European Journal of Operational Research 237 (2014) 824–835

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# Service differentiation through selective lateral transshipments

E.M. Alvarez *, M.C. van der Heijden, I.M.H. Vliegen, W.H.M. Zijm

*University of Twente, School of Management and Governance, P.O. Box 217, 7500 AE Enschede, The Netherlands*

ARTICLE INFO

*Article history:*
Received 19 October 2012
Accepted 22 February 2014
Available online 6 March 2014

*Keywords:*
Inventory
Service differentiation
Lateral transshipments
Spare parts

ABSTRACT

We consider a multi-item spare parts problem with multiple warehouses and two customer classes, where lateral transshipments are used as a differentiation tool. Specifically, *premium* requests that cannot be met from stock at their preferred warehouse may be satisfied from stock at other warehouses (so-called lateral transshipments). We first derive approximations for the mean waiting time per class in a single-item model with selective lateral transshipments. Next, we embed our method in a multi-item model minimizing the holding costs and costs of lateral and emergency shipments from upstream locations in the network. Compared to the option of using only selective emergency shipments for differentiation, the addition of selective lateral transshipments can lead to significant further cost savings (14% on average).

## 1. Introduction

In the capital intensive industry, a company's operations may fully depend on the performance of certain capital goods, such as radar systems on frigates or MRI and CT scanners in hospitals, with downtime possibly leading to severe consequences. The users of such equipment generally outsource maintenance and spare parts supply to a service provider, with targets on performance measures (such as a maximum response time to failures) formalized in service contracts through service level agreements (Al Hanbali & Van der Heijden, 2013). Because the installed base of capital goods is often geographically dispersed and waiting times for spare parts should be small, the supply chain is typically organized as multiple local stock points, each supporting a subset of the installed base, combined with a centrally located central warehouse. Cheap fast movers are typically stored at local warehouses, expensive slow movers tend to be concentrated in the central warehouse to take advantage of the risk pooling effect.

A complication for managing spare part supply chains is that service level agreements (SLAs) may vary strongly among customers to reflect the value placed on system uptime (Jalil, 2011). A key challenge for the supplier is to satisfy all SLAs at minimal costs. Part of this challenge is to position spare parts in the supply chain such that a target overall system downtime waiting for spares is met at minimal costs. Spare parts suppliers usually handle differentiated service levels by either (i) giving all customers uniform service (also known as "one-size-fits-all", Cohen, Agrawal, and Agrawal (2006)) or by (ii) keeping separate supply chains per customer segment, with premium customers served from stock points nearby and other customers served from the central warehouse. Uniform service is expensive – it must accommodate the tightest service levels – and does not induce standard customers to switch to a premium contract. Separate supply chains, on the other hand, reduce the benefit from risk pooling (Eppen & Schrage, 1981), resulting in higher stock levels in the supply chain than needed.

The literature on differentiation has mainly focused on the use of *critical levels* (Veinott, 1965). That is, a single supply chain is deployed for all customers, but stocks are reserved for premium customers once the inventory drops to a certain threshold, the critical level. Although this approach can lead to large savings in theory, there are practical drawbacks. For instance, service engineers will often not delay a repair if the required part is available, as they are generally accountable for speed of repair. To overcome these drawbacks, Alvarez, Van der Heijden, and Zijm (2013) propose an alternative differentiation approach using *selective emergency shipments*, where demand in out-of-stock settings may either be backordered or satisfied using emergency shipments depending on the customer's class and the item being considered.

In this paper, we extend the selective emergency shipment model by allowing *lateral transshipments for premium customers* as well. That is, a warehouse that is out of stock may obtain the item from a neighboring warehouse that still has the item on hand (see e.g. Kranenburg & Van Houtum, 2009). Such shipments often have shorter lead times than emergency shipments and are also cheaper. At the same time, lateral transshipments facilitate risk pooling, thereby reducing overall stock levels in the supply chain (Paterson, Kiesmüller, Teunter, and Glazebrook, 2011). We limit the use of

* Corresponding author.
*E-mail address:* e.m.alvarez@alumnus.utwente.nl (E.M. Alvarez).

http://dx.doi.org/10.1016/j.ejor.2014.02.053
0377-2217/© 2014 Elsevier B.V. All rights reserved.

lateral transshipments to premium customer requests only for the following reason. Suppose that a warehouse has only one part remaining on stock and a lateral request of a non-premium customer from another warehouse arrives. If the last item is used to fill this request, an "own" premium customer arriving a bit later will find an empty shelf. Intuitively, it is clear that this is not an adequate way to deal with high priority customers. Table 1 shows the order in which to fulfill a customer's demand at the warehouse by combining selective lateral transshipments and selective emergency shipments.

We can influence the system performance using three types of decisions: (i) the base-stock levels, (ii) the transshipment strategy, and (iii) the shipment strategy. The *transshipment strategy* specifies whether a premium request at a certain warehouse may be met through a lateral transshipment or not. Lateral transshipments cause additional shipment costs, but also result in lower stock levels since implicitly the risk pooling effect is exploited. The *shipment strategy* specifies whether an emergency shipment may be used if demand cannot be filled from stock on hand or through a lateral transshipment. If an emergency shipment is not used, the request is backordered until stock is replenished at the warehouse. If we decide not to use emergency shipments, we avoid high emergency shipments costs, but may need extra spare part inventories to meet the SLAs in terms of downtime waiting for spare parts.

We consider a supply chain of several warehouses supplying multiple spare parts to premium and non-premium customers. In turn, the warehouses are replenished from a central stock point. All locations use a continuous review, one-for-one replenishment policy. Each customer class has a distinct maximum average waiting time that applies over all parts jointly. As high waiting times for some parts might be compensated by low waiting times for other parts, we consider this as a *multi-item* problem, cf. Sherbrooke (2004). An item's waiting time and supply costs will depend on that item's stock levels in the system and on the (trans-)shipment strategy used if the nearest warehouse is out of stock. For simplicity, we assume that the central stock point has infinite stock which is also used for emergency shipments to the warehouses to satisfy customer demand as soon as possible. Shipments aimed at replenishing inventories of local warehouses only will be referred to as regular replenishments.

In Section 2, we give a literature overview and state our contribution. In Section 3, we present our multi-item model and globally describe our approach for solving this model. We then give details on the solution approach in Sections 4 and 5. Section 4 gives the analysis of a single-item building block with lateral transshipments for premium requests, whereas Section 5 details the heuristic solution of the multi-item problem. In Section 6, we discuss our extensive computational experiment. Finally, we draw conclusions and indicate further research areas in Section 7.

## 2. Literature

Our research is related to literature on service differentiation and emergency supply flexibility, i.e., using the flexibility of lateral transshipments and emergency shipments for meeting demand when the nearest warehouse is out of stock (Alfredsson & Verrijdt, 1999). Below, we focus on literature on lateral transshipments (possibly combined with emergency shipments). For literature on emergency shipments only, we refer to Alvarez et al. (2013). In the *service differentiation* area, we find contributions on both a tactical level (i.e., where the system stock levels are decision variables) and an operational level (i.e., where stock levels are given as input). Most papers on the *tactical level* apply differentiation using the critical level policy, a concept introduced by Veinott (1965). We refer to Teunter and Klein Haneveld (2008) for a literature review. Alternatively, Alvarez et al. (2013) use selective emergency shipments for differentiation with average cost savings of 4.4% over a one-size-fits-all approach. By combining selective emergency shipments with critical level policies considerably larger costs savings are possible; on average savings of 13.9% are found. The above papers only consider a single stock location. In contrast, Alvarez, Van der Heijden, and Zijm (in press) consider dedicated customer stocks as a differentiation tool, with stock possibly kept at customers' sites in addition to a central stock point. The resulting system is a two-echelon supply chain. The literature focusing on the *operational level* is limited to a few multi-location models. Jalil (2011) and Tiemessen, Fleischmann, Van Houtum, Van Nunen, and Pratsini (2012) consider single-item models with multiple warehouses and multiple customer classes, where a request can often be met from more than one warehouse. The supplier may choose to delay satisfying a low priority request or to meet such a request from a warehouse other than its nearest warehouse to reserve stock for premium requests.

The literature on *lateral transshipments* covers two types of models that differ in the way that demand is handled when it cannot be met from stock at either the nearest warehouse or through lateral transshipments from neighboring warehouses: the first model type then backorders demand, whereas the second satisfies it using emergency shipments. Models with *backordering* have initially been considered by Lee (1987) and Axsäter (1990), who consider a two-echelon setting consisting of a depot and various bases which are divided into transshipment pools. Axsäter uses an iterative analysis approach, where each base is analyzed separately over a number of iterations under the assumption that lateral transshipment requests at each base arrive according to Poisson processes. This logic has often been used in other papers, e.g. Alfredsson and Verrijdt (1999) and Van Wijk, Adan, and Van Houtum (2012). Models with *emergency shipments* have initially been considered by Dada (1992) and Alfredsson and Verrijdt (1999), who analyze similar two-echelon models. Some recent contributions are Kranenburg and Van Houtum (2009), where only a subset of warehouses can act as a lateral transshipment source, and Van Wijk et al. (2012), where a lateral transshipment request at a warehouse is only met if the stock level at that warehouse exceeds a so-called hold back level. In these latter two papers, it may in fact be that a lateral transshipment is not allowed even when some warehouses still have stock on-hand. We refer to Paterson et al. (2011) for details.

