

Reducing costs of spare parts supply systems via static priorities

Citation for published version (APA):

Adan, I. J. B. F., Sleptchenko, A. V., & Houtum, van, G. J. J. A. N. (2006). *Reducing costs of spare parts supply systems via static priorities*. (BETA publicatie : working papers; Vol. 185). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2006

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Reducing Costs of Spare Parts Supply Systems via Static Priorities

I.J.B.F. Adan

Department of Mathematics and Computer Science, Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

A. Sleptchenko

Econometric Institute, School of Economics, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

G.J. van Houtum*

Department of Technology Management, Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

October 2006

Abstract

We study static repair priorities in a system consisting of one repair shop and one stockpoint, where spare parts of multiple repairables are kept on stock to serve an installed base of technical systems. Demands for ready-for-use parts occur according to Poisson processes, and are accompanied by returns of failed parts. The demands are met from stock if possible, and otherwise they are backordered and fulfilled as soon as possible. Returned failed parts are immediately sent into repair. The repairables are assigned to static priority classes. The repair shop is modelled as a single-server queue, where the failed parts are served according to these priority classes. We show that under a given assignment of repairables to priority classes, optimal circulation stock levels follow from Newsboy-type equations. Next, we develop fast and effective heuristics for the assignment of repairables to priority classes. Subsequently, we compare the performance of the system under these static priorities to the case with a First-Come First-Served (FCFS) service discipline. We show that in many cases static priorities reduce total inventory holding and back-ordering costs by more than 40%. Finally, we analyse the effect of the number of priority classes. We show that 2 priority classes suffice to obtain 90% of the maximal savings via static priorities.

Keywords: inventory/production control, spare parts, static priorities, system-focussed performance measures.

* *Corresponding author.* Address: see above. Phone: +31 40 2475163. Email: g.j.v.houtum@tm.tue.nl.

1 Introduction

In this paper, we study the provisioning of spare parts for advanced technical equipment, such as military equipment, airplanes, large computer systems, medical equipment, baggage handling systems, and so on. Such equipment is often part of primary processes of their users, and down-times of such systems may halt large parts of these primary processes. Therefore, in many industries, high fractions of up-time are required by users, and professional service organizations are responsible for realizing those up-times. Such a service organization may be a separate department within the same company as where the systems are used (e.g., in military organizations and companies that are large enough to organize this support by themselves), a department of the Original Equipment Manufacturer (this is common in the high-tech industry), or a third party. Resources are people for call handling and remote service, service engineers, spare parts, and service tools. For high-tech systems, in general a large part (30% or more, say) of the total support costs consists of spare parts costs. This is for spare parts usage, for having spare parts on stock in locations at close distance of the installed systems, for repair of spare parts, and for transportation of the parts by fast and slower transportation modes.

The inventory control for spare parts may have a large effect on total spare parts costs. What has been studied extensively is the so-called system approach, in which the inventory control is directly focussed on availability of systems instead of target service levels for individual Stock Keeping Units (SKU-s). This has been studied in single-location and multi-echelon settings, and it has been shown that, in comparison to a straightforward item approach, the system approach may lead to large reductions (typically 20-50%) in inventory holding costs for spare parts; see Sherbrooke (2004), Thonemann et al. (2002), and Rustenburg et al. (2003). In these comparisons, leadtimes for procurement of new parts or repair of repairable parts are considered as given. Obviously, optimal total costs decrease when these leadtimes are decreased, but reducing leadtimes for all SKU-s may also require additional investments.

The goal of this paper is to study the effect of *static repair priorities* on the total system performance. We study a spare parts supply system consisting of one repair shop, one stockpoint, and multiple repairable SKU-s. Ready-for-use parts are kept on stock in the stockpoint to serve an installed base of technical systems. When a part of one of the technical systems fails, the failed part is immediately sent to the repair shop and at the same time a ready-for-use part is requested at the stockpoint. Such a request is fulfilled immediately if there is a part of the requested SKU on stock and otherwise the request is backordered and fulfilled as soon as possible. Each backordered request corresponds to a technical system that is down. The objective of our model is to minimize the average total inventory holding costs of spare parts and costs for down-time of technical systems

over an infinite horizon. This is similar to the system-focussed objective functions that have led to the system approach as discussed above. In our model, we use the FCFS discipline as the standard rule for the repair of failed parts, as FCFS reflects the common way of working in practice. By using static priorities instead of FCFS, we can decrease repair leadtimes for expensive SKU-s (by assigning them to high priority classes), which may lead to strong reduction of their circulation stocks. At the same time, repair leadtimes for cheaper SKU-s increase (if they are assigned to lower priority classes), the effect of which are higher circulation stocks. However, because these SKU-s are cheaper the latter effect may be much smaller in terms of money than the effect for the expensive SKU-s. Notice that static priorities do not require additional investments in repair capacity.

In our model, the optimization variables are the circulation stocks and the assignment of SKU-s to priority classes. To obtain representative data for the various input parameters, we collected data at the Royal Netherlands Navy. They work with specialized groups of repairman that execute preventive and corrective maintenance and repair failed parts for subsystems of technical systems such as radar systems, goalkeepers, and diesel engines. We collected data at 4 such groups. Those data will be used in our numerical experiments. With respect to the numbers of SKU-s that are repaired by one group, we observed that they varied from 13 to 99 among those 4 groups. To be able to handle those numbers, and also because priority systems are hard to analyze in general, we are forced to make some simplifying assumptions. We model the repair shop as a single exponential server and we assume that all SKU-s have exponential repair times with the same mean. Under these assumptions, we are able to optimize systems with 2 priority classes and 15 SKU-s in an exact way. Based on those instances, we test 4 different heuristic optimization algorithms. The best of these heuristics has an optimality gap of 1.1%, and this heuristic is used for larger instances with up to 5 priority classes and 50 SKU-s to compare system performance under static priorities and FCFS. This shows that static priorities may easily lead to total cost reductions of 40% or more. We also show how these savings depend on input parameters. Although these results are obtained under the simplifying assumptions, they suggest that, also under more realistic circumstances, use of static priorities may lead to substantial cost savings. Notice that the simplifying assumptions in the present paper are common in the field of capacitated production/inventory systems where capacities are modelled as queues; see e.g. Buzacott and Shantikumar (1993).

Our work contributes to a rich literature on spare parts inventory models. One stream of work started with the seminal paper of Sherbrooke (1968) on the METRIC model. In this stream ample repair capacities are assumed, the models are focussed on optimal control for multiple items in multi-echelon systems, targets are typically set in terms of system availability, and they allow

the assumption of multi-indenture structures for the technical systems. The first models in this stream were inspired by military applications. For an overview of this stream, see the references in Sherbrooke (2004) and Rustenburg et al. (2003).

The assumption of ample repair capacities in the first stream facilitates the analysis and enables that systems with many SKU-s can be optimized. But, the assumption of ample capacity is not always justified. It can lead to a poor estimation of system performance and a poor allocation of stocks in systems with highly utilized repair shops and no flexibility options to control leadtimes (cf. Van Harten and Sleptchenko, 2003, and Sleptchenko et al., 2002). Therefore, in a second stream of literature, various ways to model finite repair capacities have been studied. Most papers in this stream are based on queueing type models with exponential servers; see Gross et al. (1983), Albright and Gupta (1993), Gupta and Albright (1992), Avsar and Zijm (2002), Zijm and Avsar (2003). Other interesting contributions in this stream have been made by Aboud (1996), Diaz and Fu (1997), and Perlman et al. (2001). In much of this work, the focus is on the development of approximate evaluation algorithms, and if optimization is applied, this is generally limited to systems with limited numbers of SKU-s.

In a third stream, the effect of repair priorities has been studied. This stream consists of a few papers only, which seems to be due to the fact that priority systems are hard to analyze in general. The use of emergency repairs in case stock levels are lower than some critical level has been studied by Verrijdt et al. (1998), while Perlman et al. (2001) considered emergency repairs under predefined probabilities. Both models assume that the normal and emergency repair facilities are separated. Hausman and Scudder (1982) and Pyke (1990) studied repair priorities in systems with limited numbers of SKU-s via simulation. These studies show that static priority rules are outperformed by dynamic rules, and that priority rules may lead to significant cost savings, especially under high workloads at the repair facility. As stated above, optimization of the repair priorities via simulation models is time consuming and practically impossible for systems with many SKU-s. A model similar to the one studied in the present paper was developed by Sleptchenko et al, 2005a. They considered a two-echelon, two-indenture system with multi-server, two-priority repair shops. Based on an approximate evaluation of a multi-class, multi-server queue with two priority classes (cf. Sleptchenko et al., 2005b, and Van der Heijden et al., 2004), they developed heuristics for the optimization of circulation stocks and the assignment of repair priorities, and they showed that static priorities may lead to significant cost reductions in comparison to FCFS. As indicated, their analysis was limited to two priority classes. Moreover, due to complexity of their model, only systems with a limited number of SKU-s (up to 9 SKU-s that share the same repair capacity, and up to 28 SKU-s in total) could be analyzed.

