OR Spectrum (2017) 39:165–191
DOI 10.1007/s00291-016-0448-9

REGULAR ARTICLE

Dynamic control in multi-item production/inventory systems

H. G. H. Tiemessen[1] · M. Fleischmann[2] · G. J. van Houtum[3]

Received: 4 November 2014 / Accepted: 19 May 2016 / Published online: 10 June 2016

Abstract We consider a production/inventory system consisting of one production line and multiple products. Finished goods are kept in stock to serve stochastic demand. Demand is fulfilled immediately if there is an item of the requested product in stock and otherwise it is backordered and fulfilled later. The production line is modeled as a non-preemptive single server and the objective is to minimize the sum of the average inventory holding costs and backordering costs. We investigate the structure of the optimal production policy, propose a new scheduling policy, and develop a method for calculating base stock levels under an arbitrary but given scheduling policy. The performance of the various production policies is evaluated in extensive numerical experiments.

Keywords Inventory control · Dynamic scheduling · Simulation optimization

✉ H. G. H. Tiemessen
harold.tiemessen@fhsg.ch

M. Fleischmann
mfleischmann@bwl.uni-mannheim.de

G. J. van Houtum
g.j.v.houtum@tue.nl

1 Institute of Modeling and Simulation, St. Gallen University of Applied Sciences, St. Gallen 9001, Switzerland

2 Business School, University of Mannheim, P.O. Box 10 34 62, 68131 Mannheim, Germany

3 School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

1 Introduction

A common trend in industries where capacity investments are capital-intensive is to invest in flexible manufacturing systems. A flexible manufacturing system is capable of producing a wide range of products and rapidly changing production according to realized demand or forecast changes. Examples include auto-manufacturing, semiconductors and pharmaceuticals. An important reason for this trend is the growing product variety, which causes lower average demand volumes and greater variability in demand for the individual products. As a result, investments in dedicated capacity have become less economical. In these industries, a firm's ability to carefully manage flexible capacity is often a significant factor for its success (cf. Linebaugh 2008).

In this paper, we consider a well-known production control problem, posed in Peña Perez and Zipkin (1997). Although it is an abstract problem, it captures the two key features of any real flexible manufacturing system: (i) several products compete for limited production capacity, and (ii) production can be changed from one product to another rapidly and at no cost. The problem is to operate a single production line that produces multiple products in a make-to-stock mode to manage finished goods inventory with the objective of minimizing average inventory holding and backordering costs. The production line is controlled by a *production policy*. Each production policy answers two questions: (i) when to produce, and (ii) what to produce. Some production policies answer both questions simultaneously, but often production policies are made up of two separate control policies: an *idleness policy* that dictates when the production line is idle or busy and a *scheduling policy* that selects the good to produce in the latter case.

The goal of this study is to develop new production policies and compare their performance against existing policies. In these comparisons, we restrict ourselves to *dynamic* production policies as they usually perform considerably better than static control policies (see e.g., Veatch and Wein 1996).

Our main contribution consists of five parts:

- We propose a new dynamic production policy that is applicable for small and large problem instances. It consists of a *rolling horizon* scheduling policy and a *simulation optimization* method for calculating base stock levels. These base stock levels dictate when production is turned on and off. In extensive numerical experiments with two products, we show that the optimality gap of this production policy is usually small.
- We compare various heuristic production policies on a wide range of *large* problem instances. Such a comparison is lacking in the literature. In extensive numerical experiments, we show that our proposed production policy outperforms the other production policies. In particular, it is better capable of handling situations with non-identical production rates.
- We show that our simulation optimization method for calculating base stock levels is effective and efficient. If applied in combination with the proposed scheduling policy, it calculates base stock levels that are close to optimal. We also show that the method performs well in combination with other scheduling policies.

- We show that the optimal production policy is not necessarily of the base stock type. This is illustrated by an example with two products where the optimal *base stock* production policy performs more than 10 % worse than the overall optimal production policy.
- We show that the optimality gap of the *myopic allocation* policy proposed in Peña Perez and Zipkin (1997) can grow beyond 40 % in the case of non-identical production rates. This is an important insight and contradicts what people generally believe.

Although our work is mainly motivated by the production/inventory control problem for flexible manufacturing systems, it is also relevant for additive manufacturing (often called *3D Printing*), remanufacturing systems, and repairable inventory systems. General purpose 3D printers can build a wide variety of complex products with no change-over times. This perfectly aligns with the assumptions made in this paper. A nice introduction to 3D printing and a discussion of its impact on the supply chain can be found in Janssen et al. (2014). Liu et al. (2014) and Khajavi et al. (2014) more specifically study the impact of 3D printing on spare parts management. An interesting contribution in the field of remanufacturing systems is Lieckens et al. (2013), where the authors develop a decision support tool to optimize the global network design and the demand fulfillment strategy for a manufacturer of compressed air equipment. In their model, operational decisions such as work scheduling at the refurbishment centers are not considered since they assume infinite capacities. In the (realistic) case of capacity constraints, dynamic scheduling policies could be very useful. In Sect. 8, we discuss an interesting future research direction in the field of repairable inventory systems.

The paper is structured as follows. We start with a literature review and position our research in Sect. 2. In Sect. 3, we formulate our model and discuss important assumptions. In Sect. 4, we describe the simulation optimization method for calculating base stock levels. In Sect. 5, we describe four production policies (three new ones and one existing one). In Sect. 6, we carry out extensive numerical experiments to evaluate the performance of the four production policies. In Sect. 7, we investigate the performance of the simulation optimization method. Finally, we give our conclusions and suggestions for future research in Sect. 8.

2 Literature review

The single product version of our problem has been extensively studied in the literature; see Gavish and Graves (1980), Sobel (1982), and Li (1992). In this case, the only decision is when to produce, and under standard cost assumptions a base stock policy is optimal. This means that there is a fixed inventory target (the base stock level), and the production line is busy if inventory is below that target and idle otherwise.

The literature stream on multi-item make-to-stock queues is most relevant to our work. One of the first contributions in this stream is due to Zheng and Zipkin (1990) who consider a system consisting of two symmetric products under base stock control. They study the *Longest Queue* (LQ) policy and compare its performance against FCFS. Zipkin (1995) examines the performance of LQ on systems with more than two products. Van Houtum et al. (1997) derive lower and upper bounds for the mean waiting time for the symmetric longest queue system in order to minimize the base stock levels required to achieve a target fill rate. Wein (1992) allows for asymmetric products and derives an approximating Brownian control problem for the multi-item make-to-stock queuing problem. He uses the solution of this stochastic control problem to propose a production policy consisting of an aggregate base stock policy and a static scheduling policy. Veatch and Wein (1996) investigate several combinations of idleness policies and scheduling policies and show that the combination of the aggregate base stock policy presented in Wein (1992) and a dynamic scheduling policy performs surprisingly well. Peña Perez and Zipkin (1997) propose the *myopic*($\boldsymbol{T}$) scheduling policy. It selects the product that achieves the highest ratio of the expected cost rate reduction at a carefully chosen future time point and its average production time. They combine their scheduling policy with an idleness policy obtained from the optimal aggregate base stock level. They assume non-preemptive processing and show that the resulting production policy has a small optimality gap and outperforms the static production policies. Sanajian et al. (2010) consider a repair shop problem and show in numerical experiments that the optimality gap of the myopic(**T**) policy usually gets smaller if preemption is allowed. In Liang et al. (2013), the authors develop a myopic scheduling policy for repair shops with preemption and state-dependent arrival rates. Ha (1997) provides the theoretical justification of some of the ideas suggested by the approximation model in Wein (1992) and proposes a new dynamic rule for prioritizing products, the *switching rule*. Reported numerical tests with optimized base stock levels suggest that the switching rule has a small optimality gap and outperforms two other scheduling heuristics. Although Ha (1997) recognizes that it would be interesting to compare the switching rule with the myopic(**T**) scheduling policy, such a comparison has never been carried out. Kat and Avşar (2011) study a problem where fixed backordering costs are incurred regardless of the time needed to satisfy the backordered demand. They carried out numerical experiments that suggest that the optimal policy is a base stock policy with switching curves and fixed (not state-dependent) base stock levels. For asymmetric systems, base stock levels must be obtained via an extensive (enumeration type of) search. In a recent contribution, Arreola-Risa et al. (2011) consider a system with symmetric products and propose a heuristic that is based on simulation and regression analysis to optimize the base stock levels.

Two other relevant contributions come from the field of probability theory. Building upon the seminal work of Whittle (1988), Dusonchet (2003) and Niño-Mora (2007) formulate the multi-item make-to-stock queue with average cost criterion as a *Restless Bandit Problem*. Niño-Mora (2007) shows that the marginal productivity index obtained from the restless bandit formulation coincides with the myopic(**T**) scheduling policy in the case of linear holding and backordering costs, and further extends it to models with convex nonlinear cost rates and/or discounted costs.

We conclude this literature review with a short discussion of periodically reviewed systems. DeCroix and Arreola-Risa (1998) propose a balancing rule that uses a target inventory level for each product to divide the available production time in a period across all products. Base stock levels are obtained from a one-dimensional search along the line spanned by the vector of single period news vendor levels. The search is carried out using simulation. Janakiraman et al. (2009) extend this work and propose a weighted balancing rule. They show that this rule is optimal in two asymptotic regimes represented by high service levels and heavy traffic.

3 Problem description and model formulation

In this section, we define the problem, introduce our notation, discuss assumptions, formulate our model, and empirically investigate the structure of the optimal production policy.

3.1 System description and main assumptions

We consider a single location consisting of one production line and one stockpoint, where multiple products are kept in stock to serve stochastic demand. When demand arrives and the requested product is in stock it is immediately fulfilled. Otherwise, it is backordered and fulfilled as soon as an item of the requested product becomes available from the production line. For each product, there is an inventory holding cost and a backordering cost per item per time unit. Our objective is to develop a policy for operating the production line that minimizes the average total cost. This is exactly the setting studied in Peña Perez and Zipkin (1997). The notation of our model is given in Table 1.

