Optimization of industrial-scale assemble-to-order systems

Willem van Jaarsveld and Alan Scheller-Wolf
Erasmus School of Economics, Erasmus University Rotterdam
Tepper School of Business, Carnegie Mellon University

May 6, 2014

Abstract

Using a novel stochastic programming (SP) formulation, we develop an algorithm for inventory control in industrial sized Assemble-To-Order (ATO) systems that has unparalleled efficiency and scalability. Applying our algorithm on several numerical examples, we generate new insights with respect to the control and optimization of these systems.

We consider a continuous time model, seeking base-stock levels for components that minimize the sum of holding costs and product-specific backorder costs. Our initial focus is on first-come first-serve (FCFS) allocation of components to products; for this setting our algorithm quickly computes solutions for which the asymptotic optimality gap with the optimal FCFS base-stock policy is less than 1%. We then turn to two related questions: How do common heuristics used in practice compare to our performance, and how costly is the FCFS assumption.

For the first question, we investigate the effectiveness of ignoring simultaneous stockouts (ISS), a heuristic that has been used by companies such as IBM and Dell. We show that ISS performance, when compared to the optimal FCFS base-stock policy, improves as the average newsvendor (NV) fractiles increase, but suffers under lead time demand correlations.

For the second question, we adapt the SP formulation of Doğru et al. (2010), yielding an efficiently computable upper bound on the benefit of optimal allocation over FCFS. We find that for many large ATO systems, FCFS performs surprisingly well, and that its performance improves with decreasing NV fractile asymmetry among products and, again, with increasing average NV fractiles. We also investigate simple no-holdback allocation policies, and find that they tend to outperform the best FCFS policies.

Keywords: Continuous-time assemble-to-order systems, Industrial-scale problems, Base-stock policies, First-come-first-serve allocation, Optimal Allocation, Stochastic programming

1 Introduction

Assemble-to-Order (ATO) systems allow companies to efficiently attain short response-times for a broad assortment of products by assembling them, on demand, from multiple components. But, to fully attain the benefits of ATO systems, companies need to effectively control inventory for a large assortment of components. This is crucial, because demand fulfillment requires the simultaneous availability of the components that are needed to assemble a product, and a single component may be common to a number of products (Song and Zipkin 2003).

Specific examples of companies that manage large ATO-systems (i.e. with hundreds of components) include IBM (Swaminathan and Tayur 1998, Cheng et al. 2002) and Dell (Kapuscinski et al. 2004). Online retailers and many maintenance organizations face similar problems: The catalog of an online retailer may consist of thousands of products. They often need to satisfy customer orders
consisting of multiple products, which should preferably be shipped together (e.g. Xu et al. 2009, at Amazon). Likewise, companies that provide maintenance for capital goods typically keep inventories of many spare parts and tools, and repairs arriving over time typically require multiple spare parts and tools to complete. Specific examples include the maintenance organizations of Philips Healthcare (Kampstra 2012), Fokker Services (van Jaarsveld et al. 2011), ASML (Vliegen 2009) and an unspecified copier manufacturer (Teunter 2006). The assortments of Philips Healthcare and Fokker Services, for example, consist of thousands of spare parts.

Companies typically cope with the difficulties of managing large-scale ATO systems using pragmatic approaches. For example, companies often use first-come first-serve (FCFS) allocation of components to products. Despite being non-optimal, FCFS has many practical advantages such as ease of implementation and fairness. In addition, FCFS allows companies to guarantee a delivery date immediately upon demand arrival, which is surprisingly difficult to achieve with other simple allocation policies (Lu et al. 2010). When optimizing the component inventory control policies under FCFS, companies may simplify analysis by using base-stock policies and by approximating the probability of stock-outs by ignoring the possibility of Simultaneous Stock-outs (ISS). Companies using these pragmatic approaches have been described in various case studies (e.g. Cheng et al. 2002, Kapuscinski et al. 2004, Vliegen 2009, Xu et al. 2009, van Jaarsveld et al. 2011). The widespread use of such strategies gives rise to a number of questions:

1. Can one find provably (near-) optimal base-stock policies for FCFS ATO systems?
2. What are the costs of using ISS while optimizing the inventory policy?
3. What are the costs of using FCFS instead of an optimal allocation policy?

The goal of this paper is to develop analytical models to address these questions, for the first time, for the large-scale ATO systems that exist in practice.

Of course, some companies may use more complex \((r,q)\) or \((s,S)\) replenishment policies for ATO inventory control, for example to attain economies of scale. Nevertheless, most scholarly studies focus on base-stock policies, because “it sharpens the focus on the higher-level business issue of inventory/service trade-off, without getting into operational issues such as order sizes” (Song and Yao 2002). We focus on base-stock policies for the same reason. And, while many studies have addressed the optimization of base-stock levels in FCFS ATO systems (e.g. Zhang 1997, Song and Yao 2002, Akçay and Xu 2004, Lu et al. 2005, Lu and Song 2005, Huang and de Kok 2011), none of the proposed methods can compute close-to-optimal solutions for large-scale systems. This failure stems from not one, but two shortfalls: Current literature not only lacks methods to find high quality solutions for large systems, but also cannot provide tight lower bounds on the costs of the optimal solutions, which are necessary to guarantee solution quality. We address both of these shortfalls by developing a novel, exact, two-stage stochastic programming (SP) formulation that ensures that high quality solutions and tight asymptotic lower bounds \(^1\) are efficiently computable for large-scale systems. We emphasize that while use of sampling approximation in stochastic programming is not new, our development of an approach to solve the sampling approximation for large systems is novel, and nontrivial. Ensuring that the sampling approximation is tractable involves two contributions to the current literature. Namely, our approach differs on two key points from existing SP-based optimization methods: We focus on a tagged demand (see Section 3.2.1) and we use indicator variables in our SP formulation (see Section 3.2.2).

\(^1\)Throughout the paper we will use the term “asymptotic lower bounds” to denote sampling based bounds that asymptotically converge to lower bounds for a specified quantity. As we will be dealing with non-asymptotic settings, the actual bounds we use will be normally based 99.7\% confidence intervals for the true bound.
We consider systems in which the objective is to minimize the sum of component holding and product-specific back-order costs. The SP we propose can be used to tackle a range of modeling assumptions (see Proposition 3); but, for ease of presentation, we focus on pure Poisson demand and deterministic lead times. We verify the performance of our algorithm in a computational study: Within one hour, our algorithm computes solutions that have optimality gaps that are smaller than one percent, for problems consisting of hundreds of components and products. These problems are orders of magnitude larger than the systems for which provably close-to-optimal solutions can be found using existing approaches.

Having answered our first question in the affirmative, we turn to the second: The performance of ISS compared to the optimal policy. Utilizing our tight lower bounds on optimal costs under FCFS, we find that ISS has good performance when product newsvendor (NV) fractiles are high, but its performance degrades as NV fractiles decrease, especially when lead time demand for different components is highly correlated: Optimality losses in our experiments range from 0.1 to 30%.

We then address our third question: What are the performance benefits of optimal allocation when compared to FCFS? To answer this we develop a new SP that constitutes a lower bound on the costs of the optimal base-stock policy under optimal allocation. This SP is obtained by adapting an idea proposed by Doğru et al. (2010) to systems with unequal lead times. Results in the literature for single-component systems may lead one to believe that NV fractile asymmetry is the dominant factor determining the performance of FCFS (Topkis 1968). While it is true that as NV asymmetry increases, the relative benefit of optimal allocation over FCFS increases, we find a number of practical cases in which this benefit remains rather limited, even for significant NV asymmetry. To explore whether other simple allocation policies can close the gap between the best FCFS policy and the lower bound under optimal allocation, we investigate two easy-to-implement no-holdback policies: These only (and always) allocate components to a product when this leads to demand fulfillment. We find that these allocation policies, combined with suitable base-stock levels, can outperform the best possible FCFS policy by up to 8%. Our new lower bound on the optimal FCFS policy cost is necessary to obtain this result. No-holdback policies were investigated by other scholars (e.g. Song and Zhao 2009, Lu et al. 2010, Doğru et al. 2010), who found promising results for special cases. We appear to be the first to confirm their practical value for general systems.

Our numerical findings thus indicate that the performance of ISS when compared to the optimal base-stock levels under FCFS, and of FCFS when compared to optimal allocation, improve as the penalty costs increase. We complement these results by investigating the limit as penalty costs grow: We prove that ISS base-stock levels combined with FCFS allocation are asymptotically optimal in that limit.

In summary, we develop the first algorithm that computes provably near-optimal base-stock levels for large, FCFS ATO systems. Using this algorithm we are able to generate conclusive insights into the performance of the ISS heuristic for a number of industrial-scale systems, allowing companies to infer whether ISS will perform well in their environment. We then investigate the benefit of other relatively simple allocation rules, and optimal allocation, compared to FCFS in general industrial-sized ATO systems. This deepens and expands existing insights that were previously gained through the study of small-scale systems and special cases.

The remainder of the paper is organized as follows. In the next section we provide a literature overview. In Section 3 we formulate our model, develop our SP formulation of base-stock level optimization under FCFS, and computational procedures to solve it. We also develop our lower bound on the optimal base-stock policy under optimal allocation. In Section 4 we present the results of the computational study, and our asymptotic optimality result. Section 5 concludes.
2 Literature review

Even though closed-form expressions of performance characteristics of ATO systems controlled using base-stock policies and FCFS allocation often exist (e.g. Song 1998, 2002), exact computation of these expressions is intractable for larger systems (i.e. with more than 3 to 8 components in a product, depending on the precise setting). As a consequence, scholars have developed bounds and approximations that are tractable for larger systems (e.g. Song 1998, 2002, Vliegen and van Houtum 2009). Such bounds have also been used to develop approximate formulations of optimization problems (e.g. Zhang 1997, Song and Yao 2002, Cheng et al. 2002, de Kok 2003, Lu et al. 2005, Lu and Song 2005, van Jaarsveld et al. 2011). The use of the ISS assumption is an example along these lines: ISS gives rise to upper bounds on waiting time (Lu and Song 2005) and lower bounds on the fill-rate (Boole’s inequality). We emphasize that while approximate formulations may be tractable, the resulting solution will be sub-optimal in general. Moreover, the degree of sub-optimality has remained an open question because of a lack of quality lower bounds on the optimal solution.

So, support of the use of simplifications such as ISS in practice has relied on arguments like those in Kapuscinski et al. (2004): “because the supply chain shortages of multiple components are very infrequent, simultaneous stock-outs are rare and can be ignored.” However, Cheng et al. (2002) find feasible solutions that improve costs by 8-15% as compared to ISS solutions for a 3 product system with high fill-rate targets (in the 90-98% range). Similarly, while van Jaarsveld et al. (2011) show that simultaneous stockouts are rare for the ISS solution, they cannot rule out the possibility of solutions that significantly improve on the ISS solution.

We now review specific contributions considering the optimization of ATO systems, discussing continuous review, periodic review, and then non-FCFS allocation.

**Continuous review:** Song and Yao (2002) develop algorithms for approximate minimization of the number of back-orders in single product systems under iid lead times; which Lu et al. (2005) extend to the multi-product setting. Gullii and Koksalan (2013) consider inventory control of orthopedic implants, where demand occurs for kits of such implants. This problem is similar to an ATO system, but with a different resupply system. A greedy heuristic to optimize the base-stock levels is shown to have good performance in a numerical experiment. To solve a problem encountered at a repair shop, van Jaarsveld et al. (2011) study the optimization of \((s, S)\) policies in ATO systems using the ISS assumption. They develop an algorithm that finds close-to-optimal (under ISS) solutions for large-scale problems, that was implemented in practice.