An interesting recent contribution has been made by Caggiano et al. (2006), who developed an integrated model for real-time capacity and inventory allocation in a two-echelon repairable spare parts system with one central repair shop (see also Muckstadt, 2005). That model is a finite-horizon, periodic-review, mathematical programming model and is appropriate to support operational decisions for repairable spare parts in the exploitation phase of technical systems. Their model implicitly uses dynamic priorities for their repair shop, as SKU-s with a low actual on-hand stock of ready-for-use parts receive priority over SKU-s with high actual on-hand stocks. Our model may be seen as a complementary model to their model, as it generates insights for the initial supply of repairable spare parts at the beginning of the exploitation phase, i.e., at the moment that new technical systems are installed.

Our work fits in the third stream of research, and our main contribution consists of three parts. First, we formulate a clean model that allows exact evaluation, and we develop efficient and effective heuristics for the optimization of circulation stocks and assignment of static repair priorities. In fact, we show that under a given assignment of repair priorities, optimal circulation stocks follow from Newsboy-type equations, and next the repair priorities are optimized heuristically. For our most effective heuristic, which consists of enumeration among so-called ordered priority assignments followed by local search, an optimality gap of 1.1% is found in a test bed of 108 instances with 2 priority classes and 15 SKU-s. Second, via this effective heuristic and a test bed of 1296 instances with up to 5 priority classes and 50 SKU-s, we show that by the use of static repair priorities total costs are reduced by 40% or more in the majority of the instances, and we give insights into the parameters by which these savings are mainly determined. Third, we show that 2 priority classes suffice to obtain 90% of the maximal savings via static priorities. This result is important from a practical point of view, because priority systems with 2 classes will be much easier to implement than systems with more than 2 classes. To the best of our knowledge, this insight has not been established before in the literature.

The organization of this paper is as follows. In Section 2, we formulate our model. Next, in Section 3, we show how circulation stocks are optimized under a given assignment of repair priorities, and we formulate heuristics for the optimization of the repair priorities. In Section 4, the heuristics are tested, and we study the savings obtained by repair priorities. After that, in Section 5, we apply our heuristics to stylized cases based on data from the Royal Netherlands Navy. Finally, we conclude in Section 6.

2 Model

In this section we first describe the spare part supply system and formulate our model. Then, we show how the objective function is evaluated for given basestock levels and a given priority assignment.

2.1 System description

We consider a single location with one repair shop and one stockpoint, where spare parts of multiple repairables are kept on stock to serve an installed base of technical systems (see Figure 1). We distinguish N ($\in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$) repairables or SKU-s, which are numbered $1, \dots, N$. These SKU-s occur in the configurations of the technical systems and are subject to failures. When a part fails in one of the technical systems, immediately a demand is placed for a ready-for-use part of the same SKU at the stockpoint and the failed part is sent to the repair shop. If the requested part is on stock, then the demand is immediately fulfilled. Otherwise the demand is backordered and fulfilled as soon as a ready-for-use part of the requested SKU becomes available. We assume that, for the total installed base, failures of an SKU occur according to a Poisson process with a constant rate. This assumption is justified if the installed base is sufficiently large, or if the installed base is not that large but down-times of systems are relatively short (e.g., because of a high service level at the stockpoint). The total failure rate for SKU n is given by λ_n (> 0).

We assume that all parts can be repaired. This implies that the inventory positions of all SKU-s are kept at a constant level, or, in other words, that there is a constant circulation stock for each SKU. The constant level for the inventory position of SKU n is denoted by S_n ($\in \mathbb{N}_0$); we refer to these levels as basestock levels.

The repair shop is modelled as a single exponential server, which repairs the failed parts of all SKU-s. All repair times are exponentially distributed and mutually independent. For all SKU-s, we assume the same service rate μ . To obtain a stable system, we assume $\mu > \sum_{n=1}^N \lambda_n$. The SKU-s are divided over M ($\in \mathbb{N}$) priority classes. These classes are numbered $1, \dots, M$. The smaller the class number, the higher its priority. Further, we assume preemption, i.e., the repair of a failed part is interrupted as soon as a failed part of a higher priority arrives at the repair shop. The assignment of SKU-s to priority classes is described by a matrix \mathbf{X} of size $N \times M$. The (n, m) -th element of this matrix is denoted by $x_{n,m}$ and

$$x_{n,m} = \begin{cases} 1 & \text{if SKU } n \text{ is assigned to priority class } m; \\ 0 & \text{otherwise} \end{cases}$$

We also use the notation m_n to denote the class m to which SKU n has been assigned; i.e., m_n is

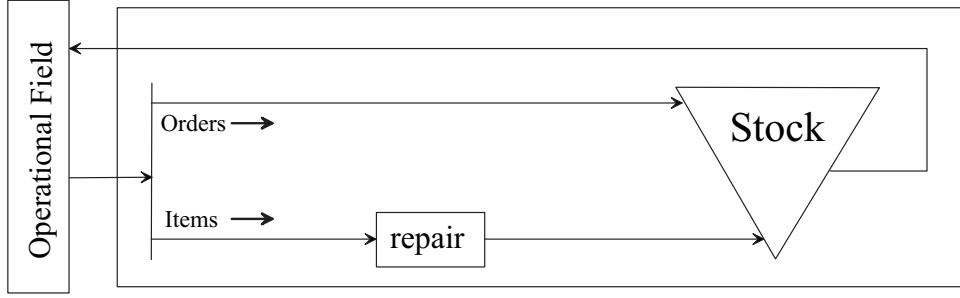


Figure 1: *The spare parts supply system and its interaction with the installed base.*

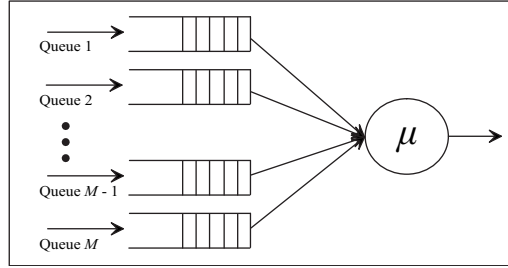


Figure 2: *The repair shop: A priority queueing system.*

the unique class index m for which $x_{n,m} = 1$. The repair shop is visualized in Figure 2, in which queue m corresponds to priority class m .

Under the above assumptions, the control of the repair shop is independent of actual on-hand stocks and of the basestock levels S_n . The repair shop has a steady-state behavior that is equivalent to the behavior of a priority queueing system. In general, for these priority systems, it is hard to derive steady-state distributions for the numbers of jobs of the various types. Our assumptions with respect to the repair shop and the repairs of failed parts are chosen such that an efficient evaluation of these steady-state distributions is possible. Obviously, these assumptions do not reflect reality, but they do capture the essential feature that different SKU-s compete for the same repair capacity.

For the costs, we distinguish inventory holding costs and backordering costs. For the circulation stock of SKU n , we pay inventory holding costs $h_n (> 0)$ per time unit per part; i.e., the inventory holding costs for SKU n are equal to $h_n S_n$. We assume that different backordered demands correspond to different technical systems. This is realistic for the same situations as needed for the Poisson failure processes with a constant rate. We pay a penalty cost b for each backordered demand per time unit, which is equivalent to paying $b (> 0)$ per time unit per technical system that is down because of a lack of spare parts. This type of penalty costs is thus system-oriented. Let $EBO_n(S_n, \mathbf{X})$ denote the mean number of backordered demands of SKU n in steady state; this

mean only depends on the steady-state number of failed parts of SKU n in the repair shop, which is determined by \mathbf{X} , and the basestock level S_n . Then the average backordering costs for SKU n are equal to $bEBO_n(S_n, \mathbf{X})$.

Our objective is to choose the basestock levels S_n and the priority assignment \mathbf{X} such that the total average costs are minimized. The mathematical formulation of this optimization problem is as follows:

$$(P) \quad \begin{cases} \min_{S_n, \mathbf{X}} \sum_{n=1}^N [h_n S_n + bEBO_n(S_n, \mathbf{X})] \\ \text{s.t.} \quad \sum_{m=1}^M x_n^m = 1, \quad n = 1, \dots, N, \\ x_n^m \in \{0, 1\}, \quad n = 1, \dots, N, \quad m = 1, \dots, M, \\ S_n \in \mathbb{N}_0, \quad n = 1, \dots, N. \end{cases}$$

This is a non-linear integer optimization problem. An underlying problem for this optimization problem is the evaluation of the $EBO_n(S_n, \mathbf{X})$ for given basestock levels S_n and a given priority assignment \mathbf{X} . This is discussed in the next subsection.

2.2 Evaluation of the expected numbers of backorders

To obtain the mean numbers of backordered parts $EBO_n(S_n, \mathbf{X})$ per SKU, we first determine the steady-state distributions for the numbers of parts per priority class in the repair shop and for the numbers of parts per SKU.

