POMS PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT POMS

Vol. 25, No. 4, April 2016, pp. 647–661
ISSN 1059-1478 | EISSN 1937-5956 | 16 | 2504 | 0647
DOI 10.1111/poms.12498

Experimental Results Indicating Lattice-Dependent Policies May Be Optimal for General Assemble-To-Order Systems

Emre Nadar
Department of Industrial Engineering, Bilkent University, Bilkent, 06800, Ankara, Turkey, emre.nadar@bilkent.edu.tr

Mustafa Akan, Alan Scheller-Wolf
Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA, akan@andrew.cmu.edu, awolf@andrew.cmu.edu

We consider an assemble-to-order (ATO) system with multiple products, multiple components which may be demanded in different quantities by different products, possible batch ordering of components, random lead times, and lost sales. We model the system as an infinite-horizon Markov decision process under the average cost criterion. A *control policy* specifies when a batch of components should be produced, and whether an arriving demand for each product should be satisfied. Previous work has shown that a *lattice-dependent base-stock and lattice-dependent rationing* (LBLR) policy is an optimal stationary policy for a special case of the ATO model presented here (the generalized *M*-system). In this study, we conduct numerical experiments to evaluate the use of an LBLR policy for our general ATO model as a heuristic, comparing it to two other heuristics from the literature: a state-dependent base-stock and state-dependent rationing (SBSR) policy, and a fixed base-stock and fixed rationing (FBFR) policy. Remarkably, LBLR yields the globally optimal cost in *each* of more than 22,500 instances of the general problem, outperforming SBSR and FBFR with respect to both objective value (by up to 2.6% and 4.8%, respectively) and computation time (by up to three orders and one order of magnitude, respectively) in 350 of these instances (those on which we compare the heuristics). LBLR and SBSR perform significantly better than FBFR when replenishment batch sizes imperfectly match the component requirements of the most valuable or most highly demanded product. In addition, LBLR substantially outperforms SBSR if it is crucial to hold a significant amount of inventory that must be rationed.

Key words: inventory management; assemble-to-order systems; Markov decision processes; mixed integer program; lost sales
History: Received: February 2012; Accepted: July 2015 by Jayashankar Swaminathan, after 3 revisions.

1. Introduction

It is common knowledge that assemble-to-order (ATO) systems are notoriously difficult to analyze: Despite the popularity of ATO systems in practice, the structure of the optimal inventory replenishment and allocation policy is still unknown for general ATO systems. Previous work has only established the optimal policy structure for very specific ATO systems—such as the *W*-system and the *M*-system; see Dogru et al. (2010), Lu et al. (2014), and Nadar et al. (2014) for example. As a result, simple heuristic control policies for general ATO systems are attracting widespread interest in practice (Lu et al. 2010). Likewise, several researchers have explored performance evaluation and optimization techniques of various heuristic policies; see, for instance, Zhang (1997), Agrawal and Cohen (2001), and Akcay and Xu (2004). We refer the reader to Song and Zipkin (2003) for a comprehensive review of this literature.

In a recent study, Nadar et al. (2014) consider a Markovian ATO "generalized *M*-system." This system involves a single master product which uses multiple units from each component, and multiple individual products each of which uses multiple units from a single unique component. They prove that if replenishment batch sizes are determined by individual product sizes, the optimal inventory replenishment policy is a *lattice-dependent base-stock production policy* and the optimal inventory allocation policy is a *lattice-dependent rationing policy*. This implies that the state space of the problem can be partitioned into disjoint lattices such that, on each lattice, (a) it is optimal to produce a batch of a particular component if and only if the state vector is less than the base-stock level of that component on the current lattice; and (b) it is optimal to fulfill a demand of a particular product if and only if the state vector is greater than or equal to the rationing level for that product on the current lattice.

In this study, we adapt the lattice-dependent base-stock and lattice-dependent rationing (LBLR) policy introduced by Nadar et al. (2014) to ATO systems with *general* product structures, evaluating its use as a *heuristic* replenishment and allocation policy. We also compare the LBLR policy to two other heuristics, both from Benjaafar and ElHafsi (2006): a state-dependent base-stock and state-dependent rationing (SBSR) policy, and a fixed base-stock and fixed rationing (FBFR) policy. We take the average cost rate as our performance criterion.

Different versions of the FBFR and SBSR policies have been extensively studied in the Markovian inventory literature; see, for instance, Ha (1997, 2000), de Véricourt et al. (2002), Frank et al. (2003), ElHafsi et al. (2008), ElHafsi (2009), Gayon et al. (2009), and Benjaafar et al. (2011). Although FBFR is a subclass of SBSR, it has the advantage of being relatively easy to understand and implement (Dekker et al. 2002). LBLR, FBFR, and SBSR are all *deterministic* policies, as opposed to *randomized* policies. (A deterministic policy always chooses the same action in a state, while a randomized policy may choose actions according to a probability distribution.) Randomized policies are often more difficult to implement, so in practice a controller may prefer to use a deterministic, but potentially suboptimal policy (Puterman 1994).

We develop a Linear Programming (LP) formulation to find the globally optimal stationary randomized policy, and Mixed Integer Programming (MIP) formulations to find the optimal stationary deterministic policy within each heuristic class (LBLR, SBSR, and FBFR). We analytically show that LBLR outperforms the other heuristics with respect to objective value, cf. Proposition 1. We then generate over 22,500 instances to numerically test efficacy of LBLR in a variety of settings. Remarkably, we find that LBLR yields the globally optimal cost in *each* of these instances.

We also find that LBLR performs better than SBSR (or FBFR) by up to 2.6% (or 4.8%) of the globally optimal cost on a test bed constructed from 350 instances. (The average distances from the optimal cost are 0.5% and 1.4%, respectively.) LBLR also has a notable computational advantage; the computation times of LBLR are shorter by up to three orders and one order of magnitude, respectively. Our numerical results indicate that LBLR and SBSR perform significantly better than FBFR when the component batch sizes imperfectly match the component requirements of the most highly demanded and/or most valuable product. In addition, LBLR has the greatest benefit over SBSR when products are highly differentiated but demand for each product should have a substantial fill rate. The latter observation is also supported by a regression study.

Our results suggest that the LBLR policies may be optimal for general ATO systems. However, we have found counter examples (see the online appendix) showing that the functional characterizations used in Nadar et al. (2014) to prove the optimality of LBLR for generalized *M*-systems need not hold for ATO systems with general product structures. Thus, showing the optimality of LBLR for general ATO systems will likely require a different methodology.

We contribute to the ATO literature in several important ways: First, our computational results reveal the practicality of LBLR as a heuristic *deterministic* policy for the general ATO problem. Second, by identifying the optimal policy structure as LBLR in our numerical experiments, we are able to uncover the role of different product characteristics in optimal control of ATO systems. Specifically, we provide Rule of Thumb 1 to guide the partitioning of the state space into disjoint lattices for LBLR. Third, we highlight when, and how, common heuristics may fall short, producing high-level guidelines for control policy choices in different environments.

The rest of this study is organized as follows: Section 2 describes the model and LP formulation. Section 3 describes the heuristics along with the MIP formulation of LBLR. Section 4 presents and interprets numerical results for the heuristics. Section 5 offers a summary and concludes. The MIP formulations of SBSR and FBFR, additional numerical results, and the structural counter examples to Nadar et al. (2014) are contained in the online appendix.

2. Model Formulation

We consider an ATO system with m components ($i = 1, 2, .., m$) and n products ($j = 1, 2, .., n$). Define $\mathbf{A}$ as an $m \times n$ nonnegative resource-consumption matrix; a_{ij} is the number of units of component i needed to assemble one unit of product j. Each component i is produced in batches of a fixed size q_i in a make-to-stock fashion. Define $\mathbf{q} = (q_1, q_2, .., q_m)$ as the vector of production batch sizes. Production time for a batch of component i is independent of the system state and the number of outstanding orders of any type, and exponentially distributed with finite mean $1/\mu_i$. Assembly lead times are negligible so that assembly operations can be postponed until demand is realized. Demand for each product j arrives as an independent Poisson process with finite rate λ_j. Demand for product j can be fulfilled only if all the required components are available; otherwise, the demand is lost, incurring a unit lost sale cost c_j. Demand may also be rejected in the presence of all the necessary components, again incurring the unit lost sale cost c_j.

The state of the system at time t is the vector $\mathbf{X}(t) = (X_1(t), .., X_m(t))$, where $X_i(t)$ is a nonnegative integer denoting the on-hand inventory for component i at time t. Component i held in stock incurs a holding cost per unit time $h_i(X_i(t))$, which is convex and strictly increasing. Denote by $h(\mathbf{X}(t)) = \sum_i h_i(X_i(t))$ the total inventory holding cost rate at state $\mathbf{X}(t)$. Since all inter-event times are exponentially distributed, the system retains no memory, and decision epochs can be restricted to times when the state changes. Using the memoryless property, we can formulate the problem as an Markov decision process and focus on Markovian policies for which actions at each decision epoch depend solely on the current state. A control policy ℓ specifies for each state $\mathbf{x} = (x_1, .., x_m)$, the action $\mathbf{u}^\ell(\mathbf{x}) = (u^{(1)}, .., u^{(m)}, u_1, .., u_n)$, $u^{(i)}, u_j \in \{0, 1\}$, $\forall i, j$; where $u^{(i)} = 1$ means produce component i, and $u^{(i)} = 0$ means do not produce component i; $u_j = 1$ means satisfy demand for product j, and $u_j = 0$ means reject demand for product j. Denote by $\mathbb{U}(\mathbf{x})$ the set of admissible actions at state $\mathbf{x}$. For any action $\mathbf{u} = (u^{(1)}, .., u^{(m)}, u_1, .., u_n) \in \mathbb{U}(\mathbf{x})$, we must have $u_j = 0$ if $\exists i$ s.t. $x_i < a_{ij}$.

