if multiple libraries have $(x_i)^- > 0$ and $x_0 < \sum_{i=1}^{n} (x_i)^-$, a choice between the two libraries is needed.

In Table 4.2 we show the optimal shipment decisions in various states, using almost the same configuration as in Table 4.1 with $\ell = 60$ instead. This higher value of $\ell$ prevents irregular behavior at the border of the state space, which we explain at the end of this section. All states in Table 4.2 are such that $x_0 = 1$, $x_1 < 0$, $x_2 < 0$, $x_3 = 0$, and $y_3 \geq 0$. We vary $y_3$ so that we can see decisions for all possible combinations of rented items at libraries 1 and 2. The reason to consider these specific states is that, since we have only one item at the support depot, it is easy to see which of the two libraries we should ship to.

In states 1-12 we have $y_1 < y_2$ and in states 13-24 we have $y_2 > y_1$. In all of these states, shipment decision $a_i' = 1$ for the library $i$ with the lowest $y_i$. An important reason to ship with priority to these libraries is that they receive the fewest returns in the future, resulting in considerable backorder costs if left unsolved. In addition, this maximizes the probability that backorders at another library are met with returns, which could lead to reduced shipment costs in future periods.

In states 25-32 the number of rented items at both libraries is equal. The number of backorders and the height of the demand rate seem to break the tie. Here, the item is always shipped to the library with the highest number of backorders. This increase the probability that an additional backorder is met by the same shipped item after the client returns it. Finally, if the number of backorders is equal, i.e., in cases 25, 28, 29, and 32, the highest demand rate is prioritized. By shipping to the library
Table 4.2: Optimal shipment decisions for an \( n = 3 \) configuration. Parameter settings: \( K = 4, \beta = 2, \lambda_1 = 0.3, \lambda_2 = 0.2, p = 0.3, h_0 = 0.7, h = 1, c = 5, b = 10, \ell = 60 \).

<table>
<thead>
<tr>
<th>#</th>
<th>( x_0 )</th>
<th>( x_1, x_2 )</th>
<th>( y_1, y_2 )</th>
<th>( a_1', a_2' )</th>
<th>#</th>
<th>( x_0 )</th>
<th>( x_1, x_2 )</th>
<th>( y_1, y_2 )</th>
<th>( a_1, a_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(0,1)</td>
<td>(1,0)</td>
<td>17</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(2,0)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(0,1)</td>
<td>(1,0)</td>
<td>18</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(2,0)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(0,1)</td>
<td>(1,0)</td>
<td>19</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(2,0)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(0,1)</td>
<td>(1,0)</td>
<td>20</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(2,0)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(0,2)</td>
<td>(1,0)</td>
<td>21</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(2,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(0,2)</td>
<td>(1,0)</td>
<td>22</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(2,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(0,2)</td>
<td>(1,0)</td>
<td>23</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(2,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(0,2)</td>
<td>(1,0)</td>
<td>24</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(2,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(1,2)</td>
<td>(1,0)</td>
<td>25</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(0,0)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(1,2)</td>
<td>(1,0)</td>
<td>26</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(0,0)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(1,2)</td>
<td>(1,0)</td>
<td>27</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(0,0)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(1,2)</td>
<td>(1,0)</td>
<td>28</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(0,0)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(1,0)</td>
<td>(0,1)</td>
<td>29</td>
<td>1</td>
<td>(-1,-1)</td>
<td>(1,1)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(1,0)</td>
<td>(0,1)</td>
<td>30</td>
<td>1</td>
<td>(-1,-2)</td>
<td>(1,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(1,0)</td>
<td>(0,1)</td>
<td>31</td>
<td>1</td>
<td>(-2,-1)</td>
<td>(1,1)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(1,0)</td>
<td>(0,1)</td>
<td>32</td>
<td>1</td>
<td>(-2,-2)</td>
<td>(1,1)</td>
<td>(1,0)</td>
</tr>
</tbody>
</table>

with the highest demand rate, the item ends up in a place where it typically can be used more effectively to meet future demand.