Next, we formulate our main assumptions:

- (i) Requests for each product follow independent Poisson processes with constant means.
- (ii) There are ample raw materials; the only limiting resource is the production capacity.
- (iii) Production times for each product are mutually independent. We do not make assumptions regarding the type of distributions. To obtain a stable system, we assume that the system utilization rate is strictly smaller than 1.
- (iv) The production facility is modeled as a single server. This assumption is made to facilitate an exact analysis. In reality, production facilities can often produce

Table 1 Notation

Parameters	
I	Number of products (products are numbered $1, \ldots, I$)
λ_i	Demand rate of product i
$1/\mu_i$	Average production time of product i
h_i	Inventory holding cost per time unit per item of product i
b_i	Backordering cost per time unit per item of product i
Variables	
$\mathbf{z}$	Vector of net inventory levels
S_i	Base stock level for product i

multiple products in parallel, thereby violating the single server assumption. Still, the assumption captures the key feature that several products compete for the same limited production capacity.

(v) Setup costs and change-over times are negligible when switching products.

(vi) Production of an item can *not* be interrupted (non-preemption).

Assumptions (i)–(iv) are common in the make-to-stock literature and are made to facilitate the analysis. Assumption (v) is also common and is justified for flexible manufacturing systems. Assumption (vi) is primarily made to allow a fair comparison of new production policies against the myopic allocation policy, but also makes sense from a practical point of view. First, it can be cumbersome to remove partially manufactured items and the unused input materials from the production line. Second, resuming the manufacturing of an interrupted product at a later time point may be very inefficient. Finally, non-preemption leads to more predictable completion times. This is in particular true in the case of deterministic production times and offers opportunities for offline preparation of the next production run.

3.2 MDP formulation

In this section, we formulate the production control problem with *exponentially* distributed production times as a continuous-time average cost MDP with finite state and control spaces (see e.g., Bertsekas 2007, pp. 310-316). State transitions and action selections take place at time instances when one of the following two event types occurs: (i) the production of an item has just been completed, or (ii) a new product demand has just arrived at the stockpoint. Note that we do *not* require that the calculated production policy belongs to the class of base stock policies.

We describe the states of the system at the moments that events occur by states $\mathbf{x} = (\mathbf{z}, j)$, with $\mathbf{z} = (z_1, \ldots, z_I)$ the I-dimensional vector of net inventory levels, and $j \in \{0, 1, \ldots, I\}$ a reference to the product that is currently being produced. If the production line was idle or if the production of an item was just completed, we set j equal to 0. We define the state space $\mathscr{S}$ as $\mathscr{S} = \{((z_1, \ldots, z_I), j) | z_i \in \mathbb{Z}, \quad i = 1, \ldots, I, \quad j = 0, \ldots, I\}$. The action space for our model is $\mathscr{A} = \{0, 1, \ldots, I\}$. Here, action $j \geq 1$ stands for the decision to produce an item of product j, and action 0 stands for the decision to produce nothing. Because of non-preemption, the following holds for the set of admissible actions $\mathscr{A}(\mathbf{z}, j)$ for each state $(\mathbf{z}, j) \in \mathscr{S}$: $\mathscr{A}(\mathbf{z}, j) = \{j\}$ if $j \geq 1$ and $\mathscr{A}(\mathbf{z}, 0) = \{0, 1, \ldots, I\}$.

Next, we describe the transitions for our MDP formulation. We assume that if the system is in state $\mathbf{x}$ and action a is applied, the next state will be $\mathbf{y}$ with probability $p_{\mathbf{x},\mathbf{y}}(a)$. Furthermore, we define $\mathbf{e}_i$ for $i > 0$ as the I-dimensional unit vector with a 1 at position i.

Transition type 1 initial state: $\mathbf{x} = (\mathbf{z}, j)$ with $j \in \{1, \ldots, I\}$, action: j, next event: arrival of a new demand for product i, next state: $\mathbf{y} = (\mathbf{z} - \mathbf{e}_i, j)$. The transition rate is λ_i and the transition probability $p_{\mathbf{x},\mathbf{y}}(j)$ is equal to $\lambda_i / [\sum_{n=1}^{I} \lambda_n + \mu_j]$.

Transition type 2 initial state: $\mathbf{x} = (\mathbf{z}, j)$ with $j \in \{1, \ldots, I\}$, action: j, next event: production order completed, next state: $\mathbf{y} = (\mathbf{z} + \mathbf{e}_j, 0)$. The transition rate is μ_j and the transition probability $p_{\mathbf{x},\mathbf{y}}(j)$ is equal to $\mu_j / [\sum_{n=1}^{I} \lambda_n + \mu_j]$.

Transition type 3 initial state: $\mathbf{x} = (\mathbf{z}, 0)$, action: $j \in \{1, \ldots, I\}$, next event: arrival of a new demand for product i, next state: $\mathbf{y} = (\mathbf{z} - \mathbf{e}_i, j)$. The transition rate is λ_i and the transition probability $p_{\mathbf{x},\mathbf{y}}(j)$ is equal to $\lambda_i / [\sum_{n=1}^{I} \lambda_n + \mu_j]$.
Transition type 4 initial state: $\mathbf{x} = (\mathbf{z}, 0)$, action: $j \in \{1, \ldots, I\}$, next event: production order completed, next state: $\mathbf{y} = (\mathbf{z} + \mathbf{e}_j, 0)$. The transition rate is μ_j and the transition probability $p_{\mathbf{x},\mathbf{y}}(j)$ is equal to $\mu_j / [\sum_{n=1}^{I} \lambda_n + \mu_j]$.
Transition type 5 initial state: $\mathbf{x} = (\mathbf{z}, 0)$, action: 0, next event: arrival of a new demand for product i, next state: $\mathbf{y} = (\mathbf{z} - \mathbf{e}_i, 0)$. The transition rate is λ_i and the transition probability $p_{\mathbf{x},\mathbf{y}}(0)$ is equal to $\lambda_i / [\sum_{n=1}^{I} \lambda_n]$.

The mean transition period lengths $\tau(\mathbf{x}, a)$ for all state/action pairs directly follow from the transition rates. We conclude our MDP formulation with the specification of the expected single stage cost $K((\mathbf{z}, j), a)$ when choosing action a in state $(\mathbf{z}, j)$. Since there is no cost associated with the action, it holds that $K((\mathbf{z}, j), a) = \tau(\mathbf{x}, a) \sum_{n=1}^{I} [\, h_n \max(z_n, 0) + b_n \max(-z_n, 0)\,]$.

Before we can calculate the optimal production policy from the MDP, we must truncate the state space. For this purpose we introduce for each product i a minimum and a maximum net inventory level (denoted as $L_i^{\min}$ and $L_i^{\max}$, respectively). This requires two small modifications in the transition probabilities and the admissible actions at the borders of the truncated state space. First, for states $(\mathbf{z}, j)$ with $z_i = L_i^{\min}$ for some $i = 1, \ldots, I$, we replace the transition from $(\mathbf{z}, j)$ to $(\mathbf{z} - \mathbf{e}_i, j)$ with a transition from $(\mathbf{z}, j)$ to itself. Second, for states $(\mathbf{z}, 0)$ with $z_i = L_i^{\max}$, we remove action i from the set of admissible actions. The state space for our model now reads as: $\mathscr{S} = \{((z_1, \ldots, z_I), j) | L_i^{\min} \le z_i \le L_i^{\max}, z_i \in \mathbb{Z},\quad i = 1, \ldots, I,\quad j = 0, \ldots, I\}$. We choose $L_i^{\min}$ and $L_i^{\max}$ such that an increase of the state space has no significant impact on the average cost obtained from the MDP with the truncated state space.

To obtain the optimal production policy we transform the continuous-time MDP into a discrete-time MDP by applying a technique called *uniformization* (see e.g., Bertsekas 2007, pp. 288–295). The uniformization procedure consists of three steps: in the first step, we determine a new transition period length τ such that $\tau \le \tau(\mathbf{x}, a)$ for all $\mathbf{x} \in \mathscr{S}$ and $a \in \mathscr{A}$. It is easy to see that $\tau = 1/[\max(\mu_j) + \sum_{n=1}^{I} \lambda_n]$ is an appropriate choice. In the second step, for each state $\mathbf{x}$ and action $a \in \mathscr{A}(\mathbf{x})$, we add an extra outgoing transition to state $\mathbf{x}$ itself and the corresponding rate is set equal to $(1/\tau) - (1/\tau(\mathbf{x}, a))$. By adding these "fictitious" transitions the total outgoing rate becomes equal to $1/\tau$ in each state, while they have no effect on the average costs under any given policy. In the third step, we recalculate the transition probabilities and the expected single stage cost taking into account the fictitious transitions added in the previous step. They are denoted as $\hat{p}_{\mathbf{x},\mathbf{y}}(a)$ and $\hat{K}(\mathbf{x}, a)$, respectively.

We now obtain the desired discrete-time MDP by replacing the exponentially distributed transition period lengths by constant transition period lengths with the same mean. The optimal average cost rate $\zeta^\star$ follows from the Bellman optimality equations for the (discrete-time) average cost MDP:

$$w^\star(\mathbf{x}) = \min_{a \in \mathscr{A}(\mathbf{x})} \left[K(\mathbf{x}, a) - \zeta^\star \tau + \sum_{\mathbf{y} \in \mathscr{S}} \hat{p}_{\mathbf{x},\mathbf{y}}(a) w^\star(\mathbf{y}) \right] \quad \forall \mathbf{x} \in \mathscr{S} \tag{1}$$

The optimal action $a^\star(\mathbf{x})$ is the action that attains the minimum in (1). To solve (1), we use relative value iteration (see e.g., Bertsekas 2007, pp. 204–229). Note that the number of optimality equations grows linearly with the number of states and thus exponentially in the number of products. Consequently, the optimality equations in (1) can only be solved for small problem instances (typically 2–3 products).