Let p_j^m be the probability that the number of parts in priority class $m = 1, \dots, M$ equals $j \in \mathbb{N}_0$ at an arbitrary moment. The corresponding mean number of parts of priority class m is denoted by L^m . Because all service times are identically distributed and our priority rule is preemptive, from the perspective of priority class m , all classes with a higher priority may be seen as one aggregated class (notice that there are no higher classes for $m = 1$) and the classes with lower priority can be ignored. For priority class 1, there are no higher classes, and its behavior follows from the classical $M/M/1$ queueing system. The probabilities p_j^1 , $j \in \mathbb{N}_0$, are equal to

$$p_j^1 = (1 - \rho_1)\rho_1^j, \tag{1}$$

where

$$\rho_1 = \frac{1}{\mu} \sum_{n=1}^N x_{n,1} \lambda_n.$$

Further, $L^1 = \rho_1/(1 - \rho_1)$.

For the lower priority groups ($m > 1$), we obtain a two-queue preemptive priority model as solved by Miller (1981), and the expressions for their probabilities p_j^m can be found as follows. We

first derive probabilities g_i^m , $i \in \mathbb{N}_0$; g_i^m represents the probability that the queue of priority class m increases with i parts while the total amount of parts of higher priority decreases with 1. Denoting the total workload of priority class m as $\rho_m = \frac{1}{\mu} \sum_{n=1}^N x_{n,m} \lambda_n$ and the total workload of higher priority classes as $\rho_m^h = \frac{1}{\mu} \sum_{n=1}^N \sum_{l=1}^{m-1} x_{n,l} \lambda_n$, we find (following the formulas of Sleptchenko et al., 2004, in which systems with two or more priority classes have been solved):

$$g_0^m = \frac{(1 + \rho_m^h + \rho_m) - \sqrt{(1 + \rho_m^h + \rho_m)^2 - 4\rho_m^h}}{2\rho_m^h}, \quad (2)$$

$$g_{i+1}^m = \frac{\rho_m g_i^m + \rho_m^h \sum_{j=1}^i g_j^m g_{i+1-j}^m}{(1 + \rho_m^h + \rho_m) - 2\rho_m^h g_0^m}, \quad i \in \mathbb{N}_0, \quad (3)$$

where, by convention, $\sum_{j=1}^i g_j^m g_{i+1-j}^m = 0$ for $i = 0$. Next, the probabilities p_j^m are obtained by the following recursive formulas:

$$p_0^m = (1 - \rho_m^h - \rho_m) + \frac{\rho_m^h}{\rho_m} (1 - \rho_m^h - \rho_m) (1 - g_0^m), \quad (4)$$

$$p_j^m = \rho_m p_{j-1}^m + \rho_m^h \sum_{i=0}^{j-1} \left[p_{j-1-i}^m \left(1 - \sum_{v=0}^i g_v^m \right) \right] + \frac{\rho_m^h}{\rho_m} (1 - \rho_m^h - \rho_m) \left(1 - \sum_{v=0}^j g_v^m \right), \quad j \in \mathbb{N}, \quad (5)$$

where $p_{-1}^m := 0$. For L^m , it holds that

$$L^m = \frac{\rho_m}{(1 - \rho_m^h)(1 - (\rho_m^h + \rho_m))}.$$

In these formulas, both ρ_m^h and ρ_m are assumed to be positive. If $\rho_m^h = 0$, then there are no higher priority jobs and thus the steady-state probabilities and L^m are as in an M/M/1 system (cf. (1)). If $\rho_m^h > 0$ and $\rho_m = 0$, then it simply holds that $p_0^m = 1$ and $p_j^m = 0$ for all $j > 0$ (which also follows from (4)-(5) when taking $\rho_m \downarrow 0$), and $L^m = 0$.

Next, we determine the distributions for the numbers of parts in the repair shop per SKU. Define $P_j^n(\mathbf{X})$ as the probability that j parts of SKU n are present in the repair shop; we now explicitly denote that these probabilities depend on \mathbf{X} because they are used in later sections. Recall that m_n denotes the class to which SKU n has been assigned. For priority class m_n , the total stream of arriving parts is a Poisson process that is constituted by Poisson arrival streams of all SKU-s l that are assigned to class m_n (i.e., for which $x_{l,m_n} = 1$). The intensity of the total stream is $\sum_{l=1}^N x_{l,m_n} \lambda_l$. Because all parts arrive according to Poisson streams, each part of class m_n in the repair shop has a probability $\lambda_n / (\sum_{l=1}^N x_{l,m_n} \lambda_l)$ that the part is of SKU n and the part is of one of the other SKU-s in class m_n otherwise. Based on this property, we find that

$$P_j^n(\mathbf{X}) = \left[\sum_{k=j}^{\infty} \binom{k}{j} \left(\frac{\lambda_n}{\sum_{l=1}^N x_{l,m_n} \lambda_l} \right)^j \left(1 - \frac{\lambda_n}{\sum_{l=1}^N x_{l,m_n} \lambda_l} \right)^{k-j} p_k^{m_n} \right], \quad j \in \mathbb{N}_0. \quad (6)$$

For each SKU n , the number of backorders is equal to the number of parts in the repair shop minus the basestock level S_n if this leads to a positive number and the number of backorders equals 0 otherwise. Hence,

$$\begin{aligned} EBO_n(S_n, \mathbf{X}) &= \sum_{j=S_n+1}^{\infty} (j - S_n) P_j^n(\mathbf{X}) \\ &= \left(\frac{\lambda_n}{\sum_{l=1}^N x_{l,m_n} \lambda_l} \right) L^{m_n} - S_n + \sum_{j=0}^{S_n} (S_n - j) P_j^n(\mathbf{X}), \quad S_n \in \mathbb{N}_0. \end{aligned} \quad (7)$$

This completes the description of how to determine all $EBO_n(S_n, \mathbf{X})$ for given basestock levels and a given priority assignment. They are obtained via the (recursive) formulas (1)-(7).

3 Analysis of Problem (P)

In this section, we describe optimization methods for Problem (P). First, in Subsection 3.1, we reduce Problem (P) to a pure priority assignment problem by deriving how the basestock levels are optimized under a given priority assignment. Next, in Subsection 3.2, we formulate various algorithms for the optimization of the priority assignment. The first of these algorithms is an enumeration algorithm and is exact. The other four algorithms are heuristics.

3.1 Optimization of the basestock levels

Suppose that a priority assignment \mathbf{X} is given. Then Problem (P) becomes a multi-dimensional optimization problem for the basestock levels S_n only:

$$(P(\mathbf{X})) \quad \begin{cases} \min_{S_n} \sum_{n=1}^N [h_n S_n + b EBO_n(S_n, \mathbf{X})] \\ \text{s.t. } S_n \in \mathbb{N}_0, \quad n = 1, \dots, N. \end{cases}$$

This problem decomposes into one-dimensional optimization problems ($P_n(\mathbf{X})$) per SKU n :

$$(P_n(\mathbf{X})) \quad \min_{S_n \in \mathbb{N}_0} [h_n S_n + b EBO_n(S_n, \mathbf{X})], \quad n = 1, \dots, N.$$

Below we show that these problems can be solved along the same lines as a Newsboy problem.

We denote the objective function of Problem ($P_n(\mathbf{X})$) by $f_n(S_n) = h_n S_n + b EBO_n(S_n, \mathbf{X})$, $S_n \in \mathbb{N}_0$. By (7), we find that the first order difference function $\Delta f_n(S_n) := f_n(S_n + 1) - f_n(S_n)$ is equal to

$$\begin{aligned} \Delta f_n(S_n) &= h_n + b \left[-(S_n + 1) + S_n + \sum_{j=0}^{S_n+1} (S_n + 1 - j) P_j^n(\mathbf{X}) - \sum_{j=0}^{S_n} (S_n - j) P_j^n(\mathbf{X}) \right] \\ &= -(b - h_n) + b \sum_{j=0}^{S_n} P_j^n(\mathbf{X}), \quad S_n \in \mathbb{N}_0. \end{aligned}$$

This shows that $\Delta f_n(S_n)$ is increasing on its whole domain, and thus $f_n(S_n)$ is convex. Hence, $f_n(S_n)$ is minimized at the first point where $\Delta f_n(S_n) \geq 0$, i.e., at the smallest $S_n \in \mathbb{N}_0$ for which

$$\sum_{j=0}^{S_n} P_j^n(\mathbf{X}) \geq \frac{b - h_n}{b}. \quad (8)$$

This optimal point is denoted by $S_n^*(\mathbf{X})$. The ratio $\frac{b-h_n}{b}$ is the Newsboy ratio for our problem; this ratio is different from the common ratio $\frac{b}{b+h}$ because in our problem also holding costs are paid for parts in repair. The sum $\sum_{j=0}^{S_n} P_j^n(\mathbf{X})$ is the fraction of time that there is no backlog for SKU n (also denoted as the α -service level, cf. Van Houtum and Zijm, 2000). Notice that $S_n^*(\mathbf{X}) = 0$ if $h_n \geq b$, and $S_n^*(\mathbf{X}) \rightarrow \infty$ if $h_n \downarrow 0$.