In the example we could sort the libraries lexicographically by lowest number of rented items, highest number of backorders, and highest demand rate. From extended additional experiments, we observed that this ordering also seems to hold for most other configurations with low values of \( \beta \), i.e., \( \beta = 1 \) or \( \beta = 2 \), as is the case for most libraries in practice. For larger values of \( \beta \), we sometimes observed that libraries with the highest number of backorders are prioritized over libraries with the least number of rented items, so that backorders can be dealt with by future returns. In addition, the height of \( \ell \) may play an important role. If the lost demand cost is low, it may be favorable to avoid shipping to libraries with \( x_i = -\beta \) or \( x_i \) close to \( -\beta \). Such irregularities at the border of the state space are due to the modeling choice of having a backorder cost per time unit and a fixed cost per lost demand. In this case losing future demand may be preferable over having backorders that remain in the system for a considerable number of periods. If \( K \) is extremely low compared to the demand, this effect may have a significantly influence on the costs. This situation does not often occur in practice since \( K \) is typically chosen high enough to fulfill most of the demand from on-hand stock. Hence, we will pay no further attention to this effect.
when we develop heuristics.

4.3.5 Summary

We have shown several examples of optimal take-back and shipment decisions. While this is not a conclusive mathematical analysis, it does provide insights into appropriate repositioning policies for stock in a public library system with a support depot. We use these insights to develop heuristics and in experiments we will see that these heuristics have near-optimal performance.

For the take-back policy we distinguish between threshold, reactive, and preventive take-backs. With threshold take-backs we aim to reduce holding costs at the libraries by storing items at the support depot. The examples indicate that we can obtain these thresholds by solving the problem with a single library. Reactive take-backs are carried out in response to current backorders. Locations with no rented items are often prioritized. Preventive take-backs are carried out to prevent future backorders. From the examples we observe that it is effective to consider the probability that stock-outs happen in the next period. It seems reasonable to take-back the items from libraries which have a relatively low probability of a stock-out.

The shipment policy can be largely explained by shipping to libraries with a low number of rented items. Especially libraries with \( y_i = 0 \) are prioritized for receiving shipments. Backorders at these libraries cannot be met through future returns, leading to incurring backorder costs for prolonged periods of time.

4.4 Shipment Heuristics

Since the MDP can only be solved for limited size instances, we develop heuristics for large instances that are easy to apply in practice. Here, we formulate two heuristics for the shipment decision: the ordering shipment (SO) heuristic and ship-to-first (SF) heuristic. The SO heuristic assigns shipment requests to libraries with backorders by using a lexicographical ordering inspired by observations from the MDP in §4.3.4. The SF heuristic assigns the request to the library with the lowest index, which is a simple policy sometimes used in practice.

The SO and SF heuristics starts with initial shipment decision \( a' = 0 \). Then iteratively a new receiving library is determined from the set of libraries with backorders and its \( a'_i \) is increased by 1. This is repeated until either the support depot is empty.
or all backorders are dealt with, i.e.,

\[ x_0 - \sum_{i=1}^{n} a_i' = 0 \text{ or } \sum_{i=1}^{n} (x_i + a_i')^- = 0. \]

At any step of the SO and SF heuristic, determine \( I^- = \{ i : x_i + a_i' < 0 \} \) as the set of libraries with remaining backorders. The SO heuristic sorts libraries \( i \in I^- \) lexicographically in descending order according to the lowest number of rented items, \( y_i + a_i' \), the highest number of backorders, \( (x_i + a_i')^- \), the highest demand rate, \( \lambda_i \), and the lowest index, \( i \). The receiving library is the first library in the lexicographical order. For the SF heuristic the index of the receiving library is \( \min_{i \in I^-} \{ i \} \), i.e., the lowest index in the set of libraries with backorders.

### 4.5 Take-back Heuristics

In this section we develop a three phase take-back heuristic (TT). This heuristic consists of three separate phases in which we deal with the threshold, reactive, and preventive take-backs as introduced in §4.3. For each of these phases we explain how to iteratively determine a heuristic take-back decision \( a'' \). Moreover, two simpler heuristics are proposed which may be easier to coordinate in practice: the base-stock take-back (TB) heuristic and take-back all (TA) heuristic.

