Abstract

We study the value of multi-stage advance demand information (MADI) in a production system in which customers place an order in advance of their actual need, and each order goes through multiple stages before it becomes due. Any order which is not immediately filled at its due date will be backordered. The producer must decide whether to produce or not based on real-time information regarding current and future orders. We formulate the problem as a Markov decision process and analyze the impact of the demand information on the production policy and the cost. We show that the optimal production policy is a state-dependent base-stock policy, and we show that it has certain monotonicity properties. We also introduce a simple heuristic policy that is significantly easier to compute and that inherits the structural properties of the optimal policy. In addition, we show that its base-stock levels bound those of the socially optimal policy. Numerical study identifies the conditions under which MADI is most beneficial and shows that the heuristic performs almost as well as the optimal policy when MADI is most beneficial.

Keywords: Multi-stage advanced demand information; state-dependent base-stock policy; Markov decision model; individually optimal policy
1 Introduction

We consider a make-to-stock firm with a single product where there is a time lapse between when an order is initially placed (i.e., the firm is informed of the customer’s intention to buy the product) and when it is due (i.e., the customer is ready to buy the product). Any new order must go through multiple stages successfully before it becomes due, and only a portion of all orders will materialize as firm orders. We call the difference between the time at which an order is placed and the time at which the order is due the demand leadtime and the evolution (status change) of an order over time until it is due or cancelled the demand realization process. The decision maker can learn the status of each existing order (i.e., which stage it is currently in) and adjust the production policy according to this information in real time. We call our model multistage advance demand information (MADI), and our goal is to analyze the impact of advance information on the optimal production policy and the optimal cost.

In particular, we show that the optimal policy is a state-dependent base-stock policy and the base stock (threshold) is monotone in the number of orders as well as in the progress of an order in the demand realization process. Another important contribution of our paper is the development of a practical heuristic policy that captures most of the benefit of advanced demand information and is significantly easier to compute. We show that the heuristic policy possesses the same monotonicity properties as the socially optimal policy, and it provides a bound for the socially optimal action. Numerical study shows that it performs very well when MADI is most beneficial. In the rest of this section, we first describe our model in detail and then an application that motivated our research along with relevant literature.

1.1 Model

Suppose that each unit is produced one at a time, and finished units can be stored as inventory. The production time for each unit is exponentially distributed with some rate $\mu$ which can be any value in $[0, \bar{\mu}]$ and is chosen by the firm. New customer orders are placed according to a Poisson process with rate $\nu$ ($\nu < \bar{\mu}$), and each order is for one unit of product. Once a new order is placed, it goes through several sequential stages until it becomes due or is cancelled. Specifically, we assume there are $m$ stage(s), the time that an order is in stage $j$, $j = 1, \cdots, m$ is exponentially distributed, and the rate of this transition from a stage may depend on the number of orders in the stage. When the order enters stage $m + 1$ (i.e., the order passes through $m$ stages without cancellation), the order is due and will be immediately filled with existing inventory if there is any. If there is no inventory to satisfy the order, it is backordered. The firm incurs holding cost for on-hand inventory and backorder cost for orders that are filled after they are due. The objective is to find a production policy that minimizes the expected discounted cost for a finite (or infinite) time horizon.

We model this problem as a Markov decision process. Let $d_j(t) \in \mathbb{Z}^+$, $j = 1, \cdots, m$, be
the number of pending orders in stage $j$ at time $t$, and let the $m$-dimensional vector $\mathbf{d}(t) = (d_1(t), \cdots, d_m(t))$ represent the collection of pending orders in the demand realization process at time $t$. We use $d_{m+1}(t)$ to represent on-hand inventory (if negative) or backorders (if positive) at time $t$. Let $\mathbf{s}(t) = (\mathbf{d}(t), d_{m+1}(t))$ denote the state of the system at time $t$ and let $S \equiv (\mathbb{Z}^+)^m \times \mathbb{Z}$ be the set of all possible states.

Depending on the triggering event, there are four different types of transitions pertinent to the evolution of the Markov chain: a new order arrival to stage 1 (changing $d_1(t)$ to $d_1(t) + 1$), completion of a production (changing $d_{m+1}(t)$ to $d_{m+1}(t) - 1$), cancellation of an order in stage $j$ (changing $d_j(t)$ to $d_j(t) - 1$), and advance of an order in stage $j$ to stage $j + 1$ (changing $d_j(t)$ and $d_{j+1}(t)$). While the first two types of transitions are intuitive the other two need further explanation.

Let $\lambda_j(d_j(t))$ be the demand realization rate at which one of the $d_j(t)$ orders leaves stage $j$. Upon leaving stage $j$, the order will be cancelled with probability $p_j$ or advance to stage $j + 1$ with probability $q_j = 1 - p_j$. We assume that the transition rate from stage $j$, $\lambda_j(d_j(t))$, is increasing\(^1\) in $d_j(t)$, $j = 1, \cdots, m$, and $\lambda_j(0) = 0$.