By substitution of the optimal basestock levels $S_n^*(\mathbf{X})$ for the S_n in Problem (P), Problem (P) reduces to the following optimization problem for the priority assignments \mathbf{X} :

$$(P') \quad \begin{cases} \min_{\mathbf{X}} \sum_{n=1}^N [h_n S_n^*(\mathbf{X}) + b EBO_n(S_n^*(\mathbf{X}), \mathbf{X})] \\ \text{s.t.} \quad \sum_{m=1}^M x_{n,m} = 1, \quad n = 1, \dots, N, \\ \quad \quad x_{n,m} \in \{0, 1\}, \quad n = 1, \dots, N, \quad m = 1, \dots, M. \end{cases}$$

3.2 Optimization of the priority assignment

Exact optimization of the assignment of the repair priorities is obtained by checking all possible priority assignments, i.e., by *total enumeration*. Each evaluation of a given assignment requires that the distributions $P_j^n(\mathbf{X})$, the optimal basestock levels $S_n^*(\mathbf{X})$, and expected numbers of backorders $EBO_n(S_n^*(\mathbf{X}), \mathbf{X})$ are determined via the formulas of Subsections 2.2 and 3.1. Since we have N SKU-s that are assigned to M priority classes, the total solution space consists of M^N different solutions (some solutions in the solution space are equivalent to others, but that is only a small fraction). This implies that the order of the computation for enumeration is M^N , which is exponential as a function of N . Hence, enumeration will quickly result into too large computation times, and it makes sense to develop heuristics with polynomial computation times, so that also instances with many SKU-s can be solved. Below, we develop these heuristics, and the enumeration algorithm will be used to study the optimality gap of these heuristics. We refer to the enumeration algorithm as *Algorithm 1*.

Many studies on optimal priority assignments in production systems (cf. Buzacott and Shan-thikumar, 1993) suggest that SKU-s with higher costs should have higher priority than the ones with lower costs. This seems also logical for our problem, as higher priorities for expensive SKU-s are expected to lead to lower basestock levels and thus significantly lower inventory holding costs.

At the same time, lower priorities for cheap SKU-s will lead to higher basestock levels for them, but that will have only a limited effect on the inventory holding costs. This leads to the formulation of the first heuristic, called *Algorithm 2*. From now on, w.l.o.g., we assume that the SKU-s are ordered such that $h_1 \geq h_2 \geq \dots \geq h_N$. Next, we define an *ordered assignment* as an assignment under which $m_1 \leq m_2 \leq \dots \leq m_N$; i.e., for each pair of SKU-s n and \tilde{n} with $n < \tilde{n}$ it holds that SKU n is assigned to the same or a higher priority class than SKU \tilde{n} . Algorithm 2 evaluates all ordered assignments and selects the best ordered assignment. Each ordered assignment is found by placing $M - 1$ gates in a row consisting of all SKU-s $1, \dots, N$; the SKU-s left of the first gate belong to class 1, the SKU-s between the first and second gate belong to class 2, and so on. Hence, the number of ordered assignments is equal to

$$\binom{N + M - 1}{M - 1} = \frac{(N + M - 1) \cdot (N + M - 2) \cdot \dots \cdot (N + 1)}{(M - 1)!}.$$

As a result, under a given M , Algorithm 2 is polynomial in N and the order of the computation time is $O(N^{M-1})$.

A faster variant of Algorithm 2 is *Algorithm 3*, which is a local search algorithm within the class of all ordered assignments. Initially, we assign all SKU-s to priority class 1. In each iteration step, we consider neighbors obtained by moving the SKU with the highest index in a non-empty class $m < M$ to class $m + 1$. Hence, each iteration requires $\leq M - 1$ evaluations of neighbors. The number of iterations is bounded from above by $(M - 1)N$ as the number of times to move a given SKU to the next priority class is at most $M - 1$. Therefore, the order of computation time for Algorithm 3 is $(M - 1)^2 N$. This order is linear in N for any given M , while Algorithm 2 has only a linear computation time for $M = 2$.

The following example shows that the optimal priority assignment also depends on the workloads of the SKU-s, to some extent, and that therefore Algorithms 2 and 3 may lead to suboptimal solutions.

Example 1. Consider a system with $N = 2$ SKU-s, $M = 2$ priority classes, and:

- $\lambda_1 = 0.75, \lambda_2 = 0.15, \mu = 1$
- $h_1 = 0.51, h_2 = 0.49, b = 1$.

In this case 4 different solutions exist, which we denote as $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4$. For these solutions the optimal basestock levels and costs are given in Table 1. Notice that under \mathbf{X}_3 and \mathbf{X}_4 , both SKU-s are in the same priority class; these solutions are equivalent and they imply that the repair shop follows a FCFS discipline.

| | | | | |
|--|---|---|---|---|
| Priority assignm. | $\mathbf{X}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ | $\mathbf{X}_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ | $\mathbf{X}_3 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ | $\mathbf{X}_4 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$ |
| $(S_1^*(\mathbf{X}_i), S_2^*(\mathbf{X}_i))$ | (2,3) | (6,0) | (5,1) | (5,1) |
| Total costs | 8.22 | 7.91 | 7.95 | 7.95 |

Table 1: Numerical results for Example 1.

The results in Table 1 show that priority assignment \mathbf{X}_2 is the unique optimal solution, but this solution is not an ordered assignment. Hence, Algorithms 2 and 3 would lead to suboptimal solutions in this case (Algorithm 2 would produce \mathbf{X}_3 or \mathbf{X}_4 as solution and Algorithm 3 would lead to \mathbf{X}_3). Apparently, in this example, it is optimal to place the SKU with the lowest workload (= lowest arrival intensity) in the highest priority class instead of the most expensive SKU. As we know from the theory of priority queues that has little effect on the mean repair leadtime for the SKU with the high workload, while it has a strong reducing effect on the mean repair leadtime for the SKU with the low workload. \square

Based on the above insights we define improved versions of Algorithms 2 and 3. To both algorithms, we add a local search, in which we also allow non-ordered assignments. In this local search, we distinguish two types of neighbors:

- We allow that an SKU n is moved from its current priority class m_n to class $m_n - 1$ (only if $m_n > 1$) or to class $m_n + 1$ (only if $m_n < M$);
- We allow that two SKU-s n and \tilde{n} with priority classes m_n and $m_{\tilde{n}} > m_n$ are swapped if m_n and $m_{\tilde{n}}$ are so-called neighboring classes, i.e., if $m_{\tilde{n}} = m_n + 1$, or if $m_{\tilde{n}} > m_n + 1$ and all priority classes m , $m_n < m < m_{\tilde{n}}$, are empty.

Algorithm 4 consists of Algorithm 2 followed by this local search procedure. Similarly, *Algorithm 5* consists of Algorithm 3 followed by this local search procedure. In this local search procedure the number of neighbors can be shown to be bounded by $N + \frac{1}{4}N^2$ (N for the first type of neighbors and $\frac{1}{4}N^2$ for the second type of neighbors). However, there is no tight bound for the number of iterations. Hence, for the orders of computation time for Algorithms 4 and 5, we only know that they are equal to or larger than the orders of computation time of the Algorithms 2 and 3.

When comparing the Algorithms 2-5 to each other with respect to accuracy, we know that Algorithm 2 dominates Algorithm 3. I.e., in all instances, the solution generated by Algorithm 2 will be at least equally good as the solution generated by Algorithm 3. Further, Algorithm 4 dominates Algorithms 2 and 3, and Algorithm 5 dominates Algorithm 3.

4 Numerical experiments

In this section, we define a large test bed in Subsection 4.1. Next, in Subsection 4.2, we present results for the optimality gap of the Algorithms 2-5 for instances with 15 SKU-s and 2 priority classes. In Subsection 4.3, we compare the performance of the Algorithms 2-5 for larger instances with up to 50 SKU-s and 5 priority classes. Finally, in Subsection 4.4, we investigate the cost savings that are obtained by using static priorities instead of the FCFS service discipline.