So far, lateral transshipments have only been considered as a service differentiation tool at an operational level, with contributions limited to single-item models (Jalil, 2011; Tiemessen et al., 2012). In contrast, we consider a multi-item model for which we

**Table 1**
Overview of order fulfillment options.

| Premium customers | Non-premium customers |
|---|---|
| 1. Stock on hand | 1. Stock on hand |
| 2. Lateral transshipment from other warehouse | 2. Emergency shipment from a central location upstream the supply chain |
| 3. Emergency shipment from a central location upstream the supply chain | 3. Backorder, wait for a replenishment order |
| 4. Backorder, wait for a replenishment order | |

calculate near-optimal values for both the stock levels and (trans-)shipment strategies, with lateral transshipments limited to premium customer requests. The use of lateral transshipments for differentiation may have significant added value: such shipments are generally both faster and less expensive than emergency shipments. Hence, if there is added value to use selective emergency shipments, it will likely be beneficial to use selective lateral transshipments as well. However, the feasibility of a lateral transshipment depends on the stock levels at other warehouses, whereas emergency shipments are always possible. Furthermore, as we aim to achieve a waiting time target over all parts jointly, the most suitable (trans-)shipment strategy for a certain item may depend on those of other items. In general, we have to decide how to allocate the waiting time allowed over the various items. So, we investigate in a multi-item setting under what conditions lateral transshipments are beneficial, for which type of items, and how often we should use each shipment option (emergency shipments, lateral transshipments, backordering). To do so, we need a new method to evaluate the performance of a single-item model when lateral transshipments are only used for premium customers. So our detailed contributions to the literature are:

1. We analyze a new single-item model in which we allow lateral transshipments to meet premium customer requests if this request cannot be met from the closest warehouse.
2. Using the above building block, we construct a multi-item model for service differentiation to calculate near-optimal values for the system stock levels and the various supply options. Specifically, we determine whether lateral transshipments are allowed if the nearest warehouse is out of stock, and whether emergency shipments are used if a request cannot be met from any warehouse (with a request backordered otherwise). Our approach, which is similar to Dantzig–Wolfe decomposition, is fast and gives good quality solutions. Although such an approach has been used on similar problems before, its application to our model is not straightforward, since we have a large number of decision variables.
3. In an extensive computational experiment, we show that the use of selective lateral transshipments in addition to selective emergency shipments leads to significant added value, especially in settings with slow movers.

## 3. Model

In Section 3.1, we describe the inventory system we consider, followed by the assumptions in Section 3.2 and the notation in Section 3.3. We conclude this section with the problem formulation in Section 3.4.

### 3.1. Description inventory system

We consider a multi-item network of multiple local warehouses and a central depot with infinite supply. Each warehouse has its own customer base consisting of premium and non-premium customers. Per customer class, there is a maximum on the average time customers of that class are willing to wait for parts, with the premium class having the most strict waiting time requirement. Direct requests at a warehouse (i.e., from its own customer base) are met from stock at the warehouse if possible, with a replenishment request sent to the depot (i.e., a continuous review, one-for-one replenishment policy). If the warehouse is out of stock, it may satisfy a *premium* customer request through a *lateral transshipment* (or *transshipment* in short) from another warehouse. If a request cannot be met from stock at any warehouse, an *emergency shipment* may be requested from the depot. Otherwise, the request is backordered. Whether such a transshipment or emergency shipment is allowed is specified by the demand fulfillment strategy that we will study. The fulfillment process of a customer request is summarized in Fig. 1.

We have three types of decision variables, whose values may vary per item and warehouse: (i) the base-stock level, (ii) the transshipment strategy, and (iii) the shipment strategy. If transshipments are allowed, transshipment requests are issued to other warehouses in a predetermined order. Such an order is common in

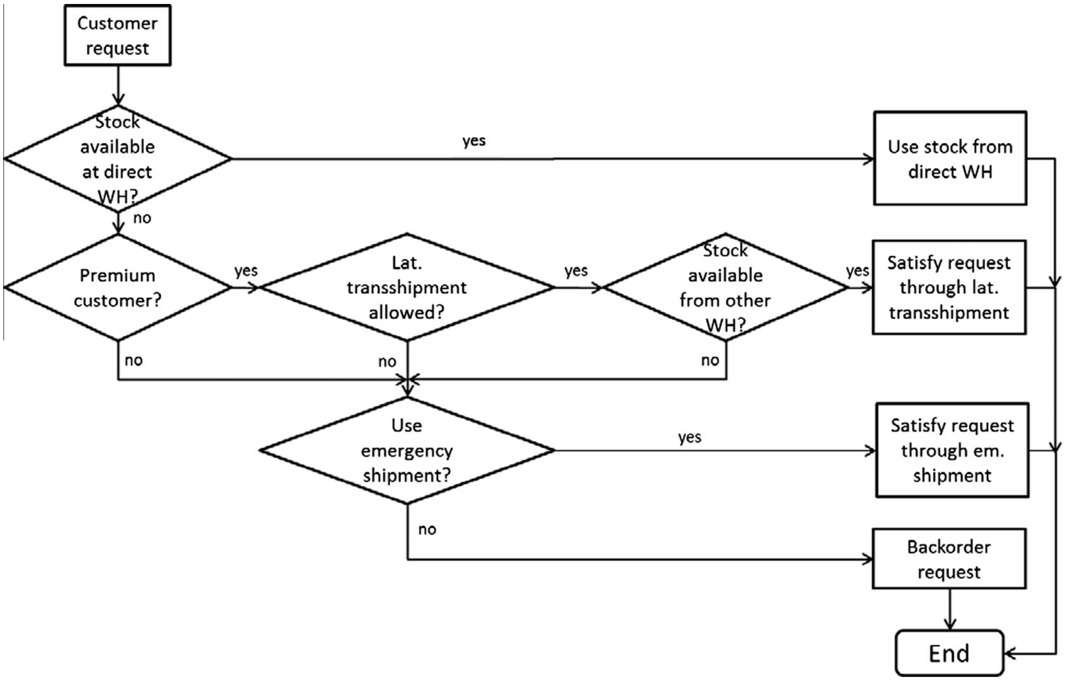
Customer request
Stock available at direct WH?
yes
no
Use stock from direct WH
Premium customer?
yes
no
Lat. transshipment allowed?
yes
no
Stock available from other WH?
yes
no
Satisfy request through lat. transshipment
Use emergency shipment?
yes
no
Satisfy request through em. shipment
Backorder request
End

**Fig. 1.** Process for handling an incoming customer request.

practice and depends on shipment times and costs between warehouses. On-hand stock is always used to satisfy a transshipment request, i.e., no stock is reserved for direct requests. In contrast, if transshipments of a specific item are *not* allowed at a warehouse, that warehouse can neither request the item at another warehouse nor receive transshipment requests. We consider the following three shipment strategies:

1. *Full backordering*: Premium and non-premium requests are backordered, with backorders cleared first-come-first-served. Premium backorders thus do not receive higher priority.
2. *Emergency shipments for premium customers and backordering for non-premium customers.*
3. *Emergency shipments for all customers.*

Note that non-premium demand may be met through emergency shipments, while cheaper lateral transshipments are not allowed. As previously mentioned, using lateral transshipments for non-premium customers depletes stock that could have been used for premium customers arriving a bit later. In contrast, the use of emergency shipments does not affect the ability of the system to meet premium requests arriving at a later moment in time, since the depot has infinite supply.

Our aim is to minimize the system holding and shipment costs, under restrictions on the mean aggregate waiting times per customer class *and* warehouse. A high waiting time at one warehouse thus cannot be compensated by a low waiting time at another warehouse, although such a variant (e.g. if a customer can be serviced from multiple warehouses) can be solved in a similar way.

### 3.2. Assumptions

1. All direct requests arrive according to mutually independent Poisson processes.
2. The replenishment lead time to any warehouse is exponentially distributed. This assumption facilitates system analysis using continuous-time Markov chains. The system performance measures also tend to be insensitive to the lead time distribution, especially when emergency shipments are used for both classes (e.g. Alfredsson & Verrijdt, 1999; Alvarez et al., 2013).
3. The lateral and emergency shipment times do not have a specific distribution: we only use the mean shipment times in our model.
4. Lateral transshipments are faster than emergency shipments and also have lower shipment costs. So, they are preferred over emergency shipments. Note that we allow to forbid certain combinations of items and warehouses to use lateral transshipments in advance by setting the related decision variable equal to zero (see the definition of $\boldsymbol{L_i}$ in Section 3.3). Also, warehouses can be excluded from lateral transshipments by creating clusters (online Appendix B).
5. Lateral and emergency shipments are sent directly to the customer and not via the warehouse.
6. Emergency shipment requests originate from the warehouse that needs the item: a second warehouse cannot request the item and then forward it to the warehouse who actually needs it.

### 3.3. Notation

In the notation below, we use index $k$ for the warehouse with ($k = 1, \dots, K$), $i$ for the item ($i = 1, \dots, I$) and $j$ for the customer class ($j = 1, 2$). The premium class is represented by class $j = 1$.