Our investigation into the quality of the ISS solutions (Question 2) builds on Lu and Song (2005), who also consider cost minimization for product specific back-order costs, focusing exclusively on FCFS allocation. For a 3-product 2-component system, they use the closed-form expressions for the waiting time in Song (2002). For larger systems, they rely on simulation and exhaustive search. In their experiments they find solutions that outperform the ISS solution by 5%, and conclude that the ISS solution has good performance. Our SP allows us to find provably near-optimal solutions for large systems, which allows us to refine and modify Lu and Song’s (2005) conclusions.

**Periodic review:** A number of early contributions use the single period/zero lead time assumption for tractability. Gerchak and Henig (1986) make an early contribution along these lines. Swaminathan and Tayur (1998) consider a case at IBM in which sub-assemblies (vanilla boxes) play a pivotal role. Van Mieghem and Rudi (2002) investigate newsvendor networks, a type of supply chain that is considerably more general than an ATO system.

For positive lead time models, different assumptions exist for the allocation of components to demands arriving in the same period, but all studies apply FCFS to demands in different periods. Hausman et al. (1998) develop a heuristic which uses an equal fill rate for each component; while simple, it cannot properly account for differences in stock-out costs. Zhang (1997) assumes...
fixed priority allocation and formulates an approximate optimization model under service level constraints. Agrawal and Cohen (2001) derive similar results under a fair share allocation rule. Cheng et al. (2002) minimize costs under product-specific fill rate constraints. They develop special purpose algorithms based on the ISS assumption, and illustrate their approach on a case at IBM. De Kok (2003) considers ATO systems with an ideal product structure: Longer lead time components are either strictly more common, or completely independent of shorter lead time components. He develops algorithms to optimize base-stock levels based on approximations.

Akçay and Xu (2004) consider weighted time-window fill-rate maximization under a budget constraint. Unlike the studies discussed earlier, they investigate the performance of a heuristic that dynamically allocates components to products arriving in the same period, using an SP to optimize the base-stock levels. Huang and de Kok (2011) note that many models exclude the holding costs of inventory committed to product demands; they include these costs in their model. Their computational results show that committed stock may comprise a large fraction of total inventory. They develop an SP to optimize base-stock levels. Both SP formulations (Akçay and Xu 2004, Huang and de Kok 2011) are structurally similar to the SPs proposed for the zero lead time case: They use the base-stock levels directly as decision variables, which gives rise to a non-linear sampling-based problem. To overcome this difficulty they use auxiliary variables to linearize sampling-based bounds, i.e. the big-M method (Huang and de Kok 2011, Akçay and Xu 2012). However, this approach gives rise to weak LP relaxations and severe scalability issues.

Our approach makes two contributions with respect to existing SP formulations of ATO systems, to address the shortcomings of these formulations: We do not adapt the zero lead time formulation by focusing on a specific period, instead we focus directly on an arbitrary product demand, determining the moment at which the components used in that demand were ordered. In addition, we use binary variables that indicate that a particular base-stock level is used for a particular component demand, instead of using the base-stock levels as decision variables. As a consequence, our SP gives rise to computational methods that for the first time do scale to large-scale systems, primarily because the sampling-based lower bound has very strong linear relaxations.

Non-FCFS allocation: Optimal control in single-component systems requires rationing levels (Topkis 1968), while optimal control of single product systems requires balanced base-stock policies (Rosling 1989). Multi-product, multi-component systems require both. A particular multi-product, multi-component system that has been of recent interest is the “W-model:” a 3-component 2-product system, in which one component is used in both products, while the other components are each unique to a product. Bernstein et al. (2011) investigate a multi-period W-model, with a single zero lead time replenishment. Other investigations typically assume multiple replenishments and positive lead times: Song and Zhao (2009) compare the costs of a single shared stock versus two separate stocks for the common component, where inventory is controlled using base-stock policies. They find that first-ready first-serve (FRFS) allocation, a no-holdback policy, tends to outperform FCFS. Lu et al. (2010) investigate base-stock control and no hold-back policies. They consider generalizations of W-models: each product consists of a product specific component and a component common to all products. They show that any no-holdback policy minimizes the total back-orders and total inventories, and thus total costs, when back-order and holding costs are symmetric.

Doğru et al. (2010) study the W-model for product specific back-order costs, assuming component lead times are deterministic and equal. This is restrictive compared to the general lead time models in Lu et al. (2010). However, Doğru et al. (2010) obtain stronger results: They develop a stochastic program that constitutes a lower bound on the costs of any replenishment policy under optimal allocation. While in general feasible solutions to their SP need not translate to feasible solutions to the original problem, for the W-model under cost symmetry, they can translate an
optimal solution to this SP into a feasible (and optimal) solution for the original problem. They show that this solution uses (independent) base-stock replenishment and no-holdback allocation. Under a balanced capacity assumption, they prove the same result.

Each using a different proof technique, Lu et al. (2012) and Reiman and Wang (2012) relax the equal lead time condition in Doğru et al. (2010). Lu et al. (2012) consider 2-component 2-product systems where one product uses both components, while the other product uses only a single component (N-systems) with deterministic unequal lead times. Under cost symmetry, they use a hybrid approach to show that the optimal policy is a coordinated base-stock policy under no-holdback allocation. Reiman and Wang (2012) consider a generalized W-model: Product specific components share the same lead time, which is larger than the lead time of the common component. They adapt the SP formulation of Doğru et al. (2010) to unequal lead times. Again, under cost symmetry, they translate an optimal solution to the SP into a feasible (and optimal) solution for the original problem. Replenishment involves coordination, while no hold-back allocation remains optimal.

Reiman and Wang (2013) develop allocation and replenishment methods for equal lead time, but otherwise general, ATO systems, based on the SP developed in Doğru et al. (2010). They show that these policies are asymptotically optimal as the lead times, or equivalently the demand rates, grow large. Plambeck and Ward (2006, 2008) also develop asymptotically optimal policies as the demand rates grow large, but their setting is quite different: Production is capacitated, and a single decision needs to be made on component production capacities and product prices. In addition, Plambeck and Ward (2008) allows for expediting of components, and salvaging of excess components.

Other scholars have used dynamic programming to investigate the structure of the optimal policy for Markovian ATO systems: Exponential make-to-stock replenishment, Poisson demand, and lost sales. Benjaafar and ElHafsi (2006), ElHafsi et al. (2008), and Nadar et al. (2013) study different system architectures. One general result is that optimal replenishment and rationing decisions should take into account the inventory of all components. However, while Benjaafar and ElHafsi (2006) and ElHafsi et al. (2008) find that state-dependent base-stock levels and rationing levels are optimal for their models, Nadar et al. (2013) show that still more complex lattice-dependent base-stock policies and lattice dependent rationing is optimal in their more general setting. For these models, finding the optimal policy is only possible for very small systems, because of the exponential growth of the state-space.

In summary, studies on non-FCFS and optimal allocation in ATO-systems have focused on special cases: The limit of high demand rates/lead times or a specially structured bill of materials. In contrast, we investigate general ATO systems, outside of the limit of long lead times. Our methodological contribution is a new and computationally tractable SP lower bound on the cost of the best base-stock level under optimal allocation. This SP is quite different from our exact SP under FCFS allocation.

3 Methods

In Section 3.1 we formalize the model and introduce notation. In Section 3.2, we develop an exact SP of the model under FCFS allocation of components to products, and computational methods to solve it. In Section 3.3 we develop a lower bound on the costs of the optimal base-stock policy under optimal allocation.  

\footnote{In the interest of brevity, we omit a number of implementation details that significantly increase the efficiency of the computational methods presented in this section. To ensure that our results can be replicated, the source code}
3.1 Model and preliminaries

We consider a continuous time ATO system: Inventory is kept for different components, while demands arrive for different products. Each product is assembled from components on demand, assembly is assumed to be instantaneous. Unsatisfied product demand is back-ordered. Inventory for each component is controlled by keeping the inventory position fixed at the component’s base-stock level. The inventory position for each component equals inventory on-hand plus inventory on order minus back-orders. Our objective is to minimize the sum of component holding costs and product back-order costs. For ease of exposition we make a number of standard assumptions: Demands for products form independent Poisson processes, component replenishment lead times are deterministic, and components of each type are used in quantity 1 (or 0) in products. (Our SP for FCFS allocation can be applied under more general assumptions, as summarized in Proposition 3. Our asymptotic lower bound under optimal allocation can also be generalized significantly, but this is not the focus of the present paper.)

Let $\mathcal{I}$ denote the set of product types, and $\mathcal{J}$ the set of component types. Throughout, we use superscript $i$ to index product types, and subscript $j$ to index component types. In addition we will use the following notation:

- $\lambda^i > 0$: demand rate for products of type $i$.
- $b^i > 0$: penalty costs per back-ordered product $i$ per time unit.
- $\mathcal{J}^i \subseteq \mathcal{J}$: The set of components used to assemble product $i$.
- $B^i$: random variable denoting the steady state number of back-ordered product $i$ demands.
- $\mathcal{I}_j \subseteq \mathcal{I}$: set of products that use a component $j$. So $i \in \mathcal{I}_j \Leftrightarrow j \in \mathcal{J}^i$.
- $h_j > 0$: holding costs per component $j$ per unit time.
- $l_j > 0$: lead time for replenishment orders of component $j$.
- $\bar{l} := \max_{j \in \mathcal{J}} l_j$: maximum replenishment lead time.
- $\lambda_j := \sum_{i \in \mathcal{I}_j} \lambda^i$: “demand” rate for components of type $i$.
- $h^i := \sum_{j \in \mathcal{J}} h_j$: “holding cost” for products of type $i$.
- $H_j$: random variable denoting the amount of on hand inventory of component $j$ in steady state (including committed inventory in the case of FCFS allocation).
- $s_j$: base-stock level used for component type $j$; we assume $s_j \in \{0, 1, 2, \ldots\}$.
- $\bar{s} := \{s_j | j \in \mathcal{J}\}$: the vector of base-stock levels.

We consider the following cost rate:

$$\sum_{i \in \mathcal{I}} b^i \mathbb{E}(B^i(\bar{s})) + \sum_{j \in \mathcal{J}} h_j \mathbb{E}(H_j(\bar{s})).$$

(1)

For any allocation policy that does not let back-orders grow to infinity, the system is positive recurrent, thus the expectations in (1) are well-defined. Apart from the base-stock levels $\bar{s}$, $B^i$ and used to compute the results in Section 4 is available from the authors upon request.
$H_j$ depend on the allocation policy that is used. In Section 3.2, we investigate the minimization of (1) under the assumption of FCFS allocation of components to product demands. In Section 3.3, we develop an asymptotic lower bound on (1) under optimal allocation.

Lu and Song (2005) show that (1) can be equivalently rewritten as

$$
\sum_{i \in \mathcal{I}} \tilde{b}_i \mathbb{E}(B^i(s)) + \sum_{j \in \mathcal{J}} h_j s_j - D
$$

where $\tilde{b}_i = b_i + h^i$ and $D = \sum_{j \in \mathcal{J}} h_j \lambda_j l_j$. While Lu and Song (2005) restrict themselves to FCFS allocation, the equivalence of (1) and (2) holds for any allocation policy.

### 3.2 Base-stock levels under FCFS

In this section we propose a novel, exact SP formulation for minimization of the cost rate (1) over the base-stock levels $\vec{s}$, when FCFS allocation is used. We then develop a sampling approximation of the SP. Next, we derive the ISS base-stock levels introduced by Lu and Song (2005), and use our SP to prove that the ISS base-stock levels are upper bounds on the optimal base-stock levels. Finally, we discuss how to extend the results in this section to more general modeling assumptions.