As each ordering decision specifies only whether or not to produce a component, there is at most one outstanding batch order for each component at any time. Also, as component orders are not part of our system state, these can in effect be cancelled upon transition to a new state. Both of these assumptions are standard in the literature (see, e.g., Ha 1997, Benjaafar and ElHafsi 2006, and ElHafsi et al. 2008). Our numerical results suggest that the latter assumption is benign: Orders are cancelled optimally in 55% of the 350 compiled instances in subsections 4.1 and 4.2. However, for those instances, if the optimal policy of our model is followed but orders are never cancelled, it increases costs by no more than 3.29%, and the average cost increase is 0.08%.

Let v denote a real-valued function defined on $\mathbb{N}_0^m$ (where $\mathbb{N}_0$ is the set of nonnegative integers and $\mathbb{N}_0^m$ is its m-dimensional cross product). For a given policy ℓ and starting state $\mathbf{x} \in \mathbb{N}_0^m$, the average cost per unit time over an infinite planning horizon $v^\ell(\mathbf{x})$can be written as follows (see, e.g., ElHafsi et al. 2008 and Nadar et al. 2014):

$$v^\ell(\mathbf{x}) = \limsup_{T\to\infty} \frac{1}{T}\left\{\int_0^T h(\mathbf{X}(t))dt + \sum_{j=1}^{n}\int_0^T c_j dN_j(t)\right\},$$

where $N_j(t)$ is the number of demands for product j that have been rejected up to time t. The objective is to identify a policy ℓ^* that yields $v^*(\mathbf{x}) = \inf_\ell v^\ell(\mathbf{x})$ for all states $\mathbf{x}$.

We next formulate a linear program to find a global optimal solution to the above problem. Define $v_{\mathbf{y}|\mathbf{x},\mathbf{u}}$ as the rate at which the system moves from state $\mathbf{x}$ to state $\mathbf{y}$ if action $\mathbf{u} \in \mathbb{U}(\mathbf{x})$ is chosen, and $\pi_{\mathbf{x},\mathbf{u}}$ as the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{u} \in \mathbb{U}(\mathbf{x})$ is chosen. As a computational requirement, we restrict the state space to be finite; define $\bar{\mathbf{x}} = (\bar{x}_1, .., \bar{x}_m)$ as a vector of upper bounds for component inventory levels. (The upper bound should be sufficiently high so that the globally optimal cost does not change with a further increase in the upper bound.) Thus, for any action $\mathbf{u} = (u^{(1)}, .., u^{(m)}, u_1, .., u_n) \in \mathbb{U}(\mathbf{x})$, we must have $u^{(i)} = 0$ if $x_i + q_i > \bar{x}_i$. The globally optimal average cost Z^* can be found by solving the following linear program (see Puterman 1994):

$$(\mathcal{LP})\ \text{minimize} \sum_{\mathbf{x}\preceq\bar{\mathbf{x}}}\sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} h(\mathbf{x})\pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x}\preceq\bar{\mathbf{x}}}\sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})}\sum_{j:u_j=0}\lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

subject to

$$\sum_{\mathbf{u}\in\mathbb{U}(\mathbf{y})}\pi_{\mathbf{y},\mathbf{u}}\sum_{\mathbf{x}\preceq\bar{\mathbf{x}}} v_{\mathbf{x}|\mathbf{y},\mathbf{u}} - \sum_{\mathbf{x}\preceq\bar{\mathbf{x}}}\sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} v_{\mathbf{y}|\mathbf{x},\mathbf{u}}\pi_{\mathbf{x},\mathbf{u}} = 0, \quad \forall \mathbf{y} \preceq \bar{\mathbf{x}}, \tag{1}$$

$$\sum_{\mathbf{x}\preceq\bar{\mathbf{x}}}\sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})}\pi_{\mathbf{x},\mathbf{u}} = 1, \tag{2}$$

$$\pi_{\mathbf{x},\mathbf{u}} \geq 0, \qquad \forall \mathbf{x} \preceq \bar{\mathbf{x}}, \qquad \forall \mathbf{u} \in \mathbb{U}(\mathbf{x}), \tag{3}$$

where "$\preceq$" denotes component-wise inequality (i.e., $\mathbf{x} \preceq \bar{x} \iff x_i \leq \bar{x}_i, \forall i$). The first term of the objective function corresponds to the time-average inventory holding cost and the second term corresponds to the time-average lost sales cost. Constraints (1) and (2) are the balance equations and normalization constraint that together yield the limiting probability values.

Notice that the above linear program may yield a randomized policy as the global optimal solution, that is, there may exist a state $\mathbf{x}$ such that $\pi_{\mathbf{x},\mathbf{u}_1} > 0$ and $\pi_{\mathbf{x},\mathbf{u}_2} > 0$, where $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{U}(\mathbf{x})$. This can indeed occur: We have found instances for which a randomized policy is optimal. But, for these instances there also exists an optimal (deterministic) LBLR policy with the same objective value.

3. Heuristic Policies and Their MIP Formulations

3.1. Lattice-Dependent Base-Stock and Lattice-Dependent Rationing

We introduce the notation $\mathbb{L}(\mathbf{p}, \mathbf{r}) = \{\mathbf{p} + k\mathbf{r} : k \in \mathbb{N}_0\}$ to denote an m-dimensional lattice with initial vector $\mathbf{p} \in \mathbb{N}_0^m$ and common difference $\mathbf{r} \in \mathbb{N}_0^m$, where $\exists i$ such that $p_i < r_i$. For any $\mathbf{r} \in \mathbb{N}_0^m$, $\mathbb{N}_0^m = \bigcup_{\mathbf{p}} \mathbb{L}(\mathbf{p}, \mathbf{r})$ and $\mathbb{L}(\mathbf{p}_1, \mathbf{r}) \cap \mathbb{L}(\mathbf{p}_2, \mathbf{r}) = \emptyset$, $\forall \mathbf{p}_1$,

$\mathbf{p}_2$ s.t. $\mathbf{p}_1 \neq \mathbf{p}_2$. In other words, we partition the state space into multiple disjoint lattices with common difference $\mathbf{r}$. We also define $\mathbf{\Delta}^i = (\Delta_1^i, \Delta_2^i, .., \Delta_m^i)$ and $\mathbf{\Delta}_j = (\Delta_{j1}, \Delta_{j2}, .., \Delta_{jm})$ as m-dimensional vectors of non-negative integers. With these we describe an LBLR policy as follows:

(i) Inventory replenishment of each component i follows a lattice-dependent base-stock policy with lattice-dependent base-stock levels $\mathbf{S}_i(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)$ such that a batch of component i is produced if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)$ is less than $\mathbf{S}_i(\mathbf{p})$; and

(ii) Inventory allocation for each product j follows a lattice-dependent rationing policy with lattice-dependent rationing levels $\mathbf{R}_j(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j)$ such that a demand for product j is satisfied if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j)$ is greater than or equal to $\mathbf{R}_j(\mathbf{p})$.

An illustration of such a policy for a 2-component 2-product system is shown in Figure 1.

We could optimize over the vectors $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ to obtain the LBLR policy with the least cost. But it is both time-consuming and unnecessary to do so, considering the optimal performance of LBLR in Section 4 when these vectors obey the following rule of thumb:

Rule of Thumb 1. *Given the parameters a_{ij} and c_j, $\forall i,j$: (i) $\Delta_i^i = \max_j a_{ij}$, and $\Delta_k^i = \min_j a_{kj}$, $\forall k \neq i$; and (ii) $\Delta_{ji} = a_{ij^*}$ where $j^* = \arg\max_{k \neq j} c_k$, $\forall i$.*

Rule of Thumb 1 builds largely upon previously established optimality results for ATO systems (see Benjaafar and ElHafsi 2006, and Nadar et al. 2014). See Figure 1 for an illustration in a 2-component 2-product system: Consider the state space partitioning scheme of component 1 for example. When we transition to a higher state on a given lattice (and a sufficient amount of component 2 exists), the total demand for *any* product that can be satisfied increases by one since we increase the inventory level of component 1 by the *maximum* of the numbers of component 1 required by a product. Thus, the desirability of producing a batch of component 1 is likely to be weakly lower at a higher state. Conversely, when we transition to a higher state on a given lattice, we increase the inventory level of component 2 by the *minimum* of the numbers of component 2 required by a product as we want to reduce the incentive to produce a batch of component 1. The inventory level of component 2 should not be increased too much, because then it may be better to produce component 1 at a higher state if the inventory level of component 1 is significantly lower than the inventory level of component 2 at this higher state. Hence, *Rule of Thumb 1* seems likely to engender a lower incentive to produce a component at a higher state on a lattice. This justifies the use of a base-stock policy.

Now consider what happens when we transition to a higher state on a given lattice by increasing the inventory levels by the component requirements of product 2: The desirability of satisfying a demand for product 1 is likely to be higher. This is because demands of product 1 compete with those of product 2 for the components, and the competition becomes less severe with the supply increase sufficient to satisfy a demand for the competitor product 2. (For ATO systems with more products the state space partitioning scheme of a product is based on the component requirements of its competitor product with the highest lost sale cost.) Hence, *Rule of Thumb 1* seems likely to engender a weakly higher incentive to satisfy a demand for a product at a higher state on a lattice, justifying the use of a rationing policy.