#### 4.5.1 Expected Stock-out Time

Before explaining the three phases, it is useful to define the expected stock-out time for a library, which will be used in the heuristic to select suitable libraries for reactive and preventive take-backs. The time until stock-out for library \( i \) can be characterized as follows. Let \( D_{is} \) denote the demand \( s \) periods from now and \( R_{is}(y_{i,s-1}) \) the returns \( s \) periods from now, which is a function of the rented items \( y_{i,s-1} \) in the preceding period \( s - 1 \). Letting \( y_{i,0} = y_i \), then the stock-out time for library \( i \) in state \( x_i, y_i \) is defined as

\[
T_i(x_i, y_i) = \min \left\{ t : \sum_{s=1}^{t} D_{is} \geq x_i + \sum_{s=1}^{t} R_{is}(y_{i,s-1}) \right\}
\]

(4.18)
i.e., the first moment that the total demand exceeds the on-hand stock plus the total returns up to time \( t \). Since \( T_i(x_i, y_i) \) is a hitting time of a finite Markov chain, its
expectation $\mathbb{E}[T_i(x_i,y_i)]$ can readily be obtained (see, e.g., Kemeny & Snell [1976]).

### 4.5.2 Phase 1: Threshold Take-backs

In the first phase threshold take-backs are carried out as discussed in §4.3.1, i.e., take-backs that lead to a favorable trade-off between holding costs and shipment costs, as in the single library case. We obtain these single library thresholds by solving the MDP for the $n = 1$ case for each library.

The number of threshold take-backs is calculated as follows. Let $x_i^*(y_i)$ denote the single library threshold level for library $i$ with $y_i$ rented items. Starting with inventory $x_i'$ and rented items $y_i'$, the threshold take-back $a_i''$ for library $i$ is

$$a_i'' = (x_i' - x_i^*(y_i'))^+.$$  

This is the starting value for $a_i''$. We continue with this initial take-back decision in the reactive take-back phase.

### 4.5.3 Phase 2: Reactive Take-backs

The idea for the reactive take-back phase is to determine the preferred number of items at the support depot to deal with current backorders, and carry out take-backs in accordance. Because of the threshold take-backs in Phase 1, the stock at the support depot may already exceed this preferred number. If the stock at the support depot turns out to be insufficient, additional items are taken back from libraries with on-hand stock according to the longest expected stock-out times.

We now describe the calculation of the preferred number of items at the support depot, denoted $a_0^*$. For each current backorder in the system we compared the expected backorder costs of no take-back with the transportation costs of a take-back. Suppose that library $i$ has a backorder. As a measure of the expected number of periods until the first item returns, define

$$f_i(y_i) = \begin{cases} 
\frac{1}{py_i} & \text{if } y_i > 0 \\
\infty & \text{if } y_i = 0
\end{cases},$$

which is the reciprocal of the average number of returns per period with $y_i$ rented items. If the choice is not to carry out a reactive take-back, a cost $b$ would be incurred
Repositioning of Stock in Rental Systems with a Support Depot

for an expected duration of $f_i(y_i)$ periods. Otherwise, a cost $2c + h_0 - h$ would be incurred once, which includes the holding cost difference from storing an item at the support depot for one period.

Let $I^- = \{i : x_i' < 0\}$ be the set of libraries with backorders. Define by $a_i^r$ the preferred number of reactive take-backs for the benefit of library $i \in I^-$, which is given by

$$a_i^r = \arg\max_{k \in \{1, \ldots, (x_i')^-\}} \{k : 2c + h_0 - h < bf_i(y_i + k - 1)\},$$  \hfill (4.19)

or $a_i^r = 0$ if the condition is never satisfied. Here, $a_i^r$ gives the maximum number of take-backs for which the direct cost of shipping are lower than the expected backorder costs. For the shipment of the $k$-th item, we assume that the previous $k - 1$ items have already been used to fulfill backorders. This accounts for the fact that shipped items can also solve additional backorders after completion of the rental period and is included to prevent shipping too much. The preferred total number of reactive take-backs is then simply