![Demand realization process](image)

Figure 1 Demand realization process

We now describe the costs, actions, and resulting optimization problem. On-hand inventory of finished units incurs holding costs at the rate of $h$ per unit time per item while backorders incur costs at the rate of $b$ per unit time per item. Let $g(d_{m+1}(t))$ be the cost rate when the inventory/backorder level is $d_{m+1}(t)$, which is

$$g(d_{m+1}(t)) = hd_{m+1}(t)^- + bd_{m+1}(t)^+$$

where $y^+ = \max\{0, y\}$ and $y^- = \max\{0, -y\}$ for any real number $y$. Note that even though we assume that the inventory cost and the backorder cost are linear, all of our results and proofs hold when $g(d_{m+1}(t))$ is a general convex function in $d_{m+1}(t)$ as long as $g(d_{m+1}(t))$ is minimized at $d_{m+1}(t) = 0$ (e.g., $g(d_{m+1}(t)) = h(d_{m+1}(t)^-) + b(d_{m+1}(t)^+)$, where $h(\cdot)$ and $b(\cdot)$ are convex functions and $g(0) = 0$).

\(^1\)Throughout the paper, we use terms in the non-strict sense so that increasing/decreasing means nondecreasing/nonincreasing and positive/negative means nonnegative/nonpositive.
For a given state $s(t)$, the firm chooses a production rate $\mu \in [0, \bar{\mu}]$, which defines the action. Let $\pi : S \rightarrow [0, \bar{\mu}]$ be a deterministic Markov stationary policy, and let $\Pi$ be the collection of all such policies. We define $V(s)$ as the expected total discounted cost-to-go function associated with the optimal policy over $\Pi$ and discount rate $\alpha > 0$ when the initial state is $s$ at time 0:

$$V(s) = \min_{\pi \in \Pi} \mathbb{E}_s \left[ \int_0^\infty e^{-\alpha t} g(d_{m+1}(t)) dt \right].$$

Our results also hold for a more general model in which new orders can arrive into any stage of the demand realization process (including stage $m+1$, which corresponds to no advance information).

1.2 Motivation and literature review

The problem was motivated by a company in the San Francisco Bay Area that manufactures and installs solar panel systems for large buildings. Rising energy costs and government subsidies for alternative energy sources have boosted demand for solar panels. Though sales were rapidly increasing, the company was also experiencing a rapid unexpected increase in inventory related costs. For most orders, there is an application for a government subsidy (up to 50% of the total cost) to the Department of Energy (DOE), who evaluates the proposal and approves (or rejects) the subsidy. The DOE’s evaluation consists of several steps including preliminary evaluation, developer qualification, data collection, site visits, and assessment. Each of these steps takes a random amount of time and cannot be controlled by the company. For most orders, the solar panel system can be installed only after the company gets approval from the DOE, so there is a significant gap between the time at which the order is first placed and the time at which it is due. In fact, the time it takes to receive approval ranges from 2 months to 2 years. In spite of this, the company currently schedules production and builds up available inventory based only on the times at which orders are placed, and does not utilize the advance information on the progress of the order before becoming due. In particular, when an order is in an early stage of evaluation, it is unlikely that the order will become due soon. On the other hand, when an order is in a late evaluation stage, it is likely that the order will become due very soon.

Similar problems arise in project-based supply chains such as in the construction industry or the customized capital equipment industry, as well as in assembly systems where the completion time of a product needs to be matched with the completion time of another product that goes through a sequential production process. In these examples, a finished product can be viewed as a project comprised of several key steps, thus the status (step) of the project provides useful information to suppliers participating in the project (Donselaar et. al. [27], Terwiesch et. al. [25]). In the construction example, customers usually finish preliminary planning and design at least several weeks (if not months or years) before the project starts, and at this point, the producer is informed of a new order placement (called a soft demand). In complex assembly systems (e.g.,
airplanes, freight vessels), matching the completion time of one component with the completion
time of another component is critical to minimize delay. The status of one component provides
useful information on when the matching component should be produced.

Our problem is related to literature on the value of information sharing in supply chain man-
agement (SCM), which has been studied extensively for quite some time. See Aviv [2], Aviv and
Federgruen [3], Cohen et. al. [5], Ehrhardt [6], Gavirneni et. al. [8], Lee et. al. [14], Scarf [20],
Simchi-Levi and Zhao [21], Cohen et. al. [5], and Song and Zipkin [23]. In this research, the
main focus has been for the producers to forecast the exogenous end-customer demand patterns
using historical data provided by the retailers. However, in some cases including our problem, the
customers might anticipate and provide information on future demand. For this, advance demand
information (ADI) has been introduced. In a typical ADI system, each order arrives with a due
date, which indicates some future time when the customer wants to receive the product. Donselaar
[27], Gallego and Ozer [7], Hariharan and Zipkin [10], Karaesmen et. al. [11], [12], Ozer and Wei
[16], and Thonemann [26] study such models using deterministic positive demand lead times and a
fixed-length information horizon. A critical assumption in all of these papers is that the customer
knows exactly when they want to receive the product. When the customer’s need for the prod-
uct depends on other exogenous factors, as in project-based supply chains studied in our paper, a
stochastic demand leadtime is more appropriate.