Under FCFS, components are allocated to product demands in the order in which the demands arrive: Upon arrival of a product demand, one uncommitted on-hand component of each type required by the product is committed to that demand. If no such components are available, then the uncommitted component on order that will arrive soonest is committed to the product demand.

#### 3.2.1 The SP formulation

Instead of focusing on the average number of back-orders $B_i$, our SP formulation is based on the waiting time $W^i$ incurred by an arbitrary demand for a product of type $i$. Note that $\mathbb{E}W^i$ is a non-separable function of the base-stock levels of all components used in product $i$. 

Figure 1: A graphical representation of $T_j^i(k)$ and $W_j^i(k)$ and $W^i$. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{A graphical representation of $T_j^i(k)$ and $W_j^i(k)$ and $W^i$.}
\end{figure}
Our first objective is to write $W^i$ as the maximum of a number of random variables, each of which depends on the base-stock level of only a single component. To this end, we introduce additional notation (for a graphical representation of this notation, see Figure 1):

- $\{T_j^i(k)\mid j \in J^i, k \in \{0, 1, \ldots\}\}$: Consider an arbitrary demand for a product of type $i$, arriving at time $t := 0$. We now examine “demands” for components of types $j \in J^i$ using component $j$ that arrived before $t = 0$. (That is, demands for products of type $i' \in I$ arriving before $t = 0$.) Then, $T_j^i(k)$ is defined as the random time at which the $k$th demand for component $j$ arrived, when counting backwards from the current demand $k := 0$ at $t = 0$. In Figure 1, for example, $T_j^i(1) = -0.5$ and $T_j^i(3) = T_j^i(2) = -2.75$.

- $\{W_j^i(k)\mid j \in J^i, k \in \{0, 1, \ldots\}\} := \{(T_j^i(k) + l_j)^+\mid j \in J^i, k \in \{0, 1, \ldots\}\}$. If a base-stock level $s_j$ is used for component $j$, the component ordered at $T_j^i(s_j)$ will be used to satisfy the demand arriving at $t = 0$. This can be understood by noting this is the component ordered exactly $s_j$ demand ago, so it will be allocated to product $i$ by FCFS. Then, $W_j^i(s_j) = (T_j^i(s_j) + l_j)^+$ represents the random time interval between the moment of arrival of an arbitrary demand for product $i$, and the moment that the component of type $j$ used to satisfy that demand is available. We emphasize that $\{W_j^i(k)\mid j \in J^i, k \in \{0, 1, \ldots\}\}$ is a collection of dependent random variables, because each variable in the collection is defined with respect to the same (arbitrary) demand for product $i$. In Figure 1, for example, $T_j^i(s_{j'}) = -4$ and $l_{j'} = 8.5$, so $W_j^i(s_{j'}) = (-4 + 8.5)^+ = 4.5$.

In light of the above, and because the product demand arriving at $t = 0$ is fulfilled when all components are available, we have the key relation

$$W^i = W^i(\{s_j\mid j \in J^i\}) = \max_{j \in J^i} W_j^i(s_j). \tag{3}$$

This relation allows us to obtain an expression for the cost rate (2) under FCFS allocation, which resembles a two-stage SP formulation:

$$C(\bar{s}) := \sum_{j \in J} h_j s_j - D + \sum_{i \in I} \lambda^i \bar{B} \max_{j \in J^i} W_j^i(s_j). \tag{4}$$

We used that $B^i = \lambda^i W^i$ by Little’s formula (1961).

### 3.2.2 The sampling approximation algorithm

We develop a sampling average approximation (SAA) algorithm to solve the optimization problem $\min_{s_j \in S_j} C(\bar{s})$. Many authors have used similar techniques to solve stochastic programs (cf. Birge and Louveaux 1997).

Samples are generated randomly, by drawing for every product $i$ a number $|N^i|$ of scenarios $\xi_n^i$, $n \in N^i$ with associated weight $p_n^i$. Each $\xi_n^i$ contains information regarding the waiting time corresponding to a demand arrival for product $i$. In particular, $\xi_n^i$ yields $\{W_j^i(s)\mid s \in S_j\}$, where $s_j$ is an upper bound on the optimal base-stock level for component $j$ (see Section 3.2.3). Samples should be drawn from a distribution that satisfies:

$$\mathbf{E} W^i(\{s_j\mid j \in J^i\}) = \mathbf{E} \sum_{n \in N^i} p_n^i \max_{j \in J^i} W_j^i(s_j)(\xi_n^i) \quad \forall i, \forall s_j \in S_j|j \in J^i. \tag{5}$$
We discuss how to generate samples in Appendix A.

Our approach for minimizing (4) will be motivated by the following result, which is a specialization of general results that have been known for quite a while (Mak et al. 1999, Kleywegt et al. 2001). (Proofs of all propositions are in Appendix B.)

**Proposition 1.** Let $C_a(\vec{s}) = \sum_{j \in J} h_j s_j - D + \sum_{i \in I} \tilde{\beta}^i \sum_{n \in N^i} p_n^i \max_{j \in J^i} W_j^i(s_j)(\xi_n^i)$.

1. When samples are drawn to satisfy (5), then
   $$\mathbb{E} \left( \min_{s_j \in S_j | j \in J} C_a(\vec{s}) \right) \leq \min_{s_j \in S_j | j \in J} C(\vec{s})$$  \hspace{1cm} (6)

2. If in addition scenarios are drawn independently and weights satisfy $p_n^i = 1/|N^i|$, then
   $$\arg \min_{s_j \in S_j | j \in J} C_a(\vec{s}) \subseteq \arg \min_{s_j \in S_j | j \in J} C(\vec{s})$$  \hspace{1cm} (7)

with probability 1, as the number of scenarios in the sample grows large for each product.

Part 1 says that solution values based on our sampling approach are in expectation lower bounds to the optimal cost. Part 2 implies that solutions to $\min_{s_j \in S_j | j \in J} C(\vec{s})$ for sufficiently large samples are likely to be high quality solutions to (1) under FCFS. In fact, it is possible to show that the probability to obtain a solution to $\min_{s_j \in S_j | j \in J} C(\vec{s})$ that is non-optimal by more than $\delta$ for the original problem $\min_{s_j \in S_j | j \in J} C(\vec{s})$ decays exponentially in the number of samples, providing theoretical support for the quality of the bound (cf. Kleywegt et al. (2001)).

Of course, a question pertaining to these results is whether we are able to compute solutions for samples consisting of “sufficiently many scenarios.” Our two primary methodological contributions allow us, for the first time, to answer this question in the affirmative. Our novel SP formulation (Section 3.2.1), combined with our use of indicator variables for base-stock levels (below), yield an MIP formulation having a strong LP relaxation that scales efficiently to very large problems. (Existing SP formulations use the base-stock levels directly as decision variables, giving rise to weak LP relaxations having severe scalability issues.) Combined with the theoretical results in Proposition 1, this allows us to answer the first question set out in the introduction in the affirmative.

Specifically, we formulate $\min_{s_j \in S_j | j \in J} C_a(\vec{s})$ as follows, where $x_{js} = 1$ implies $s_j = s$:

$$\min \sum_{j \in J} \sum_{s \in S_j} h_j s x_{js} - D + \sum_{i \in I} \tilde{\beta}^i \sum_{n \in N^i} p_n^i v_n^i,$$  \hspace{1cm} (8)

$$v_n^i \geq \sum_{s \in S_j} x_{js} W_j^i(s)(\xi_n^i), \hspace{1cm} i \in I, j \in J^i, n \in N^i,$$  \hspace{1cm} (9)

$$\sum_{s \in S_j} x_{js} = 1, \hspace{1cm} j \in J,$$  \hspace{1cm} (10)

$$v_n^i \geq 0, \hspace{1cm} i \in I, n \in N^i,$$  \hspace{1cm} (11)

$$x_{js} \in \{0, 1\}, \hspace{1cm} j \in J, s \in S_j.$$  \hspace{1cm} (12)

We need (10) because precisely 1 base-stock level per component must be selected. The auxiliary real-valued decision variables $v_n^i$ represent the waiting time incurred for sample $\xi_n^i$. Indeed, they take the minimum value allowed by (9), which equals $\max_{j \in J^i} W_j^i(s_j)(\xi_j^i)$ by the interpretation of $x_{js}$. 
It may appear that the use of indicator variables makes the problem more complex, but in fact it simplifies the solution process. For the modest price of adding $50 - 100$ variables for each component (independent of the number of scenarios in the sample), we linearize the constraints (9). This trade-off is very beneficial because MIP solvers scale to very large problems as long as the LP relaxation is sufficiently strong, and the LP relaxation for this (linearized) problem is indeed very strong. We found that the root-node gap is typically much less than 0.5%. (Theoretically, one may show that, for sufficiently many samples, the integrality gap in any node is bounded by $\sum_{j \in J} h_j$; We omit the proof for brevity.) Our SAA algorithm finds solutions by taking the component-wise average of the solution $\bar{s}$ of (8-12) for a number of samples, and rounding the resulting base-stock levels to the nearest integer. The objective value corresponding to this solution is estimated in an independent simulation run. To assess whether this solution is close-to-optimal, Part 1 of Proposition 1 is used: the mean and variance of objective functions for many independent samples are used to construct a 99.7% confidence interval for the lower bound (6), our “asymptotic lower bound.” Because of the strength of the root-node relaxation, no branching is needed to obtain the tight asymptotic lower bounds we report in our computational experiments in Section 4.

3.2.3 The ISS base-stock levels

If we ignore simultaneous stock-outs (ISS), then at most one of the random variables $W_i^j(s_j), j \in J_i$ will be nonzero for every realization. So (3) will be approximated as $W = \max_j W_i^j(s_j) \approx \sum_j W_i^j(s_j)$. Substituting into (4) and rearranging gives

$$C(\bar{s}) \approx C_{\text{iss}}(\bar{s}) = -D + \sum_{j \in J} h_j s_j + \sum_{i \in I_j} \lambda_i b_i E W_i^j(s_j).$$

(13)

$C_{\text{iss}}(\bar{s})$ separates into the sum of newsvendor type cost functions that depend on only a single base-stock level. The (greatest) base-stock levels that minimize $C_{\text{iss}}(\bar{s})$ will be referred to as the ISS base-stock levels. This solution was introduced by Lu and Song (2005). We will denote the ISS base-stock levels by $s^u_j, j \in J$ because they are upper bounds on the optimal base-stock levels under FCFS, as summarized in the following proposition. ($s_1 \lor s_2$ and $s_1 \land s_2$ denote the componentwise maximum and minimum of two vectors $\bar{s}_1$ and $\bar{s}_2$, respectively.)

**Proposition 2.** Under FCFS allocation,

1. $E W^i(\bar{s})$ and $C(\bar{s})$ are submodular in the base-stock levels $s_j$: For any $\bar{s}_1$ and $\bar{s}_2$, it holds that $E W^i(\bar{s}_1) + E W^i(\bar{s}_2) \geq E W^i(\bar{s}_1 \land \bar{s}_2) + E W^i(\bar{s}_1 \lor \bar{s}_2)$, and similarly for $C(\cdot)$.

2. Each $s^u_j$ is an upper bound on the corresponding optimal base-stock level.

Lu and Song (2005) prove a similar result, and discuss an interesting economic interpretation of submodularity of the cost function: Inventories of components are complementary. Our proof is different; we include it because it enables extensions such as non-unit usage of components in products (see Proposition 3). Lu and Song (2005) discuss that extending their method of proof to non-unit demand is difficult.