$$a_0^r = \sum_{i \in I^-} a_i^r.$$

Before carrying out these take-backs, we have to specify the libraries that supply an item. According to the observations in §4.3.3 libraries with low stock-out probabilities are typically the best candidates. We will therefore repeatedly select the library with the longest expected time until stock-out as candidate. Let $I^+ = \{i : x_i' - a_i'' > 0\}$ be the set of libraries with stock on-hand. The candidate library $j$ is then

$$j = \arg\max_{i \in I^+} \{\mathbb{E}[T_i(x_i' - a_i'' - 1, y_i)]\},$$  \hfill (4.20)

with $x_i' - a_i'' - 1$ the new stock level after the projected additional take-back.

Now iteratively increase the $a_i''$ of each subsequent candidate library from (4.20) by 1 until the support depot’s stock reaches $a_0^r$ or stock of all libraries is depleted:

$$x'_0 + \sum_{i=1}^n a_i'' \geq a_0^r \text{ or } \sum_{i=1}^n (x_i' - a_i'')^+ = 0.$$

As previously discussed, this condition may already be met due to the threshold take-backs.
4.5.4 Phase 3: Preventive Take-backs

In §4.3 a preventive take-back typically takes place when the support depot has a significant probability of failing to meet demand during the next period, provided the take-back does not increase the stock-out probability of the candidate library too much. The idea for the preventive take-backs is therefore to balance the regret of not taking back an item when it is required elsewhere in the system, against the regret of taking back an item when it is demanded at the candidate library itself in the next period.

As initial step in the preventive take-back phase, we will correct the starting stock levels for reactive take-backs. The $a_0^r$ items at the support depot from the reactive take-back phase have already been reserved for dealing with existing backorders, so this stock cannot be used for preventing future backorders. Therefore, during the preventive take-back phase we consider the inventory levels

$$\hat{x}_i = (x_i' - a_i'')^+ \text{ and } \hat{x}_0 = (x_0' + \sum_{i=1}^{n} a_i'' - a_0^r)^+,$$

i.e., the current on-hand inventory at the libraries and the current stock at the support depot not reserved for reactive take-backs.

Now we calculate the one-period regret and savings. Two events may occur. The first event is that an item is taken back from library $j$ in the current period and it is demanded at that same library in the next period. Then the regret is $2c - h + h_0$, which consists of the transportation costs minus the decrease in holding cost for storing the item at the support depot for one period. The second possible event is that no item is taken back from library $j$, but it is required by another library in the next period. Then the savings are $b + h - h_0$. This occurs whenever the item is not demanded at library $j$, while more than $\hat{x}_0$ shipments are required in total. The probability of the former event is

$$p_j = P\left(D_j - R_j > \hat{x}_j - 1\right),$$

where $\hat{x}_j - 1$ is the stock level after the take-back. The probability of the latter event is

$$P\left(D_j - R_j \leq \hat{x}_j - 1\right)P\left(\sum_{i \neq j} (D_i - R_i - \hat{x}_i)^+ > \hat{x}_0\right) = (1 - p_j)s_j$$
where \( s_j = P\left(\sum_{i \neq j} (D_i - R_i - \hat{x}_i)^+ > \hat{x}_0\right) \).

The probability \( s_j \) may be difficult to compute because it is a convolution of the demand and return distributions of \( n - 1 \) libraries. We therefore approximate \( s_j \) for large \( n \) by assuming that library \( i \) requests shipments from the support depot according to a Poisson distribution with demand rate

\[
\tilde{\lambda}_i = -\log\left(P(D_i - R_i \leq \hat{x}_i)\right).
\]

With this demand rate, the approximate probability of zero shipment requests from library \( i \) equals the exact probability, namely \( P(D_i - R_i \leq \hat{x}_i) \). The probabilities for overflow, i.e., \( D_i > R_i \), are approximate. In case \( \hat{x}_i = 0, y_i = 0 \), it is easily established that \( \tilde{\lambda}_i \) equals the exact demand rate for shipments \( \lambda_i \). The aggregate demand \( \tilde{D}_0 \) for shipments is Poisson\(\left(\sum_{i \neq j} \tilde{\lambda}_i\right) \) distributed. The approximate value for \( s_j \) is therefore given by

\[
\tilde{s}_j = P(\tilde{D}_0 > \hat{x}_0).
\]

We test the effectiveness of this approximation in the results section.