Gayon et. al. [9] consider a setting similar to ours with a make-to-stock supplier with limited
capacity in a continuous-time review system. There is a random time lag between an order place-
ment and the order realization and each order evolves independently. This is, in effect, a one-stage
demand realization process where the demand realization rate is a linear function of the number
of pending orders in our model. They include several demand classes differing in their new order
arrival rates, expected due dates, and lost sales costs. They show that a state-dependent base-stock
policy is optimal and the optimal inventory allocation policy is a multi-level rationing policy. In
our model, we do not consider multiple demand classes. However, we model the random time lag
as a multi-stage stochastic process rather than a single-stage process in Gayon et. al. [9], and we
study a problem in which orders that are not filled immediately are not lost but are backordered. In
addition, we generalize the linear transition rate in the demand realization process to an increasing
function in the number of pending orders so that the evolution of pending orders are correlated
with each other. Later, we show that in a model with a multi-stage stochastic demand realization
process and a generalized transition rate, the monotonicity properties that they show can be shown
with an extra condition. We also introduce an easily implementable heuristic that captures most
of the benefits of ADI.

Another stream of literature that is relevant to our research in terms of modeling approach
is Markov decision models for control of queueing network. See Rosberg et al. [18], Weber and
Stidham [28], [24], and Altman and Koole [1]. Our model is also a Markov decision model but it is
different in that all uncontrollable transition rates in the network are state dependent, rather than
fixed and independent of the system state.

2 The optimal policy

Before we describe our results, we discuss our assumption that the demand realization rate, $\lambda_j(d_j(t))$, is increasing in $d_j(t)$, $j = 1, \cdots, m$. This assumption is satisfied if, for example, the order departure process in each stage has faster rate as there are more orders in the stage. Without this assumption, the monotonicity properties described in Theorem 2.4 and Lemma 2.5 may not hold. For example, suppose that $\lambda_j(1) = \cdots = \lambda_j(y) \gg \lambda_j(y + 1) = \cdots = \lambda_j(\infty) = 0$ for $j = 1, \cdots, m$, that is, the departure process from any stage will stop if there are at least $y$ orders in the stage. Suppose $\lambda_j(1)$ for all $j \leq m$ is big enough so that it is optimal to produce if there is only one order in the system. Also, suppose that, as new orders arrive and progress, the system enters state $(d_1(t), \ldots, d_{m-1}(t), y + 1, d_{m+1}(t))$ where $d_i(t) < y$ for $i = 1, \ldots, m - 1$. Since no order will come due in the future, it is optimal to wait from now on, so P6 and P7 in Lemma 2.5 no longer hold true.

One important special case for the demand realization rate is when $\lambda_j(d_j(t)) = \lambda_j * d_j(t)$ for some $\lambda_j > 0$ (the linear case). That is, orders progress independently through stage $j$ at rate $\lambda_j$, $j = 1, \cdots, m$. In this case, the cancellation in the model is equivalent to assuming that demands in stage $j$ will independently be cancelled after spending some time that is exponentially distributed with rate $\beta_j$ in stage $j$ if they have not progressed to stage $j + 1$, and that they will progress independently to stage $j + 1$ after some time that is exponentially distributed with rate $\gamma_j$ if not cancelled. Then, the overall rate at which an order leaves stage $j$ is $\lambda_j := \beta_j + \gamma_j$ and the cancellation probability is simply $p_j = \frac{\beta_j}{\beta_j + \gamma_j}$. While it is possible to permit the case where the cancellation probability depends on the number of orders in stage $j$ (i.e., $p_j(d_j)$), this makes the notation and argument significantly more cumbersome with marginal additional insights, so we restrict ourselves to the case $p_j(d_j) = p_j$.

Instead of dealing with the continuous-time optimization problem, we transform the problem into an equivalent discrete-time Markov decision process using uniformization. In order to avoid a potentially infinite transition rate in the demand realization process at any time, we assume that only a finite number (say $N$) of pending orders can stay in the system. Thus, we assume that orders are no longer accepted when the system is very heavily loaded (i.e., there are $N$ orders already in the system), which is a realistic assumption for most business practices. Note that the choice of $N$ could be arbitrarily large, thus this assumption is not restrictive. Let $S_N$ be the set of feasible states when truncated by $N$.

In order to define the optimality equation, we first introduce operators that represent each of the four possible transitions. We use the same notation in the discrete-time model as in the
continuous-time model, but we suppress the time dependency in the notation; for example, we use $d_j$ instead of $d_j(t)$. Let $E_j$, $P$, $C_j$ and $R_j$ be the operators representing an extra order in stage $j$ (in particular, $E_1$ represents a new order arrival), completion of a production, cancellation of an order in stage $j$, and advance of an order in stage $j$ to stage $j + 1$, respectively:

$$E_j(s) = \begin{cases} s + e_j, & \text{if } \sum_{i=1}^{m} d_i < N \\ s, & \text{if } \sum_{i=1}^{m} d_i = N \end{cases} \quad \text{for } j = 1, \cdots, m$$

$$P(s) = s - e_{m+1}$$

$$C_j(s) = \begin{cases} s - e_j, & \text{if } d_j > 0 \\ s, & \text{if } d_j = 0 \end{cases} \quad \text{for } j = 1, \cdots, m$$

$$R_j(s) = \begin{cases} s - e_j + e_{j+1}, & \text{if } d_j > 0 \\ s, & \text{if } d_j = 0 \end{cases} \quad \text{for } j = 1, \cdots, m$$

where $e_i$ is the $i$th unit vector of dimension $m + 1$. Note that that when an order advances from stage $j$ to stage $j + 1$, the transition $R_j$ decreases $d_j$ by 1 and increases $d_{j+1}$ by 1. When $j = m$, the transition $R_m$ represents the case when an order is finally due.