**Parameters**

| | |
|---|---|
| $\overline{W}_j$ | The maximum time a class $j$ customer is on average willing to wait |
| $m_{ijk}$ | The direct demand rate for item $i$ of class $j$ customers at warehouse $k$ |
| $M_{jk}$ | The total direct demand from class $j$ customers at warehouse $k$, $M_{jk} = \sum_{i=1}^{I} m_{ijk}$ |
| $M_k$ | The total direct demand at warehouse $k$, equal to $M_{1k} + M_{2k}$ |
| $R_{ik}$ | The mean replenishment lead time of item $i$ to warehouse $k$ |
| $E_{ik}$ | The emergency shipment time of item $i$ to warehouse $k$, with $E_{ik} \leqslant R_{ik}$ |
| $T_{ik}^{l}$ | The transshipment time of item $i$ from warehouse $l$ to $k$, with $T_{ik}^{l} \leqslant E_{ik}$ |
| $\boldsymbol{\sigma}_k$ | Vector representing the order in which warehouse $k$ may issue transshipment requests to other warehouses. $\boldsymbol{\sigma}_k = \{\sigma_k(1), \dots, \sigma_k(K-1)\}$, with $\sigma_k(n)$ the $n$-th warehouse receiving the request |
| $h_i$ | The unit holding costs per time unit for item $i$ |
| $C_{ik}$ | The *extra* costs over a regular replenishment for an emergency shipment of item $i$ to warehouse $k$ |
| $O_{ik}^{l}$ | The *extra* costs over a regular replenishment for a transshipment of item $i$ from warehouse $l$ to $k$. We assume that $O_{ik}^{l} \leqslant C_{ik}$ |

**Decision variables and performance measures**

| | |
|---|---|
| $\boldsymbol{S}_i$ | Vector of stock levels for item $i$, i.e., $\boldsymbol{S}_i = [S_{i1}, \dots, S_{iK}]$, with $S_{ik}$ the stock level at warehouse $k$ |
| $\boldsymbol{L}_i$ | Vector of transshipment strategies for item $i$, i.e., $\boldsymbol{L}_i = [L_{i1}, \dots, L_{iK}]$, with $L_{ik} = 1$ when transshipments are allowed to and from warehouse $k$ and $L_{ik} = 0$ otherwise |
| $\boldsymbol{D}_i$ | Vector of shipment strategies for item $i$, i.e., $\boldsymbol{D}_i = [D_{i1}, \dots, D_{iK}]$, with $D_{ik}$ denoting the highest customer class for which emergency shipments are used at warehouse $k$ |
| $\boldsymbol{b}_i$ | Shorthand notation for item policy ($\boldsymbol{S}_i$, $\boldsymbol{L}_i$, $\boldsymbol{D}_i$) |
| $EW_{ijk}(\boldsymbol{b}_i)$ | The expected class-$j$ waiting time for item $i$ at warehouse $k$ under policy $\boldsymbol{b}_i$ |
| $TC_{ik}(\boldsymbol{b}_i)$ | The total relevant costs for item $i$ at warehouse $k$ under policy $\boldsymbol{b}_i$ |
| $TC(\boldsymbol{b})$ | The total system costs summed over all items and warehouses under policy set $\boldsymbol{b} = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_I\}$ |

Regarding the shipment strategy, $D_{ik}$ can either be 0 (full backordering), 1 (backordering for non-premium customers), or 2 (emergency shipments for all customers).

### 3.4. Problem formulation

Our problem ($P1$) is to minimize the total system costs under restrictions on the mean aggregate waiting times per customer class *and* warehouse:

$$\min \quad \min TC(\boldsymbol{b}) = \sum_{i=1}^{I}\sum_{k=1}^{K} TC_{ik}(\boldsymbol{b}_i) \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{I} \frac{m_{ijk}}{M_{jk}} EW_{ijk}(\boldsymbol{b}) \leqslant \overline{W}_j, \qquad j = 1, 2, \quad k = 1, \dots, K, \tag{2}$$

$$S_{ik} \in N_0, \qquad L_{ik} \in \{0, 1\}, \quad D_{ik} \in \{0, 1, 2\}. \tag{3}$$

Eq. (3) refers to the parameters of the item policies $\boldsymbol{b}_i = (\boldsymbol{S}_i, \boldsymbol{L}_i, \boldsymbol{D}_i)$. To solve ($P1$), we use an approach similar to Dantzig–Wolfe decomposition. Specifically, we reformulate the non-linear problem ($P1$) as an linear integer programming problem. Also, by solving the LP-relaxation of the reformulated problem, we obtain a lower bound on the system costs. We can reformulate ($P1$) by specifying a finite set of *item policies*. The reformulated problem becomes to select one item policy from the set for each item such that the system costs are minimized with the waiting time requirements still being met. Let $\boldsymbol{B}_i$ be the set of item policies considered for item $i$. Let $x_{\boldsymbol{b}_i}$ be a binary variable indicating whether $\boldsymbol{b}_i$ is selected for item $i$ ($x_{\boldsymbol{b}_i} = 1$) or not. We then find:

$$(P2)\quad \text{Min} \quad \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{\boldsymbol{b}_i\in\boldsymbol{B}_i} TC_{ik}(\boldsymbol{b}_i)x_{\boldsymbol{b}_i} \tag{4}$$

$$\text{s.t.} \quad \sum_{i=1}^{I}\sum_{\boldsymbol{b}_i\in\boldsymbol{B}_i} \frac{m_{ijk}}{M_{jk}} EW_{ijk}(\boldsymbol{b}_i)x_{\boldsymbol{b}_i} \leqslant \overline{W}_j,$$

$$j = 1, 2, \quad k = 1, \ldots, K, \tag{5}$$

$$\sum_{\boldsymbol{b}_i\in\boldsymbol{B}_i} x_{\boldsymbol{b}_i} = 1, \qquad i = 1, \ldots, I, \tag{6}$$

$$x_{\boldsymbol{b}_i} \in \{0, 1\}, \qquad i = 1, \ldots, I, \quad \boldsymbol{b}_i \in \boldsymbol{B}_i. \tag{7}$$

We find the LP-relaxation of ($P2$) by replacing (7) by $0 \leqslant x_{\boldsymbol{b}_i} \leqslant 1$. If $\boldsymbol{B}_i$ contains all item policies, ($P2$) and ($P1$) are equivalent and have the same optimal solution, with the LP-relaxation solution of ($P2$) giving a lower bound for the optimal costs of ($P1$). Our challenge is the selection of item policies to include in $\boldsymbol{B}_i$. This challenge is twofold: first, we must be able to analyze the system for a given item policy in order to determine the related costs and waiting times. The key point here is the performance evaluation of a single-item building block with selective transshipments (see Section 4). Second, we must determine which policies are beneficial to include in set $\boldsymbol{B}_i$ for each item $i$ (see Section 5).

Both elements are far from trivial: transshipments create dependencies among warehouses, with the performance – and thus the optimal decisions – at any warehouse depending on the decision variable values at the other warehouses. As a result, all warehouses must be considered jointly in a solution procedure. Any exact techniques require a lot of computation time, especially for realistic instances. Therefore, we will apply a heuristic procedure in which we decompose the overall system into single location problems. Specifically, we can analyze a single warehouse when the decision variable values at other warehouses are fixed and the transshipment rate to the warehouse is known. Under those conditions, we can also calculate near-optimal values for the decision variables of that warehouse. Naturally, the analysis of a single warehouse – and the obtained decision variable values – influence the warehouse's transshipment rate, and hence the performance of other warehouses. Therefore, we require a procedure where warehouses are iteratively analyzed until convergence occurs. We discuss the single-item building block in Section 4 and the multi-item solution method in Section 5.

## 4. Performance analysis for a given item policy

We present an approximate performance analysis for a given item policy $\boldsymbol{b}_i$, with index $i$ omitted in this section as we consider a single item. We consider the special case where transshipments are allowed among all warehouses (i.e., $\boldsymbol{L} = [1, \ldots, 1]$). The analysis under alternative values for $\boldsymbol{L}$ is simple: if $L_k = 0$, warehouse $k$ can be analyzed individually, as it does not send or receive transshipments for item $i$. As mentioned before, an exact analysis (with continuous-time Markov chains) is intractable for realistic instances. We therefore use a heuristic decomposition in single warehouse models as follows. Consider a tagged warehouse. Given the fill rates at other warehouses and the transshipment policy, we know the rate at which transshipment requests from other warehouses arrive. As an approximation, we assume that these transshipment requests arrive according to a *Poisson process*. Then, we use standard models to approximate the fill rate at the tagged warehouse. Together with the demand rate, this fill rate defines the transshipment requests issued by the tagged warehouse, influencing the demand at the other warehouses. Assuming that all warehouses operate independently of each other, we can deploy an iterative procedure in which each warehouse is approximately analyzed given the fill rates at the other warehouses, until all fill rate estimates have converged. Such an approach has led to accurate results for related models (Axsäter, 1990; Alfredsson & Verrijdt, 1999; Van Wijk et al., 2012).

Section 4.1 gives further notation for computing $EW_{jk}(\boldsymbol{b})$ and $TC_k(\boldsymbol{b})$. Section 4.2 gives the main steps of our analysis, and Section 4.3 gives details on one of these steps, namely *Step 2*: warehouse analysis. Section 4.4 numerically validates the accuracy of the approach.