Candidates are selected, as in the reactive take-back phase, according to Eq. (4.20). A take-back from candidate library \( j \) is carried out whenever its expected savings exceed its regret, that is,

\[
(2c - h + h_0)p_j \leq (b + h - h_0)(1 - p_j)s_j. \tag{4.21}
\]

If (4.21) holds, then the \( a''_j \) of the candidate is increased by one. With the updated stock levels and a new candidate, we check the criterion until it is not satisfied or until \( \sum_{i=1}^{n} \hat{x}_i = 0 \). This gives the final value of the take-back decision \( a'' \).

### 4.5.5 Alternative Take-back Heuristics

The presented take-back heuristic has several complex steps, including the calculation of a single library policy using an MDP. In order to assess the added value of these complexities, we introduce two simple alternative heuristics for comparative experiments.

The Take-back Base-stock heuristic (TB) has base-stock level \( S \). Every review period it is ensured that the support depot has \( S \) units, if the total on-hand stock in
the system so permits. Hence, we set

\[ A = \min\{S - x_0, \sum_{i=1}^{n} (x_i)^+\} \]

as the total number of take-backs and iteratively ship these from the candidate library according to (4.20). This heuristic provides a practical and intuitive way to resupply the support depot in addition to preventing backorder costs at libraries.

The Take-back All (TA) heuristic takes back all on-hand stock at the libraries to the support depot. This is achieved by setting \( a''_i = (x_i)^+ \). The TA heuristic is a special case of the TB heuristic with \( S = K \).

### 4.6 Experimental Results

In this section we evaluate the performance of the shipment and take-back heuristics. We test the heuristics in instances with small \( n \) by comparing them with the optimal solution of the MDP. For large \( n \) we run the heuristics and compare them with each other. For reference, Table 4.3 summarizes the names and main concepts of the heuristics.

**Table 4.3: Look-up table for the names of the heuristics.**

<table>
<thead>
<tr>
<th>Shipment heuristics</th>
<th>Take-back heuristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO</td>
<td>Sort by lowest rented items, highest backorders, and highest demand rate</td>
</tr>
<tr>
<td>SF</td>
<td>Ship to the first library with backorders</td>
</tr>
<tr>
<td>TT</td>
<td>Three phase heuristic</td>
</tr>
<tr>
<td>TTa</td>
<td>Three phase heuristic with an approximation for ( s_j )</td>
</tr>
<tr>
<td>TB</td>
<td>Take-back according to a base-stock policy at the support depot</td>
</tr>
<tr>
<td>TA</td>
<td>Take back all stock from the libraries</td>
</tr>
</tbody>
</table>

The experiments will be based on a Taguchi design (Taguchi, 1986), which, due to their orthogonality, are deemed suitable to test a wide range of parameter values with a relatively small number of experiments. We use a Taguchi design with 18 experiments and 3 levels per factor. The values for the parameters \( h_0, b, c, \beta \), and \( \lambda_i, i = 1, \ldots, 4 \) in the experiments are specified in Table 4.4. For instances with \( n > 4 \), we explain later how we specify the demand rates. While it seems obvious
to vary the cost and demand parameters, we also vary $\beta$ to observe the effect of the partial backordering on the effectiveness of the heuristics. In each experiment, we set $h = 1, \ell = 2b$ and $p = 0.3$. This value for $p$ is the weekly return rate as estimated from a data set with 4 million library loans in the Netherlands in the year 2013. We later specify the choice of $K$. The value iteration is run with $\epsilon = 10^{-6}$.

Table 4.4: Experimental design for each $n$.