For a given choice of $N$, let $\lambda_{\max} = \max_{1 \leq i \leq m} \lambda_i(N)$, so that $\lambda_{\max}$ is the maximal rate in the demand realization process. Without loss of generality, we use a uniformization constant of 1, so that $\alpha + \nu + \lambda_{\max} + \bar{\mu} = 1$. Applying uniformization, we can rewrite $V(s)$ as an equivalent discrete-time optimality equation. For all $s = (d, d_{m+1}) \in S_N$,

$$V(s) = g(d_{m+1}) + \nu V(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i) \left[ p_i V(C_i(s)) + (1 - p_i) V(R_i(s)) \right] + \left\{ \lambda_{\max} - \sum_{i=1}^{m} \lambda_i(d_i) \right\} V(s) + \min_{\mu \in [0, \bar{\mu}]} \left[ \mu V(P(s)) + (\bar{\mu} - \mu) V(s) \right] \quad (1)$$

Note that we have not explicitly considered any production cost associated with rate $\mu$. However, this is done without sacrificing the qualitative insights that the model provides. Suppose there is a production cost, and let $c(\mu)$ be the cost per unit time when producing at rate $\mu$. If $c(\mu)$ is concave increasing (e.g., linear) in $\mu$, the following, easily shown, Lemma shows that it is optimal to choose $\mu$ to be either 0 or $\bar{\mu}$ as opposed to an intermediate value between 0 and $\bar{\mu}$.

**Lemma 2.1.** Consider the model described in equation (1) with production cost $c(\mu)$. If $c(\mu)$ is increasing concave, then the optimal policy is of a bang-bang type, that is, $\mu^* = 0$ (wait) or $\bar{\mu}$ (produce at the maximum rate).
We note that if \( c(\mu) \) is increasing convex, then an interior value \( \mu \in (0, \bar{\mu}) \) can be chosen as an optimal production rate. While it is possible to permit such cost functions, doing so will not change the main qualitative insights on the benefit of advanced information (although the optimal production rate might vary depending on the state). For the remainder of the paper we assume that \( c(\mu) \) is concave, so we focus on a model in which the decision is either wait or produce at rate \( \bar{\mu}: \mu = 0 \) or \( \bar{\mu} \).

Now we show that, in a very strong sense, it is always optimal to produce in any state with backorders. The fairly standard sample-path proof is omitted.

**Theorem 2.2.** If \( \pi \) is a policy that waits at some time when \( d_{m+1} > 0 \), then there exists a policy \( \pi' \) that produces at that time and that has stochastically smaller cost rate at all times than \( \pi \). Therefore, for all \( s \in S_N \), if \( d_{m+1} > 0 \), the total cost up to any time of producing in state \( s \) is stochastically smaller than the cost of waiting.

From Theorem 2.2, the policy that produces when \( d_{m+1} > 0 \) costs less in every transition interval than the policy that does not until the two sample paths merge. As a result, the expected discounted cost will be smaller under the policy that produces when \( d_{m+1} > 0 \).

**Corollary 2.3.** For all \( s = (d, d_{m+1}) \in S_N \) with \( d_{m+1} > 0 \), it is optimal to produce, i.e., \( V(P(s)) \leq V(s) \).

We note that Theorem 2.2 and Corollary 2.3 hold regardless of whether orders can be cancelled during the realization process or not. In either case, we only need to determine the optimal policy for states with \( d_{m+1} \leq 0 \). In the next two subsections, we show that the optimal policy is indeed a state-dependent base-stock policy, and the threshold curve characterizing the optimal policy has some interesting monotonicity properties. We first consider a model without order cancellation because it is simpler and more intuitive. We then extend the model to the case with order cancellation and show that the optimal policy has most of the same structure as without order cancellation when a certain condition is met.

### 2.1 Model without order cancellation

We first suppose customers are not allowed to cancel their orders once they are placed (i.e., enter stage 1), so \( p_j = 0, j = 1, \cdots, m \). In this case, equation (1) can be rewritten as

\[
V(s) = g(d_{m+1}) + \nu V(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i) V(R_i(s)) + \left[ \lambda_{\max} - \sum_{i=1}^{m} \lambda_i(d_i) \right] V(s) \\
+ \min_{\mu \in \{0, \bar{\mu}\}} \left[ \mu V(P(s)) + (\bar{\mu} - \mu) V(s) \right]
\]  

(2)
For states with backorders, \( \mu = \bar{\mu} \) achieves the minimum from Corollary 2.3. Consider a state \( s = (d, d_{m+1}) \in S_N \). Note that a special case of our model, when \( m = 0 \), is a standard inventory model for which a base stock policy is optimal. That is, there exists a fixed threshold \( T \), such that it is optimal to produce if and only if the inventory level is at or below \( T \). For our more general model, we will show that there exists a state dependent threshold curve, \( T(d) \), such that if the inventory level, \(-d_{m+1}\), is less than or equal to \( T(d) \), it is optimal to produce; otherwise, waiting is optimal. We can think of \( T(d) \) as a target inventory level. Now, we will show some monotonicity properties of \( T(d) \), e.g., that the threshold, or target inventory level, increases as the number of orders increases and as they move closer to becoming due.