We proceed to the MIP formulation of this heuristic class. First, define the set $\mathbb{S}_i(\mathbf{p}, \mathbf{b}) = \{(\mathbf{x}, \mathbf{u}) : \mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i),\ \mathbf{x} \preceq \bar{\mathbf{x}},\ \mathbf{u} \in \mathbb{U}(\mathbf{x}),\ \text{and}\ \sum_{\mathbf{x},\mathbf{u}} \pi_{\mathbf{x},\mathbf{u}} = 0 \Leftrightarrow \mathbf{S}_i(\mathbf{p}) = \mathbf{b}\}$ for $\mathbf{b} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)$. The elements of the set $\mathbb{S}_i(\mathbf{p}, \mathbf{b})$ are state-action pairs $(\mathbf{x},\mathbf{u})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{u}$ is chosen should be zero when the base-stock level of component i equals $\mathbf{b}$, on the lattice with initial vector $\mathbf{p}$ and common difference $\mathbf{\Delta}^i$. Likewise, define the set $\mathbb{R}_j(\mathbf{p}, \mathbf{b}) = \{(\mathbf{x}, \mathbf{u}) : \mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j),\ \mathbf{x} \preceq \bar{\mathbf{x}},\ \mathbf{u} \in \mathbb{U}(\mathbf{x}),$ and $\sum_{\mathbf{x},\mathbf{u}} \pi_{\mathbf{x},\mathbf{u}} = 0 \Leftrightarrow \mathbf{R}_j(\mathbf{p}) = \mathbf{b}\}$ for $\mathbf{b} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j)$. The elements of the set $\mathbb{R}_j(\mathbf{p}, \mathbf{b})$ are state-action pairs $(\mathbf{x},\mathbf{u})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{u}$ is chosen should be zero when the rationing level for product j equals $\mathbf{b}$, on the lattice with initial vector $\mathbf{p}$ and common difference $\mathbf{\Delta}_j$. Lastly, define $z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}$ and $z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}$ as binary variables as follows:

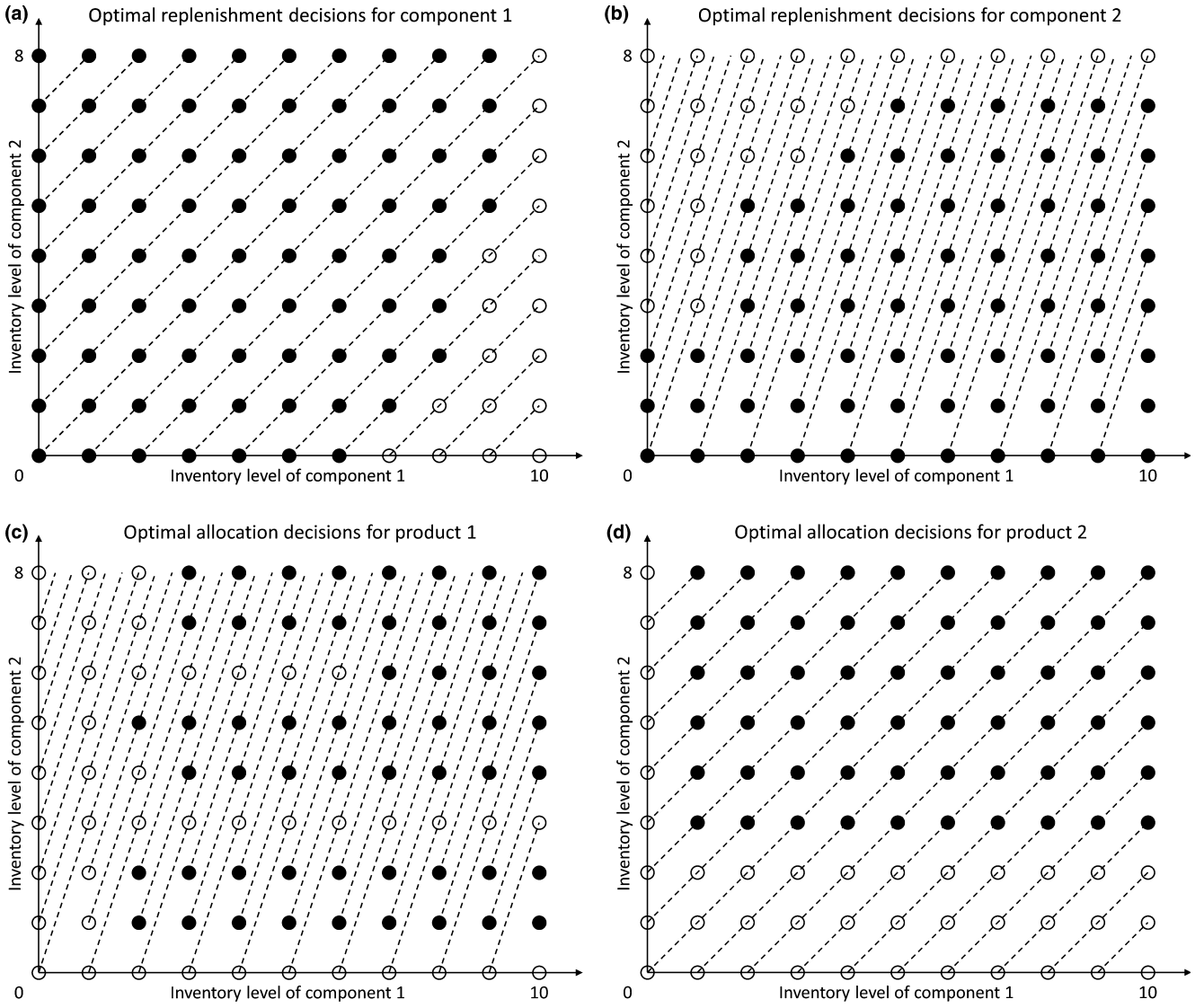
$$z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})} = \begin{cases} 1 & \text{if } \mathbf{S}_i(\mathbf{p}) = \mathbf{b}, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})} = \begin{cases} 1 & \text{if } \mathbf{R}_j(\mathbf{p}) = \mathbf{b}, \\ 0 & \text{otherwise.} \end{cases}$$

We are now ready to describe the constraints of the MIP problem. First, the optimal solution of the MIP problem should satisfy constraints (1)–(3) of the LP formulation of the optimal policy ($\mathcal{LP}$). Also, on each lattice, the optimal solution should select exactly one base-stock level for each component and one rationing level for each product. Thus we impose the following constraints:

$$\sum_{\mathbf{b} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)} z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})} = 1, \qquad \forall \mathbf{p} \text{ and } \forall i, \tag{4}$$

Figure 1 Illustration of LBLR for a 2 × 2 System with A = ((1,1),(1,3)), q = (1,3), $h_1 = 1$, $h_2 = 5$, $\mu_1 = \mu_2 = \lambda_1 = \lambda_2 = 1$, $c_1 = 20$, $c_2 = 100$, $\bar{x}_1 = \bar{x}_2 = 10$ (a) Optimal Replenishment Decisions for Component 1 (b) Optimal Replenishment Decisions for Component 2 (c) Optimal Allocation Decisions for Product 1 (d) Optimal Allocation Decisions for Product 2



Notes. In graphs (a) and (b), a filled circle means produce a batch of components at the corresponding inventory levels. In graphs (c) and (d), a filled circle means fulfill the demand at the corresponding inventory levels. In graphs (a)–(d), each dashed line forms a different lattice; its slope is determined by $\mathbf{\Delta^1} = (\mathbf{1},\mathbf{1})$, $\mathbf{\Delta^2} = (\mathbf{1},\mathbf{3})$, $\mathbf{\Delta_1} = (\mathbf{1},\mathbf{3})$, and $\mathbf{\Delta_2} = (\mathbf{1},\mathbf{1})$, respectively.

$$\sum_{\mathbf{b}\in\mathbb{L}(\mathbf{p},\mathbf{\Delta}_j)} z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})} = 1, \qquad \forall \mathbf{p} \text{ and } \forall j. \tag{5}$$

The constraints below link our binary variables to the appropriate limiting probability variables:

$$\sum_{(\mathbf{x},\mathbf{u})\in\mathbb{S}_i(\mathbf{p},\mathbf{b})} \pi_{\mathbf{x},\mathbf{u}} \le 1 - z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}, \qquad \forall \mathbf{p},\ \forall \mathbf{b}, \text{ and } \forall i, \tag{6}$$

$$\sum_{(\mathbf{x},\mathbf{u})\in\mathbb{R}_j(\mathbf{p},\mathbf{b})} \pi_{\mathbf{x},\mathbf{u}} \le 1 - z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}, \qquad \forall \mathbf{p},\ \forall \mathbf{b}, \text{ and } \forall j. \tag{7}$$

In constraint (6), if $z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}$ equals one, then all limiting probability variables corresponding to the state-action pairs in set $\mathbb{S}_i(\mathbf{p},\mathbf{b})$ are forced to equal zero. Likewise, in constraint (7), if $\mathbf{z}_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}$ equals one, then all limiting probability variables corresponding to the state-action pairs in set $\mathbb{R}_j(\mathbf{p},\mathbf{b})$ are forced to equal zero. Otherwise, these constraints become redundant. See Bhandari et al. (2008) for a similar MIP formulation in a different context. The optimal average cost of this policy Z_{LBLR} can be found by solving the following MIP problem:

(LBLR)

$$\text{minimize} \sum_{\mathbf{x} \preceq \bar{\mathbf{x}}} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x} \preceq \bar{\mathbf{x}}} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} \sum_{j: u_j = 0} \lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

subject to (1)–(7).

3.2. State-Dependent Base-Stock and State-Dependent Rationing

Define $\mathbf{x}_{-i} = (x_1, .., x_{i-1}, x_{i+1}, .., x_m)$ as a vector of the inventory levels for components $k \neq i$. With this we describe an SBSR policy as follows (as in Benjaafar and ElHafsi 2006):

(i) Inventory replenishment of each component i is governed by state-dependent base-stock levels $S_i(\mathbf{x}_{-i})$: a batch of component i is produced if and only if $x_i \leq S_i(\mathbf{x}_{-i})$; and
(ii) Inventory allocation for demand class j is governed by state-dependent rationing levels $R_{ij}(\mathbf{x}_{-i})$: a demand from class j is fulfilled if and only if $x_i \geq R_{ij}(\mathbf{x}_{-i})$, $\forall i$.

Different demand classes in Benjaafar and ElHafsi (2006) correspond to different products in our model. The SBSR policy has the following additional properties (Benjaafar and ElHafsi 2006):

(a) The base-stock level of any component is nondecreasing in the inventory level of any other component;
(b) A unit increase in the inventory level of one component leads to at most a unit increase in the base-stock level of any other component;
(c) The rationing level for any demand class at one component is nonincreasing in the inventory level of any other component;
(d) Once initiated, the production of a component is never interrupted;
(e) For each component, the rationing level for any demand class is greater than or equal to the rationing level for the demand class with the next higher lost sale cost; and
(f) Demands with the highest lost sale cost are always satisfied if sufficient inventory exists.