3.3 Optimal policy structure

In this section, we empirically investigate the structure of the optimal production policy. To the best of our knowledge, there are no results on the structure of the optimal policy for systems with non-preemptive processing. This means that we may not a priori assume that the optimal production policy belongs to the class of base stock policies. In Example 1, we actually show that the optimal production policy sometimes has a different structure and that limiting the search for good production policies to base stock policies may have a significant negative impact on the achieved average total cost.

Example 1 Consider the multi-item make-to-stock problem defined by the vectors $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) = (1.40, 0.35)$, $\boldsymbol{\mu} = (\mu_1, \mu_2) = (4, 1)$, $\mathbf{h} = (h_1, h_2) = (1.0, 0.5)$, and $\mathbf{b} = (b_1, b_2) = (80, 40)$. Figure 1a shows the optimal production policy and Fig. 1b shows the optimal base stock production policy The optimal production policy has an average total cost of 10.49, whereas the optimal base stock policy (with $\mathbf{S} = (8, 7)$) has an average total cost of 11.68. This is a cost increase of more than 10 % and shows that restricting the search for good production policies to base stock policies can have a significant negative impact on the achieved average total cost.

Figure 1 points out some remarkable properties of the optimal production policy. For net inventory vectors (7, 7), (8, 8), (8, 9), and (9, 10), we see that production is switched off when the on-hand stock of product 1 decreases. At first sight, this seems counterintuitive as for each product the stockout risk increases if on-hand stock decreases. However, the assumption of non-preemptive processing makes things dif-

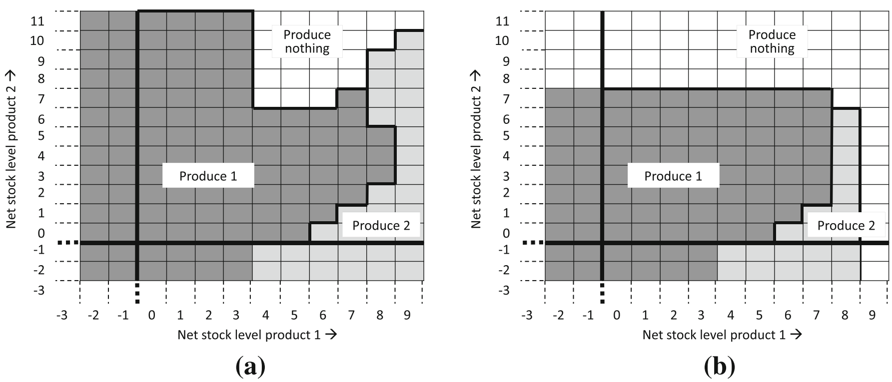

Fig. 1 Optimal production policies for Example 1

ferent. Due to this assumption, there is an option value associated with the decision to leave the production line idle because it keeps the option open to immediately start the production of either of the two products whenever necessary. Naturally, this option value increases as the on-hand stock decreases. Using this option value concept, we are now able to give an intuitive explanation of the optimal policy structure in the upper right part of Fig. 1b. If on-hand stock of product 1 is low, the optimal policy tells to produce product 1 (in an attempt to avoid future backorders of product 1). If on-hand stock of product 1 is medium, the optimal policy tells to switch off the production line (thereby keeping the option open to start producing product 1 immediately). Only if on-hand stock for product 1 increases further, it is worth to give up this option and start producing product 2. Note that $b_1 = 80 > b_2 = 40$ and $\mu_1 = 4 > \mu_2 = 1$, i.e., for product 2 backorder costs are low and average production time is long. This may explain why the effect occurs as a function of the inventory level of product 1. Example 1 also raises the question if the optimal production policy belongs to the class of base stock policies if we allow *preemptive* production scheduling. Numerical investigations suggest that this is indeed the case. However, we cannot generalize this to general preemptive production systems. For such systems, we know that base stock policies are optimal for two products, discounted cost, and identical production rates (cf. Ha 1997). De Vericourt et al. (2000) extend Ha (1997) and show that the myopic(**T**) scheduling policy is optimal in states where the product with the lower $b_i \mu_i$ value has negative stock. We are not aware of further results on the structure of optimal production policies.

Note that both production policies shown in Fig. 1 have been evaluated using MDP analysis. We have validated the MDP outcomes via discrete event simulation and observed that the simulated costs (based on 40,000,000 simulated demands) differed less than 0.3 % from the calculated costs. This is a strong indication that the results are correct.

4 Simulation optimization method for calculating base stock levels

As mentioned earlier, heuristic production policies usually consist of an idleness policy and a scheduling policy. All heuristic production policies studied in this study use idleness policies that are defined via a base stock level for each product. Production is switched on when the on-hand stock of at least one product is below its base stock level and switched off otherwise. At the same time, the scheduling policy may only select products whose on-hand stock is lower than their base stock level. An important task when developing production policies is thus to find methods to calculate appropriate base stock levels.

In this section, we propose an iterative method for calculating a base stock level vector $\mathbf{S}$ for arbitrary problem instances and an arbitrary but given scheduling policy π. It is a new method that serves as a building block for three of the four evaluated production policies. It uses discrete event simulation to examine the performance of candidate base stock level vectors and consists of three phases: initialization, greedy improvement, and local search. All simulation runs use the same random seed $r_{sim-opt}$, and thus the same stream of random inter-arrival times and random production times.

The length of each simulation run depends on the problem instance and is chosen such that the probability that the simulated average cost differs more than 1 % from its expected value is less than approximately 5 %. To achieve this, we adopt the method of *non-overlapping batch means* (see e.g., Steiger and Wilson 2001). We execute this method with a warming-up period of 500,000 demands and 20 batches, each of initial size 500,000 demands. As long as the desired 99 % accuracy is not reached, the batch size is doubled and the existing batches are reorganized. To avoid the risk of excessive run-times, we stop the procedure after the batch size has reached 2,000,000 (also in the case that the desired 99 % accuracy has not been reached). Only in very few base stock level calculations, this additional stopping criterion was triggered.

Phase 1: Initialization

In this phase, we create an initial base stock level vector $\mathbf{S}^0$ from the Brownian motion idleness policy proposed in Wein (1992) and the myopic($\mathbf{T}$) scheduling policy proposed in Peña Perez and Zipkin (1997). This production policy can be calculated very easily and performs surprisingly well (cf. Veatch and Wein 1996). The obtained base stock level is thus a reasonable start solution for our iterative procedure. We now shortly summarize how to obtain the base stock levels. For details we refer to Veatch and Wein (1996). The Brownian motion idleness policy decides to leave the production line idle when the aggregated production time represented by the stock in the system, i.e., $\sum_{i=1}^{I}(z_i/\mu_i)$, exceeds the following threshold:

$$c = \frac{\sum_{i=1}^{I} \lambda_i \frac{1}{\mu_i^2}(v_{id}^2 + v_{ip}^2)}{2(1-\rho)} \ln\left(1 + \frac{b}{h}\right), \tag{2}$$

where $\rho = \sum_{i=i}^{I}(\lambda_i/\mu_i)$, $b = \min_{1\leq i\leq I}\{b_i\,\mu_i\}$, $h = \min_{1\leq i\leq I}\{h_i\,\mu_i\}$, v_{id} is the coefficient of variation of the inter-arrival times of product i, and v_{ip} is the coefficient of variation of the production times of product i. To obtain $\mathbf{S}^0$ we must do the following. First, we set $\mathbf{S}^0 = \mathbf{0}$. Then, we call the myopic($\mathbf{T}$) scheduling policy with argument $\mathbf{S}^0$ to obtain the product n that is most attractive for being entered into production and update $\mathbf{S}^0$ according to $\mathbf{S}^0 = \mathbf{S}^0 + \mathbf{e}_n$. We repeat this procedure until the aggregated production time $\sum_{i=1}^{I}(S_i^0/\mu_i)$ exceeds the threshold c in (2).

Phase 2: Simultaneous base stock level updates

This phase consists of an iterative procedure where multiple base stock levels can be changed simultaneously. We start the procedure with the base stock level vector obtained in the initialization phase. In each iteration we carry out one simulation run. Before we discuss this phase in detail, we introduce two new variables. The variable $\hat{A}_i(\mathbf{S}, \pi, k)$ is defined as the fraction of time during the simulation run with base stock level vector $\mathbf{S}$ and scheduling policy π that the queue length (base stock level minus current net inventory) of product i is equal to k. The variable $C_i(u, \mathbf{S}, \pi)$ is an auxiliary variable that represents the average total cost of product i associated with base stock level u and the estimated queue length distribution $\hat{A}_i(\mathbf{S}, \pi, k)$, $k \geq 0$. It can be interpreted as the cost function in the classical *Newsvendor Problem* and is thus convex in u (cf. Porteus 2002).

$$C_i(u, \mathbf{S}, \pi) = \sum_{k=0}^{u-1}\left[\hat{A}_i(\mathbf{S}, \pi, k)\cdot(u-k)\cdot h_i\right] + \sum_{k=u+1}^{\infty}\left[\hat{A}_i(\mathbf{S}, \pi, k)\cdot(k-u)\cdot b_i\right] \quad (3)$$

Let $\mathbf{S}^m$ represent the base stock level vector in iteration m. In each iteration m, we simulate the system with scheduling policy π and base stock level vector $\mathbf{S}^m$ and calculate the average cost and the normalized queue length histograms $\hat{A}_i(\mathbf{S}^m, \pi, k)$, $k \geq 0$ for all products $i = 1, \ldots, I$. In each iteration, we update all base stock levels simultaneously according to:

$$S_i^{m+1} = \arg\min_u \left[C_i(u, \mathbf{S}^m, \pi)\right] \quad 1 \leq i \leq I \quad (4)$$

Equations (3) and (4) simply tell that the base stock level of each product in the next iteration is chosen such that it is optimal with respect to its normalized queue length histogram in the current iteration. Substituting (3) into (4) and using the convexity of $C_i(u, \mathbf{S}, \pi)$ in u, we can derive following (newsvendor) formula for the base stock level vector in iteration m:

$$S_i^{m+1} = \inf\left\{u \in \mathbb{N}_0 : \sum_{k=0}^{u} \hat{A}_i(\mathbf{S}^m, \pi, k) \geq \frac{b_i}{b_i + h_i}\right\} \quad 1 \leq i \leq I \quad (5)$$

We repeat this procedure until all base stock levels have converged (i.e., $S_i^{m+1} = S_i^m$ for all products $i = 1, \ldots, I$) *or* until the simulated average cost in the current iteration is higher than the simulated average cost in the previous iteration. (We have no proof for the convergence of this procedure, but we obtained convergence in all instances of our numerical experiments.)