To characterize \( T(d) \), we first extend (and abuse) our notation for the operators, \( E_j, P, C_j, R_j \), defined earlier. Let \( \Gamma(s) = (\Gamma_1(s), \ldots, \Gamma_m(s), \Gamma_{m+1}(s)) \) where \( s = (d, d_{m+1}) \), \( \Gamma \in \{E_j, P, C_j, R_j\} \) and \( \Gamma_i(s) \) is the \( i \)th component of \( \Gamma(s) \). Let \( \hat{\Gamma}(d) = (\Gamma_1(s), \ldots, \Gamma_m(s)) = (\Gamma_1(d, \cdot), \ldots, \Gamma_m(d, \cdot)) \), so \( \hat{\Gamma}(d) \) is the same as \( \Gamma(s) \) except the last component.

We now give our main result for the optimal policy. Since we will introduce a heuristic policy called the individually optimal policy in Section 3, and this policy optimizes the cost associated with one particular order, we will call the solution of the optimality equation (2) the socially optimal policy.

**Theorem 2.4.** The socially optimal policy is a state-dependent base-stock threshold policy. It has the following properties.

**P1** There exists a threshold function \( T(d) \geq 0 \) such that for any \( s \in S_N \), if the inventory level, \(-d_{m+1}\), is less than or equal to \( T(d) \), it is optimal to produce; otherwise, waiting is optimal.

**P2** The optimal threshold \( T(d) \) is increasing in \( d_j \) so if it is optimal to produce in state \( s \), then it is also optimal to produce in a state with one more order in stage \( j \), \( E_j(s), j = 1, \ldots, m \). In particular, if it is optimal to produce in the current state, then it will still be optimal to produce if a new order is placed. However, in any case, the optimal threshold does not increase by more than 1. That is, \( T(d) \leq T(\hat{E}_j(d)) \leq T(d) + 1, j = 1, \ldots, m \).

**P3** \( T(d) \) is increasing as orders move downstream, except for the movement from stage \( m \) (order is due). Again the optimal threshold does not increase by more than 1. Therefore, \( T(d) \leq T(\hat{R}_j(d)) \leq T(d) + 1 \) for \( j = 1, \ldots, m - 1 \). When a pending order in stage \( m \) is due, since we have one less order in the demand realization process, \( T(d) \) decreases but not by more than 1. That is, \( T(\hat{R}_m(d)) \leq T(d) \leq T(\hat{R}_m(d)) + 1 \). Hence, if it is optimal to produce in state \( s \), then it is also optimal to produce in state \( R_j(s), j = 1, \ldots, m \).

**P4** \( T(d) \) is increasing in \( b, \nu \), and \( \lambda_j(\cdot) \), \( j = 1, \ldots, m \), and it is decreasing in \( h \) and \( \bar{\mu} \).
Note that properties \( P_2 \) and \( P_3 \) guarantee that if it is optimal to produce, then it is optimal to continue producing until the unit is complete. That is, the production policy is \textit{transition monotone} so production will never be interrupted by any transition under the optimal policy.

The threshold curve for a single stage model is illustrated in Figure 2. From the figure, we see that the threshold, or target inventory level, \( T(d) \) is increasing, but for a unit increase in \( d_1 \), it increases by at most 1. The figure also includes the optimal base-stock level (the no-MADI threshold) which characterizes the policy for the standard inventory model with no advance information. The other curve represents the individually optimal threshold will be explained later.

Our proof of Theorem 2.4 uses the following Lemma, which describes the convexity and supermodularity properties of the socially optimal cost function. We use ‘\( \circ \)’ to denote the composition of two or more operators: For example, \( E_j \circ P(s) = E_j(P(s)) \). The proof of Lemma 2.5 is provided in Appendix A.

\textbf{Lemma 2.5.} The cost function under the SO policy has the following properties.

\begin{itemize}
  \item \textbf{P5} \( V(s) \) is convex in \( d_{m+1} \) for \( s = (d, d_{m+1}) \in S_N \).
  \item \textbf{P6} The marginal benefit of having one more unit in on-hand inventory, \( V(s) - V(P(s)) \), is increasing in the number of pending orders in stage \( j \), \( d_j \), \textit{ceteris paribus}. That is,
    \[ V(s) - V(P(s)) - V(E_j(s)) + V(P \circ E_j(s)) \leq 0 \] for \( s \in S_N, j = 1, \cdots, m \).
  \item \textbf{P7} The marginal benefit of having one more unit in on-hand inventory, \( V(s) - V(P(s)) \), increases as an order moves downstream. That is,
    \[ V(s) - V(P(s)) - V(R_j(s)) + V(P \circ R_j(s)) \leq 0 \] for \( s \in S_N, j = 1, \cdots, m \).
\end{itemize}