4.1 Test bed

We use a factorial design for our test bed. The number of SKU-s is chosen equal to 15, 25, and 50, and we have 2, 3, 4, and 5 priority classes. Next, regarding the holding cost, we fix the highest holding cost (h^{\max}) at 1000. The lowest holding cost (h^{\min}) is taken equal to 1, 10, and 100. Hence the ratio of highest and lowest inventory holding cost varies from 10 to 1000, which are common ratios for spare parts (see also Section 5). The dependence between the inventory holding cost parameters and demands rates for the SKU-s is another factor taken into account. We test three different variants for this dependence. In all three variants, we choose the demand rates λ_n for all SKU-s as independent samples from a uniform distribution on $[1,100]$. Then, in the first variant, the holding cost parameters and demand rates are chosen independently. In this case, the holding cost parameters are picked as independent samples from a uniform distribution on $[h^{\min}, h^{\max}]$ (see Figure 3A). In the second variant, we assume a hyperbolic relation between demand rates and holding cost parameters (see Figure 3B), as this reflects what one typically sees in practice. The holding cost parameters h_n are obtained by the following function:

$$h_n = \max \left\{ h^{\min}, a \frac{1}{c\lambda_n + d} + b + \xi_n \right\}. \quad (9)$$

In this function, c and d are such that $0.1 \leq c\lambda + d \leq 1$ for all $\lambda \in [1, 100]$, a and b are such that $a \frac{1}{c+d} + b = h^{\min}$ and $a \frac{1}{100c+d} + b = h^{\max}$, and $\xi_n \in U[-v, v]$ with $v = 0.025(h^{\max} - h^{\min})$. In the third variant, we assume the same relation as in the second variant, but we include some SKU-s that are picked differently. For most of the SKU-s ($\frac{2}{3}N$), we assume the hyperbolic dependence as in the second variant (cf. (9)), for some of the SKU-s ($\frac{2}{9}N$) we take low demand rates and low inventory holding cost parameters, and for some SKU-s ($\frac{1}{9}N$) we take high demand rates and high inventory holding cost parameters (see Figure 3C). Further, the mean repair time is chosen such that the utilization rate of the repair shop is equal to 0.7, 0.82, 0.9, and 0.95, respectively. Finally, for the penalty cost parameter b , we choose the values 1000, 10000, and 100000. In Table 2, we summarize our choices for all parameters.

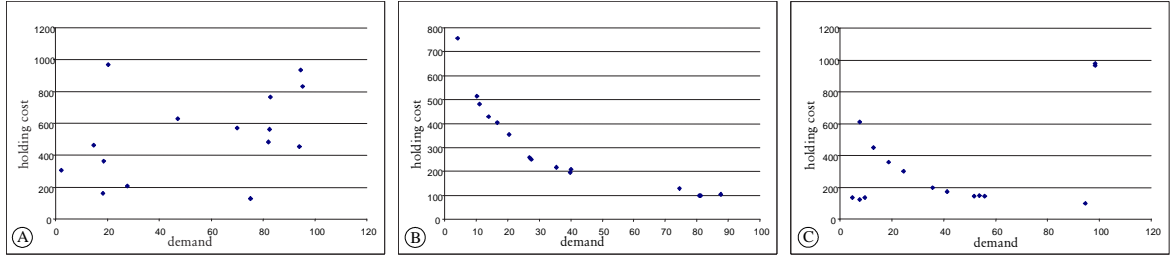


Figure 3: Relations between holding costs and demands relations.

| Name of parameter | No. of values | Values |
|---|---------------|---|
| No. of SKU-s (N) | 3 | 15, 25, 50 |
| No. of priority classes (M) | 4 | 2, 3, 4, 5 |
| Lowest holding cost (h^{\min}) | 3 | 1, 10, 100 |
| Highest holding cost (h^{\max}) | 1 | 1000 |
| Demand rates (λ_n) Holding costs (h_n) | 3 | $\left\{ \begin{array}{l} 1) \lambda_n \in U[1, 100], h_n \in U[h^{\min}, h^{\max}], \\ 2) \lambda_n \in U[1, 100], h_n = h(\lambda_n) \text{ (see (9))}, \\ 3) \left\{ \begin{array}{ll} \lambda_n \in U[1, 100], & h_n = h(\lambda_n), & 1 \leq n \leq \frac{2}{3}N \\ \lambda_n \in U[1, 10], & h_n \in U[h^{\min}, h^{\min} + v], & \frac{2}{3}N < n \leq \frac{8}{9}N \\ \lambda_n \in U[90, 100], & h_n \in U[h^{\max} - v, h^{\max}], & \frac{8}{9}N < n \leq N \end{array} \right. \end{array} \right.$ |
| Utilization rate (ρ) | 4 | 0.7, 0.82, 0.9, 0.95 |
| Penalty cost (b) | 3 | 1000, 10000, 100000 |

Table 2: Parameter choices for the test bed.

The total number of all possible combinations for these parameters is $3 \times 4 \times 3 \times 1 \times 3 \times 4 \times 3 = 1296$. In addition, for each of the three variants for the λ_n and h_n we generated 5 sets of values as there are uniform distributions involved in the generation of these values (notice that optimal solutions may be very sensitive for small changes in input parameters as we are dealing with an integer optimization problem). This gives us in total $1296 \times 5 = 6480$ instances.

4.2 Optimality gap of the heuristic algorithms

Algorithms 2-5 generate heuristic solutions, and their quality may be tested by comparison of their costs to the optimal costs. For sufficiently small values of M and N , the optimal costs may be obtained by Algorithm 1. We applied all 5 algorithms to all instances with $N = 15$ SKU-s and $M = 2$ priority classes. Per instance we computed the optimality gap as the relative distance of the costs of the heuristic solutions to the optimal costs. The averages over different subsets of instances for these optimality gaps are listed in Table 3. E.g., in part (a) of this table, the average optimality gaps for all instances with $h^{\min} = 1, 10, \text{ and } 100$ are given in the second, third, and fourth column,

| (a) Aver. opt. gaps as a function of h^{\min} | | | | (b) Aver. opt. gaps as a function of λ_n / h_n | | | |
|---|-------|-------|-------|--|---------|---------|---------|
| | 1 | 10 | 100 | | Case 1. | Case 2. | Case 3. |
| Algorithm 2 | 5.1% | 5.4% | 5.9% | Algorithm 2 | 5.6% | 5.0% | 5.8% |
| Algorithm 3 | 23.3% | 24.7% | 19.7% | Algorithm 3 | 13.8% | 26.7% | 27.2% |
| Algorithm 4 | 1.1% | 1.3% | 0.9% | Algorithm 4 | 1.0% | 1.5% | 0.8% |
| Algorithm 5 | 9.7% | 9.4% | 7.9% | Algorithm 5 | 5.0% | 11.6% | 10.4% |

| (c) Aver. opt. gaps as a function of ρ | | | | | (d) Aver. opt. gaps as a function of b | | | |
|---|-------|-------|-------|------|--|------|-------|--------|
| | 0.7 | 0.82 | 0.9 | 0.95 | | 1000 | 10000 | 100000 |
| Algorithm 2 | 10.8% | 5.7% | 3.6% | 1.8% | Algorithm 2 | 0.8% | 4.7% | 10.9% |
| Algorithm 3 | 35.8% | 27.8% | 18.8% | 7.9% | Algorithm 3 | 5.9% | 27.8% | 34.0% |
| Algorithm 4 | 1.8% | 1.0% | 1.1% | 0.5% | Algorithm 4 | 0.0% | 0.3% | 3.0% |
| Algorithm 5 | 9.5% | 13.6% | 9.8% | 3.1% | Algorithm 5 | 1.4% | 10.5% | 15.1% |

| (e) Aver. opt. gaps over all instances | |
|--|-------|
| Algorithm 2 | 5.5% |
| Algorithm 3 | 22.6% |
| Algorithm 4 | 1.1% |
| Algorithm 5 | 9.0% |

Table 3: Average optimality gaps (for the test bed limited to $N = 15$ and $M = 2$).

respectively. This shows how the optimality gap depends on the choice for h^{\min} . In the last part of the table, part (e), the averages over all instances are given.

The results in part (e) show that Algorithm 4 has the best performance. This algorithm performs very well with an average optimality gap of 1.1%. Algorithms 2 and 5 perform reasonably, and the performance of Algorithm 3 is relatively bad. When we look into further detail in the parts (a)-(d), we observe that the gap for Algorithm 4 is not so sensitive for the choice of the lowest inventory holding cost h^{\min} . It is somewhat sensitive for the relation that is assumed between the demand rates and inventory holding cost parameters. Further, the gap decreases significantly for increasing workloads ρ , and increases significantly for increasing values of the penalty cost parameter b . For the Algorithms 2, 3, and 5, we see similar behavior in the parts (a), (c), and (d), while the picture in part (b) is more mixed.

In Figure 4, the distribution of the optimality gaps over all instances is depicted for the Algorithms 2-5. In this figure, we see e.g. that Algorithm 5 had an optimality gap of 0% (i.e., an optimal solution was found) in 55% of all instances. Algorithm 4 had an optimality gap of 0% in 80% of all instances, and an optimality gap of at most 7% in 95% of all instances. Also from this figure, we clearly see that Algorithm 4 is the best one. The average computation times per instance for the Algorithms 2-5 were equal to 0.15, 0.07, 1.47, and 1.43 seconds, respectively. So,

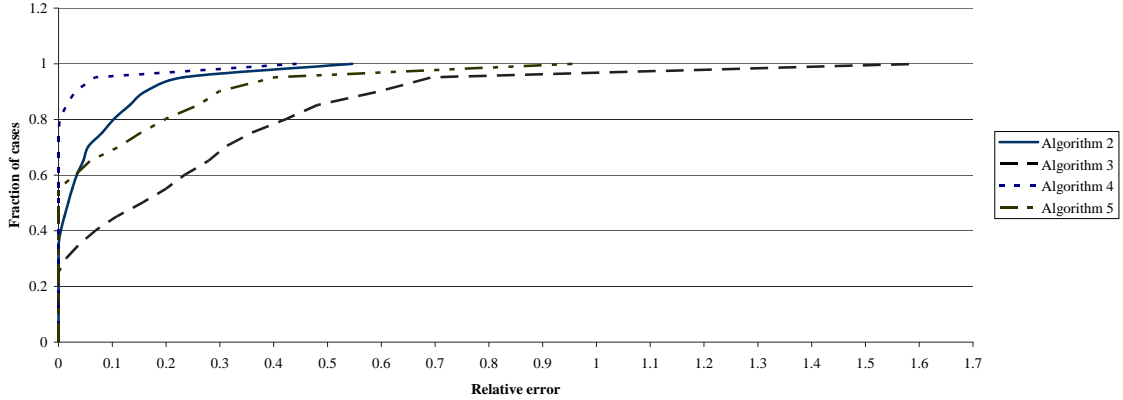


Figure 4: *Statistics for the optimality gaps (for the test bed limited to $N = 15$ and $M = 2$).*

these times were small for all instances.