Properties (e) and (f) are inapplicable to our general model, as our products differ not only in their lost sale costs but also in their component usage rates, and thus we do not enforce these properties. We also omit property (d) from SBSR to keep the state space manageable; this can only improve the performance of SBSR. The MIP formulation for our "relaxed" SBSR policy is contained in the online appendix; define Z_{SBSR} as the optimal average cost of this policy.

Benjaafar and ElHafsi (2006) showed that, under Markovian assumptions, the SBSR policy is optimal when the system involves a *single* end-product that requires one unit from multiple components and is demanded by multiple demand classes.

3.3. Fixed Base-Stock and Fixed Rationing

Lastly, we describe an FBFR policy as follows (as in Benjaafar and ElHafsi 2006):

(i) Inventory replenishment of each component i is governed by a fixed base-stock level S_i: a batch of component i is produced if and only if $x_i \leq S_i$; and
(ii) Inventory allocation for each product j is governed by a vector of fixed rationing levels $\mathbf{R}_j = (R_{1j}, R_{2j}, .., R_{mj})$: a demand for product j is satisfied if and only if $x_i \geq R_{ij}$, $\forall i$.

We also provide the MIP formulation of this heuristic class in the online appendix. Define Z_{FBFR} as the optimal average cost of this policy.

3.4. Analytical Comparison of Heuristic Policies

The proposition below ranks our heuristic policies in terms of their optimal costs:

PROPOSITION 1. $Z^* \leq Z_{LBLR} \leq Z_{SBSR} \leq Z_{FBFR}$

PROOF OF PROPOSITION 1. The first and third inequalities hold since $\mathcal{LP}$ is a relaxation of all the other MIP formulations and since FBFR is a subclass of SBSR. To prove the second inequality, we will show that SBSR is a subclass of LBLR.

Recalling the definitions of $\mathbf{\Delta}^i = (\Delta_1^i, \Delta_2^i, .., \Delta_m^i)$ and $\mathbf{\Delta}_j = (\Delta_{j1}, \Delta_{j2}, .., \Delta_{jm})$ from subsection 3.1, we choose any specific $\mathbf{\Delta}^i$ such that $\Delta_i^i \geq \sum_{k \neq i} \Delta_k^i$, $\forall i$ (recall LBLR chooses the optimal $\mathbf{\Delta}^i$). The only constraint LBLR places on inventory replenishment decisions is that if a batch of component i is not produced at inventory level $\mathbf{x}$, then it is not produced at inventory level $\mathbf{x} + \mathbf{\Delta}^i$. This is also true under an SBSR policy: If a batch of component i is not produced at inventory level $\mathbf{x}$, then the base-stock level of component i is less than x_i at inventory level $\mathbf{x}$. Property (b) of SBSR implies that if the inventory level of component $k \neq i$ increases by Δ_k^i, $\forall k$, then the base-stock level of component i increases by at most $\sum_{k \neq i} \Delta_k^i$ units. Consequently, the base-stock level of component i is less than $x_i + \sum_{k \neq i} \Delta_k^i$ at inventory level $\mathbf{x} + \mathbf{\Delta}^i$. As we assume $\Delta_i^i \geq \sum_{k \neq i} \Delta_k^i$, a batch of component i is not produced at inventory level $\mathbf{x} + \mathbf{\Delta}^i$.

The only constraint on inventory allocation decisions is that if a demand for product j is satisfied at inventory level $\mathbf{x}$, then it is satisfied at inventory level $\mathbf{x} + \mathbf{\Delta}_j$. Property (c) of SBSR guarantees that if a demand for product j is satisfied at inventory level $\mathbf{x}$, then it is also satisfied at inventory level $\mathbf{y} \geq \mathbf{x}$.

Hence, any SBSR policy can be replicated by LBLR with an appropriate $\mathbf{\Delta}^i$. □

4. Numerical Experiments

We examine the performance of LBLR relative to SBSR and FBFR, investigating how system parameters affect the relative costs of each policy. For ease of exposition, we initially focus on 2-component 2-product systems in which either (1) products are *nested*—one product requires a subset of components used by the other product (Subsection 4.1), or (2) products are not nested (Subsection 4.2). Our regression results indicate that the gap between LBLR and SBSR decreases with the ratio of lost sale costs in the nested structure (p-value of 0.001), while there is no such monotonic relationship in the non-nested structure.

Altogether we examine 350 instances in subsections 4.1 and 4.2. After comparing computational efforts in subsection 4.3, we report numerical results for 24 selected larger instances in subsection 4.4. Finally, to draw more general conclusions about LBLR, we compare its cost to that of the optimal policy on 22,500 instances in subsection 4.5.

To construct our 2-component 2-product systems, we select two products from a set of four (A, B, C, and D), each of which requires different amounts of two different components (ϕ and γ):

	A	B	C	D
ϕ	1	1	2	1
γ	1	2	1	3

For each of our 2-component 2-product systems we generate instances by varying values of q_i, h_i, c_j, and λ_j, assuming linear holding cost rates (i.e., $h_i(x_i) = h_i x_i$). We impose $\bar{x}_i = 10$, $\forall i$, in all instances. For each instance, we solve the LP and MIP problems to find the minimum average costs and corresponding product fill rates (denoted by f_j). We compare the heuristic policies in terms of (i) their percentage differences from optimal cost Z^*, calculated as $100 \times \frac{Z_H - Z^*}{Z^*}$ where $H \in \{\text{LBLR,SBSR,FBFR}\}$; and (ii) their computation times. We coded the LP and MIP formulations in the Java programming language, incorporating CPLEX 12.5 optimization package, and used a dual processor WinNT server, with Intel Core i7 2.67 GHz processor and 8 GB of RAM. We restricted the computation time of any instance to be no more than 1000 seconds.

If we increase $\bar{x}_i$ from 10 to 11, $\forall i$, the globally optimal cost decreases by no more than 2.67% and the average percentage decrease is 0.31% for the 350 compiled instances in subsections 4.1 and 4.2. This may suggest that we should impose a larger bound such that the globally optimal cost stays the same. However, since the SBSR computation times exceed 1000 seconds in some instances when $\bar{x}_i = 10$, $\forall i$, increasing the upper bound can lead to greater costs for SBSR.

Although our configurations in subsections 4.1, 4.2, and 4.4 violate the sufficient conditions ensuring the optimality of LBLR in Nadar et al. (2014), *LBLR, using Rule of Thumb 1, yields the globally optimal cost in each of those instances that could be solved within 5 hours*. (We verified that this result holds for the instances in subsections 4.1 and 4.2 with $\bar{x}_i = 40, \forall i$.) This motivates our examination in subsection 4.5, where we generate 22,500 instances of general 2-component 2-product systems: *LBLR yields the globally optimal cost in all of these instances as well.*

4.1. Nested Structure

We consider three different examples: (a) An ATO system with products A and D, $q_\phi = 1$, and $q_\gamma = 3$; (b) an ATO system with products A and B, $q_\phi = 1$, and $q_\gamma = 2$; and (c) an ATO system with products A and B, and $q_\phi = q_\gamma = 1$. In each example we vary the holding cost rates of the components and the ratio of lost sale costs of the products, all else being equal. Also, we vary demand rates, all else being equal. *LBLR yields the globally optimal cost in all instances.* (LBLR continues to yield the globally optimal cost in all instances even when $q_\phi, q_\gamma \in \{1,2,3,4,5\}$.) The percentage differences for SBSR and FBFR are only sufficiently large to convey meaningful information in Example (a), so we relegate the numerical results for Examples (b) and (c) to the online appendix. However, we will study each example in a separate regression analysis. An explanation of the lower percentage differences in Examples (b) and (c) is that smaller component usage rates lead to fewer lattices, making use of LBLR less important.

4.1.1. LBLR vs. SBSR. We observe from Table 1 that, for fixed holding cost rates, the largest two gaps always occur when the ratio of lost sale costs is 0.2 or 0.4: Products become less differentiated when the ratio increases, and therefore they should be treated as if they are almost equally important in stock allocation decisions, decreasing the benefit of a lattice-dependent rationing policy. An important insight here is that product differentiation is driven *both* by differences in lost sale costs and component usage rates. Thus, when the ratio of lost sale costs is sufficiently large but lower than 1 (say 0.6 and 0.8), we expect products A and D to be only slightly differentiated, since product A requires fewer components. But, when the ratio is 1, products again become significantly differentiated, due to the difference in