### 4.1. Additional notation for a single-item building block

We introduce notation that is specific for the single-item building block:

| | |
|---|---|
| $e_k^l$ | Rate at which warehouse $k$ receives transshipment requests from warehouse $l$ |
| $e_k$ | Rate at which transshipment requests arrive at warehouse $k$, i.e., $e_k = \sum_{l\mid k\in\boldsymbol{\sigma}_l} e_k^l$ |
| $\lambda_k$ | The *effective* demand rate at warehouse $k$ when that warehouse has stock on-hand, consisting of direct requests $m_{1k} + m_{2k}$ and transshipment requests $e_k$., i.e., $\lambda_k = m_{1k} + m_{2k} + e_k$ |
| $\beta_k(\boldsymbol{b})$ | The fill rate, i.e., the fraction of effective demand met from stock at warehouse $k$ under policy $\boldsymbol{b}$ |
| $\alpha_{jk}^l(\boldsymbol{b})$ | The fraction of direct demand of customer class $j$ at warehouse $k$ met through transshipments from warehouse $l$ under policy $\boldsymbol{b}$ |
| $\gamma_{jk}(\boldsymbol{b})$ | The fraction of direct demand of customer class $j$ at warehouse $k$ met through emergency shipments under policy $\boldsymbol{b}$ |
| $EBO_{jk}(\boldsymbol{b})$ | The mean backorder level for customer class $j$ at warehouse $k$ under policy $\boldsymbol{b}$ |

Using these performance measures, we find $EW_{jk}(\boldsymbol{b})$ and $TC_k(\boldsymbol{b})$ as follows:

$$EW_{jk}(\boldsymbol{b}) = EBO_{jk}(\boldsymbol{b})/m_{jk} + \gamma_{jk}(\boldsymbol{b})E_k + \sum_{l\in\boldsymbol{\sigma}_k} \alpha_{1k}^l(\boldsymbol{b})T_k^l, \tag{8}$$

$$TC_k(\boldsymbol{b}) = hS_k + \sum_{l\in\boldsymbol{\sigma}_k} \alpha_{1k}^l(\boldsymbol{b})m_{1k}O_k^l + \sum_{j=1}^{2} \gamma_{jk}(\boldsymbol{b})m_{jk}C_k. \tag{9}$$

The first term of $EW_{jk}(\boldsymbol{b})$ arises from backordering (using Little's formula), whereas the second and third term denote the waiting time arising from emergency and lateral transshipments. The holding costs in (9) are computed over both the on-hand stock and the items in the pipeline. All these parts have initially been procured, and so we incur holding costs over this total investment. This approach is common on literature, see e.g. Kranenburg and Van Houtum (2009).

### 4.2. Overview analysis procedure

Our main analysis steps are:

1. **Initialization**: $e_k = 0$, $k = 1..K$, so we initially ignore lateral transshipments.

2. **Warehouse analysis**: Compute fill rates $\beta_k(\boldsymbol{b})$ and the expected number of backorders $EBO_{jk}(\boldsymbol{b})$ for each warehouse $k$ and class $j$ given the current value of $e_k$. This step is discussed in detail in Section 4.3.
3. **Update of transshipment rates**: Update rates $e_k$ $\forall k$ given the current values of $\beta_k(\boldsymbol{b})$.
4. **Finish**: Stop if the change in $e_k$ is smaller than some small $\varepsilon$ $\forall k$. Otherwise, go to *step 2*.

In *Step 3*, we update $e_k^l$, and thus $e_k$, as follows: if $k = \sigma_l(n)$ for any integer $n$, $k$ receives transshipment requests from $l$ when $l$ and all warehouses $\sigma_l(1)$ up to $\sigma_l(n-1)$ are out of stock or do not allow lateral transshipments. Assuming independence among warehouses, we find:

$$e_k^l = m_{1l}(1-\beta_l(\boldsymbol{b}))\prod_{x=1}^{n-1}(1-\beta_{\sigma_l(x)}(\boldsymbol{b})). \tag{10}$$

We obtain $\alpha_{1k}^l$ by multiplying the fraction of premium demand at $k$ forwarded to $l$ (i.e., $e_l^k/m_{1k}$) by the probability that this demand can be met from on-hand stock at $l$ (i.e., $\beta_l(\boldsymbol{b})$). Eq. (12) below makes sure that all demand is met, either directly from stock or by transshipment or by emergency shipment. Note that (12) only applies if emergency shipments are used for the class.

$$\alpha_{1k}^l(\boldsymbol{b}) = \beta_l(\boldsymbol{b})e_l^k/m_{1k}, \tag{11}$$

$$\beta_k(\boldsymbol{b}) + \gamma_{jk}(\boldsymbol{b}) + \sum_{l\in\boldsymbol{\sigma}_k}\alpha_{jk}^l(\boldsymbol{b}) = 1 \quad j \leqslant D_k. \tag{12}$$

### 4.3. Step 2: Warehouse analysis

When warehouse $k$ has stock on hand, the effective demand rate is $\lambda_k$. When the warehouse is out of stock, we denote the demand rate under shipment strategy $D_k$ by $\theta_k(D_k)$, where:

- $\theta_k(2) = 0$: demand is not backordered, but met through lateral or emergency shipments.
- $\theta_k(1) = m_{2k}$: non-premium demand is backordered, premium demand is met through lateral or emergency shipments.
- $\theta_k(0) = \pi_{1k}m_{1k} + m_{2k}$: premium requests are backordered when the item cannot be obtained elsewhere in the system, which coincides with all warehouses in $\sigma_k$ being out of stock. Hence, the probability $\pi_{1k}$ of a premium backorder equals $\prod_{l\in\boldsymbol{\sigma}_k}(1-\beta_l(\boldsymbol{b}))$.

Let $\mu_k = 1/R_k$ be the regular shipment rate. Fig. 2 shows the Markov chain of the number of outstanding orders for a warehouse $k$.

When $D_k = 2$, the Markov chain simplifies to an Erlang loss system with $S_k$ servers. Using the notation $\rho_k = \lambda_k/\mu_k$, we then have (see amongst others Gross, Shortle, Thompson, & Harris, 2008):

$$\beta_k(\boldsymbol{b}) = 1 - \frac{\rho_k^{S_k}\big/S_k!}{\sum_{w=0}^{S_k}\rho_k^w/w!}. \tag{13}$$

When $D_k \leqslant 1$, we solve balance equations to find the steady-state probabilities $p_n$ of $n$ outstanding orders. With $\rho_k = \lambda_k/\mu_k$ and $\rho_{1k} = \theta_k(D_k)/\mu_k$, we get the following expressions:

$$p_0 = \left\{\sum_{w=0}^{S_k}\frac{\rho_k^w}{w!} + \left(\frac{\lambda_k}{\theta_k(D_k)}\right)^{S_k}\left(e^{\rho_{1k}} - \sum_{w=0}^{S_k}\frac{\rho_{1k}^w}{w!}\right)\right\}^{-1}, \tag{14}$$

**Table 2**
Parameter values for testing the analysis approach.

| Parameter | Values |
|---|---|
| $K$ | 6; 18 |
| $\left[R_k, T_k^l, E_k\right]$ | [8, 1, 2] |
| $f$ | 0.1; 0.2; 0.3; 0.5 |
| $M_k$ | 0.05; 0.5 |
| $S_k$ | 1; 2 ($M_k = 0.05$)<br>4; 8 ($M_k = 0.5$) |

$$p_n = \rho_k^{\min\{n,S_k\}}\rho_{1k}^{[n-S_k]^+}\frac{1}{n!}p_0, \tag{15}$$

$$\beta_k(\boldsymbol{b}) = \sum_{n=0}^{S_k-1}p_n, \tag{16}$$

$$\begin{aligned} EBO_k(\boldsymbol{b}) &= \sum_{n=S_k+1}^{\infty}(n-S_k)p_n \\ &= \left(\frac{\lambda_k}{\theta_k(D_k)}\right)^{S_k}p_0\left\{\rho_{1k}\left(e^{\rho_{1k}} - \sum_{n=0}^{S_k-1}\frac{\rho_{1k}^n}{n!}\right) - S_k\left(e^{\rho_{1k}} - \sum_{n=0}^{S_k}\frac{\rho_{1k}^n}{n!}\right)\right\}. \end{aligned} \tag{17}$$

When $D_k = 1$, $EBO_{1k}(\boldsymbol{b}) = 0$ and $EBO_{2k}(\boldsymbol{b})$ is given by (17). When $D_k = 0$, in contrast, the expected backorder level (17) should be disaggregated over premium and non-premium customers at rates $\pi_{1k}m_{1k}$ and $m_{2k}$ respectively, resulting in expressions (18) and (19):

$$EBO_{1k}(\boldsymbol{b}) = \frac{EBO_k(\boldsymbol{b})\pi_{1k}m_{1k}}{\pi_{1k}m_{1k}+m_{2k}}, \tag{18}$$

$$EBO_{2k}(\boldsymbol{b}) = \frac{EBO_k(\boldsymbol{b})m_{2k}}{\pi_{1k}m_{1k}+m_{2k}}. \tag{19}$$

### 4.4. Approximation accuracy

We compare our method to simulation on estimates for $\alpha_{1k}(\boldsymbol{b}) = \sum_{l\in\boldsymbol{\sigma}_k}\alpha_{1k}^l(\boldsymbol{b})$, $\beta_k(\boldsymbol{b})$ and $EW_{jk}(\boldsymbol{b})$, $(j = 1, 2)$. We test 32 problem instances with either 6 or 18 warehouses, where 18 warehouses depict a practical setting (see e.g. Section 7 in Kranenburg and Van Houtum (2009). Table 2 gives the remaining parameter values. In all instances, the shipment strategies are spread evenly over the warehouses, i.e., one third of all warehouses uses full backordering, one third uses emergency shipments for premium customers only, and one third uses emergency shipments for all customers. The demand rates and shipment times are the same at all warehouses, with a fraction $f$ of demand coming from premium customers.