4.3 Comparison of the heuristics for large instances

In this subsection, we compare the performance of the Algorithms 2-5 for the full test bed. This time we cannot compare to Algorithm 1, since enumeration requires too much computation time for the instances with large N or M . Because in the previous subsection Algorithm 4 was the best algorithm, here the performance of the other three algorithms is expressed relative to Algorithm 4. Per instance, we compute the relative distance of the costs of the heuristic solution generated by Algorithm 2 to the costs of the heuristic solution generated by Algorithm 4; and, similarly for the Algorithms 3 and 5. By definition, nonnegative relative differences are obtained for the Algorithms 2 and 3 as these algorithms are dominated by Algorithm 4. In the case of Algorithm 5, also negative relative differences may be obtained. The results are listed in Table 4, where again averages over different subsets of instances are listed.

We see that the performance of Algorithm 4 is on average only 5.1% worse relative to Algorithm 2, Algorithm 5 is 25.9% worse, and Algorithm 3 is 46.0% worse (see part (g)). Although Algorithm 5 is much worse than Algorithm 4 on average, Algorithm 5 generated a better solution than Algorithm 4 in 8% of all instances. The performance of Algorithm 2 is rather stable when input parameters are varied. The performance of the Algorithms 3 and 5 strongly depends on various parameter, among which the number of SKU-s.

In Figure 5, the distribution of the relative differences over all instances is given for the Algorithms 2, 3, and 5. This shows that large differences (of 100% and more) may occur for the performance of the Algorithms 3 and 5 relative to Algorithm 4, while the relative difference between Algorithm 2 and 4 does not become large in general. In 95% of all instances it is less than

| (a) Rel. diff. as a function of h^{\min} | | | | (b) Rel. diff. as a function of λ_n / h_n | | | |
|--|--------|--------|--------|---|---------|---------|---------|
| | 1 | 10 | 100 | | Case 1. | Case 2. | Case 3. |
| Alg.2 vs. Alg.4 | 5.4 % | 5.8 % | 4.2 % | Alg.2 vs. Alg.4 | 5.7 % | 2.4 % | 7.3 % |
| Alg.3 vs. Alg.4 | 54.6 % | 50.5 % | 33.0 % | Alg.3 vs. Alg.4 | 31.5 % | 57.7 % | 48.9 % |
| Alg.5 vs. Alg.4 | 30.7 % | 28.2 % | 18.8 % | Alg.5 vs. Alg.4 | 16.0 % | 38.7 % | 22.9 % |

| (c) Rel. diff. as a function of ρ | | | | | (d) Rel. diff. as a function of b | | | |
|--|--------|--------|--------|--------|-------------------------------------|--------|--------|--------|
| | 0.7 | 0.82 | 0.9 | 0.95 | | 1000 | 10000 | 100000 |
| Alg.2 vs. Alg.4 | 6.3 % | 5.6 % | 4.5 % | 4.1 % | Alg.2 vs. Alg.4 | 0.9 % | 3.2 % | 11.2 % |
| Alg.3 vs. Alg.4 | 43.5 % | 48.1 % | 46.5 % | 46.0 % | Alg.3 vs. Alg.4 | 12.0 % | 63.9 % | 62.2 % |
| Alg.5 vs. Alg.4 | 16.8 % | 26.6 % | 29.4 % | 30.7 % | Alg.5 vs. Alg.4 | 4.0 % | 35.9 % | 37.7 % |

| (e) Rel. diff. as a function of M | | | | | (f) Rel. diff. as a function of N | | | |
|-------------------------------------|--------|--------|--------|--------|-------------------------------------|--------|--------|--------|
| | 2 | 3 | 4 | 5 | | 15 | 25 | 50 |
| Alg.2 vs. Alg.4 | 6.0 % | 5.8 % | 5.4 % | 5.1 % | Alg.2 vs. Alg.4 | 2.7 % | 6.2 % | 6.5 % |
| Alg.3 vs. Alg.4 | 35.9 % | 45.6 % | 46.2 % | 46.0 % | Alg.3 vs. Alg.4 | 33.7 % | 44.8 % | 59.6 % |
| Alg.5 vs. Alg.4 | 18.4 % | 25.5 % | 26.0 % | 25.9 % | Alg.5 vs. Alg.4 | 18.0 % | 25.3 % | 34.4 % |

| (g) Aver. rel. differences over all instances | |
|---|--------|
| Algorithm 2 vs. Algorithm 4 | 5.1 % |
| Algorithm 3 vs. Algorithm 4 | 46.0 % |
| Algorithm 5 vs. Algorithm 4 | 25.9 % |

Table 4: *Relative difference between total costs obtained by different heuristics.*

20%.

Table 5 shows the computation times (in seconds) of the Algorithms 2-5 executed on a PC with a PIII-1000MHz processor. It shows that Algorithm 2 and Algorithm 4 have limited computation times for instances with $M = 2$ or 3 priority classes, and the computation times become much larger for $M = 4$ and 5. Algorithms 3 and 5 have small computation times for all M .

From the above results for our full test bed, we conclude that Algorithm 4 is most appropriate, at least for instances with 2 or 3 priority classes. Algorithm 5 may constitute an attractive alternative if computation times for Algorithm 4 become too large (this also applies when we would relax the assumptions of a single exponential server and equal mean repair times for all SKU-s, in which case simulation would be needed for the evaluation of a given priority assignment).

4.4 Cost savings relative to FCFS

In this subsection, we investigate the cost savings that are obtained via the use of priority classes. We take again the full test bed, and for all instances, we apply our best heuristic, Algorithm 4 (recall that our experiment in Subsection 4.2 showed a small optimality gap of 1.1%), and we

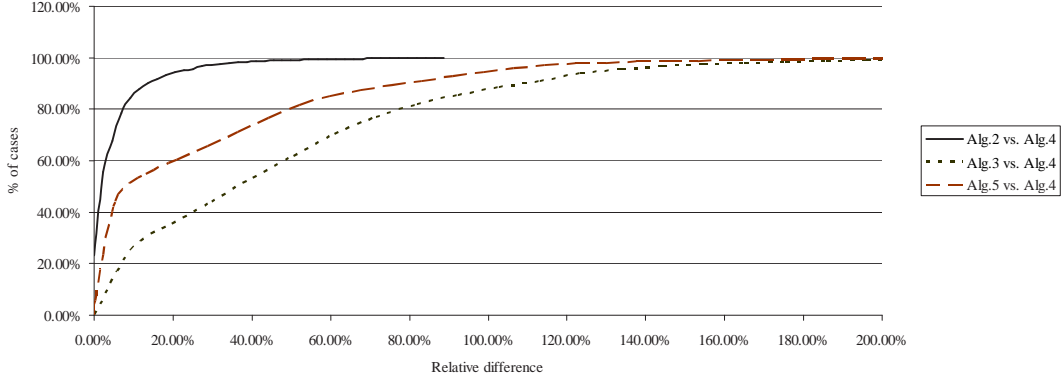


Figure 5: Average relative differences between total costs obtained by different heuristics.

| (a) Algorithm 2 | | | | (b) Algorithm 3 | | | |
|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | $N = 15$ | $N = 25$ | $N = 50$ | | $N = 15$ | $N = 25$ | $N = 50$ |
| $M = 2$ | 0.15 | 0.28 | 0.69 | $M = 2$ | 0.07 | 0.10 | 0.17 |
| $M = 3$ | 1.17 | 3.83 | 18.34 | $M = 3$ | 0.12 | 0.18 | 0.30 |
| $M = 4$ | 6.75 | 37.74 | 367.70 | $M = 4$ | 0.17 | 0.25 | 0.42 |
| $M = 5$ | 26.59 | 268.10 | 5520.94 | $M = 5$ | 0.22 | 0.32 | 0.54 |

| (c) Algorithm 4 | | | | (d) Algorithm 5 | | | |
|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | $N = 15$ | $N = 25$ | $N = 50$ | | $N = 15$ | $N = 25$ | $N = 50$ |
| $M = 2$ | 1.47 | 4.75 | 29.25 | $M = 2$ | 1.43 | 3.77 | 19.19 |
| $M = 3$ | 3.12 | 10.97 | 66.72 | $M = 3$ | 2.10 | 5.57 | 25.99 |
| $M = 4$ | 9.13 | 46.47 | 429.54 | $M = 4$ | 2.26 | 5.72 | 26.51 |
| $M = 5$ | 29.36 | 278.73 | 5360.88 | $M = 5$ | 2.31 | 5.79 | 26.73 |

Table 5: Average computation times (in seconds) of the Algorithms 2-5.

determine the optimal costs that are obtained if the FCFS service discipline is used in the repair shop. The latter case corresponds to using only one priority class, which is a special case of our model; notice that then only the basestock levels have to be optimized (cf. Subsection 3.1). By definition, the costs obtained via Algorithm 2 are lower than or equal to the optimal costs under the FCFS discipline, and we study the relative savings in costs that are obtained via the heuristic solution of Algorithm 2. These number denote how profitable it may be to work with static priority classes. The results are listed in Table 6.