Table 1 Numerical Results for Nested Structure

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_A/c_D	Average cost	f_A	f_D	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	54.974	0.160	0.707	0.000	1.397	1.481	0.39	138.74	2.44
–	–	0.4	69.827	0.300	0.669	0.000	1.054	1.293	0.41	112.54	2.09
–	–	0.6	83.416	0.340	0.642	0.000	0.495	0.513	0.37	20.03	1.65
–	–	0.8	96.217	0.407	0.582	0.000	0.178	0.312	0.38	18.16	2.93
–	–	1.0	106.280	0.631	0.360	0.000	0.123	0.993	0.44	17.05	2.31
–	3	0.2	63.591	0.213	0.690	0.000	2.000	2.511	0.40	**1000**	4.02
–	–	0.4	78.085	0.316	0.651	0.000	1.550	2.456	0.38	68.87	2.21
–	–	0.6	91.221	0.381	0.599	0.000	0.850	1.879	0.37	18.71	2.86
–	–	0.8	102.751	0.474	0.508	0.000	0.364	1.789	0.36	14.59	2.62
–	–	1.0	111.582	0.629	0.356	0.000	0.218	2.273	0.43	11.29	2.46
–	5	0.2	71.364	0.244	0.668	0.000	2.476	3.444	0.41	**1000**	4.56
–	–	0.4	85.140	0.358	0.610	0.000	1.711	3.390	0.40	132.15	3.87
–	–	0.6	97.362	0.423	0.551	0.000	1.014	2.668	0.39	81.19	2.30
–	–	0.8	107.718	0.511	0.466	0.000	0.738	2.288	0.37	14.17	2.17
–	–	1.0	116.091	0.623	0.358	0.000	0.129	3.165	0.36	9.99	2.24
3	1	0.2	61.368	0.112	0.689	0.000	0.917	1.121	0.32	60.17	3.29
–	–	0.4	76.644	0.328	0.632	0.000	0.620	0.677	0.43	23.43	2.73
–	–	0.6	89.403	0.389	0.598	0.000	0.377	0.391	0.35	13.77	1.72
–	–	0.8	101.044	0.451	0.541	0.000	0.223	0.297	0.43	16.10	2.40
–	–	1.0	110.406	0.608	0.385	0.000	0.033	0.604	0.36	13.98	1.67
–	3	0.2	70.509	0.132	0.670	0.000	1.103	2.088	0.33	18.15	3.86
–	–	0.4	85.362	0.337	0.617	0.000	1.023	1.812	0.35	110.82	3.83
–	–	0.6	97.654	0.429	0.555	0.000	0.668	1.589	0.37	24.11	3.06
–	–	0.8	108.447	0.500	0.487	0.000	0.351	1.647	0.43	16.14	2.97
–	–	1.0	116.867	0.615	0.375	0.000	0.436	2.347	0.42	15.42	2.33
–	5	0.2	78.196	0.150	0.652	0.000	1.270	2.136	0.40	127.20	1.97
–	–	0.4	92.564	0.373	0.578	0.000	1.481	2.639	0.38	183.87	3.79
–	–	0.6	104.024	0.466	0.515	0.000	0.710	2.623	0.37	19.34	4.33
–	–	0.8	113.995	0.531	0.453	0.000	0.409	2.683	0.37	17.70	2.82
–	–	1.0	122.202	0.622	0.365	0.000	0.222	2.885	0.37	16.22	2.53
5	1	0.2	65.655	0.123	0.664	0.000	0.786	1.250	0.31	18.92	3.03
–	–	0.4	81.147	0.328	0.607	0.000	0.755	0.927	0.39	20.05	3.11
–	–	0.6	93.924	0.407	0.561	0.000	0.103	0.152	0.35	16.73	2.15
–	–	0.8	105.146	0.483	0.505	0.000	0.444	0.587	0.35	24.70	3.00
–	–	1.0	114.037	0.588	0.406	0.000	0.030	0.349	0.37	11.03	1.19
–	3	0.2	75.148	0.137	0.646	0.000	0.816	2.354	0.32	29.74	5.13
–	–	0.4	90.404	0.336	0.590	0.000	0.907	1.584	0.40	27.17	5.09
–	–	0.6	102.664	0.450	0.518	0.000	0.363	1.257	0.43	12.98	3.12
–	–	0.8	113.290	0.553	0.429	0.000	0.180	1.832	0.36	12.95	7.91
–	–	1.0	121.537	0.606	0.384	0.000	0.310	2.120	0.44	11.71	2.26
–	5	0.2	82.579	0.151	0.612	0.000	0.679	1.672	0.32	34.04	5.68
–	–	0.4	97.557	0.355	0.556	0.000	1.332	2.829	0.40	167.97	6.36
–	–	0.6	109.376	0.478	0.484	0.000	0.434	2.200	0.40	37.17	3.11
–	–	0.8	119.242	0.576	0.397	0.000	0.174	2.486	0.42	14.57	2.62
–	–	1.0	127.367	0.612	0.373	0.000	0.422	2.850	0.43	15.80	3.01

Notes. $q_\phi = 1$, $q_\gamma = 3$, $\lambda_A = \lambda_D = 1$, $\mu_\phi = \mu_\gamma = 1$, $c_D = 100$. Computation times equal to 1000 seconds indicate termination of the algorithm.

component usage rates. This explains why the fill rates of product D are lower than those of product A when the ratio is 1. However, such differentiation results in smaller optimal cost gaps than when the ratio is 0.2 and 0.4.

We next examine the percentage gaps under different holding cost rates when c_A/c_D is equal to 0.2. As h_ϕ increases while h_γ is fixed, the gap declines. However, as h_γ increases while h_ϕ is fixed, the gap increases (there is a minor exception at $h_\phi = 5$). As h_ϕ increases, inventory control decisions rely more heavily on component ϕ, and therefore, since products A and D use the same number of component ϕ (but different numbers of component γ), SBSR better mimics LBLR and the gap diminishes. But the reverse is true as h_γ increases. Also note that the gap declines as both h_ϕ and h_γ increase: Higher holding cost rates lead to less inventory in the system, shrinking the action space and the number of actions in which LBLR and SBSR differ.

We list computation times for the heuristics in the last three columns of this and subsequent tables. It is clear LBLR has distinct computational advantage over SBSR, and a slight one over FBFR. We discuss computation times in greater detail in subsection 4.3.

We next vary demand arrival rates, in Table 2. For a fixed demand rate of product A, the largest two gaps always occur when the demand rate of product D is 0.5 and 1. When λ_D takes greater values, the cost of rejecting the demand per unit time for product D relative to the cost of rejecting all demands per unit time (i.e., $\frac{\lambda_D c_D}{\lambda_A c_A + \lambda_D c_D}$) is higher. Since product D has a greater impact on total costs, product D dominates product A and the system is close to the one with a single product (where SBSR is optimal). But, when λ_D is 0.5 or 1, since product D has a higher lost sale cost, the effect of product dominance is less significant and LBLR can outperform SBSR by a couple of percent. Also, observe that as λ_A increases while λ_D is 0.5, the gap declines (there is a minor exception at $\lambda_A = 1.5$), but as λ_A increases while λ_D is 1, the gap first increases and then decreases. Our explanation is again related to dominance; when the arrival rates are comparable, system performance can be improved by LBLR. Finally, as λ_A increases while λ_D is 1.5, the gap increases, again for the same reason. We expect the gap to fall at higher values of λ_A, since product A will eventually dominate product D.

Another important observation from Table 2 is that, as both demand arrival rates go from 0.5 to 2.5, the gap first increases and then declines. When capacity is high relative to demand (i.e., $\lambda_A = \lambda_D = 0.5$), it is optimal to hold less inventory and therefore the benefit of LBLR is lower. When capacity is scarce (i.e., $\lambda_A, \lambda_D \geq 1.5$), the system focuses more on filling the high value item, even under high base-stock levels. Consequently, it is not critical to ration inventory in a sophisticated manner, and again the benefit of LBLR is lower.

Our overall conclusion is that LBLR may substantially outperform SBSR when demands for both products are fulfilled in significant quantities, when products are highly differentiated, or when products differ mainly in their lost sale costs. Thus we predict that the gap between LBLR and SBSR will increase as the fill rates of both products increase, as the difference of fill rates increases, or as the ratio of lost sale costs decreases. To test these predictions we use the data in Tables 1 and 2 in a regression model for the percentage gap between SBSR and LBLR with the following independent variables: (i) f_A, (ii) $f_D - f_A$, and (iii) c_A/c_D. As we report in Table 3, variables (i)–(iii)

Table 2 Numerical Results for Nested Structure

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_A	λ_D	Average cost	f_A	f_D	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	38.387	0.506	0.712	0.000	2.157	2.670	0.30	78.55	1.99
–	1.0	62.807	0.199	0.693	0.000	0.777	1.616	0.35	358.24	2.83
–	1.5	96.090	0.000	0.559	0.000	0.081	1.157	0.35	7.68	1.47
–	2.0	138.053	0.000	0.440	0.000	0.056	1.107	0.37	3.96	1.82
–	2.5	183.980	0.000	0.359	0.000	0.035	1.088	0.35	3.27	2.00
1.0	0.5	44.757	0.544	0.679	0.000	1.562	3.702	0.32	84.76	5.45
–	1.0	71.364	0.244	0.668	0.000	2.476	3.444	0.41	**1000**	4.56
–	1.5	106.032	0.045	0.552	0.000	0.128	1.103	2.80	11.18	2.95
–	2.0	148.053	0.000	0.440	0.000	0.052	1.032	0.37	4.60	1.48
–	2.5	193.980	0.000	0.359	0.000	0.034	1.032	0.33	8.08	4.32
1.5	0.5	52.369	0.433	0.667	0.000	1.566	4.785	0.34	30.30	3.97
–	1.0	80.498	0.194	0.659	0.000	2.052	4.129	0.36	**1000**	9.53
–	1.5	115.877	0.035	0.551	0.000	0.251	1.143	2.98	225.49	4.02
–	2.0	158.053	0.000	0.440	0.000	0.049	0.967	0.36	7.62	1.53
–	2.5	203.980	0.000	0.359	0.000	0.032	0.981	0.36	7.02	1.64
2.0	0.5	61.127	0.326	0.671	0.000	1.276	4.370	0.39	81.51	2.90
–	1.0	90.017	0.157	0.644	0.000	1.714	4.064	0.35	730.47	3.76
–	1.5	125.770	0.031	0.550	0.000	0.316	1.139	3.62	55.29	2.87
–	2.0	168.053	0.000	0.440	0.000	0.046	0.909	0.35	4.13	2.12
–	2.5	213.980	0.000	0.359	0.000	0.030	0.936	0.35	15.42	1.48
2.5	0.5	70.416	0.259	0.678	0.000	0.980	3.616	0.36	43.09	5.89
–	1.0	99.717	0.129	0.640	0.000	1.451	3.745	0.39	548.25	3.57
–	1.5	135.675	0.031	0.549	0.000	0.363	1.125	4.01	221.41	3.03
–	2.0	178.053	0.000	0.440	0.000	0.043	0.858	0.33	7.86	3.71
–	2.5	223.980	0.000	0.359	0.000	0.029	0.894	0.34	5.84	1.47

Notes. $q_\phi = 1$, $q_\gamma = 3$, $h_\phi = 1$, $h_\gamma = 5$, $\mu_\phi = \mu_\gamma = 1$, $c_A = 20$, $c_D = 100$. Computation times equal to 1000 seconds indicate termination of the algorithm.

have the predicted sign and are statistically significant at $p = 0.001$. The results continue to hold when stepwise regression is used by including all the candidate variables (i.e., system parameters) in the model and eliminating those that are statistically insignificant.