For the simulation, we use a replication/deletion approach with at least 0.3 million requests for both premium and non-premium customers (average values are one million premium and five million non-premium requests). Table 3 shows that average accuracy is high for systems with 18 warehouses – which corresponds to practical instances – and slow movers. In systems with six warehouses and low stock levels (resulting in fill rates below 50%), the estimate of the transshipment fraction $\alpha_{1k}(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ can be poor. This situation, however, will almost never occur in practice. Fast movers contribute greatly to the overall waiting time. Therefore, waiting times for these items should be low, and stock

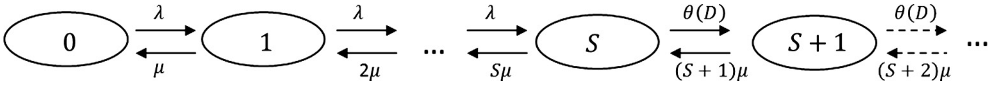

**Fig. 2.** Markov chain of the number of outstanding orders at warehouse under shipment strategy *D*.

**Table 3**
Relative errors of the analysis approach to simulation.

| Settings | | | Average relative error | | | | Maximum relative error | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $M_k$ | $K$ | $S_k$ | $\beta_k$ (%) | $\alpha_k$ (%) | $EW_{1k}$ (%) | $EW_{2k}$ (%) | $\beta_k$ (%) | $\alpha_k$ (%) | $EW_{1k}$ (%) | $EW_{2k}$ (%) |
| 0.05 | 6 | 1 | 0.1 | 1.1 | 1.4 | 0.1 | 0.2 | 2.5 | 5.4 | 0.5 |
| | | 2 | 0.0 | 0.2 | 0.2 | 0.1 | 0.0 | 0.7 | 0.7 | 0.5 |
| | 18 | 1 | 0.0 | 0.1 | 0.1 | 0.0 | 0.2 | 0.4 | 0.4 | 0.1 |
| | | 2 | 0.0 | 0.2 | 0.2 | 0.1 | 0.0 | 0.5 | 0.5 | 0.6 |
| 0.5 | 6 | 4 | 1.6 | 7.7 | 4.5 | 0.9 | 6.2 | 19.7 | 9.0 | 4.8 |
| | | 8 | 0.0 | 0.5 | 0.5 | 0.4 | 0.1 | 1.2 | 1.2 | 1.0 |
| | 18 | 4 | 0.3 | 0.7 | 1.1 | 0.2 | 1.9 | 2.9 | 4.3 | 1.2 |
| | | 8 | 0.0 | 0.5 | 0.5 | 0.6 | 0.1 | 1.3 | 1.3 | 1.4 |

levels will be (relatively) high. The maximum computation time for an instance is 12 milliseconds. Clearly, our approach is accurate and requires little computation time. As a result, it will be a suitable building block for solving multi-item problems in the next section. We refer to online Appendix A for details on the performance for each of the 32 instances.

## 5. Solving the multi-item problem

Our solution to the multi-item problem consists of two steps. First we solve the LP relaxation (Section 5.1). Second, we use this solution to find a near-optimal solution to the integer problem (P2) (Section 5.2).

### 5.1. Solving the LP-relaxation

To find suitable item policies to include in the LP-relaxation, we use a similar iterative technique to Alvarez et al. (2013) and Kranenburg and Van Houtum (2008). We first construct an initial set of item policies, for which we solve the LP-relaxation. Then, we use column generation to find new item policies that we may add to the policy set in order to improve the objective value of the LP-relaxation. As we will show, the attractiveness of an item policy depends on the policies already included in the LP-relaxation. Therefore, solving the LP relaxation with an extended policy set may reveal new attractive item policies to be added to the policy set. So, we end up with an iterative procedure in which we alternately solve the LP-relaxation and add item policies until no further interesting policies can be found. We first discuss the construction of the initial policy set, and then focus on finding new interesting policies using column generation.

#### 5.1.1. Constructing an initial policy set

An initial policy set should lead to a feasible solution to the *integer* problem (*P*2). One option to find such a set is to select a policy per item *i* such that $EW_{ijk}(\boldsymbol{b}_i) \leqslant \overline{W}_j$ for each class *j* and warehouse *k*, which guarantees $\sum_{i=1}^{I}\frac{m_{ijk}}{M_{jk}}EW_{ijk}(\boldsymbol{b}_i) \leqslant \overline{W}_j$. As that option may lead to relatively large stock levels, we instead look for a policy over all items simultaneously. We use a “biggest-bang-for-the-buck” algorithm, where we satisfy all unmet demand using emergency shipments, i.e., $L_{ik} = 0$ and $D_{ik} = 2$. This is justified since we only need a reasonable feasible solution as starting point for finding further interesting policies. In each step of our algorithm, we increase the stock level $S_{ik}$ by one unit at the item-warehouse combination $(i, k)$ that leads to the greatest added value. We continue until all waiting time restrictions are met. To choose an option $(i, k)$, we compute the decrease in waiting time relative to the extra investment needed. We find the decrease in waiting times for a unit stock increase at $(i, k)$ (denoted by $\boldsymbol{S}_i + U_{ik}$), as follows:

$$\Delta W(\boldsymbol{S}_i+U_{ik}) = \sum_{j=1}^{2}\sum_{k=1}^{K}\left\{\left(\sum_{i=1}^{I}\frac{m_{ijk}}{M_{jk}}EW_{ijk}(\boldsymbol{S}_i,\boldsymbol{L}_i,\boldsymbol{D}_i)-\overline{W}_j\right)^{+} - \left(\sum_{i=1}^{I}\frac{m_{ijk}}{M_{jk}}EW_{ijk}(\boldsymbol{S}_i+U_{ik},\boldsymbol{L}_i,\boldsymbol{D}_i)-\overline{W}_j\right)^{+}\right\}. \qquad (20)$$

Here $[a]^+ = \max\{0, a\}$, which ensures that we only consider waiting time reductions above their respective thresholds. The extra investment $\Delta TC(\boldsymbol{S}_i + U_{ik}) = TC(\boldsymbol{S}_i + U_{ik}) - TC(\boldsymbol{S}_i)$ follows from (9). Options $(i, k)$ may exist where both waiting times *and* costs decrease: a stock increase may lead to lower waiting times and fewer transshipments or emergency shipments (and hence lower shipment costs). Then, we select the option with the largest $\Delta W(\boldsymbol{S}_i + U_{ik})$ among those with lower costs (i.e., $\Delta TC(\boldsymbol{S}_i + U_{ik}) < 0$). Otherwise, we select the option with the largest $\Delta W(\boldsymbol{S}_i + U_{ik})/\Delta TC(\boldsymbol{S}_i + U_{ik})$. During the procedure, we remove all dominated policies that have both higher costs and higher waiting times at all warehouses than at least one other policy in the set. Note that the obtained policy set might contain more than one policy for each item: we expect a large initial policy set to limit the amount of time needed to generate extra policies through column generation.

#### 5.1.2. Finding additional policies through column generation

Column generation focuses on finding unconsidered item policies with negative reduced costs. Per item, we iteratively add the policy with minimal reduced costs to the policy set if these costs are negative. We stop once we cannot find further policies with negative reduced costs for any item. To compute an item's reduced costs, we require the shadow prices associated with solving the LP-relaxation for a given set of item policies. We now omit item index *i* for simplicity of notation, and let $u_{jk}$ ($\leqslant 0$) and $v$ ($\geqslant 0$) denote the shadow price values for constraints (3) and (4) respectively (with $v$ denoting the shadow price for the item being considered). The reduced costs $Z(\boldsymbol{b})$ for a policy $\boldsymbol{b}$ are now found as follows:

$$Z(\boldsymbol{b}) = \sum_{k=1}^{K}\left\{TC_k(\boldsymbol{b}) - \sum_{j=1}^{2}u_{jk}\frac{m_{jk}}{M_{jk}}EW_{jk}(\boldsymbol{b})\right\} - v. \qquad (21)$$

Since transshipments result in dependencies among warehouses, we can only find the policy with minimum reduced costs with certainty by setting the decision variable values over all warehouses jointly. Such an approach, however, requires too much time for problems of realistic size: instances of 4 warehouses and 10 items require nearly 3 days. Instead, we disaggregate the overall problem into single warehouse problems. Specifically, we can find the decision variable values at a warehouse *k* that minimize $Z(\boldsymbol{b})$, if the values of the variables at the other warehouses are given. Clearly, the choice of the decision variables at warehouse *k* will influence the performance at other warehouses. Therefore, we iteratively set the decision variable values at each warehouse separately until convergence occurs.