3.2.4 Extensions

The following proposition shows that results in this section can be extended to a (much) more general setting.
Proposition 3. The cost formulation (2), the key relation (3), the SP formulation (4), the sample approximation (8-12), and Proposition 2 can be extended to incorporate the following assumptions:

- Non-unit (possibly stochastic) requirements of components in products.
- Make-to-stock, or stochastically sequential lead times (Svoronos and Zipkin 1991).
- Exogenous batching of resupply orders, i.e. \((r,q)\) policies with \(q\) exogenous and \(r \geq -q\).

The results also extend when the costs \(\tilde{b}^iEB^i\) in (2) are replaced by \(Ec^i(W^i)\), for any non-decreasing function \(c^i(\cdot)\). This allows the results to be extended, for example, to time window fill-rate penalties.

3.3 A lower bound on the costs under optimal allocation

The purpose of this section is to develop an asymptotic SP lower bound on the costs of the optimal base-stock levels under optimal allocation (as opposed to the lower bound under FCFS allocation discussed in Section 3.2.2). The SP lower bound propose corresponds to minimizing the (expected) cost rate (1) incurred at a pre-specified moment in time, without taking into consideration the cost-rate before or after that point, instead of minimizing the average cost rate over time. Doğru et al. (2010) and Reiman and Wang (2012) use this idea to also derive a lower bound. The SP we develop differs from their SPs because our SP is two-stage, even for cases in which different components have different lead times, and because our SP restricts attention to base-stock policies. Obtaining a two-stage SP is important for our purposes, because multi-stage SPs are typically very challenging computationally. Additionally, we increase computational efficiency by using a different formulation that halves the number of second-stage decision variables.

Slightly abusing the notation introduced in Section 3.1, we define the following:

- \(D^i(t)\), for \(t > -\bar{l}\): Random demand for products of type \(i\) in period \((-\bar{l}, t]\). For convenience, let \(D^i_j(t) := \sum_{i \in I^i_j} D^i(t)\).
- \(B^i(t)\): product \(i\) back-orders at \(t\).
- \(H^j_0\): on hand component \(j\) inventory at \(t = 0\).
- \(z^i\): Total product demands of type \(i\) satisfied during \((-\bar{l}, 0]\). This may include demands that arrived before \(-\bar{l}\).

We have the following relation:

\[
H^j_0 = s^j + \sum_{i \in I} B^i(-\bar{l}) + D^i(-l^i_j) - \sum_{i \in I^i_j} z^i \geq 0. \tag{14}
\]

This relation is valid because at \(-\bar{l}\), the inventory position of component type \(j\) is \(s^j\), while back-orders equal \(\sum_{i \in I^i_j} B^i(-\bar{l})\). (Because back-orders are subtracted from the inventory position, additional inventory on-hand or on order is kept as a consequence of these back-orders.) Also, “demand” \(D^i_j(-l^i_j)\) arriving between \(-\bar{l}\) and \(-l^i_j\) results in additional purchase orders that arrive before 0. Finally, any satisfied product demands \(z^i\) of type \(i \in I^i_j\) result in withdrawals of type \(j\) inventory. For product \(i\) back-orders at time 0 we have:

\[
B^i(0) = B^i(-\bar{l}) + D^i(0) - z^i \geq 0. \tag{15}
\]
For our SP, we take into account the constraints (14) and (15) that must be satisfied by any base-stock policy under any allocation policy. Thus, the cost incurred at 0 depends on the random variables \( D_j(-l_j) \) and \( D^i(0) \). We denote a realization of these random variables (scenario) by \( \xi \). Then, using (2), we find the following two-stage SP for cost minimization at \( t := 0 \):

\[
\min_{s_j \geq 0, j \in J} \sum_{j \in J} h_j s_j - D + EC_{SP}(\bar{s}, \xi).
\]

The second stage costs \( C_{SP}(\bar{s}, \xi) \) are expressed in the decision variables \( x^i := B^i(0) \) and \( B^i(-l) \) as:

\[
C_{SP}(\bar{s}, \xi) = \min \sum_{i \in I} \tilde{b}^i x^i,
\]

\[
\text{s.t. } \sum_{i \in I} x^i + s_j \geq D_j(0)(\xi) - D_j(-l_j)(\xi),
\]

\[
0 \leq x^i \leq D^i(0)(\xi) + B^i(-l).
\]

Here, (18) and (19) correspond to (14) and (15), respectively. The minimization is over \( x^i \) and \( B^i(-l) \). Clearly, since \( B^i(-l) \) appears only in (19), setting \( B^i(-l) = \infty \) relaxes (19) without affecting the objective function. So only \( x^i \) remains as a second stage decision variable.

Because (1) is the average cost rate over time, while (16) minimizes the costs at one point in time, (16) constitutes a lower bound on the cost rate (1) under optimal allocation. This lower bound can in general not be attained by any feasible allocation policy for (1).

To find a lower bound to (16), with corresponding asymptotic confidence interval, we use a sampling-based approach, similar to the approach described in Section 3.2.2. Details are available from the authors upon request.

4 Results

In this section we use the algorithms developed in Section 3 to investigate the performance of heuristic resupply and allocation rules commonly applied in practice to industrial-scale ATO systems. We focus on the ISS heuristic to optimize the base-stock levels, and the FCFS heuristic for allocation, but for comparison purposes we also investigate other heuristics. We first summarize the different policies that will be investigated in Section 4.1, and the lower bounds that will be investigated in Section 4.2; we then give the performance of these policies from experiments with different ATO systems in Sections 4.3-4.5. The results give rise to a question about asymptotic performance; we answer this question in Section 4.6. We summarize results and discuss the managerial insights gained though our study in Section 4.7.

Because computation times are quite a different concern from the performance of the policies, we first benchmark the scalability of the SAA algorithm developed in Section 3.2 (vJ&S-W) against the algorithms proposed by Akçay and Xu (2004) (A&X) and Huang and De Kok (2011) (H&dK). In order to test for scalability, we generate test cases with various numbers of components, keeping the number of products equal to the number of components. As the three algorithms are based on different modeling assumptions, we generate test cases that can be interpreted as instances of the various models. Furthermore, due to the different modeling (and cost) assumptions, we do not compare the policies on cost.

For each test case, we determine the amount of computation time that is needed to obtain a solution of good quality using the different algorithms (i.e. within 1% of the corresponding
Table 1: The computation times of the tested algorithms (vJ&S-W, A&X, H&dK) to get within 1% of the asymptotic lower bound for test problems with various number of components $|J|$. We choose $|I| = |J|$ for all test problems. A “-” indicates that computation time exceeds 100 hours.

| Size $|J|$ of test case | Computation time (Hrs.) |
|-----------------------|-------------------------|
|                       | vJ&S-W | A&X | H&dK |
| 5                     | 0.01   | 0.11 | 1.2  |
| 10                    | 0.03   | 0.11 | 4.5  |
| 15                    | 0.04   | 1.5  | 40.1 |
| 20                    | 0.07   | 88.8 | -    |
| 100                   | 0.4    | -    | -    |
| 200                   | 0.88   | -    | -    |
| 300                   | 1.45   | -    | -    |

Table 1: The computation times of the tested algorithms (vJ&S-W, A&X, H&dK) to get within 1% of the asymptotic lower bound for test problems with various number of components $|J|$. We choose $|I| = |J|$ for all test problems. A “-” indicates that computation time exceeds 100 hours.

Asymptotic lower bound). For details, see Appendix C.

Table 1 shows that our algorithm computes near-optimal base-stock levels for systems of hundreds of products and components within a 1.5 hours, while the computation times of the benchmarks become prohibitive as problem size increases above 15-20 components. The main reason for the difference in scalability is that our sample approximation has strong linear relaxations, while the root node optimality gap of the benchmark algorithms is typically more than 50% from the best integer solution, leading to a huge branch and bound trees. We emphasize that this performance of our SAA algorithm is key in establishing the results in the remainder of this section.

4.1 The investigated policies

In our numerical experiments we compare the performance of eight heuristics, each arising by pairing a heuristic to set the base-stock levels with a heuristic for allocating components to product demands. The performance of these heuristics will be compared with the asymptotic lower bounds developed in Sections 3.2.2 and 3.3.

For allocation heuristics, our focus will be on FCFS. But since FCFS is not the only simple allocation rule that is applied in practice, and because other scholars have found promising results for simple no-holdback allocation rules in special cases (e.g. Song and Zhao 2009, Doğru et al. 2010, Lu et al. 2010), we will also investigate the performance of selected no-holdback rules. We test two different no-holdback rules that differ in the method by which back-orders are cleared: The first-ready first-serve (FRFS) allocation rule clears back-orders first-come first-serve, and the no-holdback with priority clearing (NHB-PR) rule clears back-orders in order of decreasing modified penalty costs $\tilde{b}$ (see (2)).

For setting the base-stock levels, our focus will be on the ISS base-stock levels defined in Section 3.2.3, and the base-stock levels computed using the SAA algorithm developed in Section 3.2.2. These latter base-stock levels are based on an SP formulation that is exact under FCFS. Hence, they will be referred to as SP-FC base-stock levels. We test one additional heuristic to determine the base-stock levels, motivated by Doğru et al. (2010): For the equal lead time case, they prove that the base-stock levels attaining the minimum in (16) are optimal under any no-holdback allocation rule for the W-system under cost symmetry. They also find promising numerical results for the same model under cost asymmetry. Determining the optimal solution to (16) appears to be intractable for the systems we consider. Therefore, we test the base-stock levels obtained by solving (16) using the SAA algorithm proposed in Section 3.3. These base-stock levels will be referred to
Table 2: The various heuristics tested in our experiments are obtained by combining a heuristic to set the base-stock levels and a heuristic to allocate components to demands.

Combining the three heuristics for setting the base-stock levels and three allocation heuristics gives us nine different policies, of which eight will be tested, as summarized in Table 2. (We discard drw-fc because they are dominated by the spfc-fc policy for all investigated cases.) The average cost rate (1) for the obtained solution for each policy will be estimated using simulation.

### 4.2 The investigated lower bounds

Clearly, estimators of the cost rates of the policies in Table 2 alone give only limited insight into the performance of these policies, because it is unclear what performance can be hoped for. For that reason, we also developed asymptotic lower bounds in Section 3:

- **lb-fc**: An asymptotic lower bound on the best cost rate (1) that can be attained under FCFS allocation of components to products and base-stock policies (Section 3.2.2 and Proposition 1).

- **lb-opt**: An asymptotic lower bound on the optimal objective function (16), which constitutes a lower bound on the best cost rate (1) that can be attained under optimal allocation of components to products, and base-stock policies (Section 3.3).

By comparing the performance of our eight policies with lb-fc and lb-opt, we are able to provide much more insightful results on the performance of these policies.

We report the relative difference between the performance of different policies; reporting the relative difference with lb-fc is particularly insightful: For the FCFS policies (iss-fc and saa-fc) such relative differences give us an upper bound on how much we lose by applying them instead of the optimal FCFS policy, allowing us to answer the first two questions set out in the introduction. For the non-FCFS policies these relative differences tell us how much we gain (or lose) by applying the heuristic allocation policy instead of the best FCFS policy, which is a relevant benchmark for heuristic allocation policies because of FCFS’s prevalence in practice. And the relative difference between lb-opt and lb-fc gives us an upper bound on how much we can gain by applying optimal allocation instead of FCFS. So the relative performance of the non-FCFS policies and lb-opt in comparison to lb-fc allows us to answer the third question set out in our introduction.