Figure 2 The Threshold Functions (\( m = 1 \)).

\[ -d_{m+1} \]

\textbf{Wait}

\textbf{SO}

\textbf{IO}

\textbf{No-MADI}

\textbf{Produce}

\textbf{d}_1
\textbf{P8} \( V'(s) - V'(P(s)) - V(s) + V(P(s)) \leq 0 \) for \( s \in S_N \), where \( V'(s) \) is the expected discounted cost associated with the optimal policy when all the parameters are the same as \( V(s) \) except the backorder cost rate, \( b' \leq b \), the demand release rate, \( \nu' \leq \nu \), or the demand realization rate in stage \( j \), \( \lambda_j'(x) \leq \lambda_j(x) \), \( 1 \leq x \leq N \), \( j = 1, \ldots, m \).

On the other hand, \( V''(s) - V''(P(s)) - V(s) + V(P(s)) \geq 0 \) for \( s \in S_N \), where \( V''(s) \) is the expected discounted cost associated with the optimal policy when all the parameters are the same as \( V(s) \) except the inventory cost rate, \( h'' \leq h \), or the maximum production rate \( \bar{\mu}'' \leq \bar{\mu} \).

\textbf{Proof of Theorem 2.4}

\textbf{Proof of P1} Because \( V(s) = V(d, d_{m+1}) \) is convex in \( d_{m+1} \) (P5), there exists \( T(d) \) that is both a local and global minimizer. When \( -d_{m+1} \leq T(d) \), \( V(s) - V(P(s)) \geq 0 \) due to the convexity, so it is optimal to produce. Otherwise, \( V(s) - V(P(s)) < 0 \), so it is optimal to wait.

\textbf{Proof of P2} From P6, we know that if the optimal policy in state \( s = (d, d_{m+1}) \) is produce, then it is optimal to produce in \((\tilde{E}_j(d), d_{m+1})\). Therefore, \( T(d) \leq T(\tilde{E}_j(d)) \). In order to show that \( T(\tilde{E}_j(d)) \leq T(d) + 1 \), suppose \( T(\tilde{E}_j(d)) > T(d) + 1 \), that is, it is optimal to produce in state \( s = (\tilde{E}_j(d), -(T(d) + 2)) \). From P7, it should be optimal to produce in state \( R_m \circ R_{m-1} \circ \ldots \circ R_j \circ E_j(s) = (d, -(T(d) + 1)) \), which contradicts the definition of \( T(d) \). Therefore, \( T(d) \leq T(\tilde{E}_j(d)) \leq T(d) + 1 \) for \( j = 1, \ldots, m \).

\textbf{Proofs of P3 and P4} We omit the proofs because the arguments are similar to the proof of P2.

Note that if we allow a new order arrival not only to stage 1 but also to any stage in the demand realization process, then \( E_j \) will represent an arrival to stage \( j \), \( j \leq m + 1 \). Suppose that there are \( m + 1 \) different types of orders, and type \( j \) order arrives to stage \( j \), \( j \leq m + 1 \) (e.g., type \( j \) orders provide more advance information than type \( j + 1 \) orders). Also suppose that a type \( j \) orders arrive according to a Poisson process with rate \( \nu_j \) (see Figure 3). Then, the same result can be shown in this more general model with a similar argument.

\begin{figure}[h]
\centering
\begin{tikzpicture}
\node (d1) at (0,0) [shape=circle,draw] {$d_1$};
\node (d2) at (1,0) [shape=circle,draw] {$d_2$};
\node (d3) at (2,0) [shape=circle,draw] {$d_3$};
\node (dm) at (3,0) [shape=circle,draw] {$d_m$};
\node (dm1) at (4,0) [shape=circle,draw] {$d_{m+1}$};
\node (v1) at (-1,1) {$v_1$};
\node (v2) at (1,1) {$v_2$};
\node (v3) at (3,1) {$v_3$};
\node (v4) at (5,1) {$v_{m+1}$};
\node (v5) at (-1,-1) {$p\lambda(d_1)$};
\node (v6) at (1,-1) {$p\lambda(d_2)$};
\node (v7) at (3,-1) {$p\lambda(d_m)$};
\node (v8) at (5,-1) {$p\lambda(d_{m+1})$};
\node (v9) at (0,0) [shape=circle,draw] {$d_1$};
\node (v10) at (1,0) [shape=circle,draw] {$d_2$};
\node (v11) at (2,0) [shape=circle,draw] {$d_3$};
\node (v12) at (3,0) [shape=circle,draw] {$d_m$};
\node (v13) at (4,0) [shape=circle,draw] {$d_{m+1}$};
\node (v14) at (-1,1) {$v_1$};
\node (v15) at (1,1) {$v_2$};
\node (v16) at (3,1) {$v_3$};
\node (v17) at (5,1) {$v_{m+1}$};
\node (v18) at (-1,-1) {$p\lambda(d_1)$};
\node (v19) at (1,-1) {$p\lambda(d_2)$};
\node (v20) at (3,-1) {$p\lambda(d_m)$};
\node (v21) at (5,-1) {$p\lambda(d_{m+1})$};
\path[->]
(d1) edge node {$q\lambda(d_1)$} (v1)
(d2) edge node {$q\lambda(d_2)$} (v2)
(d3) edge node {$q\lambda(d_m)$} (v3)
(dm) edge node {$q\lambda(d_{m+1})$} (v4)
(d1) edge node {$p\lambda(d_1)$} (v5)
(d2) edge node {$p\lambda(d_2)$} (v6)
(dm) edge node {$p\lambda(d_m)$} (v7)
(dm1) edge node {$p\lambda(d_{m+1})$} (v8);
\end{tikzpicture}
\caption{Model with new arrivals to any stage.}
\end{figure}
Our model seems to fit in the framework of Altman and Koole [1], and the transition monotonicity of the optimal policy appears to follow their submodularity. We provide a direct proof because it is easier than checking the conditions of their framework, and it provides a more intuitive understanding of the particular structure of our model.