Fig. 3 shows the column generation steps for a single item. First, we construct a start (i.e., initial) item policy. This policy – specifically the decision variable values for warehouses $l > 1$ – serves as input for setting the decision variables at warehouse 1 a first time. Then, we iteratively set the decision variable values at another warehouse $k$, with the variable values at warehouses $n \neq k$ fixed to their most recent values. Each time we find a new item policy, we verify whether it has the lowest reduced costs so far and store it if this is the case. In an iteration, all warehouses in the system are considered. Convergence occurs when the decision variable values for all warehouses remain unchanged from one iteration to the next. We now give details on steps 1 and 2, with ($\boldsymbol{S}^*$, $\boldsymbol{L}^*$, $\boldsymbol{D}^*$) being the best item policy found.

***Step 1*: finding a start item policy for the column generation procedure.**

We can find a start policy in two extreme ways: either we allow transshipments at all warehouses (i.e., $L_k = 1\ \forall k$) or we do not allow them at any warehouse ($L_k = 0\ \forall k$). In the second option, the absence of transshipments allows us analyze each warehouse separately. Therefore, we easily find the values for $S_k$ and $D_k$ that minimize $Z(\boldsymbol{b})$. On the other hand, the first option will likely result in a more suitable start policy: we expect it to be easiest to move from a policy where transshipments are allowed at all warehouses to one where transshipments are only allowed at a subset of warehouses. In contrast, a move from a policy where transshipments are not used to one where transshipments are allowed can only occur if it is beneficial to transshipment among two or more warehouse (transshipments will not occur if they are only allowed at one warehouse).

These arguments prompt us to combine the options to find a start policy: first, we set $L_k = 0$ and determine values for $S_k$ and $D_k\ \forall k$. Then, we set $L_k = 1\ \forall k$ to obtain the start policy. In this way, we easily find values for $S_k$ and $D_k$, while still obtaining a start policy where transshipments are allowed among all warehouses. Note that the values found for $S_k$ and $D_k$ result in a valid item policy both when $L_k = 0$ and when $L_k = 1$. Therefore, we analyze the system under both settings and store the policy with the lowest reduced costs $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ as the best policy so far ($\boldsymbol{S}^*$, $\boldsymbol{L}^*$, $\boldsymbol{D}^*$).

Given that $L_k = 0$, we first set $S_k$ for each value of $D_k \in \{0, 1, 2\}$ separately. Subsequently, we select the combination ($S_k$, $D_k$) leading to the lowest value for $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$. Given a value for $D_k$, we start with $S_k = 0$. We then iteratively increase $S_k$ by one unit until a further increase has no benefit. Each time we increase $S_k$, we store the combination ($S_k$, $D_k$) if it leads to the lowest value for $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ so far (denoted by $Z^{min}(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$). A further increase of $S_k$ has no benefit once $h\left(\sum_{n=1}^{K} S_n + 1\right) - v \geqslant Z^{min}(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$. Then, the minimal reduced costs for $S_k + 1$ (consisting of the system holding costs minus the item shadow price) already exceed the best reduced costs found so far. Note that the actual costs for $S_k + 1$ will be larger than that minimum value, as we ignore the shipment and waiting time costs.

***Step 2*: setting decision variable values at warehouse *k*.**

We aim to find the values for $S_k$, $L_k$ and $D_k$ that minimize the reduced costs $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ in the entire system. We do so, because the decision variable values at warehouse $k$ influence the service levels at all warehouses. This influence can be significant: in particular, if stock is mainly (or even only) kept at warehouse $k$, the value of $L_k$ is crucial, since it influences whether other warehouses have access to this stock. First, we fix $L_k$ and determine what values for $S_k$ and $D_k$ minimize $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$. Then, we select the combination ($S_k$, $L_k$, $D_k$) with the smallest value for $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$. When $L_k = 0$, the values for $S_k$ and $D_k$ that minimize $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ are the same as those when looking for the start item policy (*step 1*), as the warehouse service levels are not influenced by transshipments. When $L_k = 1$, we find $S_k$ and $D_k$ using the approach given in *step 1*.

Given values for $S_k$, $L_k$ and $D_k$, we can estimate $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ either (i) by using the full analysis approach of Section 4 or (ii) by only analyzing warehouse $k$ (as in Section 4.3) and updating the estimates of $\alpha_{jl}^{k}(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ and $\gamma_{jl}(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ for the other warehouses $l$ in the system through Eqs. (11) and (12). The first option gives the most accurate estimate of $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$, but it is also very time-consuming. Furthermore, the second option leads to sufficiently good solutions, as we show in Section 5.1.3. Therefore, we use that latter option for our computational experiment.

Once we have found the values of $S_k$, $L_k$ and $D_k$ that minimize the rough estimate of $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$, we use the approach of Section 4 to determine the actual value of $Z(\boldsymbol{S}, \boldsymbol{L}, \boldsymbol{D})$ for to the newfound policy. Using this actual value, we determine whether the new policy is the best so far (i.e. *step 3*) and store it if this is the case.

### *5.1.3. Quality of the obtained lower bound*

Our column generation method does not necessarily find the item policy with the lowest reduced costs. Therefore, we cannot ensure that our solution to the LP-relaxation of (*P*2) is optimal. We therefore compare the lower bound found with our method to that when using a complete enumeration method. As the latter method is time-consuming, we test small problem instances. Note that complete enumeration might still not give an exact lower bound, the only reason being that we use an approximate approach for system analysis. We tested 192 problem instances, each with 5,

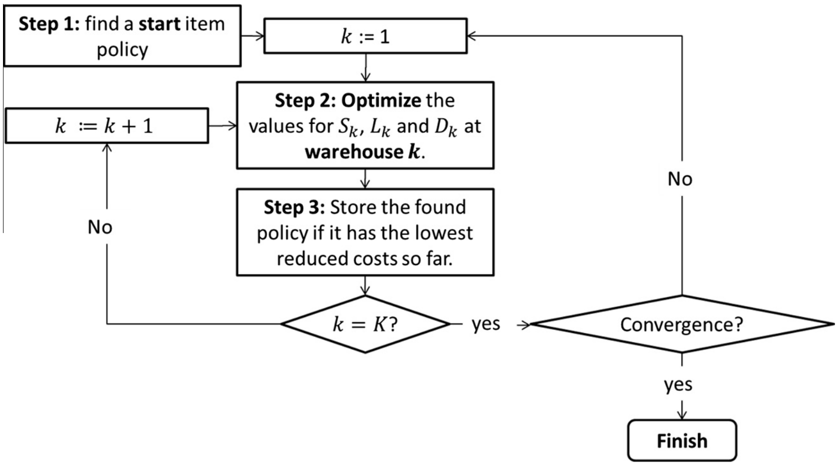
Step 1: find a start item policy
$k := 1$
Step 2: Optimize the values for $S_k$, $L_k$ and $D_k$ at warehouse $k$.
$k := k + 1$
Step 3: Store the found policy if it has the lowest reduced costs so far.
No
No
$k = K$?
yes
Convergence?
yes
Finish

**Fig. 3.** Column generation approach to find a near-optimal item policy for a particular item.

10 or 20 items, and 2 or 4 warehouses. We considered a single sample of demand rates and holding costs. The remaining parameter values have been marked by an asterisk in Table 5 (Section 6.1). Table 4 shows that our approach has a relative error of at most 2.29% to the enumeration solution. Also, the errors tend to decrease with the number of warehouses and items, with an error of at most 0.48% for instances with 20 items and 4 warehouses. Although instances with 6 or more warehouses require computation times of several days, we have tested 6 instances with 6 warehouses and 5 or 10 items, for which we found a maximum relative error of 0.01%.

### 5.2. Finding a near-optimal integer solution

In the solution to the LP-relaxation, $x_{\boldsymbol{b}_i}$ is not restricted to be integer, making it possible to select multiple item policies per item. Therefore, we need an approach to find a near-optimal solution to the integer problem ($P2$) proceeding from the solution of the LP relaxation. A simple option would be the intelligent rounding of the fractional values $x_{\boldsymbol{b}_i}$ of the LP-relaxation solution. However, such rounding will not be trivial, as we can have many items for which multiple policies are used: ($P2$) has $2K + I$ constraints, leading to $2K + I$ item policies $\boldsymbol{b}_i$ being basis variables (i.e., where $x_{\boldsymbol{b}_i} > 0$). For each item, at least one policy will be selected. We thus can have up to $2K$ items for which multiple policies are selected. Also, even if rounding is used to find a starting point for a local search procedure, the resulting solution is usually inferior to that obtained by solving the integer problem using software such as CPLEX (see Alvarez et al. (2013)). Therefore, we also solve ($P2$) using CPLEX.