We will denote these relative differences with lb-fc by adding a % symbol, e.g. for iss-fc:

\[
\text{iss-fc\%} := \frac{\text{iss-fc} - \text{lb-fc}}{\text{lb-fc}} \times 100%.
\]

(20)

We base our asymptotic normal estimators of policy performance (e.g. iss-fc) and lower bounds (lb-fc, lb-opt) on many samples and realizations (e.g. 100 samples of \(350 \times I\) realizations for lb-fc), making it reasonable to assume normality. This allows us to compute an estimator for iss-fc\% (and the other relative differences) that can be safely treated as a normal estimator, because the coefficients of variation of our estimators for lb-fc are small (Hayya et al. 1975). To convey
Figure 2: Performance of iss-fc and spfc-fc for PC assembly, under different demand levels and back-order asymmetry. Bars denote 99.7\% asymptotic confidence intervals.

information about the variance of these estimators, we give 99.7\% asymptotic confidence intervals around our point estimators when presenting our results.

4.3 PC assembly case

We first test the performance of the different algorithms on Test Problem 2 in Akçay and Xu (2004), a PC assembly system with realistic problem data from the IBM personal systems group. The case consists of 17 components and 6 products. We use the bill of material (BOM) data \{J^i | i \in \mathcal{I}\}, lead times and holding costs as given in the online appendix of Akçay and Xu (2004). Like Akçay and Xu (2004), we assume the same demand distribution for all products \(\forall i: \lambda^i = \lambda\).

For setting the back-order penalties, we use the well-known concept of newsvendor (NV) fractiles \(\delta_b\) to modulate the average fractiles and back-order cost asymmetry, respectively. To this end, we index the products \(i \in \{1, \ldots, 6\} = \mathcal{I}\), where the index is increasing in \(h^i\). We set the product penalty costs for product \(i\) equal to \(b^i/(h^i + b^i)\) for product \(i\). (Recall that \(h^i := \sum_{j \in \mathcal{J}^i} h_j\) corresponds to the “holding costs” for product \(i\).) We use two parameters, \(f\) and \(\delta_b\), to modulate the average fractiles and back-order cost asymmetry, respectively. To this end, we index the products \(i \in \{1, \ldots, 6\} = \mathcal{I}\), where the index is increasing in \(h^i\). We set the product penalty costs for product \(i\) equal to \(b^i/(f/(1-f))h^i(1+x^i)\), where \(\{x^1, \ldots, x^6\} = \{-0.55\delta_b, -0.33\delta_b, -0.16\delta_b, 0.1\delta_b, 0.3\delta_b, 0.5\delta_b\}\). As \(\delta_b\) increases, products that are more expensive to produce will have a higher NV fractile. Penalty asymmetry is quite sensitive to \(\delta_b\): e.g. when \(\delta_b = 0.5\) and \(f = 0.9\) penalty costs are \(\sim \{6.7h^1, 7.6h^2, 8.5h^3, 9.4h^4, 10.3h^5, 11.2h^6\}\). Individual penalty costs thus differ by more than 100\% since \(h^6 = 1.3h^1\) (Akçay and Xu 2004).

For FCFS policies, we vary \(\lambda \in \{0.5, 2.0, 8.0\}\), \(\delta_b \in \{0, 0.5, 1\}\), and \(f \in \{0.8, 0.85, 0.9, 0.95, 0.97, 0.99\}\), to obtain a test bed of 54 parameter settings. Results are depicted in Figure 2. Results for the intermediate values \(\lambda = 2.0\) and \(\delta_b = 0.5\) are omitted because they are in line with the results reported in the figure. The figure shows that SAA base-stock levels are near-optimal (within 0.5\% of LB) among FCFS for all considered cases, while ISS performs well for high NV fractiles, but its performance deteriorates as the NV fractiles decrease. In addition, the figure shows that these results are largely insensitive to changes in demand rate \(\lambda\) and back-order cost asymmetry \(\delta_b\).

Results for non-FCFS allocation policies under symmetric (\(\delta_b = 0\)) and asymmetric (\(\delta_b = 1\)) back-order costs are depicted in Figures 3 and 4, respectively. We report results for \(\lambda = 2\), which corresponds to a coefficient of variation of lead time demand of about \(1/\sqrt{20} \approx 22\%\) for lead times.

\(^3\)To improve legibility of the confidence intervals, we slightly disalign the \(x\) coordinates of some plot markers in some figures with the value of \(f\) to which they correspond.
of 10 days, which are typical in this system. The results for other demand rates are quite similar.

Figure 3 shows that for small back-order asymmetry, using FCFS constitutes a limited optimality loss - about 4% when \( f = 0.8 \) - which decreases as \( f \) increases. (Recall that the spfc-fc solutions were very close to lb-fc, which is about 4% above lb-opt.) Furthermore, NHB policies with SPFC or DRW base-stock levels practically attain the lower bound, implying that these policies are near optimal for this case. Finally, the ISS base-stock levels have inferior performance for this problem under non-FCFS allocation, especially if \( f \) is small.

For larger back-order asymmetry, Figure 4 shows that the optimality loss of using FCFS increases, but remains rather limited considering that cost asymmetry is quite significant. Not surprisingly, we also see that prioritized backlog clearing (i.e. drw-pr and spfc-pr) now becomes important.
Figure 5: The performance of different policies for the repair shop case, for asymmetric penalty costs ($\delta b = 1$). Bars denote 99.7% asymptotic confidence intervals.

4.4 Maintenance Organisation

In the introduction we discussed that many maintenance organizations face a problem that bears similarities to an ATO problem, with maintenance tasks playing the role of products and spare parts playing the role of components. In this section we will test the performance of the different policies on a specific problem encountered during a project at a repair shop owned by Fokker Services (van Jaarsveld et al. 2011), with characteristics that are typical for the maintenance industry.

At the repair shop, maintenance tasks are carried out on different types of equipment that arrive at the shop over time. After initial inspection, defective parts are replaced by spare parts, which are purchased from vendors. Once all defective spare parts are replaced, the equipment is sent back to the customer. Customers expect short repair turnaround times, but spare part lead times may be significant, so a local inventory of spare parts is kept at the repair shop. An important difference between the ATO system considered in Section 4.3 and the problem considered here is that different repairs of the same type may use different spare parts, because maintenance is carried out by replacing (only) the defective spare parts.

We conduct our tests on a small case based on data from Fokker Services. The case consists of three repair types ($a, b$ and $c$) and 110 spare parts. Usage probabilities of spare parts in each repair type, repair type arrival rates, as well as spare part lead times and holding costs are given in Appendix D. As in section 4.3, the penalty costs of each repair type will depend on the NV fractile $f$ and NV asymmetry $\delta b$. For each repair type $i \in a, b, c$, we extend the definition of $h^i$ to take into account the usage probabilities $p^i_j$: $h^i := \sum_{j=1}^{10} p^i_j h_j$, which gives $h^a = 300, h^b = 289$ and $h^c = 268$. We then define $b^i = h^i(f/(1 - f))(1 + x^i)$, where $x^a = 0.5\delta b, x^b = 0, x^c = -0.5\delta b$.

The performance of the different policies for asymmetric penalty costs ($\delta b = 1$) is given in Figure 5. (Results for the $\delta b = 0$ case are omitted because they are similar.) The performance of $drw-pr$ is quite poor (+50-100%), and the lower bound $lb-opt\%$ is as low as $-60\%$. Therefore, these values were plotted in a separate figure to increase legibility: Figure 6. Figure 5 shows that the $spfc-fc$ policy is near-optimal among the class of FCFS base-stock policies for all tested NV fractiles, while the $iss-fc$ policy is near-optimal for FCFS for high NV fractiles, and only slightly non-optimal as $f$ decreases to 0.8. The figures also show that FRFS and NHB-PR policies $iss-fr$, $spfc-fr$, $iss-pr$ and $spfc-pr$ outperform their FCFS counterparts by 1-5%. Differences for $\delta b = 0$ were somewhat less.
Figure 6: The lower bound on optimal allocation $lb-opt$ for the repair shop case for different values of $f$ and $\delta b$. The 99.7\% asymptotic confidence intervals do not exceed the size of the markers.

For low NV fractiles, none of the policies come close to $lb-opt$ in Figure 6. Only for high NV fractiles can we conclude that the considered policies perform at least reasonably well when compared with the best base-stock policy under optimal allocation.

However, there is no (theoretical) guarantee $lb-opt$ can be attained by any feasible policy. In fact, a careful examination of the analysis in Section 3.3 reveals that $lb-opt$ can back-order any component repair to deal with a spare part shortage at the moment in time, $t = 0$, for which the cost rate is minimized under $lb-opt$, including products that were not demanded in $(-\bar{I}, 0]$. This is accomplished by letting $B_i(-\bar{I}) \rightarrow \infty$: A large number of back-orders of all products is kept at $-\bar{I}$. In the repair shop case, since repairs (products) use spare parts (components) with a certain probability, every combination of spare parts may be kept as a back-order at $-\bar{I}$, which gives the lower bound many degrees of freedom to deal with component shortages. We believe that this explains why the lower bound $lb-opt$ is weak for this case. We have observed weak lower bounds $lb-opt$ for other (unreported) cases where the effective number of demand types is very large. Thus, it is at least plausible that the examined policies perform relatively close to optimal.

4.5 Assembly of products of multiple families

Many OEMs divide their products into product families (e.g. medical equipment or wafer steppers, see De Kok (2003)). Products in each family are assembled from group-specific expensive components, which may be combined with relatively inexpensive components that are common over all groups. In this section, we investigate the performance of the developed policies on test problems developed along these lines. Instead of using data from a company as in Sections 4.3 and 4.4, the test problems in this section are randomly generated with certain characteristics. This allows us to investigate the effect of BOM structure and other problem aspects on policy performance.

We consider an ATO system consisting of 3 product families, each consisting of 12 products. Each product family has 8 product-family specific components that are only used in that family. There are also 20 components that are common to all families. Each product in each family is assembled from $n_s \in \{1, \ldots, 8\}$ family specific components, chosen at random from the 8 components that are specific for that product family. In addition, each product uses $n_c = 5$ common components, chosen at random from the 20 common components. Component lead times are chosen.
uniformly on \([1.0 - \delta l, 1.0 + \delta l]\), and holding costs for common components are chosen uniformly on \([0.5, 1.5]\). Holding costs for family specific components are higher: They are chosen uniformly on \([2.5, 7.5]\). Product demand rates \(\lambda^i\) are 8, giving a coefficient of variation of product demand during a typical component lead time of about 35%.

To set product penalties, we again use the newsvendor fractile \(f\) and penalty asymmetry \(\delta b\). We explore two different types of penalty asymmetry: 1) To test penalty asymmetry within product families, we set \(x^i\) for each product as \(-0.5\delta b\) (low criticality), 0 (medium criticality) or \(0.5\delta b\) (high criticality) with equal probability; 2) to test penalty asymmetry between product families, we set \(x^i = -0.5\delta b\) for products in the first family, \(x^i = 0.0\) for products in the second family, and \(x^i = 0.5\delta b\) for products in the third family. We then define \(b^i = h^i(f/(1 - f))(1 + x^i)\). We remark that while some parameters in this problem design have been chosen somewhat arbitrarily, additional experiments have shown that results are qualitatively insensitive to our choices.

Figures 7 and 8 show the performance of iss-fc and spfc-fc for the problem, respectively. We vary \(n_s\), \(\delta l\) and \(f\), and fix \(\delta b = 0.0\) (no penalty asymmetry), so all NV fractiles are exactly \(f\). Figure 7 shows that as \(n_s\) increases, the performance of iss-fc degrades significantly, especially for low NV fractiles \(f\) and lead time asymmetry \(\delta l\). We excluded from the figure the extreme value \(n_s = 8\); for that case with \(\delta l = 0\), iss-fc% increases to 33% for \(f = 0.8\), to 19% for \(f = 0.95\), and to 11% for \(f = 0.99\). The poor performance of iss-fc for some cases is in sharp contrast with the performance of spfc-fc: the latter is within 0.5% of optimality for all cases, as shown in Figure 8.