From Table 6, we draw the following conclusions:

1. First of all, from part (f), we conclude that the average relative savings are more than 40%.

| (a) Rel. savings as a function of h^{\min} | | | | (b) Rel. savings as a function of λ_n / h_n | | | |
|--|-------|-------|-------|---|---------|---------|---------|
| | 1 | 10 | 100 | | Case 1. | Case 2. | Case 3. |
| $M = 2$ | 49.2% | 46.0% | 33.1% | $M = 2$ | 32.2% | 45.9% | 50.2% |
| $M = 3$ | 54.2% | 49.8% | 34.8% | $M = 3$ | 35.9% | 50.1% | 52.9% |
| $M = 4$ | 54.5% | 50.1% | 34.8% | $M = 4$ | 36.2% | 50.3% | 52.9% |
| $M = 5$ | 54.5% | 50.1% | 34.8% | $M = 5$ | 36.2% | 50.4% | 52.9% |

| (c) Rel. savings as a function of ρ | | | | | (d) Rel. savings as a function of b | | | |
|--|-------|-------|-------|-------|---------------------------------------|-------|-------|--------|
| | 0.7 | 0.82 | 0.9 | 0.95 | | 1000 | 10000 | 100000 |
| $M = 2$ | 39.0% | 41.4% | 44.1% | 46.5% | $M = 2$ | 39.9% | 49.1% | 39.2% |
| $M = 3$ | 40.2% | 44.1% | 48.0% | 52.9% | $M = 3$ | 41.6% | 52.9% | 44.4% |
| $M = 4$ | 40.2% | 44.1% | 48.4% | 53.2% | $M = 4$ | 41.7% | 53.1% | 44.6% |
| $M = 5$ | 40.2% | 44.1% | 48.4% | 53.2% | $M = 5$ | 41.7% | 53.1% | 44.6% |

| (e) Rel. savings as a function of N | | | | (f) Rel. savings over all instances | |
|---------------------------------------|-------|-------|-------|-------------------------------------|-------|
| | 15 | 25 | 50 | | |
| $M = 2$ | 38.9% | 42.4% | 46.9% | $M = 2$ | 42.8% |
| $M = 3$ | 43.2% | 46.2% | 49.5% | $M = 3$ | 46.2% |
| $M = 4$ | 43.6% | 46.4% | 49.4% | $M = 4$ | 46.5% |
| $M = 5$ | 43.6% | 46.4% | 49.4% | $M = 5$ | 46.5% |

Table 6: *Relative savings in total system costs via static priorities.*

In 42% of the instances, the savings were between 40% and 60%, and in 24% they were even more than 60%. These percentages imply that static priorities may lead to enormous cost savings in absolute terms for real-life situations (see also Section 5).

- Second, the results in part (f) show that, for an average instance, the maximal savings are around 46.5%, and these savings are reached at 4 priority classes. More than 90% of the maximal savings is already obtained when one works with 2 priority classes (42.6% for $M = 2$ vs. 46.5% as maximum). And, via 3 priority classes almost 100% of the maximal savings is obtained. This is an important observation from both an algorithmic and practical point of view. It implies that effective and efficient heuristics are only needed for systems with 2 or 3 priority classes, and Algorithm 4 meets this requirement. Further, in practice, implementing priority control rules with 2 or 3 priority classes will be much easier than rules with 4 or more priority classes.
- Third, the parts (a)-(e) show how the relative savings depend on the various input parameters. They strongly increase for increasing cost differences in inventory holding cost parameters (part (a)) and for increasing workloads (part (c)). Further, correlated demand rates and

inventory holding cost parameters, as common in practice, lead to significantly larger savings than when these parameters are not correlated; see Cases 2 and 3 vs. Case 1 in part (b) of the table. Part (d) shows that savings first increase and then decrease for increasing penalty costs. From part (e), we learn that the savings slightly increase as a function of the number of SKU-s.

5 Stylized Cases

Based on the insights of the previous section, we executed an additional experiment with real life data of the Royal Netherlands Navy. We obtained data of so-called capacity groups that are responsible for preventive and corrective maintenance and for the repair of spare parts for different subsystems of technical systems installed at frigates. Reliable data that we could collect from their information systems were for the numbers of SKU-s that were handled per capacity group, the prices of these SKU-s (which we could easily translate to inventory holding cost parameters), and the total demand rates. No reliable data were available on (mean) repair times for SKU-s. We therefore constructed one stylized case per capacity group. We pretend that at each capacity group there is a separate group of engineers that only works on repairs of spare parts, and further we make the same assumptions as in our model. Per case, we chose the N , λ_n , and h_n according to the above data. The mean repair time was chosen such that the workload was equal to 0.8 and 0.9, respectively. We considered both the use of FCFS and the use of static priorities with 2 priority classes for the repair shop. We limited ourselves to 2 priority groups as that brings most of the maximal possible savings (cf. Subsection 4.4). The optimization of the system with static priorities was solved via Algorithm 4. Per case, the penalty cost parameter was tuned such that under the FCFS service discipline an aggregate fill rate AFR_{FCFS} of 90% was obtained. It holds that

$$AFR_{FCFS} = \sum_{n=1}^N \left[\frac{\lambda_n}{\sum_{l=1}^N \lambda_l} \sum_{j=0}^{S_n-1} P_j^n(j, \mathbf{X}_{FCFS}) \right],$$

where \mathbf{X}_{FCFS} corresponds to a priority assignment in which all SKU-s are assigned to the same priority class. The larger the b is chosen, the larger the optimal basestock levels will be (cf. (8)), and thus the larger AFR_{FCFS} (see also Theorem 2 of Van Houtum and Zijm, 2000). We chose b as the lowest value for which $AFR_{FCFS} \geq 0.90$.

In Table 7, the number of SKU-s per case is found back. In Figure 6, the data the inventory holding cost parameters and demand rates are plotted (for the lists with all values, see Dirkwager, 2004). For the Cases 1-3, we see similar patterns as in the cases 2 and 3 of the test bed of Section 4. For Case 4, we see a rare pattern, with several SKU-s that have high inventory holding costs

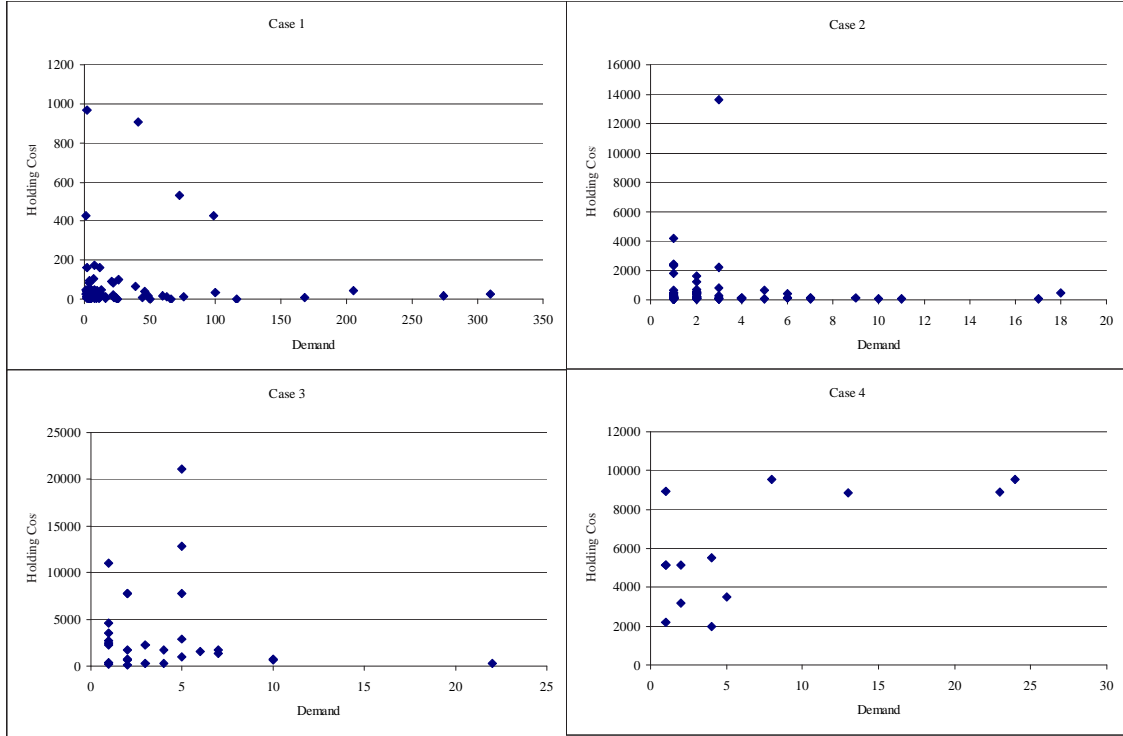


Figure 6: *Holding cost parameters (in EURO-s/year) and yearly demand rates for Cases 1-4.*

and high demand rates, and there no SKU-s with low inventory holding costs and high demand rates. Also, in Cases 1-3, we see a large ratio between the highest and lowest inventory holding cost parameter (the ratio is 1138 for Case 1, 457 for Case 2, and 458 for Case 3). In Case 4, this ratio is small (equal to 5).