Note that in the case the scheduling decisions do not (directly or indirectly) depend on the base stock level vector $\mathbf{S}$, the normalized queue length histograms $\hat{A}_i(\mathbf{S}^m, \pi, k)$, $k \geq 0$ do not change and the iterative process terminates after only one simulation run. Examples of scheduling policies where only one simulation run is needed are FCFS and LQ. However, in most scheduling policies the calculated decisions depend on the net inventory levels (and thereby on the base stock levels) and then typically more than one simulation run is needed to reach convergence.

Phase 3: Local search

In this phase, we execute a simple local search procedure. The neighborhood of a solution $\mathbf{S}$ consists of all non-negative base stock level vectors $\mathbf{S}'$ that can be obtained from $\mathbf{S}$ by increasing or decreasing the base stock level of exactly one product. For a given base stock level $\mathbf{S}$, we evaluate all its neighbors via simulation and jump to the neighbor with the lowest average cost. The local search procedure terminates when we have reached a local optimum. In Sect. 7, we investigate the contribution of the local search phase (in terms of realized cost reductions and computational effort) to the overall performance.

5 Heuristic production policies

In this section, we describe four heuristic production policies for our problem. All four production policies can be applied to problem instances of arbitrary size. They are described in detail in Sects. 5.1–5.4. The first heuristic production policy is a production policy proposed in Peña Perez and Zipkin (1997). For problem instances where a simple enumeration over all base stock level vectors is computationally infeasible, it is the best production policy in the existing literature. The authors simply denote their production policy as *myopic allocation*. In this paper, however, we denote it as My-EPC for two reasons. First, we want to indicate that it consists of a scheduling policy *and* an idleness policy. Second, it makes it easier to distinguish this production policy from other production policies that use the same scheduling policy but a different method for calculating base stock levels (cf. Sect. 5.2). The other three heuristic production policies use the new simulation optimization method described in Sect. 4 to calculate base stock levels. One method uses the myopic(**T**) scheduling policy and is denoted as My-Sim. Another method uses the existing switching rule for production scheduling and is denoted as SR-Sim. Finally, we present a production policy that uses a new (rolling horizon) policy for scheduling. This policy is denoted as RH-Sim.

All scheduling policies that we consider in this paper are index policies. Let $o^{\pi}(\mathbf{z}) \in \{1, \ldots, I\}$ denote the product selected by scheduling policy π when the vector of net inventory levels is equal to $\mathbf{z}$. Then, scheduling policy π is an index policy if it can be written as:

$$o^{\pi}(\mathbf{z}) = \arg\min_{i} G_i^{\pi}(\mathbf{z}), \tag{6}$$

where the $G_i^{\pi}(\mathbf{z}), \quad 1 \leq n \leq I$, are real-valued (index) functions. We assume that ties are broken with equal probabilities.

5.1 My-EPC

The My-EPC policy of Peña Perez and Zipkin (1997) consists of the myopic(**T**) scheduling policy and a one-dimensional optimization along the so-called *equal priority curve* (EPC) to determine appropriate base stock levels. In this search, candidate base stock level vectors are evaluated using simulation. In this paper, we do not discuss the myopic(**T**) scheduling policy in detail, but just give the expression for the index functions:

$$G_i(\mathbf{z}) = \begin{cases} -b_i\,\mu_i \;+ (h_i + b_i)\,\mu_i\,(1 - (\lambda_i/\mu_i)^{z_i+1}) & \text{if } z_i \geq 0 \\ -b_i\,\mu_i & \text{otherwise} \end{cases} \tag{7}$$

The equal priority curve is parameterized by $\theta \in \mathbb{R}$ and consists of all real vectors $\mathbf{S} \in \mathbb{R}^I$ that meet the following condition:

$$G_1(\mathbf{S}) = G_2(\mathbf{S}) = \cdots = G_I(\mathbf{S}) = \theta \tag{8}$$

Since base stock levels must be integer, Peña Perez and Zipkin (1997) round off **S** when evaluating a particular value of θ. Unfortunately, they do not provide information on how they have rounded. In order to allow the reader to reproduce our results, we now describe the method we have used.

First, we add $\mathbf{S}_0 = \mathbf{0}$ to the curve. Next, we use expression (8) to calculate $n_1 = \arg\min_i G_i^{\pi}(\mathbf{S}_0)$ and we add $\mathbf{S}_1 = \mathbf{S}_0 + \mathbf{e}_{n_1}$ to the curve. Then, we calculate $n_2 = \arg\min_i G_i^{\pi}(\mathbf{S}_1)$ and we add $\mathbf{S}_2 = \mathbf{S}_1 + \mathbf{e}_{n_2}$ to the curve. We repeat this procedure until we have added so many vectors to the curve that we are sure that all further base stock level vectors will have higher cost than the best base stock level vector added so far.

5.2 My-Sim

The My-Sim policy is a new production policy that only differs from the My-EPC production policy in the method that is used to calculate the base stock levels. Instead of searching along the equal priority curve, My-Sim uses the new simulation optimization method discussed in Sect. 4.

5.3 SR-Sim

The SR-Sim policy is a new production policy that consists of the switching rule for calculating scheduling decisions and the simulation optimization method for calculating base stock levels. The switching rule has been proposed in Ha (1997) as a heuristic for scheduling production in a single server make-to-stock queue with two products and a preemption. It is an index policy that prefers products that are backordered over other products. Among all products with at least one backorder the switching rule selects the product with the highest $b_i\,\mu_i$. If there are no backorders, priority is given to the product with the highest $b_i\,\mu_i\,(1-(z_i/S_i))$. Below, we give a formal description of the index functions $G_i(\mathbf{z})$ for the switching rule:

$$G_i(\mathbf{z}) = \begin{cases} -b_i\,\mu_i\,(1-(z_i/S_i)) & \text{if } \mathbf{z} \geq 0 \\ -b_i\,\mu_i & \text{if } \min(z_j) < 0 \text{ and } z_i < 0 \\ 0 & \text{if } \min(z_j) < 0 \text{ and } z_i \geq 0 \end{cases} \tag{9}$$

In the few contributions that apply the switching rule, enumeration has been used to obtain optimal base stock levels. For problem instances with more than two products, enumeration is however usually computationally prohibitive. Since we are interested in production policies that are also applicable to large problem instances, we combine the switching rule with our simulation optimization method for calculating base stock levels.

5.4 RH-Sim

The RH-Sim policy is one of the main contributions of this paper. It consists of a rolling horizon scheduling policy and the simulation optimization method for calculating base stock levels. The scheduling part of the RH-Sim policy consists of an index policy that uses the expected cost of schedules consisting of two products. The expected cost of a schedule is defined as the expected total cost arising during the time needed to produce both products. The idea behind this new scheduling policy is a sample path argument. Consider two products, i and j. If the expected cost of production schedule (i, j) is lower than the expected cost of production schedule (j, i), then product i seems to be needed more urgently than product j and we favor product i over product j. In the case of more than two products, the index for product i is the weighted sum of the costs of all schedules (i, j) and (j, i) with $j \neq i$. In Sect. 5.4.1 we discuss the two-product case, and in Sect. 5.4.2 the multi-product case.

The advantage of comparing the schedules (P_i, P_j) and (P_j, P_i) is that both schedules have the same duration and lead to the same end state (for all possible realizations of inter-arrival times and production times). Consequently, we can make a fair comparison between these two schedules without having to look beyond the completion time of these schedules. Before we explain the rolling horizon scheduling policy in detail, we introduce some useful definitions in Table 2.

5.4.1 Two products case

Assume we have two products, i and j. The RH-Sim policy selects product i if production schedule (i, j) has lower expected total cost than schedule (j, i), and product j otherwise. Instead of comparing $C_{ij}(\mathbf{z})$ with $C_{ji}(\mathbf{z})$, we compare $\Delta C_{ij}(\mathbf{z})$ with $\Delta C_{ji}(\mathbf{z})$ because this simplifies the calculations considerably. Note that this transformation is allowed since $\widetilde{C}_{ij}(\mathbf{z}) = \widetilde{C}_{ji}(\mathbf{z})$. We derive an exact expression for $\Delta C_{ij}(\mathbf{z})$ by calculating the expected total cost of production schedule (i, j) over all possible realizations of inter-arrival times and the production times P_i and P_j. The variable $\Delta C_{ij}(\mathbf{z})$ can be interpreted as the cost savings associated with executing production schedule (i, j) compared to producing nothing. Figure 2 illustrates how we can calculate $\Delta C_{ij}(\mathbf{z})$. In this figure, the solid lines show possible evolutions of the net

Table 2 Notation and definitions for RH-Sim production policy

Random variables	
P_j	Production time of product j
Probabilities	
$f_{ij}(u)$	Probability that during a random production time of product j the demand of product i is equal to u
Cost variables	
$C_{ij}(\mathbf{z})$	Expected cost caused by products i and j during production schedule (i, j) as a function of the initial inventory levels $\mathbf{z}$
$\widetilde{C}_{ij}(\mathbf{z})$	Expected cost caused by products i and j in time interval $[0, P_i + P_j]$ as a function of the initial inventory levels $\mathbf{z}$ if no items produced during the time interval $[0, P_i + P_j]$
$\Delta C_{ij}(\mathbf{z})$	$C_{ij}(\mathbf{z}) - \widetilde{C}_{ij}(\mathbf{z})$

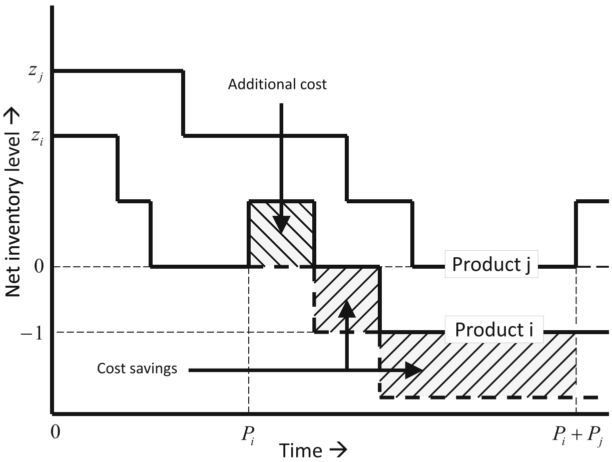

Fig. 2 Evolution of net inventory levels during production schedule (i, j)

inventories of product i and product j during the execution of production schedule (i, j).