In Figures 9 and 10 we report experiments for cases with penalty asymmetry between product families, and within product families, respectively. We fix \(n_s = 6\) and \(\delta l = 1.0\). We use \(\delta b = 1.0\), which corresponds to significant asymmetry: the high criticality products have a 3 times higher penalty cost than the low criticality products. By comparing Figures 9 and 10, it becomes clear that asymmetry within product families harms FCFS performance significantly more relative to the no-holdback policies spfc-pr and drw-pr and the lower bound lb-opt, than asymmetry between different product families. This is to be expected, because there is more competition for common and expensive components when there is asymmetry within families. We discuss this, and other insights, in Section 4.7.
Figure 8: The performance of spfc-fc for the assembly case, for different values of \( n_s \) and \( \delta l \). The bars denoting 99.7% asymptotic confidence intervals appear large because of the scale of the figure.

Figure 9: The performance of various heuristics for the assembly case compared to lb-fc and lb-opt for penalty asymmetry between product families. The bars denote 99.7% asymptotic confidence intervals.

4.6 Asymptotic Results

For all cases studied in this section, we found that the cost of the ISS base-stock levels is relatively close to the cost of the optimal base-stock levels under FCFS, for higher NV fractiles. Intuitively, this is sensible because if the number of stock-outs decreases, then the number of simultaneous stock-outs will typically decrease even faster, so that they can be safely ignored. A less intuitive observation is that the relative cost benefit of optimal allocation compared to FCFS also decreases with increasing NV fractiles.

Because this behavior was observed for all cases, we ask the question whether this result holds in general. The following proposition answers this question positively.

**Proposition 4.** Consider a range of ATO problems indexed by a parameter determining the back-order cost, denoted by \( p \in \{1, 2, \ldots\} \): Back-order costs are given by \( b^p_i = pb^1_i \), with \( b^1_i > 0 \). Other problem characteristics can be arbitrary, and remain fixed for all \( p \). Denote the cost of iss-fc for problem \( p \) by \( C^u_p \), the cost of the optimal FCFS policy for problem \( p \) by \( C^f_p \), and the cost of the optimal replenishment policy under optimal allocation by \( C^*_p \).

1. The iss-fc policy is asymptotically optimal for FCFS as \( p \to \infty \): \( (C^u_p - C^f_p)/C^f_p \to 0 \).
2. The iss-fc policy is asymptotically optimal as \( p \to \infty \): \( (C^u_p - C^*_p)/C^*_p \to 0 \).

Note that \( C^*_p \) is the cost of the optimal replenishment policy under optimal allocation. Thus, this proposition leads to the following corollary:

**Corollary 5.** Consider a range of ATO problems as in Proposition 4. Then FCFS allocation and independent base-stock control are asymptotically optimal as \( p \to \infty \).

### 4.7 Discussion

We first discuss the results for FCFS ATO systems: The ISS performance, iss-fc, is most strongly influenced by the NV fractiles, component demand correlation, and similarity of lead times. Figure 7 shows this dependence best: When \( n_s \) approaches 8, demand correlation between different family-specific components in the same family increases significantly (because they are almost always used together), which causes poor performance of iss-fc. Performance degrades further if all lead times are the same (\( \delta_l = 0 \)), causing even larger lead time demand correlation. Similarly, for the PC-assembly case, we observe that there is significant demand correlation between a number of components (e-appendix of Akçay and Xu 2004). For instance, component 1, 2, 3, and 11 are always used together, and component 2, 3, and 11 all have a lead time of 8. This explains the mediocre performance of iss-fc observed in Figure 2. Finally, because of the low demand probabilities of spare parts the maintenance case (see Appendix D), demand correlation is very small, explaining the excellent performance of iss-fc observed in Figure 5. So, it appears safe to use ISS as long as NV fractiles are high, or if demand correlation is low. But for lower NV fractiles and higher demand correlation, it is not safe to ignore simultaneous stockouts. Thus in these cases, it is crucial to use the spfc base-stock levels, as their performance appears to be close to the optimal FCFS policy (within 0.5% of lb-fc) for all considered experiments.

We now discuss non-FCFS policies, and the performance of FCFS policies when compared to optimal allocation. FCFS performs well compared to optimal allocation when NV fractiles are in the higher range, and the back-order cost asymmetry is limited, as in these cases the benefits of prioritization are inherently lower. The maintenance organization case is possibly an exception to this rule, but this may also be due to the fact that lb-opt is weak for that case, as discussed in Section 4.4. With decreasing NV fractiles, FCFS performance deteriorates to a limited extent, even
Table 3: A summary of the influence of average NV fractiles and other important problem parameters on the performance of various allocation policies and methods for setting the base-stock levels.

when NV fractiles are symmetric. For these cases, switching to saa-fr can be an easy win of 2-4%, considering that FRFS should be easily implementable in practice.

For asymmetric cost cases, the performance of FCFS compared to the lower bound for optimal allocation deteriorates. However, deterioration is rather limited, even if penalty costs for different products differ by a factor of 2-3. Typical loss of optimality is $3 - 5\%$ for $f = 0.99$, increasing to $8 - 16\%$ for $f = 0.8$. For such cases, NHB-PR allocation combined with the spfc or drw base-stock levels can significantly outperform FCFS, and may be an attractive alternative to FCFS in practice because, like FCFS, it is easy to implement.

We emphasize that for all cases, for both FRFS and NHB-PR, we have identified base-stock levels which outperform our asymptotic lower bound $lb-fc$ on the performance of the best possible FCFS policy. This implies that, at least for the cases considered, the performance of these no-holdback policies is superior to FCFS. Considering the wide range of systems for which we have run experiments, we argue that if there are no reasons to prefer FCFS over FRFS or NHB-PR (such as guaranteed maximum waiting times), then the latter policies should be preferred in practice. We summarize our insights in Table 3. Note that the results for FCFS vs $lb-opt$ are given separately for the repair shop case.

5 Conclusions and future research

We developed an algorithm for optimizing base-stock levels in realistically sized ATO systems with general system architectures under FCFS allocation; experiments suggest that it finds close-to-optimal base-stock levels. We used the algorithm to gain insights into the performance of ISS - a heuristic used by companies to determine base-stock levels - for a number of realistic practical examples. We found that its performance is excellent in many cases, but deteriorates for lower NV fractiles or high lead time demand correlations.

We also investigated the impact of FCFS allocation on the performance of ATO control policies. To this end, we developed a lower bound on the costs of optimal allocation. We found that FCFS performs surprisingly well for a number of realistic practical examples. We also found that heuristics based on no-holdback policies may outperform the best FCFS policies.
The investigated no-holdback policies were less effective for cases with high penalty asymmetry and low NV fractiles. Effective policies for these situations would likely involve rationing, and future research should thus investigate how to ration in large-scale systems. Also, because promising results regarding order synchronization have been found for special cases (Lu et al. 2012), the next logical step would be to investigate how to synchronize ordering for general ATO systems. We largely ignored this question in the present paper. Additionally, the open questions regarding FCFS performance in the repair shop case in Section 4.4 should be addressed. Furthermore, even though the SAA algorithm (8-12) solves rather sizable problems, problems in practice may be even larger, calling for even more efficient algorithms. Finally, as the developed SP methodology is promising because of its generality, it may be interesting to investigate whether similar SP techniques can be applied for more general supply chains.

References


P. Kampstra. Email communication, 2012. (Mr Kampstra is senior modality performance manager at the service parts supply chain of Philips Healthcare).


A Sample generation

By simulating the ATO system and inspecting $T_{ij}(k)$ upon arrival of a demand for product $i$, we can obtain realizations of the original random variable \{\$i_j(s)|j \in J^i, s \in S_j\}, giving us scenarios $\xi^n_i$. Because lead times are deterministic, a warm-up period of $l$ before taking samples is sufficient. Because experiments indicated that (significant) dependence adversely impacts the quality of the bounds, we skip $n$ orders for product $i$ between each scenario for that product. We use $n = \lceil 5 + \lambda_i l + 7\sqrt{\lambda_i l} \rceil$, which gives a probability of two drawings occurring within $l$ smaller than $10^{-9}$ for all $\lambda_i$ and $l$. (We cannot simply skip $l$ because of the inspection paradox.) A sample consisting of scenarios of the original random variable obviously satisfies (5).

However, when newsvendor fractiles are (moderately) high, most scenarios $\xi^n_i$ from the original distribution are boring: They give zero associated waiting time $W_i(\xi^n_i)$ for any solution $\bar{s}$ of reasonable quality. A small fraction of scenarios are interesting: It is more likely that they give a positive waiting time for reasonable solutions. As a consequence, we need a large number of scenarios from the original distribution to obtain good lower bounds. But solving the SAA for such large samples is time-consuming, especially for large systems. We propose a simple yet effective method to generate skewed samples, i.e. samples with a large fraction of interesting scenarios: 1) Generate a large number of scenarios; 2) Use a heuristic to divide the scenarios into interesting and boring scenarios; 3) (Randomly) select a majority of boring scenarios to drop; 4) Adapt the weight of the remaining boring scenarios, to ensure that condition (5) holds.

Regardless of the heuristic used in Step 2, this approach will generate samples that satisfy the necessary conditions for Proposition 1 to hold, so that the algorithm generates asymptotic lower and upper bounds that converge to the optimum as the sample size grows.

Of course, using a low quality heuristic in step 2 will not improve the quality of the samples. Our heuristic first determines base-stock levels with reasonable quality by solving (8)-(12) for a sample with equal weights. This is repeated for a small number (five) of independent samples; $\bar{s}$ is chosen as the component-wise minimum of the resulting solutions. (The solution values for these five samples are not used in any statistics.) We then generate skewed samples: A scenario $\xi$ is qualified as boring if and only if $W_i(\xi)(\bar{s}) = 0$. Samples generated in this manner give rise to superior lower and upper bounds, when compared with samples of the same size consisting of independent equal-weight realizations.

B Proof of propositions

In this appendix, we provide the proofs of the propositions in Sections 3 and 4.6.

Proof of Proposition 1. 1) See Mak et al. (1999, Theorem 1).
2) Let \( \bar{s}_1 \notin \arg \min C(\bar{s}) \). Then \( C(\bar{s}_1) = C(\bar{s}_2) + 2\epsilon \) with \( \epsilon > 0 \) for some \( \bar{s}_2 \). But this gives
\[
P[\bar{s}_1 \in \arg \min C_a(\bar{s})] = P[C_a(\bar{s}_1) \leq C_a(\bar{s}_2)] \leq P[|C_a(\bar{s}_1) - C(\bar{s}_1)| \geq \epsilon] + P[|C_a(\bar{s}_2) - C(\bar{s}_2)| \geq \epsilon],
\]
which approaches 0 as \( |N^i| \to \infty \) for all \( i \) by the weak law of large numbers. (Note that \( \mathbb{E} \max_{j \in J} W_j^i(s_j)(\xi^i) = \mathbb{E} \xi^i|\{s_j|j \in J\} \) by (5).) Because we need only consider finitely many \( \bar{s}_1 \notin \arg \min C(\bar{s}) \), the result follows (cf. Kleywegt et al. (2001)).

This proof extends to the sampling scheme proposed in Appendix A, because the Central Limit Theorem still guarantees \( P(|C_a(\bar{s}) - C(\bar{s})| > \epsilon) \to 0 \).