<table>
<thead>
<tr>
<th>#</th>
<th>$h_0$</th>
<th>$b$</th>
<th>$c$</th>
<th>$\beta$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>10</td>
<td>7</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>0.15</td>
<td>0.15</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>15</td>
<td>4</td>
<td>2</td>
<td>0.25</td>
<td>0.25</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>15</td>
<td>7</td>
<td>3</td>
<td>0.05</td>
<td>0.05</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>7</td>
<td>0.5</td>
<td>20</td>
<td>1</td>
<td>2</td>
<td>0.05</td>
<td>0.25</td>
<td>0.15</td>
<td>0.25</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>20</td>
<td>4</td>
<td>3</td>
<td>0.15</td>
<td>0.05</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>20</td>
<td>7</td>
<td>1</td>
<td>0.25</td>
<td>0.15</td>
<td>0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>10</td>
<td>0.8</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>0.25</td>
<td>0.15</td>
<td>0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>11</td>
<td>0.8</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>0.05</td>
<td>0.25</td>
<td>0.25</td>
<td>0.15</td>
</tr>
<tr>
<td>12</td>
<td>0.8</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>0.15</td>
<td>0.05</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>13</td>
<td>0.8</td>
<td>15</td>
<td>1</td>
<td>2</td>
<td>0.25</td>
<td>0.05</td>
<td>0.25</td>
<td>0.15</td>
</tr>
<tr>
<td>14</td>
<td>0.8</td>
<td>15</td>
<td>4</td>
<td>3</td>
<td>0.05</td>
<td>0.15</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>15</td>
<td>0.8</td>
<td>15</td>
<td>7</td>
<td>1</td>
<td>0.15</td>
<td>0.25</td>
<td>0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>16</td>
<td>0.8</td>
<td>20</td>
<td>1</td>
<td>3</td>
<td>0.15</td>
<td>0.25</td>
<td>0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>17</td>
<td>0.8</td>
<td>20</td>
<td>4</td>
<td>1</td>
<td>0.25</td>
<td>0.05</td>
<td>0.15</td>
<td>0.25</td>
</tr>
<tr>
<td>18</td>
<td>0.8</td>
<td>20</td>
<td>7</td>
<td>2</td>
<td>0.05</td>
<td>0.15</td>
<td>0.25</td>
<td>0.05</td>
</tr>
</tbody>
</table>

4.6.1 Performance of Heuristics in Small Instances

The shipment and take-back heuristics are compared to the optimal policy in an experiment with small instances. We use the following approach: given the fixed decisions of each shipment (take-back) heuristic, the MDP selects the cost-minimizing take-backs (shipments) corresponding to this heuristic. This approach generally gives a too optimistic estimate of the performance, since in practice all decisions are taken heuristically. Therefore, we also show the performance of combining the best shipment heuristic with the best take-back heuristic.

The number of libraries in the experiments varies between $n = 2, 3, 4$. For a specific $n$, only demand parameters $\lambda_i, i \leq n$ will be included. The number of items
$K$ is specified according to the rule

$$K = \left\lceil 1.8 \frac{\sum_{i=1}^{n} \lambda_i}{p} \right\rceil,$$

which can roughly be interpreted as having 1.8 times the stock required to meet the total system demand, scaled by the average return time $\frac{1}{p}$ of a rented item. The base-stock level $S$ of the TB policy is determined by calculating the average cost per time unit for $S = 1, \ldots, K$ and taking the cost-minimizing value.

Table 4.5 gives the average deviations from the optimal policy for each of the heuristics in the 18 scenarios. TT and TTa show the results for threshold take-back heuristics without and with the approximation for the preventive take-back. The column SO/TTa shows the result of combining the SO heuristic with the TTa heuristic. In detail, Figure 4.5 shows the average deviations for each experiment and each value of $n$. The deviations for the TT and TTa heuristic are almost the same, hence the graph only shows the TTa heuristic to avoid confusion.