A net inventory decrease represents a demand arrival and a net inventory increase represents a product completion. The dashed lines show the evolution of the net inventories if no items would be produced. To obtain $\Delta C_{ij}(\mathbf{z})$, we must calculate for product i and product j the cost difference between the solid line and the dashed line during the forecast horizon $P_i + P_j$. Note that for product j this difference is always 0 because its net inventory is only increased at the very end of the interval $[0,\ P_i + P_j]$. Consequently, $\Delta C_{ij}(\mathbf{z})$ only depends on z_i and thus we write from now on $\Delta C_{ij}(z_i)$.

To calculate $\Delta C_{ij}(z_i)$ we condition on the demand for product i during P_i (denoted as u) and the demand for product i during P_j (denoted as w) and calculate the expected additional backordering cost and the expected additional inventory holding cost. We distinguish three cases: (i) the next backorder for product i occurs before $t = P_i$, (ii) the next backorder for product i occurs after $t = P_i + P_j$, and (iii) the next backorder for product i occurs at $P_i \le t \le P_i + P_j$.

In the first case, we save the backordering cost of one item of product i during the entire production time of one item of product j. In the second case, we have additional inventory holding cost one item of product i during the entire production time of one item of product j. In the third case, the cost savings depend on the moment when the first backorder of product i would occur if no items would be produced. At time $t = P_i$ the net inventory of product i is equal to $(z_i - u)$. Let $E(P_j|w)$ represent the expected production time of one item of product j under the condition that during this production time the demand for product i is equal to w. Then, we have additional inventory holding costs until the arrival of the $(z_i - u + 1)$th demand during the production time $E(P_j|w)$. During the remaining part of $E(P_j|w)$ we save backordering costs. Using the assumption that demand inter-arrival times are independent and identically exponentially distributed, we can argue that on average the $(z_i - u + 1)$th demand will arrive $E(P_j|w)(z_i - u + 1)/(w + 1)$ time units after

$t = P_i$. Combining the three components of the expected total cost savings leads to the following expression for $\Delta C_{ij}(z_i)$:

$$\Delta C_{ij}(z_i) = \left[1 - \sum_{u=0}^{z_i} f_{ii}(u)\right] b_i \frac{1}{\mu_j} - \sum_{u=0}^{z_i} \sum_{w=0}^{z_i-u} f_{ii}(u)\ f_{ij}(w)\ E(P_j|w)\ h_i \\ + \sum_{u=0}^{z_i} \sum_{w=z_i-u+1}^{\infty} f_{ii}(u) \quad f_{ij}(w) \quad E(P_j|w) \quad \left[-h_i \cdot \frac{z_i-u+1}{w+1} + b_i \cdot \frac{w-z_i+u}{w+1}\right] \tag{10}$$

Usually, equation (10) can be calculated easily because $f_{ii}(u)$, $f_{ij}(w)$ and $E(P_j|w)$ have simple closed-form expressions for Poisson demand and exponential or deterministic production times. First, consider exponential production times. From basic probability theory, we know that the number of Poisson arrivals (with rate λ) during an exponentially distributed time interval with mean $1/\mu$ has a geometric distribution on the set of natural numbers (including zero) with parameter $p = 1/(1 + (\lambda/\mu))$. This immediately implies that $f_{ii}(u) = (1-p)^u\, p$, $u \in \mathbb{N}_0$, with $p = 1/(1 + (\lambda_i/\mu_i))$ and $f_{ij}(u) = (1-p)^u\, p$, $u \in \mathbb{N}_0$, with $p = 1/(1 + (\lambda_i/\mu_j))$. In Appendix 9 we show that for exponentially distributed production times it holds that $E(P_j|w) = (w+1)/(\lambda_i + \mu_j)$. Next, consider deterministic production times. Then, it is easy to see that $f_{ii}(u)$ and $f_{ij}(u)$ have Poisson distributions with parameters λ_i/μ_i and λ_i/μ_j, respectively, and that $E(P_j|w) = E(P_j) = 1/\mu_j$. We conclude this section with a formal description of the index functions $G_i(\mathbf{z})$ and $G_j(\mathbf{z})$ for the two products case.

$$G_i(\mathbf{z}) = \Delta C_{ij}(z_i) \tag{11a}$$
$$G_j(\mathbf{z}) = \Delta C_{ji}(z_j) \tag{11b}$$

5.4.2 Multi-product case

We now extend the RH-Sim policy to systems with more than two products. The idea is to calculate for each product i an index that is the weighted sum of $[\Delta C_{ij}(z_i) - \Delta C_{ji}(z_j)]$ over all products $j \neq i$. Since we consider an average cost problem, it is reasonable to weigh the expected relative cost $\Delta C_{ij}(z_i)$ of production schedule (i, j) with the inverse of the average duration of that schedule. This yields the following general expression for the index functions $G_i(\mathbf{z})$ for the RH-Sim policy with I products:

$$G_i(\mathbf{z}) = \sum_{j \neq i} \frac{1}{\frac{1}{\mu_i} + \frac{1}{\mu_j}} [\Delta C_{ij}(z_i) - \Delta C_{ji}(z_j)] \quad 1 \leq i \leq I \tag{12}$$

6 Numerical experiments

In this section, we investigate the performance and the structure of the new production policies via numerical experiments. We have the following objectives in conducting numerical experiments. First, we investigate the optimality gap of My-Sim, SR-Sim, and RH-Sim and compare it against the optimality gap of the existing

My-EPC production policy. Second, we investigate the potential cost savings of the three new production policies compared to My-EPC on a wide range of large problem instances and explore how the relative performance depends on the various problem characteristics.

To meet the first objective, we define a numerical experiment with a test bed containing problem instances with two products and exponentially distributed production times (Experiment I, Sect. 6.1). For all problem instances in this test bed, we can calculate the optimal production policy. In this experiment, we use relative value iteration to evaluate and compare the production policies. In Sect. 6.2, we carry out a detailed optimality gap analysis and characterize the proposed RH-Sim policy.

To meet the second objective, we define a numerical experiment with a test bed containing a wide range of problem instances with ten products (Experiment II, Sect. 6.3). In this experiment, we cannot determine the optimal production policy due to the large problem sizes. Therefore, we use discrete event simulation to evaluate and compare the heuristic production policies.

6.1 Experiment I

In this experiment, we consider problem instances with two products, independent Poisson demand, and independent exponentially distributed production times. We assume that $\lambda_1/\mu_1 = \lambda_2/\mu_2 = (1/2)\,\rho$, and $b_1/h_1 = b_2/h_2 = b/h$. Cost asymmetry is measured by the ratio h_2/h_1, and the inventory holding cost of product 1 is always equal to 1. We examine three production rate vectors $\boldsymbol{\mu}$, namely $\boldsymbol{\mu} = (1, 1)$, $\boldsymbol{\mu} = (1, 4)$ and $\boldsymbol{\mu} = (4, 1)$; three values of ρ (0.90, 0.80, and 0.70); three values of h_2/h_1 (0.9, 0.7, and 0.5); and two values of b/h (20 and 80). This test bed only differs in two minor aspects from the test bed used in Peña Perez and Zipkin (1997): (i) we use three utilization rates instead of eight, and (ii) we have replaced the production rate vectors $\boldsymbol{\mu} = (1, 2)$ and $\boldsymbol{\mu} = (2, 1)$ by $\boldsymbol{\mu} = (1, 4)$ and $\boldsymbol{\mu} = (4, 1)$ in order to gain better insights into the impact of non-identical production rates on the performance of the various production policies. The optimal average costs as well as the optimality gaps of the heuristic production policies are shown in Tables 3, 4 and 5.