Proof of Proposition 2. 1) We prove a more general result, to facilitate the extensions in Proposition 3. We will prove that \( \mathbb{E} c^i(W_i^j) \) is submodular for any nondecreasing function \( c^i(\cdot) \). By the definition of \( T_j^i \), we know that \( W_j^i \) is non-increasing in \( s_j \), which implies for every scenario \( \xi \) that
\[
\forall s_j \leq s_j' : c^i \left( W_j^i(s_j)(\xi) \right) \geq c^i \left( W_j^i(s_j')(\xi) \right).
\]
In addition, (3) implies that
\[
c^i \left( W_i^j(\xi) \right) = \max_{j \in J^i} c^i \left( W_j^i(s_j)(\xi) \right) = -\min_{j \in J^i} -c^i \left( W_j^i(s_j)(\xi) \right).
\]
By Topkis (1998, example 2.6.2 f), (21) and (22) imply that \( -c^i \left( W_i^j(\xi) \right) \) is supermodular in the base-stock levels, and hence \( c^i \left( W_i^j(\xi) \right) \) submodular. Because taking expectations preserves submodularity, \( \mathbb{E} c^i(W_i^j) \) is submodular (see e.g. Topkis 1998, Corollary 2.6.2). Submodularity of \( C \) follows from (4) because taking linear combinations with positive weights preserves submodularity (e.g. Topkis 1998, Lemma 2.6.1), and because the holding costs are linear, and hence submodular, in the base-stock levels.

2) Submodularity of \( C \) implies that component inventory is complementary: The cost-minimizing base-stock level for each part is nondecreasing in the base-stock level of other parts (Topkis 1998, Corollary 2.8.1). Thus, if we let all \( s_{j'} \) with \( j' \neq j \) approach infinity, then the resulting cost-minimizing base-stock level \( s_j \) for component \( j \) is an upper bound on the optimal base-stock level for that component. But \( s_{j'} \to \infty \) implies \( P(W_j^i(s_{j'}) = 0) \to 1 \). As a consequence, it can be verified from (4) that as \( s_{j'} \to \infty \) for all \( j' \neq j \), the cost minimization problem for \( s_j \) approaches \( \min_s h_j s_j + \sum_{i \in I_j} \lambda_i b_i^j \mathbb{E} W_j^i(s_j) \), which corresponds to the ISS minimization problem (13) for component \( j \).

Proof of Proposition 3. Since base-stock policies are equivalent to \( (r,q) \) policies with \( q = 1 \) and \( r = s-1 \), we will show our results for \( (r,q) \) policies. Denote the inventory policy used for component \( j \) by \( (r_j,q_j) \). The inventory position of component \( j \) at time \( t \) will be written as \( r_j + R_j(t) \), where \( R_j(t) \) takes on values on \( \{1,\ldots,q_j\} \). We assume that initial inventories are equal to \( r_j + 1 \), implying that \( R_j(t) \) is independent of \( r_j \). On-hand component \( j \) inventory \( H_j(t) \) at time \( t \) can be written as
\[
H_j(t) = r_j + R_j(t) - O_j(t) + B_j(t).
\]
\( B_j(t) \) are component \( j \) back-orders at \( t \): the components \( j \) needed to satisfy all unsatisfied product demands. \( O_j(t) \) is the number of components \( j \) on order at \( t \). (For make-to-stock replenishment, we assume production orders are queued if a previous batch is still being produced. \( O_j(t) \) thus consists of the components in production and any component orders in the queue. Of course, we need to assume that production rates are large enough to cover average demand, so that \( \mathbb{E} \lim_{T \to \infty} \frac{1}{T} \int_0^T O_j(t)dt \) exists. Similarly, we need stochastic sequential lead times to have a
finite mean. ) Component ordering is independent of \( r_j \), so \( O_j(t) \) is independent of \( r_j \), also for make-to-stock and stochastically sequential lead times.

We assume without loss of generality that the components required to assemble a product are deterministic: A demand stream for a product that uses components stochastically can be split into a number of demand streams for products that use components deterministically. We denote the number of components \( j \) that are used to assemble product \( i \) by \( z^i_j \). Using (23), we can rewrite the cost rate \( C(t) = \sum_j h_j H_j(t) + \sum_i b^i B^i(t) \) as \( C(t) = \sum_j h_j r_j + \sum_i \hat{b}^i B^i(t) - O(t) \) where \( \hat{b}^i = b^i + \sum_{j \in J^i} z^i_j h_j \) and \( O(t) = \sum_j h_j (O_j(t) - R_j(t)) \). Using Little’s formula (1961), we can rewrite the cost formulation (2) to include all of the extensions in the proposition.

\[
C := E \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} C(t) dt = \sum_{j \in J} h_j r_j + \sum_{i \in I} \hat{b}^i \lambda^i E \hat{W}^i - \hat{D},
\]

where \( \hat{D} = E \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} O(t) dt. \) (Note that existence of \( E \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} O_j(t) dt \) and the fact that \( R_j(t) \) is uniform on \( \{1, \ldots, q_j\} \) as \( t \to \infty \) implies existence of \( \hat{D} \).) So we have generalized the cost formulation (2) to include all of the extensions in the proposition.

We next extend the definition of \( W^i_j(s_j) \). Consider an arbitrary order for product \( i \) arriving at \( t := 0 \) (the current order). We focus on component \( j \in J^i \). The order constitutes \( z^i_j \) “demands” for component \( j \), which will be labeled \( \{-z^i_j + 1, -z^i_j + 2, \ldots, -1, 0\} \), with labels in the order in which the component demands will be satisfied by committing components to the product demand. So by FCFS, the last “demands” for component \( j \) occurring before \( t \), constituted by a demand \( i' \in I_j \) requiring \( z^i_j \) components, are labeled \( \{-z^i_j - z^{i'}_j + 1, \ldots, -z^i_j\} \), etc. And the first “demands” for component \( j \) occurring after \( t \), constituted by a demand \( i'' \in I_k \), are labeled \( \{1, \ldots, z^{i''}_j\} \), etc.

For convenience of exposition, we pretend that component \( j \) demands arriving in the same batch arrive in rapid succession (rather than simultaneously), and that replenishment orders may be triggered after each component demand. Considering these now “individual” demands, define \( R_j(n) = R_j(t_n) \), where \( t_n \) represents the moment directly after demand \( n \), but before demand \( n+1 \). So \( R_j(n+1) = R_j(n) - 1 + q_j \) if \( R_j(n) = 1 \), and \( R_j(n+1) = R_j(n) - 1 \) otherwise. And for any \( n \), demand \( n' = n - q_j \) triggers a replenishment. The components in this replenishment are committed to demands \( \{n' + r_j + 1, \ldots, n' + r_j + q_j\} \), by FCFS. So one component in the order triggered by \( n'' = (-r_j - 1) - q_j + R_j(-r_j - 1) \) will be used to satisfy demand \( 0 \) for component \( j \).

Let \( T^i_j(n) \) be the time at which demand \( n - q_j + R_j(n) \) arrives, and let \( A^i_j(n) \) be the time at which the replenishment order triggered by that demand arrives. (So \( A^i_j(n) = T^i_j(n) + l_j \) for deterministic lead times.) Fixing an arbitrary demand \( i \) on any sample path fixes \( A^i_j(n) \) for all \( j \) and \( n \). Note that this also holds if lead times are make-to-stock or stochastically sequential, because lead times are fixed on any sample path. Let \( \tilde{W}^i_j(r_j) = (A^i_j(-r_j - 1))^+, \) which thus represents the waiting time on component \( j \) of the current arbitrary product \( i \) demand arriving at \( t := 0 \), when reorder point \( r_j \) is used for \( j \). \( \tilde{W}^i_j \) thus extends \( W^i_j \). Since the current demand will be satisfied if all components needed to satisfy it are available, this allows us to extend the key relation (3) to all of our generalizations:

\[
\tilde{W}^i_j = W^i_j(\{r_j | j \in J^i\}) = \max_{j \in J^i} \tilde{W}^i_j(r_j).
\]
Combining (24) and (25) allows us to extend the SP formulation (4) as

\[
\hat{C}(\bar{s}) := \sum_{j \in \mathcal{J}} h_j r_j - \hat{D} + \sum_{i \in \mathcal{I}} \lambda_i \hat{b}_i^j \mathbb{E} \max_{j \in \mathcal{J}_i} \hat{W}_j(r_j). \tag{26}
\]

By these developments, the proof of Proposition 2 Part 1 goes through without modifications because \(\hat{W}_j(r_j)\) continues to be monotonous (21 extends), because (25) implies (22), and because (26) is a linear combination of \(\mathbb{E}W^i\) and a term linear in \(r_j | j \in \mathcal{J}\). So we find ISS reorder points, denoted by \(r_j^u\), which are an upper bound on the optimal reorder points under FCFS. The proof of Proposition 2 Part 2 then also follows without modifications.

Let \(\bar{S}_j = \{-q_j, -q_j + 1, \ldots, r_j^u\}\), which lets us develop a sample average approximation of (26) along the lines of (8-12). Samples may be generated by simulating the demand and replenishment system, and obtaining for each \(i \in \mathcal{I}, n \in |N^i|\) scenarios \(\xi_{ij}^n\) yielding \(\{A_j^i(r)(\xi_{ij}^n)| j \in \mathcal{J}, r \in \bar{S}_j\}\). Note that it is straightforward to extend samples to make-to-stock or stochastically sequential lead times.

Finally, when \(\hat{b}^j \mathbb{E}B^j\) in (2) is replaced by \(\mathbb{E}c(W^i)\) for non-decreasing \(c(\cdot)\), then the key relation (25) may be extended as (22), which allows the extension of the SP formulation (26) and the sample average approximation.

Proof of Proposition 4. The ISS costs \(C_{\text{iss}}(\bar{s})\) are given by (13). So the ISS base-stock level \(s^u_{jp}\) for component \(j\) in problem \(p\) is the (greatest) minimum of

\[
h_j s_j + f_{jp} \mathbb{E}W_j(s_j), \tag{27}
\]

where \(f_{jp} = \sum_{i \in \mathcal{I}} \lambda_i \hat{b}_i^j = \sum_{i \in \mathcal{I}} \lambda_i (p \hat{b}_i^j + h^i)\) and \(\mathbb{E}W_j := \mathbb{E}W^j(= \mathbb{E}W^j_{\hat{r}})\). Component \(j\) demand during lead time, denoted by \(D_{it}\), is Poisson distributed with mean \(\lambda_j l_j\). By Little’s formula (1961), \(\mathbb{E}W_j = \mathbb{E}((D_t - s)^+)/\lambda_j\), which implies that \(\mathbb{E}W_j\) and thus (27) is convex. So \(s^u_{jp}\) is the unique base-stock level satisfying:

\[
\Delta_j(s^u_{jp}) \geq h_j / f_{jp} > \Delta_j(s^u_{jp} + 1), \tag{28}
\]

where

\[
\Delta_j(s) := \mathbb{E}W_j(s - 1) - \mathbb{E}W_j(s) = e^{-\lambda_j l_j} / \lambda_j \sum_{n=s}^{\infty} (\lambda_j l_j)^n / n! . \tag{29}
\]

By (28), \(s^u_{jp} \to \infty\) as \(p \to \infty\).

The first term in \(\sum_{n=s}^{\infty} (\lambda_j l_j)^n / n!\) dominates the other terms as \(s \to \infty\). (The other terms are bounded by a geometric series, the sum of which vanishes in comparison to the first term.) So for every \(\epsilon > 0\), we can choose \(s'\) such that for every \(s \geq s'\), it holds that

\[
\frac{(\lambda_j l_j)^s}{s!} < e^{\lambda_j l_j} \lambda_j \Delta_j(s) = \sum_{n=s}^{\infty} (\lambda_j l_j)^n / n! < (1 + \epsilon) \frac{(\lambda_j l_j)^s}{s!} . \tag{30}
\]

We fix \(\epsilon\), and assume from now on that \(p\) is large enough for (30) to apply for all \(s \geq s^u_{jp} - 2\).