<table>
<thead>
<tr>
<th>Shipment</th>
<th>Take-back</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SO</td>
<td>TT</td>
</tr>
<tr>
<td>$n = 2$</td>
<td>0.00%</td>
<td>0.63%</td>
</tr>
<tr>
<td>$n = 3$</td>
<td>0.11%</td>
<td>0.64%</td>
</tr>
<tr>
<td>$n = 4$</td>
<td>0.26%</td>
<td>0.54%</td>
</tr>
<tr>
<td>Total</td>
<td>0.12%</td>
<td>0.60%</td>
</tr>
</tbody>
</table>

From Figure 4.5 we can immediately see that the SO heuristic outperforms the SF heuristic in each experiment and for each value of $n$. Its average deviation is 0.12% and its maximum deviation is below 1%. The SO heuristic seems to perform slightly worse as $n$ grows, however shipping with priority to libraries with the least rented items seems appropriate in most instances. In any case, it outperforms shipping with priority to the first library with a backorder, as in the SF heuristic. The SF heuristic has an average deviation of 2.04% over all experiments and its performance is extremely volatile.

For the more complex take-back decisions, the take-back heuristics (TT, TTa, TB, and TA) typically have larger deviations from optimality than the shipment heuristics. The three phase take-back heuristics, TT and TTa, are within 1% from
optimality on average and almost all deviations are below 2%. In most experiments TTa and TT have the same deviation, although there are some minor differences in a few experiments. Overall the difference between using the exact or approximate stock-out probability seems to have negligible effect on costs.

From Figure 4.5 it seems that the take-back heuristics have the following order in terms of performance: TTa outperforms TB, which in turn outperforms TA. The TA is on average 13.01% from the optimal solution, implying that there are high costs associated with overestimating the number of required take-backs. Taking back all on-hand stock can be optimal for scenarios with low demand rates at all libraries and low shipment costs, but leads to high costs in most other scenarios. The Take-back Base-stock heuristic (TB) has a relatively large deviation of 2.66%. Since the TB heuristic has a static rule independent of the state, it is incapable of adapting to the situation when required, leading to high deviations in some scenarios. However, its
deviations seem to decrease as $n$ increases.

SO/TTa gives the deviation from optimality when shipments as well as take-backs are carried out heuristically. The gap with the optimal solution remains below 1% on average. In all experiments the total deviation from this combined policy is close to the sum of the deviations of the individual policies. The interaction between the shipment and take-back heuristic thus appears limited.

### 4.6.2 Take-back Heuristics in Larger Instances

In order to gain insight in performance of the take-back heuristics in instances with a higher number of libraries, we carry out a simulation experiment. The costs of the TTa, TA, and TB heuristic are compared with each other in instances with $n = 5, 10, 20, 50,$ and 100 libraries. For all these take-back heuristics, we take the shipment decisions according to the SO heuristic. Common random numbers are used, to the extent possible, to reduce the variability of the results. All heuristics face common Poisson demand $D_t$ in period $t$ of a given simulation run. Since the state variable $y_t$ varies between heuristics, we chose not to take common random returns $R_t$. Instead, each $R_t$ is drawn independently from a Binomial distribution.

As before, the experiments follow the design from Table 4.4. Since there are only four values for demand rates in this table, we need a new way to specify the demand rates. The following method is applied. The base demand rate for each library is $\lambda_1$ from Table 4.4, now denoted $\bar{\lambda}$. For each library, we set

$$\lambda_i = \bar{\lambda} + u_i,$$

with $u_i \sim N(0, 0.2\bar{\lambda})$ distributed. Adding this normal random noise leads to libraries with varied demand rates in the experiment. The number of items $K$ is again set according to (4.22).

For each experiment in the L18 array, we take 10 different random draws of the demand rates. For each random draw, 1000 simulation runs are carried out, giving 10000 runs in total for each scenario. In every run the system is simulated for 1000 weeks, excluding a warmup period of 100 weeks. The starting state for each policy is $x_0 = K, x_i = 0$ for $i = 1, \ldots, n$. For each random draw, the base-stock level $S$ of the TB policy is determined by simulation. For $S = 1, \ldots, K$, we calculate the average costs of 100 simulation runs and stop as soon as the average costs increase.
Table 4.6 shows the outcome of the experiment. The columns for the TA and TB heuristic give the percentage cost increase in each experiment relative to the TTa heuristic.

Table 4.6: The percentage increase in costs for the TA and TB heuristic compared to the TTa heuristic for a varying number of libraries.