The policy set used for solving the LP-relaxation serves as a starting point for the integer problem policy set, as this set has worked well before (e.g. Alvarez et al. (2013)). From the LP-relaxation set, we remove dominated policies (i.e., policies with both higher costs and waiting times than at least one other policy) and policies $\boldsymbol{b}_i$ where $\frac{m_{ijk}}{M_{jk}} EW_{ijk}(\boldsymbol{b}_i) > \overline{W}_j$ for at least one item $i$ and warehouse $k$ (the overall waiting time $\sum_{i=1}^{I} \sum_{r=1}^{|\boldsymbol{B}_i|} \frac{m_{ijk}}{M_{jk}} EW_{ijk}(\boldsymbol{b}_i) x_{\boldsymbol{b}_i}$ also exceeds $\overline{W}_j$ then). Unfortunately, computation times can still amount to several hours under this smaller set of policies. To decrease computation times, we consider two options, namely (i) further reducing the number of item policies or (ii) setting a time limit for CPLEX to find a solution. We choose option (ii) because computation times remained large under option (i), irrespective of the policy selection criterion (e.g., removing all policies whose reduced costs exceed a certain threshold). Also, the solution quality could be very poor (e.g. a gap to the lower bound of 14%). Option (ii) outperformed (i) both on solution quality and computation times. The reason is that CPLEX often finds a good solution in the first few minutes, with improvements being minor from then on. Most time is spent on evaluating options that turn out to be infeasible. In an experiment with 80 problem instances – with 20–50 items and 10–20 warehouses – we considered time limits from 15 to 60 minutes. We found a limit of 15 minutes to be effective, with an average gap to the lower bound of 0.85%. Further improvements in quality were negligible under larger time limits (e.g., under 60 minutes the average gap reduced to 0.84%).

**Table 4**
Relative error to the true lower bound.

| $I$ | $K$ | Relative error to LB (%) | |
|---|---|---|---|
| | | Average | Maximum |
| 5 | 2 | 0.24 | 2.23 |
| | 4 | 0.13 | 1.23 |
| 10 | 2 | 0.24 | 2.29 |
| | 4 | 0.06 | 0.46 |
| 20 | 2 | 0.26 | 1.36 |
| | 4 | 0.06 | 0.48 |

**Table 5**
Tested parameter values.

| | Parameter | Value |
|---|---|---|
| 1 | $I$ | 20, 50 |
| 2 | $K$ | 10, 20 |
| 3 | $R_{ik}$ (days) | 8*, 16* |
| 4 | $E_{ik}$ (days) | 2*, 4* |
| 5 | $[\overline{W}_1; \overline{W}_2]$ (hours) | [0.5; 2]*, [3; 24]* |
| 6 | $O_{ik}^{l}$ | 100*, 500 |
| 7 | Avg. $M_{ik}$ – interval (p. day) | [0.002; 0.05]*, [0.002; 0.5]* |
| 8 | Avg. $f$ | 0.2*, 0.5 |
| 9 | $h_i$ – interval (p. day) | [0.1; 10]*, [0.1; 50]* |

## 6. Computational experiment

In an extensive experiment, we investigate (i) the solution quality and computation time of our heuristic procedure, (ii) the added value of the selective transshipment approach, (iii) the suitability of the various shipment and transshipment strategies, and (iv) the impact of transshipment clusters, where transshipments are only allowed among subsets of warehouses.

### 6.1. Experiment design

We construct 1024 problem instances, with $T_{ik}^{l} = 1$ day and $C_{ik} = 1000$. Table 5 gives the other parameter values. The asterisks specify the values considered when evaluating the quality of our lower bound estimate (Section 5.1.3). Shipment times and costs are the same for all items and warehouses in a problem instance, with the lateral times and costs equal for any warehouse pair. Using a uniform distribution, the holding costs $h_i$ are randomly drawn on the specified interval. Below, we explain in detail how we obtain values for demand rates $m_{ijk}$.

Our demand rates $m_{ijk}$ should differ among warehouses *and* items, with the *overall* fraction of premium demand in the system equal to $f$. We find $m_{ijk}$ in three steps: first, (1) we draw a value on the $M_{ik}$ – interval (using a uniform distribution) to obtain the average demand rate for item $i$ at one warehouse. By multiplying this value by $K$ we find the total *system* demand rate $M_i$. Then, (2) we find the total premium demand in the system $M_i^p$ by multiplying $M_i$ by $f$, with $M_i^n$ denoting the remaining non-premium demand. Finally, (3) we disaggregate $M_i^p$ and $M_i^n$ over the warehouses to obtain $m_{ijk}$. Each warehouse is assigned a fraction of $M_i^p$ and $M_i^n$ (using a normal distribution), with normalization ensuring that $\sum_{k=1}^{K} m_{i1k} = M_i^p$ and $\sum_{k=1}^{K} m_{i2k} = M_i^n$.

Our parameter values are similar to those used by Kranenburg and Van Houtum (2008, 2009), as their values are based on practice. In particular, Kranenburg and Van Houtum (2008) serves as a basis for the demand rates and the parameters related to the customer classes (such as parameters 5 and 9), while Kranenburg and Van Houtum (2009) serves as input for the number of warehouses, and the various shipment times and cost elements. We consider items that have both high and low values, and high and low demand rates. The annual demand rates are between 0.7 units and 183 units. In practice, an item's annual holding cost is a fraction (about 25%) of its value. We thus consider item values between 146 and 73,000 euro's.

For simplicity, a warehouse $k$ sends transshipment requests to other warehouses in the same order in all problem instances:

**Table 6**
Solution quality and computation times of solution procedure.

| Parameter | Values | Gap to lower bound estimate (%) | | Computation time (minutes) | |
|---|---|---|---|---|---|
| | | Average | Maximum | Average | Maximum |
| $I$ | 20 | 1.3 | 5.5 | 7 | 21 |
| | 50 | 0.3 | 1.3 | 17 | 34 |
| $K$ | 10 | 0.6 | 2.9 | 7 | 16 |
| | 20 | 1.0 | 5.5 | 16 | 34 |
| Grand total | | 0.8 | 5.5 | 12 | 34 |

**Table 7**
Relative savings of ST_SES over SES.

| Parameter | Values | Average savings (%) | Maximum savings (%) |
|---|---|---|---|
| $E_{ik}$ | 2 | 12 | 28 |
| | 4 | 17 | 34 |
| $[\overline{W}_1; \overline{W}_2]$ | [0.5; 2] | 11 | 19 |
| | [3; 24] | 18 | 34 |
| Max. $M_{ik}$ | 0.05 | 19 | 34 |
| | 0.5 | 9 | 20 |
| Grand total | | 14 | 34 |

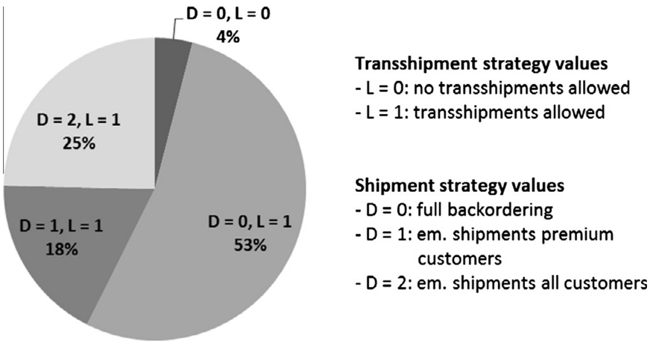

**Fig. 4.** The fraction of items and warehouses using a particular (trans-)shipment combination.

$\sigma_k = \{k+1, k+2, \ldots, K, 1, 2, \ldots\}$. So, if warehouse $k$ is out of stock, it first requests an item at warehouse $k+1$, then at warehouse $k+2$, etc.

For each combination of parameters in Table 5, we construct 2 sets of item demand rates and holding costs to ensure that our results are not dependent on the specific values of one sample. Combined with $2^9 = 512$ possible parameter combinations, we thus have 1024 instances in total.

### 6.2. Performance of the solution procedure

Table 6 shows the solution quality – expressed as a relative gap to the lower bound estimate – and computation times of the solution procedure. We used a Dell optiplex 760 with Intel quad core 2.83 gigahertz processor. Overall, the relative gap is 0.8% on average, with a maximum of 5.5%. The average and maximum gap decrease greatly as the number of items increases. We therefore expect the approach to work very well in realistic settings with many items. The instance computation time is 12 minutes on average and at most 34 minutes. Of these times, at most 15 minutes are used for solving the integer problem using CPLEX (see Section 5.2). The computation time mainly increases with the number of items and warehouses in an instance.

### 6.3. The added value of selective lateral transshipments

We estimate the added value of the selective transshipment model (ST_SES) in terms of relative cost savings over the selective emergency shipment model (SES), which is the special case of ST_SES with transshipments not allowed. Table 7 shows that ST_SES has significant savings over SES: 14% on average and 34% maximum. The savings are extensive if we have many slow movers; for fast movers, transshipments are not beneficial, as we show in Section 6.4. Savings are also large when emergency shipment times are large and waiting times are not very strict, although the influence of these parameters is mainly large in cases with expensive slow movers.

### 6.4. Suitability of shipment and transshipment strategies

For each combination $(L_{ik}, D_{ik})$, Fig. 4 shows the overall fraction of items and warehouses for which that combination is used. Clearly, lateral supply is very suitable for premium requests: overall, transshipments are used at 96% of all item-warehouse combinations. If lateral supply is not allowed, we always use full backordering. This is logical: if transshipments are not beneficial, more expensive (and slower) emergency shipments will not be beneficial either. We can thus limit the combinations $(L_{ik}, D_{ik})$ that we should consider. The instances where transshipments are not beneficial have inexpensive fast moving items, high transshipment costs and loose waiting time restrictions, making lateral supply expensive and unnecessary.