The above prediction remains true in Example (b), but not in Example (c). We report the regression results of Example (b) in Table 3. The ambiguity in Example (c) arises because the lower batch sizes in Example (c) require less flexibility in inventory control decisions, enabling SBSR to perform very well.

4.1.2. LBLR vs. FBFR. As expected, the percentage gaps between LBLR and FBFR are higher than the ones between LBLR and SBSR. In Table 1, in contrast to the comparison of LBLR and SBSR, we observe significant gaps between LBLR and FBFR when products differ only in their component usage rates (i.e., when $c_A/c_D = 1$). This benefit comes from the coordination of the components achieved by LBLR and SBSR but not FBFR: Since batch sizes for components ϕ and γ are 1 and 3, respectively, it is easier to match supply with the demand of product D (using 1 and 3 units of components ϕ and γ), compared to product A (using 1 unit of each component). Hence, it becomes more crucial to coordinate inventory decisions when product A becomes more important, as is the case when $c_A/c_D = 1$. Likewise, Table 2 indicates that the gaps between FBFR and the other heuristics are noticeably higher when product A is more highly demanded (especially when $\lambda_D \leq 1 \leq \lambda_A$). These observations underscore the importance of the coordinated inventory decisions when the component batch sizes imperfectly match the component usage rates of the most valuable and/or mostly demanded product.

Table 3 Regression Results

Variable	Estimate	SE	*t*-statistic	*p*-value
4.1(a). Products A and D, $q_\phi = 1$, and $q_\gamma = 3$				
Intercept	−0.5117	0.3697	−1.3839	0.1711
c_A/c_D	−2.0086	0.4289	−4.6835	0.0000*
f_A	5.1516	0.4806	10.7194	0.0000*
$f_D - f_A$	2.4109	0.5836	4.1308	0.0001*
$N = 70$, $R^2 = 69.62\%$, and adjusted-$R^2 = 68.24\%$.				
4.1(b). Products A and B, $q_\phi = 1$, and $q_\gamma = 2$				
Intercept	−0.2409	0.2174	−1.1081	0.2718
c_A/c_B	−1.8301	0.2808	−6.5185	0.0000*
f_A	4.0457	0.3309	12.2249	0.0000*
$f_B - f_A$	1.6889	0.3043	5.5500	0.0000*
$N = 70$, $R^2 = 76.82\%$, and adjusted-$R^2 = 75.76\%$.				
4.2(a). Products B and C, $q_\phi = 2$, and $q_\gamma = 2$				
Intercept	−1.6409	0.2311	−7.1005	0.0000*
f_B	3.2893	0.3740	8.7961	0.0000*
$f_C - f_B$	3.5478	0.3588	9.8868	0.0000*
$N = 70$, $R^2 = 59.37\%$, and adjusted-$R^2 = 58.15\%$.				
4.2(b). Products C and D, $q_\phi = 2$, and $q_\gamma = 3$				
Intercept	−2.0301	0.4879	−4.1613	0.0000*
f_C	4.4066	0.8245	5.3445	0.0000*
$f_D - f_C$	4.4873	0.8490	5.2856	0.0000*
$N = 70$, $R^2 = 30.43\%$, and adjusted-$R^2 = 28.35\%$.				

*The corresponding variable is statistically significant at probability of 0.001.

4.2. Non-Nested Structure

We consider two different examples: (a) An ATO system with products B and C, and $q_\phi = q_\gamma = 2$; and (b) an ATO system with products C and D, $q_\phi = 2$, and $q_\gamma = 3$. *LBLR yields the globally optimal cost in all instances.* (LBLR continues to yield the globally optimal cost in all instances even when $q_\phi, q_\gamma \in \{1, 2, 3, 4, 5\}$.) Since the basic insights gained from Example (a) can be extended to Example (b), we relegate the numerical results of Example (b) to the online appendix. However, we will again study each example in a separate regression study.

4.2.1. LBLR vs. SBSR. We note from Table 4 that, for fixed holding costs, LBLR provides the least savings when c_B/c_C is 0.6 (there is a minor exception when $h_\phi = 5$ and $h_\gamma = 3$). For smaller values of c_B/c_C, products are highly differentiated and therefore lattice-dependent rationing greatly improves the system performance. For higher values of c_B/c_C, products are almost equally important since the total numbers of components they require are equal. Nevertheless, when c_B/c_C is greater than 0.6, there are cases where the optimal cost gaps between LBLR and SBSR are comparatively large. To understand why this happens, we examined the optimal solutions when c_B/c_C is 1: If inventory levels are equal and sufficiently great to satisfy any demand, it is optimal to satisfy demands of both products. However, if the inventory level of one component is much greater than that of the other, it may be optimal to reject demand of the product that uses a greater number of the scarce component. SBSR cannot induce this kind of structure, but LBLR does.

We next consider the percentage gaps between LBLR and SBSR under different holding cost rates when c_B/c_C is 0.2. In these cases LBLR provides the greatest cost advantage when $h_\phi = 5$ and $h_\gamma = 1$, and the smallest cost advantage when $h_\phi = 1$ and $h_\gamma = 5$. These correspond to the cases when the fill rate of product B takes the greatest and lowest values, respectively. Any increment in h_γ (or h_ϕ) hurts product B (or C) more since product B (or C) requires a greater number of component γ (or ϕ). Hence, when h_γ is higher, product C is so valuable that demands for product B are rejected most of the time and stock rationing becomes less critical.

We now vary demand arrival rates, in Table 5. Our conclusions from the nested structure remain valid:

Table 4 Numerical Results for Non-Nested Structure

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_B/c_C	Average cost	f_B	f_C	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	45.970	0.135	0.800	0.000	1.416	2.291	0.36	**1000**	2.32
–	–	0.4	61.586	0.313	0.743	0.000	0.671	1.416	0.41	345.36	2.50
–	–	0.6	72.943	0.505	0.654	0.000	0.090	0.106	0.39	10.51	1.88
–	–	0.8	82.243	0.560	0.615	0.000	0.554	0.554	0.39	23.26	2.21
–	–	1.0	90.722	0.589	0.589	0.000	0.257	0.257	0.38	13.45	2.11
–	3	0.2	52.497	0.138	0.778	0.000	1.158	2.101	0.36	51.67	1.62
–	–	0.4	68.437	0.294	0.727	0.000	0.576	2.588	0.36	21.51	2.04
–	–	0.6	80.874	0.456	0.653	0.000	0.117	0.904	0.41	14.29	2.39
–	–	0.8	90.622	0.565	0.595	0.000	0.559	0.818	0.36	20.31	1.93
–	–	1.0	98.944	0.605	0.567	0.000	0.609	0.658	0.37	21.56	2.74
–	5	0.2	57.452	0.122	0.764	0.000	0.782	2.256	0.35	39.97	2.85
–	–	0.4	73.412	0.255	0.726	0.000	0.646	2.921	0.37	36.04	2.33
–	–	0.6	86.437	0.411	0.660	0.000	0.174	1.459	0.36	13.73	1.52
–	–	0.8	97.158	0.516	0.604	0.000	0.899	1.608	0.35	26.59	2.96
–	–	1.0	106.030	0.587	0.555	0.000	0.277	0.705	0.38	22.90	2.91
3	1	0.2	54.913	0.199	0.783	0.000	1.377	2.509	0.37	728.96	3.73
–	–	0.4	69.340	0.346	0.733	0.000	0.798	1.196	0.41	56.62	2.85
–	–	0.6	80.220	0.499	0.658	0.000	0.131	0.143	0.39	18.52	1.93
–	–	0.8	90.026	0.539	0.629	0.000	0.190	0.231	0.37	23.94	2.27
–	–	1.0	98.944	0.567	0.605	0.000	0.609	0.658	0.36	21.55	2.80
–	3	0.2	61.973	0.183	0.763	0.000	1.272	3.135	0.37	273.83	4.18
–	–	0.4	76.955	0.306	0.725	0.000	0.720	2.196	0.34	31.56	1.93
–	–	0.6	88.944	0.473	0.651	0.000	0.111	0.768	0.42	15.74	1.99
–	–	0.8	98.931	0.554	0.607	0.000	0.171	0.620	0.38	24.46	2.97
–	–	1.0	107.503	0.586	0.586	0.000	0.649	0.680	0.37	20.58	2.71
–	5	0.2	67.262	0.165	0.748	0.000	0.990	2.795	0.37	494.28	2.88
–	–	0.4	82.624	0.272	0.721	0.000	0.736	2.585	0.36	166.12	2.57
–	–	0.6	95.092	0.426	0.656	0.000	0.160	1.287	0.34	15.08	2.06
–	–	0.8	105.871	0.493	0.627	0.000	0.397	1.269	0.37	22.49	2.67
–	–	1.0	115.073	0.570	0.584	0.000	0.655	1.122	0.36	30.54	3.48
5	1	0.2	62.755	0.227	0.750	0.000	1.613	2.680	0.34	**1000**	3.75
–	–	0.4	76.537	0.366	0.708	0.000	0.784	1.303	0.36	251.68	3.02
–	–	0.6	87.039	0.500	0.642	0.000	0.227	0.511	0.36	19.24	1.72
–	–	0.8	96.899	0.517	0.621	0.000	0.071	0.517	0.38	19.39	2.57
–	–	1.0	106.030	0.555	0.587	0.000	0.277	0.705	0.35	22.28	3.07
–	3	0.2	69.789	0.173	0.724	0.000	1.550	2.015	0.34	940.33	2.42
–	–	0.4	84.698	0.312	0.703	0.000	0.702	1.975	0.37	73.35	3.45
–	–	0.6	96.246	0.481	0.630	0.000	0.062	1.007	0.34	16.87	2.90
–	–	0.8	106.112	0.516	0.613	0.000	0.098	0.812	0.36	20.54	3.01
–	–	1.0	115.073	0.584	0.570	0.000	0.656	1.122	0.37	28.36	2.95
–	5	0.2	74.924	0.144	0.703	0.000	0.814	2.156	0.35	73.13	2.07
–	–	0.4	90.553	0.284	0.694	0.000	0.550	2.294	0.37	40.73	2.74
–	–	0.6	102.748	0.446	0.621	0.000	0.169	1.373	0.34	15.12	1.96
–	–	0.8	113.464	0.490	0.608	0.000	0.296	1.385	0.37	18.11	3.23
–	–	1.0	123.004	0.564	0.564	0.000	0.599	1.729	0.42	19.59	3.49

Notes. $q_\phi = q_\gamma = 2$, $\lambda_B = \lambda_C = 1$, $\mu_\phi = \mu_\gamma = 1$, $c_C = 100$. Computation times equal to 1000 seconds indicate termination of the algorithm.