2.2 Model with cancellation

We now suppose any order leaving stage $j$ can be cancelled with probability $p_j$ (and if cancelled, it fails to advance to stage $j + 1$ and leaves the system) without any explicit cost. Specifically, if there are $d_j$ orders in stage $j$, the rate at which one of $d_j$ orders will be cancelled is $\lambda_j(d_j)p_j$. A similar argument as in the proof of property $P5$ in Lemma 2.5 shows that $P5$ is preserved for the system with order cancellation. Therefore, the SO policy is still defined by a threshold function, $T(d)$ ($P1$). However, we need a condition on the cancellation rates to prove that the optimal policy preserves the same monotonicity properties, $P2$-$P8$, when there is more than one stages in the demand realization process.

Define $\delta_j(d_j)$ to be the increment in the demand realization rate in stage $j$ as the number of orders in stage $j$ increases from $d_j - 1$ to $d_j$, that is, $\delta_j(d_j) = \lambda_j(d_j) - \lambda_j(d_j - 1)$ for all $j = 1, \cdots, m$ and $d_j = 1, \cdots, N$. Thus, $\delta_j(d_j)p_j = (\lambda_j(d_j) - \lambda_j(d_j - 1))p_j$ represents the increment in the demand cancellation rate in stage $j$ as the number of orders increases from $d_j - 1$ to $d_j$ in stage $j$. To prove Theorem 2.6, we require condition A below, which says that total cancellation rates decrease as demands move downstream. Under condition A, the proof is similar to that of Theorem 2.4, so we omit it.

**Condition A** $\delta_j(d_j)p_j \geq \delta_{j+1}(d_{j+1} + 1)p_{j+1}$ for $j = 1, \cdots, m$ and $d_j = 0, \cdots, N - 1$.

**Theorem 2.6.** $P1$ and $P5$ hold for the system with order cancellation, i.e., the socially optimal policy is a state-dependent base-stock policy. Furthermore, if condition A is met, then $P1$-$P8$ hold. In addition,

$P9$ $T(d)$ is decreasing in the cancellation probability $p_j$, $j = 1, \cdots, m$.

When an order advances from stage $j$ to stage $j + 1$, this transition will increase the cancellation rate at stage $j + 1$ by $\delta_{j+1}(d_{j+1} + 1)p_{j+1}$ and will decrease the cancellation rate at stage $j$ by $\delta_j(d_j)p_j$. Condition A simply stipulates that the total cancellation rate will not increase as an order advances downstream in the demand realization process. Condition A is not restrictive. If the demand realization process has only one stage ($m = 1$), as in Gayon et. al. [9], the condition is trivially satisfied. If the demand realization rate in stage $j$ is linear in $d_j$, that is $\lambda_j(d_j) = \lambda_j d_j$,
j = 1, \ldots, m$, condition $A$ reduces to $\lambda_j p_j \geq \lambda_{j+1} p_{j+1}$. So, for example, if $\lambda_j \equiv \lambda$ is the same for all stages, and if $p_j$ is the same for all $j$ or decreasing in $j$, i.e., an order is less likely to be cancelled as it approaches its due date, then condition $A$ holds.

If condition $A$ does not hold, then we can not rule out the possibility that it is optimal to stop production when an order moves from stage $j$ to stage $j+1$ (without being cancelled) in some states. This would imply that $T(d) < T(R_j(d))$, and claim $P2$ would no longer hold. In particular, in our induction proof of $P2$, if

$$\{\delta_j(d_j)p_j - \delta_{j+1}(d_{j+1} + 1)p_{j+1}\} \{V_n(C_j(s)) - V_n(C_j(P(s))) - V_n(R_j(s)) + V_n(P(R_j(s)))\} \quad (3)$$

is negative, then we can show that $V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(R_j(s)) + V_{n+1}(P \circ R_j(s)) \leq 0$, so $P2$ can be shown by a similar argument as in the model without cancellation. Note that if condition $A$ holds, (3) is negative from condition $A$ and the induction hypothesis for $P6$ and $P7$. However, if condition $A$ does not hold, it is not clear whether $V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(R_j(s)) + V_{n+1}(P \circ R_j(s))$ is negative or not because (3) is positive. Although condition $A$ is used in our proof of $P2$ for an algebraic reason, we believe it may not be necessary. As any particular order advances to a later stage, it is more likely that the order will ultimately become due without cancellation so we conjecture that $P1$-$P9$ hold for the system with cancellation, even when condition $A$ is not met. However, in any case, the optimal policy is no longer transition monotone; it may be optimal to interrupt a production when a cancellation occurs.