Overall, full backordering ($D = 0$) is the most common shipment strategy (see Fig. 4). This strategy is especially beneficial when emergency shipments are slow relative to regular supply, and when items are mostly cheap fast movers, as shown in Fig. 5. Then, that strategy is used for roughly 85% of all items and warehouses. This coincides with findings by Alvarez et al. (2013). Clearly, backordering should be considered in addition to emergency shipments, even though the latter option is commonly the only shipment mode considered both in literature and in business.

Fig. 6 shows for various problem instances how the strategies $(L_{ik}, D_{ik})$ are distributed over the items in each instance. We focus on instances with an $M_{ik}$ – interval of [0.002; 0.5] and a holding cost interval of [0.1; 50]; the results are similar for other parameter

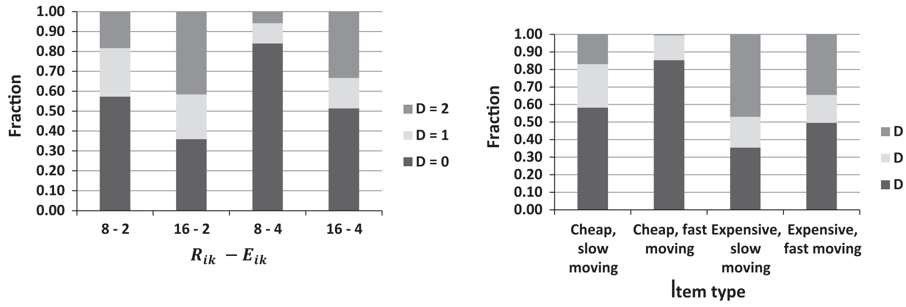

**Fig. 5.** The influence of shipment times (left) and item type (right) on the use of various shipment strategies.

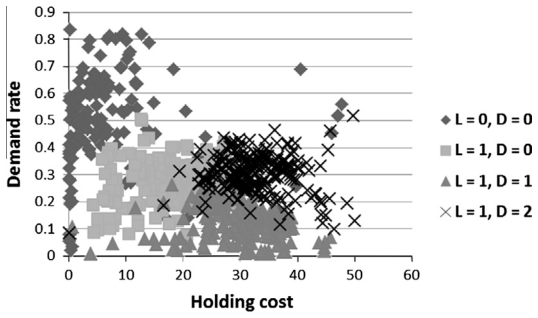

**Fig. 6.** Item characteristics per (trans-)shipment strategy.

values. As expected, neither lateral transshipments nor emergency shipments are used for inexpensive fast movers, with both transshipments and (partial) emergency shipments used for expensive slow movers.

## 7. Conclusions and further research

We first summarize our conclusions in Section 7.1, and then discuss directions for further research in Section 7.2.

### *7.1. Conclusions*

We considered a system with two customer classes where both lateral transshipments and emergency shipments are used for service differentiation purposes. For a single-item setting, we developed an analysis approach for the situation where selective transshipments may only be used for premium requests. We also developed a heuristic approach similar to Dantzig–Wolfe decomposition to set stock levels and (trans-)shipment strategies the multi-item system under class-specific waiting time restrictions, using the single-item building block. Key conclusions are:

- Our multi-item solution approach gives near-optimal solutions in little computation time.
- Selective lateral transshipments lead to significant cost savings when combined with selective emergency shipments. The savings are 14% on average and can amount to 34%. The savings can be particularly large (19% on average) if we have many expensive slow movers.
- Backordering should also be considered as a shipment option in spare parts settings. This is in contrast to the practice of always using emergency shipments for unmet demand.

Furthermore, we find it is not necessary to use transshipments among all warehouses: we may find comparable savings by allowing transshipments among a subset of warehouses. See online Appendix B for details.

### *7.2. Further research*

The selective transshipments model can be extended in various ways. First, we can consider more than two customer classes. Transshipment and emergency shipments are then used for a subset of customers (where the subset may vary per item). The analysis approach for such a system follows directly from that the approach described in Section 4. However, we obtain additional decision variables (i.e., for what customer classes do we allow transshipments and emergency shipments) and extra constraints. Further research is thus needed to carefully select relevant item policies for the solution approach.

Second, we may allow transshipments for non-premium customers if neighboring warehouses have plenty of stock, with extra decision variables specifying the threshold value per warehouse from which such shipments may occur (similar to holdback levels in Van Wijk et al. (2012)). Such an extension requires adjustments to the analysis of a single warehouse. Also, the additional decision variables result in a large set of item policies to consider, making it even more difficult to solve the multi-item problem. It is also uncertain whether such an extension will lead to large cost savings: for expensive slow mowers little stock is kept, so transshipments generally will not be allowed for non-premium requests. Conversely, for inexpensive fast movers it will be too expensive to even use transshipments for premium requests.

Finally, we may extend the transshipment model with a critical level policy, where some stock at each warehouse is reserved for premium requests (either direct or transshipment requests). In the simplest case, we may combine a positive critical level at a warehouse with emergency shipments for both classes (i.e., $D_{ik} = 2$ when the critical level $C_{ik} > 0$). Alvarez, Van der Heijden, Vliegen, and Zijm (2012) consider this extension and show that this combination has similar savings to the selective transshipment model (averages of 15% and 14% respectively). Combinations of positive critical levels and (partial) backordering are more complicated, because we cannot adjust the analysis model in a straightforward manner: we require a two-dimensional state space to analyze a warehouse, as we may have on-hand stock and class 2 backorders simultaneously.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ejor.2014.02.053.

## References

Al Hanbali, A., & Van der Heijden, M. C. (2013). Interval availability analysis of a two echelon, multi-item system. *European Journal of Operational Research, 228*, 494–503.

Alfredsson, P., & Verrijdt, J. (1999). Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science, 45*, 1416–1431.

Alvarez, E. M., Van der Heijden, M. C., & Zijm, W. H. M. (2013b). Service differentiation in spare parts supply through dedicated stocks. *Annals of Operations Research* (in press, doi:10.1007/s10479-013-1362-z).

Alvarez, E. M., Van der Heijden, M. C., Vliegen, I. M. H., & Zijm, W. H. M. (2012). Service differentiation through selective lateral transshipments. *Beta Working Paper Series, 395*. http://beta.ieis.tue.nl/node/2050.

Alvarez, E. M., Van der Heijden, M. C., & Zijm, W. H. M. (2013). The selective use of emergency shipments for service-contract differentiation. *International Journal of Production Economics, 143*(2), 518–526.

Axsäter, S. (1990). Modelling emergency lateral transshipments in inventory systems. *Management Science, 36*, 1329–1338.

Cohen, M., Agrawal, N., & Agrawal, V. (2006). Winning in the aftermarket. *Harvard Business Review, 84*, 129–138.

Dada, M. (1992). A two-echelon inventory system with priority shipments. *Management Science, 38*, 1140–1153.

Eppen, G., & Schrage, L. (1981). Centralized ordering policies in a multi-warehouse system with lead times and random demand. In L. B. Schwarz, (Ed.), *Multi-level production/inventory control systems: Theory and practice* (pp. 51–67). North-Holland.

Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). *Fundamentals of queuing theory* (4th ed.). Hoboken, New Jersey: John Wiley & Sons.

Jalil, M. N. (2011). *Customer information driven after sales service management: lessons from spare parts logistics*. PhD thesis, Erasmus Research Institute of Management (ERIM). <http://repub.eur.nl/res/pub/22156/>.

Kranenburg, A. A., & Van Houtum, G. J. (2008). Service differentiation in spare parts inventory management. *Journal of the Operational Research Society, 59*, 946–955.

Kranenburg, A., & Van Houtum, G. J. (2009). A new partial pooling structure for spare parts networks. *European Journal of Operational Research, 199*, 908–921.

Lee, H. L. (1987). A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science, 33*, 1302–1316.

Paterson, C., Kiesmüller, G., Teunter, R., & Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research, 210*, 125–136.

Sherbrooke, C. C. (2004). *Optimal inventory modeling of systems* (2nd ed.). Kluwer Academic Publishers.

Teunter, R. H., & Klein Haneveld, W. K. (2008). Dynamic inventory rationing strategies for inventory systems with two demand classes, Poisson demand and backordering. *European Journal of Operational Research, 190*, 156–178.

Tiemessen, H. G. H., Fleischmann, M., Van Houtum, G. J., Van Nunen, J. A. E. E., & Pratsini, E. (2012). Dynamic demand fulfillment in spare parts networks with multiple customer classes. *Beta Working Paper Series, 377*. http://beta.ieis.tue.nl/node/2002.

Van Wijk, A. C. C., Adan, I. J. B. F., & Van Houtum, G. J. (2012). Approximate evaluation of multi-location inventory models with lateral transshipments and hold back levels. *European Journal of Operational Research, 218*, 624–635.

Veinott, A. F. (1965). Optimal policy in a dynamic, single product, nonstationary inventory model with several demand classes. *Operations Research, 13*, 761–778.