The main results for these cases are also presented in Table 7. In the fourth column, the tuned penalty cost parameter b is given. In the fifth and sixth column, we have listed the costs under FCFS and the use of 2 priority classes, respectively (these costs are include penalty costs). The cost reduction obtained via priority classes is given in the last column.

We find that large cost reductions are obtained for the Cases 1-3, with percentages varying from 39% to 55%. For workloads of 90%, the savings are significantly larger than for workloads of 80%. For Case 4, the cost reductions are relatively small. This is due to the small ratio between the highest and lowest inventory holding cost parameter.

6 Conclusions

We studied the use of static priority classes in a repairable spare parts system consisting of one repair shop and one stockpoint. We made simplifying assumptions for the repair shop and repair times.

| | N | Util. rate | Penalty (b) | TC_{FCFS} | TC_{Prior} | Reduction |
|--------|-----|------------|-----------------|-------------|--------------|-----------|
| Case 1 | 67 | 0.8 | 3596 | 3239 | 1723 | 47 % |
| | | 0.9 | 2406 | 4507 | 2007 | 55 % |
| Case 2 | 99 | 0.8 | 13958 | 25701 | 15612 | 39 % |
| | | 0.9 | 23427 | 32052 | 16612 | 48 % |
| Case 3 | 29 | 0.8 | 124624 | 124694 | 69975 | 44 % |
| | | 0.9 | 66657 | 158011 | 70619 | 55 % |
| Case 4 | 13 | 0.8 | 136249 | 196215 | 176989 | 10 % |
| | | 0.9 | 106656 | 333435 | 271821 | 18 % |

Table 7: Overview of the results for the Royal Netherlands Navy for $M = 2$.

We showed that under a given assignment of SKU-s to priority classes, optimal basestock levels follow from Newsboy equations, which reduced our full optimization problem to an optimization problem for the priority assignments (this reduction also holds under relaxed assumptions for the repair shop and repair times). We developed one heuristic, called Algorithm 4, that was effective (an optimality gap of 1.1% was measured in an experiment with 15 SKU-s and 2 priority classes) and efficient for 2 or 3 priority classes. The latter is sufficient, as we also observed that having 4 or more priority classes instead of 2 or 3 classes does not lead to much lower optimal costs. The basic ideas behind Algorithm 4 are simple (enumeration among so-called ordered assignments, followed by local search), and thus it would not be hard to extend the algorithm for systems with relaxed assumptions (the definition of ordered assignments has to be extended as well, which may be less straightforward). Finally, we investigated the costs savings that one may obtain by the use of static priorities instead of the FCFS service discipline. Savings of 40-60% are possible in many cases, among which in 3 of 4 stylized cases with real-life data. As investments in repairable spare parts are huge in many companies (tens or hundreds of million EURO-s), this suggests that many companies could save a lot of money via static priorities.

References

- Aboud, N.E. 1996. The Markovian two-echelon repairable item provisioning problem. *Journal of the Operational Research Society* 47, 284-296.
- Albright, S.C., A. Gupta. 1993. Steady-state approximation of a multi-echelon multi-indentured repairable-item inventory system with a single repair facility. *Naval Research Logistics* 40, 479-493.
- Avsar, Z.M., W.H.M. Zijm. 2002. Capacitated two-echelon inventory models for repairable item systems. In: Gershwin, S.B., et al. (eds.), *Analysis and Modeling of Manufacturing Systems*, Kluwer, Boston, 1-36.
- Buzacott, J.A., J.G. Shanthikumar. 1993. *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliffs, NJ.

- Caggiano, K.E., J.A. Muckstadt, J.A. Rappold. 2006. Integrated real-time capacity and inventory allocation for repairable service parts in a two-echelon supply system. *Manufacturing and Service Operations Management* 8, 292-319.
- Diaz, A., M.C. Fu. 1997. Models for multi-echelon repairable item inventory systems with limited repair capacity. *European Journal of Operational Research* 97, 480-492.
- Dirkzwager, A.A. 2004. On the Use of Priority Classes in a Capacitated Model of Repairable Spare Parts at the Royal Netherlands Navy. M.Sc. thesis, Technische Universiteit Eindhoven.
- Gross D., D.R. Miller, R.M. Soland. 1983. A closed queueing network model for multi-echelon repairable item provisioning. *IIE Transactions* 15, 344-352.
- Gupta, A., S.C. Albright. 1992. Steady-state approximations for a multi-echelon multi-indentured repairable-item inventory system. *European Journal of Operational Research* 62, 340-353.
- Hausman, W., G. Scudder. 1982. Priority scheduling rules for repairable inventory systems. *Management Science* 28, 1215-1232.
- Miller, D. 1981. Computation of steady-state probabilities for $M/M/1$ priority queues. *Operations Research* 29, 945-958.
- Muckstadt, J. 2005. *Analysis and Algorithms for Service Parts Supply Chains*. Springer, Berlin.
- Perlman, Y., A. Mehrez, M. Kaspi. 2001. Setting expediting repair policy in a multi-echelon repairable-item inventory system with limited repair capacity. *Journal of the Operational Research Society* 52, 198-209.
- Pyke, D.F. 1990. Priority repair and dispatch policies for repairable-item logistics systems. *Naval Research Logistics* 37, 1-30.
- Rustenburg, W.D., G.J. van Houtum, W.H.M. Zijm. 2002. Exact and approximate analysis of multi-echelon, multi-indenture spare parts systems with commonality. In: Shanthikumar, J.G., D.D. Yao, W.H.M. Zijm (eds.), *Stochastic Modeling and Optimization of Manufacturing Systems and Supply Chains*, Kluwer, Boston, 143-176.
- Sherbrooke, C.C. 1968. METRIC: Multi-Echelon Technique for Recoverable Item Control. *Operations Research* 16, 122-141.
- Sherbrooke, C.C. 2004. *Optimal Inventory Modeling of Systems: Multi-Echelon Techniques*. Kluwer, Dordrecht, The Netherlands.
- Sleptchenko, A., I.J.B.F. Adan, G.J. van Houtum. 2004. Joint queue length distribution of multi-class, single-server queues with preemptive priorities. EURANDOM Report 2004-045, Technische Universiteit Eindhoven.
- Sleptchenko, A., M.C. van der Heijden, A. van Harten. 2002. Effects of finite repair capacity in multi-echelon, multi-indenture service parts supply systems, *International Journal of Production Economics* 79, 209-230.
- Sleptchenko, A., M.C. van der Heijden, A. van Harten. 2005a. Using repair priorities to reduce stock investment in spare part networks, *European Journal of Operations Research* 163, 733-750.

- Sleptchenko, A., A. van Harten, M.C. van der Heijden. 2005b. An exact solution for the state probabilities of the multi-class, multi-server queue with preemptive priorities. *Queueing Systems* 50, 81-107.
- Thonemann, U.W., A.O. Brown, W.H. Hausman. 2002. Easy quantification of improved spare parts inventory policies. *Management Science* 48, 1213-1225.
- Van der Heijden, M.C., A. van Harten, A. Sleptchenko. 2004. Approximations for Markovian multi-class queues with preemptive priorities. *Operations Research Letters* 32, 273-282.
- Van Harten, A., A. Sleptchenko. 2003. On Markovian multi-class, multi-server queueing. *Queueing Systems* 43, 307-328.
- Van Houtum, G.J., W.H.M. Zijm. 2000. On the relation between cost and service models for general inventory systems. *Statistica Neerlandica* 54, 127-147.
- Verrijdt, J., I.J.B.F. Adan, A.G. de Kok. 1998. A trade-off between emergency repair and inventory investment. *IEE Transactions* 30, 119-132.
- Zijm, W.H.M., Z.M. Avsar. 2003. Resource-constrained two-echelon inventory models for repairable item systems. *International Journal of Production Economics* 81-82, 573-588.