In the discussion of the results of Experiment I, we distinguish between identical production rates (Table 3) and non-identical production rates (Tables 4, 5). For identical production rates, we see that the optimality gaps of all evaluated production policies are very small. For every evaluated production policy, the average optimality gap is less than 1 % and the maximum optimality gap is less than 3 %. We see that all four heuristic production policies outperform the other three on at least one problem instance. This means that none of the four evaluated production policies dominates any other. For non-identical production rates, the differences are much bigger. We see that for all production policies the optimality gaps increase when the ratio between the higher and the lower production rate increases. The maximum optimality gaps for My-EPC, My-Sim, SR-Sim, and RH-Sim are, respectively, 112.3, 56.9, 15.8, and 11.6 %. It is remarkable to see how the performance of My-Sim, and in particular My-EPC, degrades dramatically when the production rates differ. SR-Sim and SH-Sim show much more robust behavior in this respect. Further, we see that RH-Sim outperforms

Table 3 Experiment I: gap analysis for $\boldsymbol{\mu} = (1, 1)$

Input parameters			Avg cost	Optimality gap (%)			
ρ	h_2/h_1	b/h	Opt	My-EPC	My-Sim	SR-Sim	RH-Sim
0.9	0.9	20	26.85	0.2	0.2	0.8	0.9
0.9	0.9	80	38.60	0.1	0.1	1.1	0.8
0.9	0.7	20	22.21	0.3	0.3	0.5	1.4
0.9	0.7	80	31.79	0.5	0.5	0.7	0.7
0.9	0.5	20	17.30	1.7	1.4	0.8	1.0
0.9	0.5	80	24.70	0.9	0.8	0.9	2.7
0.8	0.9	20	13.14	0.1	0.1	0.4	0.1
0.8	0.9	80	18.82	0.1	0.1	0.3	0.7
0.8	0.7	20	11.29	0.4	0.4	1.0	0.4
0.8	0.7	80	16.10	0.4	0.4	1.1	1.0
0.8	0.5	20	9.28	1.0	1.0	0.7	0.8
0.8	0.5	80	13.22	1.5	1.0	1.1	1.0
0.7	0.9	20	8.59	0.0	0.0	0.0	0.0
0.7	0.9	80	12.26	0.1	0.2	0.2	0.2
0.7	0.7	20	7.53	0.8	0.8	0.8	0.6
0.7	0.7	80	10.73	0.0	0.0	0.7	0.0
0.7	0.5	20	6.42	0.3	0.3	0.3	0.0
0.7	0.5	80	9.12	2.0	1.0	0.9	1.0
Average				0.6	0.5	0.7	0.7
Maximum				2.0	1.4	1.1	2.7

all other three heuristic production policies on all 36 problem instances with non-identical production rates. Finally, we see larger optimality gaps for the instances in Table 5, where the cheap product has the long production time, than for the instances in Table 4, where the expensive product has the long production time. This seems to be related to option value effect as described in Sect. 3.3. This effect seems to be stronger for the instances where the cheap product has the long production time.

6.2 Optimality gap analysis

As described in Sect. 3.3, the optimal production policy must not necessarily belong to the class of base stock policies. This means that we can distinguish three root causes for sub-optimality of heuristic production policies: (i) sub-optimality of the scheduling logic, (ii) sub-optimality of the method for calculating base stock levels, and (iii) sub-optimality due to the imposed base stock structure. To examine the contribution of each of these three factors to the reported optimality gaps, we carry out a deeper optimality gap analysis on those problem instances in Tables 3, 4 and 5.

To investigate the contribution of the three root causes to the observed optimality gaps, we compare the four heuristic production policies against the optimal *base stock*

Table 4 Experiment I: gap analysis for $\boldsymbol{\mu} = (1, 4)$

Input parameters			Avg cost	Optimality gap (%)			
ρ	h_2/h_1	b/h	Opt	My-EPC	My-Sim	SR-Sim	RH-Sim
0.9	0.9	20	23.65	20.0	17.0	3.9	1.6
0.9	0.9	80	33.81	48.5	30.5	3.5	2.5
0.9	0.7	20	22.12	17.5	14.5	3.4	1.6
0.9	0.7	80	31.65	25.1	18.2	3.0	1.9
0.9	0.5	20	20.40	16.1	13.0	3.3	2.8
0.9	0.5	80	29.29	21.4	14.8	2.9	1.8
0.8	0.9	20	13.32	34.5	23.6	6.7	4.1
0.8	0.9	80	19.00	45.4	27.9	6.1	4.5
0.8	0.7	20	12.11	30.6	19.9	4.9	2.9
0.8	0.7	80	17.29	39.5	23.3	4.7	3.3
0.8	0.5	20	10.83	19.2	14.4	3.6	2.7
0.8	0.5	80	15.49	33.2	18.6	3.8	2.6
0.7	0.9	20	9.41	30.1	21.2	8.2	6.4
0.7	0.9	80	13.45	55.4	28.5	8.1	6.6
0.7	0.7	20	8.40	27.6	18.5	6.3	4.9
0.7	0.7	80	12.03	54.7	25.1	7.0	5.1
0.7	0.5	20	7.39	22.0	13.4	3.9	2.9
0.7	0.5	80	10.56	44.7	18.5	4.7	3.8
Average				32.5	20.0	4.9	3.4
Maximum				55.4	30.5	8.2	6.6

production policy. The optimal base stock production policy differs from the optimal production policy in that it must belong to the class of base stock policies (cf. Fig. 1). To obtain the optimal *base stock* production policy, we enumerate over a large set of candidate base stock level vectors and solve for each candidate base stock level vector an MDP that is very similar to the one described in Sect. 3.2. In fact, we only need to make a small adjustment to the set of admissible actions in order to guarantee that the solution of the MDP is indeed a base stock policy. Instead of $\mathscr{A}(\mathbf{z}, 0) = \{0, 1, \ldots, I\}$, it is sufficient to use $\mathscr{A}(\mathbf{z}, 0) = \{1 \leq i \leq I \,|\, z_i < S_i\}$. The optimal base stock production policy is denoted as B-Opt.

For the selected problem instances, we also calculate the production policies consisting of one of the three heuristic scheduling policies (myopic($\mathbf{T}$), switching rule, rolling horizon) together with base stock levels that have been optimized with respect to the applied scheduling policy. These policies are denoted as My-Opt, SR-Opt, and RH-Opt, respectively, and have been obtained via full enumeration over a wide range of candidate base stock level vectors. The results are shown in Table 6.

Note that B-Opt, My-Opt, SR-Opt, and RH-Opt can only be calculated for problem instances with a small number of products (typically 2–3) because its derivation requires solving multiple MDPs. Table 6 shows that the performance of RH-Sim is very close to the performance of the optimal base stock policy. The maximum gap

Table 5 Experiment I: gap analysis for $\boldsymbol{\mu} = (4, 1)$

Input parameters			Avg cost	Optimality gap (%)			
ρ	h_2/h_1	b/h	Opt	My-EPC	My-Sim	SR-Sim	RH-Sim
0.9	0.9	20	22.66	23.3	19.9	4.9	2.0
0.9	0.9	80	32.37	55.8	37.1	4.7	2.1
0.9	0.7	20	19.19	27.4	23.5	6.5	3.2
0.9	0.7	80	27.41	64.0	41.8	5.2	2.2
0.9	0.5	20	15.69	54.6	42.2	9.0	3.4
0.9	0.5	80	22.39	77.8	52.0	7.1	3.7
0.8	0.9	20	13.10	39.0	27.6	8.5	4.9
0.8	0.9	80	18.70	51.8	32.9	7.1	5.5
0.8	0.7	20	11.48	44.6	32.6	10.7	6.3
0.8	0.7	80	16.39	64.3	42.7	8.8	5.6
0.8	0.5	20	9.83	51.0	37.2	14.0	6.9
0.8	0.5	80	14.05	112.3	56.9	11.0	7.0
0.7	0.9	20	9.38	32.3	23.7	9.9	8.2
0.7	0.9	80	13.43	60.6	32.1	9.6	7.8
0.7	0.7	20	8.36	36.8	27.9	12.4	8.0
0.7	0.7	80	11.97	68.1	37.5	11.2	9.4
0.7	0.5	20	7.32	53.5	39.2	15.8	9.8
0.7	0.5	80	10.49	78.2	43.4	13.8	11.6
Average				55.3	36.1	9.5	6.0
Maximum				112.3	56.9	15.8	11.6

Table 6 Experiment I: detailed gap analysis for selected problem instances

Input parameters				Optimality gap (%)							
μ	ρ	h_2/h_1	b/h	B-Opt	My-Opt	My-EPC	My-Sim	SR-Opt	SR-Sim	RH-Opt	RH-sim
(1, 1)	0.9	0.7	20	0.1	0.2	0.3	0.3	0.5	0.5	1.4	1.4
(1, 1)	0.9	0.5	80	0.1	0.8	0.9	0.8	0.9	0.9	2.6	2.7
(1, 4)	0.7	0.9	20	5.5	21.2	30.1	21.2	7.9	8.2	5.5	6.4
(1, 4)	0.7	0.9	80	6.6	28.5	55.4	28.5	8.1	8.1	6.6	6.6
(1, 4)	0.7	0.7	80	5.1	24.7	54.7	25.1	6.3	7.0	5.1	5.1
(4, 1)	0.7	0.7	80	9.4	37.1	68.1	37.5	11.2	11.2	9.4	9.4
(4, 1)	0.7	0.5	20	9.7	33.9	53.5	39.2	13.4	15.8	9.8	9.8
(4, 1)	0.7	0.5	80	11.4	43.4	78.2	43.4	13.8	13.8	11.4	11.6

between RH-Sim and B-Opt is only 2.6 % and for 6 out of 9 problem instances it is even less than 1 %. Table 6 also indicates that the simulation optimization method for calculating base stock levels performs very well. If we compare RH-Sim with RH-Opt, we see that the simulation optimization method manages to find the optimal base stock levels in 6 out of 9 problem instances. The table also shows that this method does not only perform very well in combination with the rolling horizon scheduling policy, but also in combination with the switching rule and the myopic(**T**) scheduling policy. Furthermore, we see that—even under optimized base stock levels—the optimality gap of the myopic(**T**) scheduling policy can grow to 44.3 %. This is much bigger than the reported optimality gaps in Peña Perez and Zipkin (1997). The reason for this discrepancy is that our test bed contains problem instances where the production rates differ by a factor four whereas in Peña Perez and Zipkin (1997) production rates never differ by more than a factor two. Apparently, the myopic(**T**) scheduling policy has difficulties handling products with (highly) different average production times.