As one product grows more dominant, it becomes less critical to ration inventory, and the gap between LBLR and SBSR decreases. Likewise, when capacity becomes scarce or high relative to demand, it is not critical to ration inventory in a sophisticated manner, and therefore the gap shrinks. Also, notice that the gap between LBLR and SBSR is significant even when λ_B is 2.5 and λ_C is 0.5, due to the lower lost sale cost of product B.

Based on the previous findings, we again predict that the gap between LBLR and SBSR increases with the product fill rates or difference of fill rates. To test this prediction, we use the data in Tables 4 and 5, and develop a regression model with two independent variables: (i) f_B and (ii) $f_C - f_B$. Unlike the nested case, we excluded c_B/c_C from the regression model due to its nonmonotonic relationship with our dependent variable, the percentage gap between LBLR and SBSR.

Table 5 Numerical Results for Non-Nested structure

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_B	λ_C	Average cost	f_B	f_C	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	32.743	0.506	0.794	0.000	1.426	2.147	0.45	22.59	2.20
–	1.0	53.926	0.259	0.758	0.000	1.643	1.901	0.41	**1000**	3.73
–	1.5	84.862	0.026	0.626	0.000	0.015	0.516	0.38	19.98	2.69
–	2.0	126.450	0.000	0.488	0.000	0.000	0.275	0.40	8.51	3.19
–	2.5	172.774	0.000	0.395	0.000	0.002	0.149	0.38	11.32	1.63
1.0	0.5	39.922	0.456	0.803	0.000	1.265	2.167	0.35	110.82	2.32
–	1.0	62.755	0.227	0.750	0.000	1.613	2.680	0.41	**1000**	3.66
–	1.5	94.745	0.034	0.622	0.000	0.137	0.586	0.39	28.89	2.60
–	2.0	136.450	0.000	0.488	0.000	0.000	0.255	0.38	6.95	2.75
–	2.5	182.774	0.000	0.395	0.000	0.000	0.141	0.38	8.91	1.94
1.5	0.5	47.885	0.380	0.794	0.000	1.134	2.381	0.35	39.28	2.62
–	1.0	72.092	0.176	0.745	0.000	1.364	2.283	0.35	**1000**	3.14
–	1.5	104.645	0.029	0.621	0.000	0.220	0.626	0.36	139.78	3.06
–	2.0	146.450	0.000	0.488	0.000	0.000	0.238	0.38	8.78	2.03
–	2.5	192.774	0.000	0.395	0.000	0.000	0.134	0.38	8.04	2.71
2.0	0.5	56.723	0.310	0.778	0.000	0.883	1.849	0.35	8.51	5.21
–	1.0	81.721	0.132	0.751	0.000	1.202	1.948	0.39	**1000**	4.15
–	1.5	114.577	0.024	0.620	0.000	0.260	0.631	0.35	704.68	3.27
–	2.0	156.450	0.000	0.488	0.000	0.003	0.222	0.41	9.31	3.71
–	2.5	202.774	0.000	0.395	0.000	0.002	0.127	0.37	7.43	1.97
2.5	0.5	66.026	0.261	0.773	0.000	0.729	1.558	0.37	31.06	2.50
–	1.0	91.469	0.109	0.748	0.000	1.092	1.705	0.39	**1000**	5.19
–	1.5	124.528	0.021	0.620	0.000	0.279	0.620	0.36	484.24	3.28
–	2.0	166.450	0.000	0.488	0.000	0.000	0.209	0.35	11.83	3.44
–	2.5	212.774	0.000	0.395	0.000	0.000	0.121	0.35	7.59	1.67

Notes. $q_\phi = q_\gamma = 2$, $h_\phi = 5$, $h_\gamma = 1$, $\mu_\phi = \mu_\gamma = 1$, $c_B = 20$, $c_C = 100$. Computation times equal to 1000 seconds indicate termination of the algorithm.

All the variables have the predicted sign and are statistically significant (see Table 3). The above prediction remains true in Example (b) (see Table 3).

4.2.2. LBLR vs. FBFR. FBFR performs, on average, better than in the nested structure. As component usage rates of both products are closer to component batch sizes, it is easier to match supply with demand, and thus coordination of inventory decisions is less crucial. Furthermore, no matter which product is more valuable or dominant, the degree of difficulty of inventory coordination remains the same since products *B* and *C* are symmetric. Hence, the performances of SBSR and FBFR are closer, although SBSR again significantly outperforms FBFR in many instances.

Table 6 Computation Times (in Seconds)

	LBLR	SBSR	FBFR
Numerical instances in subsection 4.1			
Average	0.59	129.79	3.58
SD	0.87	282.11	1.65
Minimum	0.28	2.58	1.19
Maximum	5.56	1000.00	9.90
Numerical instances in subsection 4.2			
Average	0.37	120.11	2.44
SD	0.23	260.48	0.80
Minimum	0.27	1.98	1.01
Maximum	2.98	1000.00	5.21

Notes. Subsection 4.1 contains 210 compiled instances. Subsection 4.2 contains 140 compiled instances.

4.3. Computational Effort

In Table 6 we report the average, standard deviation, minimum, and maximum computation time for the 350 instances in subsections 4.1 and 4.2 within each heuristic class. (Computation times for the global optimal solution are instantaneous.) Table 6 indicates that LBLR outperforms the other heuristics in terms of average computation times. The computational advantage of LBLR over FBFR is interesting because LBLR has a significantly larger number of base-stock and rationing levels than FBFR. This can be explained by the optimality of LBLR: The MIP typically first solves an LP relaxation, which in all instances yields an integral, and thus optimal solution, with LBLR form. In addition, the range of LBLR computation times is lower within each example, implying that the computation time of LBLR is more robust to parameter change in our instances.

4.4. Selected Larger Instances

We next generate several instances with more components and/or products to determine the maximum problem size that can be solved within a reasonable time for each heuristic:

	Products																
Components	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	q_i	h_i	μ_i
ϕ	1	1	2	1	2	3	2	3	1	1	2	1	2	3	2	1	1
γ	1	2	1	3	2	1	3	2	1	2	1	3	2	1	2	1	1
η	1	1	2	2	1	1	2	2	1						2	1	1
θ	2	2	1												2	1	1
ϑ	1	2													2	1	1
c_j	30	50	40	70	60	50	80	70	25	45	35	65	55	45			
λ_j	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2			

Table 7 exhibits our numerical results; the components and products that we select to construct our instances are shown in the first two columns. We restricted the computation time of each instance to be no more than 5 hours; *LBLR again yields the globally optimal cost in all the instances that could be solved within 5 hours*. (Even when $q_i = q$, $\forall i$, and $q \in \{1,3,4,5\}$, LBLR continues to yield the globally optimal cost in all the instances that could be solved within 5 hours.)

Computation times for each of our heuristics increase considerably with the number of components and/or products. Relatively speaking, an increment in the number of components increases computation times more than an increase in the number of products, since both the state and action spaces rapidly grow with the number of components. For LBLR, we could solve instances with two components and thirteen products, three components and eight products, or four components and two products, within 5 hours. For SBSR, we could solve an instance with two components and seven products within 5 hours. For FBFR, we could solve instances with two components and ten products, or three components and three products, within 5 hours. (We could find global optimal solutions for instances with two components and fifteen products, three components and eleven products, or four components and seven products.)

Below we report the average numbers of lattices per component/product that we need to implement into the MIP formulation of LBLR in our instances of various sizes:

	n												
m	2	3	4	5	6	7	8	9	10	11	12	13	14
2	26	31	38	39	41	46	47	48	48	48	48	48	48
3	371	446	522	545	563	620	633	637					
4	5941	6027											
5	76,065												

The average number of lattices converges as the number of products grows. Thus, for LBLR, the primary cause of the increase in the computation time as the number of *products* increases is due to the growing action space for inventory allocation. However, the average number of lattices rapidly grows with the number of *components*. This conceivably leads to a significant increase in the computation required by LBLR. Nevertheless, the average number of points on any lattice does *not* increase with the number of components, and thus the MIP constraints of LBLR individually become no more burdensome as the state space grows. This gives an explanation for the much lower computation times of LBLR, in comparison with FBFR, in systems with more components.

4.5. LBLR vs. Optimal Policy on a Larger Test Bed

We generated 22,500 instances for five different 2-component 2-product systems in which the products, *j* and *k*, are: (i) *A* and *B*, (ii) *A* and *D*, (iii) *B* and *C*, (iv) *B* and *D*, and (v) *C* and *D*, respectively. For each of these systems, we consider 4,500 instances in which $q_\phi, q_\gamma \in \{1,2,3,4,5\}$, $h_\phi, h_\gamma \in \{1,3,5\}$, $\mu_\phi = \mu_\gamma = 1$, $c_j \in \{20,40,60,80,100\}$, $c_k = 100$, and $\lambda_j, \lambda_k \in \{0.5,1\}$. (Some of these instances overlap with those in subsections 4.1 and 4.2.) *LBLR, using Rule of Thumb 1, continues to yield the globally optimal cost in all of these instances.*

5. Concluding Remarks

We have studied the LBLR policy for Markovian ATO systems with general product structures. We analytically and numerically compare the LBLR policy to two other heuristics from the literature: the SBSR policy and the FBFR policy, establishing the superiority of LBLR. In addition, we numerically show that LBLR minimizes the average costs in each of the more than 22,500 instances of general ATO problems we tested.