<table>
<thead>
<tr>
<th>exp\n</th>
<th>Cost increase of TA in %</th>
<th>Cost increase of TB in %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>-0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>10.13</td>
<td>9.58</td>
</tr>
<tr>
<td>3</td>
<td>48.38</td>
<td>50.96</td>
</tr>
<tr>
<td>4</td>
<td>1.54</td>
<td>2.52</td>
</tr>
<tr>
<td>5</td>
<td>20.63</td>
<td>27.04</td>
</tr>
<tr>
<td>6</td>
<td>-0.57</td>
<td>0.17</td>
</tr>
<tr>
<td>7</td>
<td>0.18</td>
<td>0.13</td>
</tr>
<tr>
<td>8</td>
<td>3.44</td>
<td>5.90</td>
</tr>
<tr>
<td>9</td>
<td>51.13</td>
<td>62.39</td>
</tr>
<tr>
<td>10</td>
<td>4.03</td>
<td>5.97</td>
</tr>
<tr>
<td>11</td>
<td>0.41</td>
<td>0.91</td>
</tr>
<tr>
<td>12</td>
<td>22.52</td>
<td>25.37</td>
</tr>
<tr>
<td>13</td>
<td>3.90</td>
<td>7.88</td>
</tr>
<tr>
<td>14</td>
<td>-1.20</td>
<td>0.17</td>
</tr>
<tr>
<td>15</td>
<td>25.64</td>
<td>26.71</td>
</tr>
<tr>
<td>16</td>
<td>-0.42</td>
<td>0.34</td>
</tr>
<tr>
<td>17</td>
<td>41.42</td>
<td>52.10</td>
</tr>
<tr>
<td>18</td>
<td>0.23</td>
<td>1.44</td>
</tr>
<tr>
<td>avg.</td>
<td>12.85</td>
<td>15.54</td>
</tr>
</tbody>
</table>

The TA heuristic leads, as before, to relatively high costs. The TA heuristic has the lowest backorder and holding costs of all heuristics, but this is at the expense of extreme transportation costs. Because libraries hold no stock, there may also be high costs for lost demands. Taking back all stock seems appropriate in scenarios with low demand. In these scenarios the single library thresholds are zero, hence all heuristics lead to the same take-back decisions. However, the deviations in scenarios with higher demand are significant. The performance of the TA policy becomes relatively worse compared to the TTa heuristic as the number of libraries increases, because for high \( n \) the total system demand has relatively low variance, requiring a lower amount of stock at the support depot.

It should be clear that the TB policy performs reasonably well in this simulation. While the TTa policy is better than the TB policy in many of the scenarios, the
differences becomes smaller as $n$ increases. For small $n$ demand is volatile, requiring a policy which is capable of adapting to the situation. For large $n$ an invariant base-stock level will usually suffice, because demand for shipments from the support depot becomes quite constant. This can be regarded as a certain kind of pooling of demand. The TB policy could therefore be a suitable alternative in situations with a large number of libraries.

4.7 Conclusion

This chapter considers a library system with a low-cost support depot from which items are shipped in case of stock-outs at the libraries. In contrast to regular inventory systems, 100% of the rented items in a library system return. Therefore, restocking of the support depot must be achieved by taking back items from the libraries. We formulate and solve an MDP for several scenarios and use the obtained insights to create heuristics for large size problems. With this work we target two specific issues that are of interest to library organizations with such a system: storage of low-demand items and resupplying the support depot for future shipment requests.

The single library problem provides insights for storage of low-demand items. We observe that it is optimal to take back all on-hand items above a certain threshold which decreases in the number of rented items. These threshold take-backs lead to a favorable trade-off between the reduced holding costs and the extra transportation costs. Knowledge of this single library take-back policy can assist in practice in making decisions for removing rental items from libraries to create room for new and popular rental items.

The optimal resupply of the support depot follows from studying the MDP. We find that the concepts of reactive and preventive pooling from inventory theory are of importance in library systems: we observed reactive and preventive take-backs. These take-backs are largely explained by economic trade-offs between backorder costs and shipment costs. For library systems with many libraries, we have shown that a base-stock policy at the support depot may yield reasonable results.