We conclude this section with a characterization of RH-Sim. For this purpose, we have compared the production decisions calculated by RH-Sim with the optimal production decisions for problem instances in Experiment I where the optimality gap of RH-Sim is relatively big. We made two important observations. First, RH-Sim belongs—by construction—to the class of base stock policies whereas the optimal production policy for several problem instances does not. Second, RH-Sim more often selects the product with the lower inventory holding cost than the optimal production policy. This behavior particularly occurs when on-hand stock is high and the system utilization rate ρ is also high. An intuitive reason for this behavior would be that RH-Sim does not look more than two scheduling decisions ahead. Assume on-hand stocks are high, then the stockout probabilities during the forecast horizon (i.e., during the next two scheduling decisions) are very low. Consequently, RH-Sim will select the product with the lower inventory holding cost (cf. expression (10)). However, especially in the case of high utilization rates, it may be wiser to produce an item of the product with the higher inventory holding cost rate in order to avoid high backordering costs in the future.

6.3 Experiment II

In this experiment, we compare the three new production policies (My-Sim, SR-Sim, and RH-Sim) against My-EPC on a test bed containing problem instances with ten products. In contrast to Experiment I, the problem instances in this experiment are too large for an MDP analysis. Consequently, all results in this experiment have been obtained via discrete event simulation. When evaluating the performance of different production policies on a particular problem instance, we use the same random seed r_{eval} and the same simulation length. In order to guarantee unbiased cost estimates, the random seed r_{eval} has a different value than the random seed $r_{sim-opt}$ that is used inside the simulation optimization method for calculating base stock levels.

The length of each simulation run depends on the problem instance and is chosen such that the probability that the simulated average cost differs more than 1 % from its expected value is less than approximately 5 %. To achieve this, we adopt the method of non-overlapping batch means (see e.g., Steiger and Wilson 2001). We execute this method with a warming-up period of 2,000,000 demands and 20 batches, each of initial size 2,000,000 demands. As long as the desired 99 % accuracy is not reached, the batch size is doubled and the existing batches are reorganized. For all problem instances and all production policies in this experiment, the desired 99 % accuracy has been reached with a batch size of 8,000,000 maximum.

The test bed in this experiment is a full factorial design on five parameters: (i) the maximum demand rate across all products ($\lambda^{\max}$); (ii) the type of coupling between demand rates, production rates, and inventory holding costs ('coupling'); (iii) the utilization rate (ρ); (iv) the ratio between backordering costs and inventory holding costs (b/h); and (v) the shape of the production times distributions ('shape'). The minimum demand rate across all products ($\lambda^{\min}$) is set equal to 1. We consider three values for $\lambda^{\max}$ (2,4, and 8); four couplings between the demand rates, production rates, and inventory holding costs ('A', 'B', 'C', and 'D'); four values for the utilization rate ρ (0.70, 0.80, 0.90, and 0.95); two values for the ratio b/h (20 and 80); and two shapes of the production time distributions (deterministic and exponential). The total number of all possible combinations for these parameters is thus $3 \times 4 \times 4 \times 2 \times 2 = 192$. The following procedure describes how to construct a problem instance from the values of these five parameters.

Step 1 Set $\lambda_i = \lambda^{\min} + (\lambda^{\max} - \lambda^{\min})\,(i-1)\,/\,9, i = 1, \ldots, 10$.
Step 2 Calculate production rates, and inventory holding costs.
If coupling = 'A': $\mu'_i = 1, h_i = 1, i = 1, \ldots, 10$.
If coupling = 'B': $\mu'_i = \lambda_i, h_i = 1/\mu'_i, i = 1, \ldots, 10$.
If coupling = 'C': $\mu'_i = 1, i = 1, 3, \ldots, 9; \mu'_i = \lambda^{\min}/\lambda^{\max}, i = 2, 4, \ldots, 10$; $h_i = 1/\mu'_i, i = 1, \ldots, 10$.
If coupling = 'D': $\mu'_i = 1, i = 1, 3, \ldots, 9; \mu'_i = \lambda^{\min}/\lambda^{\max}, i = 2, 4, \ldots, 10$; $h_i = 1/\lambda_i, i = 1, \ldots, 10$.
Step 3 $\mu_i = \mu'_i \cdot \left[(1/\rho)\ \sum(\lambda_i/\mu_i)\right], i = 1, \ldots, 10$.
Step 4 $b_i = h_i\ \cdot\ b/h, i = 1, \ldots, 10$.

In Experiment II, we take the existing My-EPC policy as the reference solution and calculate for the three new production policies (My-Sim, SR-Sim, RH-Sim) the average cost reductions over all 192 problem instances and all subsets where one of the factorial design parameters is fixed to one of its admissible values. Besides the average cost reductions (shown in bold face), we also show the minimum and the maximum cost reductions (shown in parenthesis). The results are shown in Table 7.

Table 7 shows that all three new production policies outperform the existing myopic allocation approach. The average cost reductions for My-Sim, SR-Sim, and RH-Sim are, respectively, 2.3, 3.5, and 4.5 %. We also see that RH-Sim has a better average performance than the other three production policies on all 15 subsets, except the subset with 'coupling' = 'D' where SR-Sim performs 0.2 % better. Further, we see that sometimes the new production policies do slightly worse than the existing myopic allocation approach (indicated by the negative numbers in the table). However, the cost increases are low (in particular for RH-Sim and My-Sim), whereas the cost reductions that can be achieved are high (20–30 %). So in sum, RH-Sim seems most attractive as it realizes the highest average cost reduction and is very robust. This is very much in line with the results obtained in Experiment I (cf. Sect. 6.2).

Table 7 Experiment II: aggregated results

Test bed subset	N	Cost reduction (%)		
		My-Sim	SR-Sim	RH-Sim
All instances	192	2.3 (−1.6, 19.3)	3.5 (−4.4, 32.0)	4.5 (−1.3, 32.7)
$\lambda^{\max} = 2$	64	1.3 (−1.6, 15.3)	1.7(−3.0, 20.8)	2.5 (−1.2, 19.9)
$\lambda^{\max} = 4$	64	1.9 (−1.3, 12.5)	2.8 (−4.4, 22.0)	3.9 (−1.3, 22.2)
$\lambda^{\max} = 8$	64	3.5 (−1.0, 19.3)	5.9 (−3.7, 32.0)	7.0 (−0.8, 32.7)
Coupling = 'A'	48	0.1 (−1.2, 1.7)	−1.6 (−4.4, 1.4)	0.1 (−1.2, 1.7)
Coupling = 'B'	48	2.3 (−1.3, 17.4)	2.9 (−0.4, 17.4)	4.3 (−1.3, 18.8)
Coupling = 'C'	48	0.4 (−1.0, 2.1)	−0.4 (−2.4, 2.3)	0.7 (−0.8, 3.3)
Coupling = 'D'	48	6.3 (−1.6, 19.3)	13.0 (0.0, 32.0)	12.8 (0.0, 32.7)
$\rho = 0.70$	48	2.7 (−0.7, 17.4)	2.8 (−3.2, 24.0)	4.3 (0.0, 24.7)
$\rho = 0.80$	48	3.1 (−1.6, 19.3)	3.6 (−4.4, 32.0)	5.3 (−0.4, 32.7)
$\rho = 0.90$	48	2.1 (−1.2, 12.8)	3.8(−3.5, 30.5)	4.6 (−1.2, 30.3)
$\rho = 0.95$	48	1.1 (−1.3, 8.6)	3.7 (−1.7, 25.7)	3.7 (−1.3, 23.2)
$b/h = 20$	96	2.0 (−1.6, 16.8)	3.0 (−4.4, 28.9)	4.1 (−0.8, 28.6)
$b/h = 80$	96	2.6 (−1.2, 19.3)	3.9 (−3.7, 32.0)	4.9 (−1.3, 32.7)
shape = 'det.'	96	1.7 (−1.6, 17.4)	1.9 (−4.4, 19.8)	3.2 (−1.2, 20.0)
shape = 'exp.'	96	2.9 (−1.3, 19.3)	5.0 (−2.9, 32.0)	5.8 (−1.3, 32.7)

An interesting observation from a technical point of view, is the strong performance of SR-Sim. In contrast to the other production policies, it does not use the demand rates when calculating scheduling decisions. So apparently, the simulation optimization method for calculating base stock levels anticipates the scheduling logic so well that this information is not needed in the scheduling phase.

We conclude this section with an investigation on how the relative performance of the three new policies (and in particular RH-Sim) depends on various problem characteristics. Our most important observation is that the relative performance strongly depends on the coupling between the demand rates, production rates, and inventory holding costs. In line with the results obtained in Experiment I, we see that all heuristic production policies perform about equally well in the case of identical production rates (case 'A'). If the production rates grow linearly with the demand rates (case 'B'), RH-Sim outperforms the existing myopic allocation approach. However, if we compare the results for case 'C' with the results for case 'D', we see that the spread in the production rates cannot be the only reason for the poor performance of the myopic allocation approach since we use the same demand rates and production rates. Apparently, the cost structure also plays a role (in case 'C' the inventory holding costs and backordering costs scale with the average production time, where in case 'D' they scale with the average inter-arrival time).

A related observation is that the achieved cost reductions of the three new production policies get bigger if the spread in the demand rates (measured via λ^{max}) gets bigger. In light of the previous discussion on the couplings, this is not really a surprise as the spread in the production rates and the spread in the cost parameters get bigger if the spread in the demand rates gets bigger. Finally, we see that the cost reductions are bigger in the case of exponentially distributed production times than in the case of deterministic production times and that the utilization rate ρ and the ratio b/h have little impact on the relative performance.

7 Performance of the simulation optimization method

In this section, we investigate the computation times of the simulation optimization method for calculating base stock levels (cf. Sect. 4) and the effectiveness of its three phases: initialization, greedy improvement, and local search. In Table 8, we show for each scheduling policy, the performance of the method in Experiment II. In the column *Greedy*, we show the average cost reduction (shown in bold face) as well as the minimum and the maximum cost reduction (shown in parenthesis) obtained at the end of the greedy phase compared to the cost obtained at the end of the initialization phase. In the column *Local search*, we show the average, minimum, and maximum additional cost reduction obtained in the local search phase. Finally, we show the maximum and the average required CPU time (in minutes) for the entire simulation optimization approach (i.e., sum of the CPU times of all three phases). The reported CPU times are obtained on an Intel 2.16 GHz processor with 8 GB RAM.