Identifying the optimal policy structure in our numerical experiments enables us to uncover the role of different product characteristics in optimal control of ATO systems. Our numerical experiments also

Table 7 Numerical Results for Selected Larger Instances

		Optimal solution		Heuristic solutions			Computation times		
Components	Products	Average cost	Computation time	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
ϕ and γ	*A* and *B*	7.076	0.13	7.076	7.076	7.097	0.39	10.29	1.23
–	*A*,*B*, and *C*	9.765	0.27	9.765	9.765	9.825	0.57	22.94	3.60
–	*A*–*D*	14.674	0.39	14.674	14.674	14.745	1.42	71.67	8.37
–	*A*–*E*	19.434	0.52	19.434	19.434	19.564	2.14	178.66	21.32
–	*A*–*F*	24.996	1.01	24.996	25.049	25.141	5.07	3941.82	69.53
–	*A*–*G*	35.976	2.13	35.976	36.026	36.112	10.58	13,122.17	242.00
–	*A*–*H*	47.412	4.88	47.412	90.000	47.513	23.17	**18,000**	853.98
–	*A*–*I*	51.870	10.30	51.870	51.987	52.007	53.92	**18,000**	11,410.01
–	*A*–*J*	59.723	22.20	59.723	104.000	60.640	141.52	**18,000**	20,750.36
–	*A*–*K*	66.177	49.83	66.177	111.000	*	300.10	**18,000**	*
–	*A*–*L*	78.074	108.40	78.074	124.000	*	1438.19	**18,000**	*
–	*A*–*M*	88.648	236.45	88.648	*	*	11,407.78	*	*
–	*A*–*N*	97.126	521.87	*	*	*	*	*	*
ϕ,γ, and η	*A* and *B*	9.471	0.79	9.471	16.000	9.570	10.04	**18,000**	1815.83
–	*A*, *B*, and *C*	13.548	2.37	13.548	24.000	13.660	222.76	**18,000**	12,453.59
–	*A*–*D*	20.141	5.31	20.141	38.000	65.000	773.80	**18,000**	**18,000**
–	*A*–*E*	25.464	11.04	25.464	50.000	*	3783.45	**18,000**	*
–	*A*–*F*	31.283	25.15	31.283	62.000	*	2298.46	**18,000**	*
–	*A*–*G*	42.429	55.32	42.429	103.000	*	1566.59	**18,000**	*
–	*A*–*H*	53.995	128.41	53.995	117.000	*	5492.18	**18,000**	*
–	*A*–*I*	58.525	267.37	105.000	*	*	**18,000**	*	*
ϕ,γ,η, and θ	*A* and *B*	12.489	25.59	12.489	52.000	*	11,342.35	**18,000**	*
–	*A*, *B*, and *C*	17.883	172.24	26.000	60.000	*	**18,000**	**18,000**	*
ϕ,γ,η,θ, and ϑ	*A* and *B*	*	*	*	*	*	*	*	*

*The MIP solver fails to report a feasible solution as it runs out of memory. Computation times equal to 18,000 seconds indicate termination of the algorithm.

reveal when SBSR and FBFR significantly deviate from the optimal policy, producing high-level guidelines for control policy choices in a variety of settings. Specifically, LBLR performs significantly better than SBSR (by up to 2.6% of the optimal cost) when products are highly differentiated and it is optimal to fulfill a significant fraction of the demand for each product. FBFR performs substantially worse than the other two heuristics (by up to 4.8% of the optimal cost) when replenishment batch sizes imperfectly match the component requirements of the most valuable and/or most highly demanded product. LBLR, despite its complicated structure, could also be easily implemented in practice since the basic easy-to-understand principles of FBFR still hold for LBLR after state space partitioning. For instance, FBFR specifies one rationing level on the entire state space for a particular product. LBLR specifies one rationing level on each of the multiple disjoint lattices of the state space.

We can modify the ATO model in this study by allowing the controller to produce any number of units of each component at any time, extending the replenishment policy of LBLR to this case as follows: Produce κ units of component i (i) if the inventory level is less than the base-stock level on the current lattice, (ii) if the inventory level is less than the base-stock level on the lattice that we reach after producing z units of component i, for all $z \leq \kappa - 1$, and (iii) if the inventory level is no less than the base-stock level on the lattice that we reach after producing κ units of component i. This extended version of LBLR again minimizes the average costs in all the instances in Section 4 that could be solved within 5 hours.

The evidence from our study leads naturally to the conjecture that *LBLR may be optimal for ATO systems with general product structures and lost sales under Markovian assumptions on production and demand.* Furthermore, for LBLR, *the state space of the ATO problem may be optimally partitioned into disjoint lattices based on products' component requirements and lost sales costs, as stated in Rule of Thumb 1.* Our conjecture may guide future research aimed at characterizing the optimal policy structure for general ATO systems. However, the existence of counter examples shows that the functional characterizations used to show the optimality of LBLR in Nadar et al. (2014) need not hold for general product structures. Thus, if LBLR is to be shown to be optimal for general ATO systems, a different methodology will likely be required.

Another direction for future research is to study the performance of LBLR in ATO systems with backordering and/or general component production and demand interarrival times. We could generalize LBLR

and its MIP formulation to models with phase-type component production times and/or compound Poisson demand. But such generalizations come at the expense of increased computational burden since the state and/or action spaces become extremely large. Lastly, future research could develop effective solution procedures for the optimization of lattice-dependent base-stock and rationing levels in high-dimensional ATO problems for which even solving the linear program formulation to optimality might prove problematic. The structural knowledge of the optimal policy gained from our study can potentially inspire and facilitate future research on smarter computational methods.

Acknowledgments

The authors are grateful to the Department Editor, the Senior Editor, and three anonymous referees for their constructive comments and suggestions. They also thank the National Science Foundation (CMMI 1351821 and CMMI 1334194), Bilkent University, and Carnegie Mellon University for financial support.

References

Agrawal, M., M. Cohen. 2001. Optimal material control and performance evaluation in an assembly environment with component commonality. *Nav. Res. Logist.* **48**(5): 409–429.

Akcay, Y., S. H. Xu. 2004. Joint inventory replenishment and component allocation optimization in an assemble-to-order system. *Manage. Sci.* **50**(1): 99–116.

Benjaafar, S., M. ElHafsi. 2006. Production and inventory control of a single product assemble-to-order system with multiple customer classes. *Manage. Sci.* **52**(12): 1896–1912.

Benjaafar, S., M. ElHafsi, C. Y. Lee, W. Zhou. 2011. Optimal control of an assembly system with multiple stages and multiple demand classes. *Oper. Res.* **59**(2): 522–529.

Bhandari, A., A. Scheller-Wolf, M. Harchol-Balter. 2008. An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Manage. Sci.* **54**(2): 339–353.

Dekker, R., R. M. Hill, M. J. Kleijn, R. H. Teunter. 2002. On the $(S-1,S)$ lost sales inventory model with priority demand classes. *Nav. Res. Logist.* **49**(6): 593–610.

Dogru, M. K., M. I. Reiman, Q. Wang. 2010. A stochastic programming based inventory policy for assemble-to-order systems with application to the W model. *Oper. Res.* **58**(4): 849–864.

ElHafsi, M., H. Camus, E. Craye. 2008. Optimal control of a nested-multiple-product assemble-to-order system. *Int. J. Prod. Res.* **46**(19): 5367–5392.

ElHafsi, M. 2009. Optimal integrated production and inventory control of an assemble-to-order system with multiple non-unitary demand classes. *Eur. J. Oper. Res.* **194**(1): 127–142.

Frank, K., R. Q. Zhang, I. Duenyas. 2003. Optimal policies for inventory systems with priority demand classes. *Oper. Res.* **51**(6): 993–1002.

Gayon, J. P., F. de Vericourt, F. Karaesmen. 2009. Stock rationing in an $M/E_r/1$ multi-class make-to-stock queue with backorders. *IIE Trans.* **41**(12): 1096–1109.

Ha, A. 1997. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Manage. Sci.* **43**(8): 1093–1103.

Ha, A. 2000. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Manage. Sci.* **46**(1): 77–87.

Lu, Y., J. S. Song, Y. Zhao. 2010. No-holdback allocation rules for continuous-time assemble-to-order systems. *Oper. Res.* **58**(3): 691–705.

Lu, L., J. S. Song, H. Zhang. 2014. Optimal and asymptotically optimal policies for assemble-to-order N- and W-systems. Working paper, Columbia Business School, New York, NY.

Nadar, E., M. Akan, A. Scheller-Wolf. 2014. Optimal structural results for assemble-to-order generalized *M*-systems. *Oper. Res.* **62**(3): 571–579.

Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY.

Song, J. S., P. Zipkin. 2003. Supply chain operations: Assemble-to-order systems. T. De Kok, S. Graves, eds. *Handbooks in Operations Research and Management Science*, Vol. 11: Supply Chain Management: Design, Coordination, and Operation. Elsevier, Amsterdam, 561–596.

de Véricourt, F., F. Karaesmen, Y. Dallery. 2002. Optimal stock allocation for a capacitated supply system. *Manage. Sci.* **48**(11): 1486–1501.

Zhang, A. X. 1997. Demand fulfillment rates in an assemble-to-order system with multiple products and dependent demands. *Prod. Oper. Manag.* **6**(3): 309–324.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Appendix S1: Formulation of the Heuristics

Appendix S2: Additional Numerical Results

Appendix S3: Counter Examples in the Discounted Cost Case