We now develop an upper bound on \(C^u_{\text{iss}}\) as \(p \to \infty\). By the definition of \(\Delta_j(s)\), \(\mathbb{E}W_j(s) = \sum_{s'=s+1}^{\infty} \Delta_j(s')\). By (30), the first term of \(\sum_{s'=s+1}^{\infty} \Delta_j(s')\) dominates the other terms as \(s \to \infty\), so we can assume that \(p\) is large enough such that \(\mathbb{E}W_j(s^u_{jp}) < (1 + \epsilon') \Delta_j(s^u_{jp} + 1)\). So \(\mathbb{E}W_j(s^u_{jp}) < \)
\[(1 + \epsilon') \Delta_j (s_{jp}^u + 1) < (1 + \epsilon') h_j / f_{jp}, \] where the last inequality is due to (28). So, from (13)

\[C_p^u \leq C_{\text{iss}}(s_p^u) = -D + \sum_j h_j s_{jp}^u + \sum_j f_{jp} \mathbf{E}W_j(s_{jp}^u) \leq -D + \sum_j h_j s_{jp}^u + \sum_j (1 + \epsilon') h_j. \quad (31)\]

We now show that a policy can only outperform iss-fc if it keeps “sufficient” inventories. Under any policy, let \(J_i \subseteq J\) be the set of components \(j\) for which the inventory position is at or below \(s_{jp}^u - 3\) a fraction \(\gamma\) of the time with probability \(\phi\), for some \(\gamma > 0\) and \(\phi > 0\). We will show that as \(p \to \infty\), a policy can only outperform iss-fc if \(\mathcal{J}_i = \emptyset\).

Consider a candidate policy for which \(\mathcal{J}_i \neq \emptyset\). Let \(k \in \arg\max_{j \in \mathcal{J}_i} s_{jp}^u\). For the average number of component \(k\) back-orders \(\mathbf{E}B_k\) under the candidate policy, we find

\[\mathbf{E}B_k \geq \gamma \phi \mathbf{E}((D_k - (s_{kp}^u - 3))^+) = \lambda_k \gamma \phi \mathbf{E}W_k(s_{kp}^u - 3) > \lambda_k \gamma \phi \Delta_k(s_{kp}^u - 2). \quad (32)\]

Here, the first inequality holds because an inventory position \(IP \leq s_{kp}^u - 3\) at \(t\) causes the expected number of back-orders \(B_k\) at \(t + h_k\) to be \((D_k - IP)^+ \geq (D_k - (s_{kp}^u - 3))^+\); the equality holds because \(\mathbf{E}((D_k - s)^+)\) is precisely the number of backorders if a base-stock level \(s\) is used; the last inequality holds by definition of \(\Delta_k(s)\). By applying (30) once for \(s = s'\) and once for \(s = s' - 1\), we find \(\Delta((s' - 1)/\Delta(s')) > s' / ((1 + \epsilon') \lambda_k')\). So \(\Delta(s_{kp}^u - 2) > \Delta(s_{kp}^u) s_{kp}^u (s_{kp}^u - 1) / ((1 + \epsilon') \lambda_k')^2\), while \(\Delta_k(s_{kp}^u) \geq h_k / f_{kp}\) by (28). Combining with (32), this implies \(\mathbf{E}B_k \geq s_{kp}^u (s_{kp}^u - 1) h_k / f_{kp} d\), where \(d\) is some positive number independent of \(p\). Under any allocation policy, back-orders of component \(k\) must correspond to back-orders of products that use \(k\), and each back-order costs at least \(p b_i\) for some \(i \in \mathcal{I}_k\). Since \(b_i > 0\), \(p b_i / f_{kp}\) is bounded below by a strictly positive constant as \(p \to \infty\). So the penalty costs of the candidate policy are at least quadratic in \(s_{kp}^u\): \(p b_i \mathbf{E}B_k \geq (s_{kp}^u)(s_{kp}^u - 1)d^\prime\) as \(p \to \infty\), where \(d^\prime\) is a positive constant. Now, let \(C'\) be the expected average costs of the candidate policy, and consider the cost difference

\[C' - C_p^u \geq (s_{kp}^u) (s_{kp}^u - 1) d' - \sum_{j \in \mathcal{J}} (1 + \epsilon') h_j - \sum_{j \in \mathcal{J} \setminus \mathcal{J}_i} 2h_j - \sum_{j \in \mathcal{J}_i} h_j s_{jp}^u. \quad (33)\]

In this expression, the first term is the lower bound on the penalty costs of our candidate policy, and the second term is minus an upper bound on the penalty costs of iss-fc by (31). Finally, the third and fourth term constitute an upper bound on the difference in holding costs between iss-fc and the candidate policy. The third term arises because the inventory position of the candidate policy is bigger than \(s_{jp}^u - 3\) a fraction 1 of the time a.p. 1 for components \(j \in \mathcal{J} \setminus \mathcal{J}_i\), by definition of \(\mathcal{J}_i\). The fourth term arises because holding costs for components \(\mathcal{J}_i\) must be nonnegative for the candidate policy, while holding costs are given by \(h_j s_{jp}^u\) for component \(j\) in the ISS solution. As \(p \to \infty\), we have \(s_{jp}^u \to \infty\). So, since \(k \in \arg\max_{j \in \mathcal{J}_i} s_{jp}^u\), the first term dominates all other terms as \(p \to \infty\). We conclude that \(C' - C_{\text{iss}}(s_p^u) > 0\). This shows that policies can only outperform iss-fc as \(p \to \infty\) if \(\mathcal{J}_i = \emptyset\).

Since the optimal policy outperforms iss-fc, \(\mathcal{J}_i = \emptyset\) for the optimal policy. So \(C_p^u \geq \sum h_j (s_{jp}^u - 2) - D\). Using (31) and \(s_{jp}^u \to \infty\) as \(p \to \infty\), we conclude that \((C_p^u - C_p^*) / C_p^* \to 0\) as \(p \to \infty\), which proves part 2. (In fact, we obtain the stronger result that \(C_p^u - C_p^*\) is bounded by \(\sum_j (3 + \epsilon') h_j\) in the limit \(p \to \infty\).) Since \(C_p' \leq C_p^*\), part 1 follows immediately.

Proof of Corollary 5. Note that iss-fc is an independent base-stock policy under FCFS. The result thus follows from Proposition 4.
C Algorithm scalability

Table 1 summarizes the results of an investigation of the scalability of the sampling-based algorithm proposed in Section 3.2.2 (vJ&S-W). In this appendix, we discuss the setup of this investigation.

The algorithms proposed by Akçay and Xu (2004) and Huang and de Kok (2011), hereafter referred to as A&X and H&dK, are used as benchmarks. These algorithms optimize base-stock levels and allocation in periodic review ATO systems. (H&dK also present an algorithm for which the base-stock levels are exogenous, but we focus on the version of the algorithm in which the base-stock levels are treated as decision variables.) However, they assume FCFS allocation of components to product demands arriving in different periods, and optimize component allocation to product demands that arrive simultaneously (in the same period). So, as the review period grows short, these algorithms assume FCFS. H&dK study a minimization problem, and their objective function is comparable to ours. A&X study a maximization problem with the objective function corresponding to the weighted fill rate. They apply a budget constraint that does not include committed inventories. To facilitate an appropriate and more uniform presentation of optimality gaps, we transform the latter problem into an equivalent minimization problem with the objective corresponding to the non-fill rate (=1 minus the fill rate).

We now discuss how the test cases with different sizes used for Table 1 were generated. For simplicity, we keep the number of products equal to the number of components. For each product, \( J^i \) is generated by randomly selecting 6 components. (For cases with \(|I| < 6\), we select only 2 components to generate \( J^i \). This discontinuity may explain why computation times for the A&X algorithm for the size 5 and size 10 cases are equal.) Since A&X and H&dK are periodic review algorithms, we generate component lead times randomly on \{4, 5, 6, \ldots, 18\}. Higher (lower) maximum lead times significantly increase (decrease) computation times for the H&dK algorithm, without affecting the qualitative observations in Table 1. The maximum lead time of 18 corresponds to the maximum lead time in the PC-assembly case studied in Cheng et al. (2002), and to similar systems studied in Akçay and Xu (2004) and Lu and Song (2005). We let \( \lambda^i = 10 \) for all products, and generate \( h_j \) uniformly for each component on \([0.5, 1.5]\). We set \( b^i = f/(1-f)h^i \). \( b^i \) is also used to weight the fill rate of different products for the A&X model, and to penalize waiting customer demands for the H&dK model. We use \( f = 0.9 \). Likewise, we choose the inventory budget for the A&X model such that the weighted fill-rate that is attained is about 90%.

The computational complexity of the sampling approximation increases with the number of scenarios, while Proposition 1 reveals that it is critical to use sufficiently many scenarios per sample in order to produce good solutions and tight lower bounds. Every scenario for the periodic review systems corresponds to one demand realization for each product in a period. We thus tested the scalability of the algorithms using \( n = 200 \) scenarios for the periodic review algorithms, and \( n = 200 \) scenarios per product (\( n|I| \) scenarios in total) for the vJ&S-W algorithm. This approach was motivated by the results in Table 4, which reports estimators of the gap between the asymptotic upper and lower bounds obtained for a size 3 problem, for different values of \( n \). The asymptotic lower bound is obtained based on the solution values of the sample approximation for 200 samples, each consisting of \( n \) scenarios. The upper bound is obtained by taking the point-wise average of the base-stock levels obtained for these 200 samples, and estimating its costs using simulation. The table shows that the algorithms yield similar bounds for similarly sized samples, and that \( n = 200 \) yields gaps on the order of 1%.

To obtain a reliable estimate of a lower bound, the objective function obtained for multiple independent samples is averaged. The computation times in Table 1 correspond to solving the sample approximation for 50 samples, each consisting of 200 scenarios for the A&X and H&dK algorithms and of \( 200|I| \) scenarios for the vJ&S-W algorithm. (For the experiments in Sections 4.3-
<table>
<thead>
<tr>
<th>$n$</th>
<th>vJ&amp;S-W</th>
<th>A&amp;X</th>
<th>H&amp;dK</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>2.09± 0.84</td>
<td>2.64± 2.33</td>
<td>4.52± 1.09</td>
</tr>
<tr>
<td>100</td>
<td>1.66± 0.6</td>
<td>1.45± 1.61</td>
<td>3.03± 0.87</td>
</tr>
<tr>
<td>200</td>
<td>0.8± 0.44</td>
<td>0.71± 1.09</td>
<td>1.38± 0.65</td>
</tr>
</tbody>
</table>

Table 4: The gaps obtained for the algorithms using different numbers of scenarios per sample.

4.5, we typically used even more scenarios per sample to further strengthen the lower bound.)

Computations were performed using Cplex 12.2 on a Intel Core i5 CPU with 4GB of RAM.

D Data for the maintenance organization problem

Data for the maintenance organization problem are given in this section. The problem data were estimated based on data in the ERP system of the repair shop, cf. van Jaarsveld et al. (2011). Costs and lead times have been rescaled and rounded to prevent any sensitive information from being retrievable from the data. This did not affect the main insights gained through this study.

The case consists of three repair types $a$, $b$ and $c$, with associated arrival rates 0.13, 0.10 and 0.35 per unit of time, respectively. Spare parts are characterized by their lead time $l$, their holding cost per unit of time $h$, and their associated usage probabilities $p_a$, $p_b$ and $p_c$ in each of the repair types. Table 5 gives this data for all 110 parts. A dash (-) indicates that the spare part is never used in that particular repair type. When a repair of a given type arrives, it uses each spare part with the probability prescribed in Table 5, independent of the usage of other spare parts in the repair.
Table 5: Spare part data for the maintenance organization case.