Table 8 shows that the greedy phase in the simulation optimization method for calculating base stock levels often realizes big cost reductions. The table also shows that local search phase is much less effective, although it still achieves significant improvements for certain problem instances. The last two columns show that the computation times of our simulation optimization method are acceptable. Only for a few problem instances (all with a utilization rate of 0.95) it needs more than 1 h, but never more than 5 h. Not shown in the table is that the greedy phase usually only requires 1, 2, or 3 simulation runs (and never more than 7) and that the local search phase is often responsible for more than 90 % of the total CPU time.

Table 8 Performance simulation optimization method for calculating base stock levels

Test bed	Policy	Greedy impr (%)	Local search impr (%)	CPU time (min)	
				Avg	Max
Experiment II (exp. prod. times)	My-Sim	10.8 (0.0, 37.7)	1.6 (0.0, 8.5)	13.7	68.8
	SR-Sim	13.6 (0.0, 38.2)	1.3 (0.0, 4.7)	13.2	78.5
	RH-Sim	12.2 (0.0, 37.8)	1.6 (0.0, 6.7)	17.3	64.5
Experiment II (det. prod. times)	My-Sim	21.6 (0.0, 74.1)	1.7 (0.0, 7.9)	16.1	158.1
	SR-Sim	23.5 (0.0, 74.1)	0.9 (0.0, 5.5)	13.7	155.6
	RH-Sim	22.0 (0.0, 74.7)	1.6 (0.0, 7.9)	18.6	258.9

8 Conclusions

We conclude by summarizing our main results and pointing out opportunities for future research. We studied a production/inventory system consisting of one production line and multiple products. Finished goods are kept in stock to serve stochastic demand. Demand is fulfilled immediately if there is an item of the requested product in stock and otherwise it is backordered and fulfilled later. The objective is to minimize holding and backordering costs. We developed a new dynamic production policy that consists of a rolling horizon scheduling policy and a simulation optimization method for calculating base stock levels. The developed simulation optimization method is generic in the sense that it can be combined with any scheduling policy.

In this study, we compared the performance of our proposed production policy against the myopic allocation policy proposed in Peña Perez and Zipkin (1997). To the best of our knowledge, this is the best existing production policy that is applicable to problem instances with more than just 2 or 3 products. First, we showed on a test bed with two products that the optimality gap of our proposed production policy is small. The average and the maximum optimality gap over all examined problem instances are 3.4 and 11.6 %, respectively. This is a clear improvement compared to the myopic allocation policy. Second, we showed that the simulation optimization method finds near-optimal base stock levels for all evaluated scheduling policies. Third, we showed that the difference in total cost between the optimal production policy and the optimal base stock production policy can be more than 10 % for certain problem instances. Finally, we showed on a test bed with ten products that the proposed production policy outperforms all other evaluated production policies. In particular, it can achieve cost reductions of up to 30 %, compared to the myopic allocation approach in the case of (very) heterogeneous products.

For future research, we suggest to further investigate the structure of the optimal production policy for systems with a non-preemptive discipline. The remarkable observations in Example 1 in Sect. 3.3 may serve as a good starting point for this investigation. Furthermore, we believe that there are interesting opportunities to apply the rolling horizon scheduling policy and the simulation optimization method for calculating base stock levels to other environments, in particular repairable inventory systems. Adan et al. (2009) consider such a system for expensive spare parts. They present an exact method for (simultaneously) optimizing (i) the initial spare parts supplies and (ii) *static* priorities for scheduling the outstanding repair jobs. It would be interesting to compare their static approach with our dynamic approach.

Acknowledgements We kindly thank the associate editor and two anonymous referees for their helpful suggestions to improve the paper.

9 Calculation of $E(P|w)$

Let P represent an exponentially distributed time variable with mean $1/\mu$. Consider a queuing system where customers arrive according to a Poisson distribution with constant rate λ. In this appendix, we show that $E(P|w)$, the expectation of a time interval given that during this time interval w customer arrivals occur, is equal to $(w+1)/(\lambda+\mu)$. Using the definition of conditional expectation and the property that the number of Poisson arrivals during an exponentially distributed time interval with mean $1/\mu$ has a geometric distribution with success probability $\frac{1}{1+\frac{\lambda}{\mu}}$, we obtain the following expression for $E(P|w)$:

$$E(P|w) = \frac{\int_0^\infty t \quad \mu e^{-\mu t} \quad \frac{(\lambda t)^w}{w!} \quad e^{-\lambda t} \quad dt}{(1-\frac{1}{1+\frac{\lambda}{\mu}})^w \quad (\frac{1}{1+\frac{\lambda}{\mu}})} \tag{13a}$$

Rearranging terms, and using that $\frac{(\lambda+\mu)^{w+2} \quad t^{w+1}}{(w+1)!} \quad e^{-(\lambda+\mu)\,t}$ is the probability density function of the Erlang distribution with parameters $(w+2)$ and $(\lambda+\mu)$, we can rewrite expression (13a) as follows:

$$E(P|w) = \frac{\frac{\mu\,\lambda^w\,(w+1)}{(\lambda+\mu)^{w+2}} \int_0^\infty \frac{(\lambda+\mu)^{w+2} \quad t^{w+1}}{(w+1)!} \quad e^{-(\lambda+\mu)\,t}}{\left(1-\frac{1}{1+\frac{\lambda}{\mu}}\right)^w \quad \left(\frac{1}{1+\frac{\lambda}{\mu}}\right)} \tag{13b}$$

$$= \frac{\frac{\mu\,\lambda^w\,(w+1)}{(\lambda+\mu)^{w+2}}}{\left(1-\frac{1}{1+\frac{\lambda}{\mu}}\right)^w \quad \left(\frac{1}{1+\frac{\lambda}{\mu}}\right)} \tag{13c}$$

Finally, we obtain the desired result $E(P|w) = (w+1)/(\lambda+\mu)$ from (13c) via straightforward algebraic manipulations.

References

Adan I, Sleptchenko A, Van Houtum G (2009) Reducing costs of spare parts supply systems via static priorities. Asia-Pacific J Oper Res 26(04):559–585

Arreola-Risa A, Giménez-García V, Martínez-Parra J (2011) Optimizing stochastic production-inventory systems: a heuristic based on simulation and regression analysis. Eur J Oper Res 213(1):107–118

Bertsekas D (2007) Dynamic programming and optimal control, 3rd edn, vol. II. Athena Scientific, USA

De Vericourt F, Karaesmen F, Dallery Y (2000) Dynamic scheduling in a make-to-stock system: a partial characterization of optimal policies. Oper Res 48(5):811–819

DeCroix G, Arreola-Risa A (1998) Optimal production and inventory policy for multiple products under resource constraints. Manag Sci 44(7):950–961

Dusonchet F (2003) Dynamic scheduling for production systems operating in a random environment. PhD thesis, École Polytechnique Fédérale de Lausanne

Gavish B, Graves S (1980) A one-product production/inventory problem under continuous review policy. Oper Res 28(5):1228–1236

Ha A (1997) Optimal dynamic scheduling policy for a make-to-stock production system. Oper Res 45(1):42–53

Janakiraman G, Nagarajan M, Veeraraghavan S (2009) Simple policies for managing flexible capacity. Working paper

Janssen R, Blankers I, Moolenburgh E, Posthumus B (2014) The impact of 3-d printing on supply chain management. Tech. rep, TNO

Kat B, Avşar Z (2011) Using aggregate fill rate for dynamic scheduling of multi-class systems. Ann Oper Res 182(1):87–117

Khajavi S, Partanen J, Hölmstrom J (2014) Additive manufacturing in the spare parts supply chain. Comput Ind 65(1):50–63

Li L (1992) The role of inventory in delivery-time competition. Manag Sci 38(2):182–197

Liang W, Balcıoğlu B, Svaluto R (2013) Scheduling policies for a repair shop problem. Ann Oper Res 211(1):273–288

Lieckens KT, Colen PJ, Lambrecht MR (2013) Optimization of a stochastic remanufacturing network with an exchange option. Dec Supp Syst 54(4):1548–1557

Linebaugh K (2008) Honda's flexible plants provide edge. Wall Street J 22:B1

Liu P, Huang SH, Mokasdar A, Zhou H, Hou L (2014) The impact of additive manufacturing in the aircraft spare parts supply chain: supply chain operation reference (scor) model based analysis. Prod Plann Control 25(13–14):1169–1181

Niño-Mora J (2007) Dynamic priority allocation via restless bandit marginal productivity indices. Top 15(2):161–198

Peña Perez A, Zipkin P (1997) Dynamic scheduling rules for a multiproduct make-to-stock queue. Oper Res 45(6):919–930

Porteus EL (2002) Foundations of stochastic inventory theory. Stanford University Press, Palo Alto

Sanajian N, Abouee-Mehrizi H, Balcıoğlu B (2010) Scheduling policies in the M/G/1 make-to-stock queue. J Oper Res Soc 61(1):115–123

Sobel M (1982) The optimality of full service policies. Oper Res 30(4):636–649

Steiger N, Wilson J (2001) Convergence properties of the batch means method for simulation output analysis. INFORMS J Comput 13(4):277–293

Van Houtum G, Adan I, Van der Wal J (1997) The symmetric longest queue system. Stoch Models 13(1):105–120

Veatch M, Wein L (1996) Scheduling a make-to-stock queue: index policies and hedging points. Oper Res 44(4):634–647

Wein L (1992) Dynamic scheduling of a multiclass make-to-stock queue. Oper Res 40(4):724–735

Whittle P (1988) Restless bandits: activity allocation in a changing world. J Appl Probab 25:287–298

Zheng Y, Zipkin P (1990) A queueing model to analyze the value of centralized inventory information. Oper Res 38(2):296–307

Zipkin P (1995) Performance analysis of a multi-item production-inventory system under alternative policies. Manag Sci 41(4):690–703