### 3 Individually optimal base-stock policy

Utilizing the information on the multi-stage demand realization process improves system performance. However, the computational complexity of finding the socially optimal (SO) policy becomes a new challenge as we increase the number of stages in order to build a more accurate model. In most of this section, we assume there is no order cancellation but, later, explain what changes when order cancellation is allowed. We introduce a myopic policy that only considers the first order to become due for which a product is not yet ready (which is the first backorder if there is one). We call it the individually optimal (IO) policy and show that it has the same monotonicity properties as the socially optimal (SO) policy. It also provides a lower bound for the socially optimal threshold function $T(d)$. Our numerical study shows that when MADI is beneficial, the IO policy performs significantly better than the optimal base-stock policy without MADI.

Later, we introduce an algorithm to compute the IO policy. In the algorithm, we assume that there is an upper bound on the inventory level, say $H$, which is finite but can be arbitrarily large, in order to avoid an infinite state space. This is realistic since for most businesses, there is finite warehouse capacity. The computational complexity of the IO algorithm is $O(mN^m \log H)$; it depends
on the size of the buffer \((N)\), the number of stages in the demand realization process \((m)\), and the upper limit on the inventory level \((H)\). On the other hand, the principal method for calculating the SO policy is value iteration, which requires \(O(mN^mH)\) computations for each iteration, and may take many iterations to converge. In our numerical study, we used an error bound of 0.05%; if the difference by computing one more iteration is less than 0.05% of the cost, the algorithm is terminated. In more than 100 different scenarios in our numerical study, the number of iterations to find the SO policy ranged between 7000 and 1,000,000. On the other hand, computing only one iteration is enough to find the IO policy.

It is easy to see that when there is a backorder \((d_{m+1} > 0)\), so that the IO policy minimizes the cost for the first backorder, it is optimal to produce, as in the SO policy. Therefore, unless we explicitly state otherwise, we assume that \(d_{m+1} \leq 0\) in state \(s\).

Without loss of generality, the first \(|d_{m+1}|\) orders to become due will take the \(|d_{m+1}|\) units that are already in the inventory; therefore, there are no future decisions to be made for them. We call these first \(|d_{m+1}|\) orders sunk orders, and we call the other \(\sum_{i=1}^{m} d_i - |d_{m+1}|\) orders and all future orders outstanding orders. The next item to be produced will be given to the \(|d_{m+1}| + 1\) order to become due, so, without loss of generality, we call it the priority order, and we call the outstanding orders except the priority the remaining orders. In order to compute the IO policy, which is a policy that is optimized only for the priority order, we first define \(\Psi(s)\) as the time until the priority order is due when the current state is \(s\). Note that when \(\sum_{i=1}^{m} d_i = N\), the order buffer is full and any new orders will be lost, therefore \(\Psi(E_1(s)) = \Psi(s)\). The operators \(E_1\) and \(R_j\)'s are defined as in the previous section. Then,

\[
\Psi(s) = \frac{1}{\sum_{i=1}^{m} \lambda_i(d_i) + \alpha + \nu} \left\{ 1 + \nu \Psi(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i) \Psi(R_i(s)) \right\} \quad (4)
\]

Let \(M^{IO}(s)\) (\(M\) for “Me first”) be the expected cost for the priority order under the IO policy. In the IO policy, the priority order chooses to produce or not in order to minimize its own cost. The priority order remains as a priority order until a unit is produced for it. Once production of a new unit completes, the priority order becomes a sunk order incurring the cost \(h\Psi(s)\). Therefore, the cost for the priority order is

\[
M^{IO}(s) = \frac{1}{\sum_{i=1}^{m} \lambda_i(d_i) + \alpha + \nu + \bar{\mu}} \left[ \nu M^{IO}(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i) M^{IO}(R_i(s)) + \min_{\mu \in (0, \bar{\mu})} \left\{ \mu h\Psi(s) + (\bar{\mu} - \mu) M^{IO}(s) \right\} \right]. \quad (5)
\]

For \(d_{m+1} > 0\), the priority order is past due, therefore \(\Psi(s) = 0\). The cost for the priority order is the backorder cost until the production of one unit is complete, that is, \(M^{IO}(s) = \frac{b}{n+\alpha}\).
3.1 Properties of the IO policy

The IO threshold function has the same monotonicity properties as the SO threshold function. The proof is similar to that of Theorem 2.4 and Lemma 2.5 so is omitted.

**Theorem 3.1.** The individually optimal policy is a state-dependent base-stock policy, i.e., there exists a threshold function $T_{IO}(d)$ such that for any $s \in S_N$, if the inventory level is less than or equal to $T_{IO}(d)$, it is IO to produce; otherwise, waiting is IO. $T_{IO}(d)$ has the properties, $P_1$-$P_8$, described in Theorem 2.4 and Lemma 2.5.

Note that, as in the SO policy, the IO production policy is transition monotone, so once production is started under the IO policy, it will never be interrupted until completion.

We now show that $T_{IO}(d)$ is a lower bound for $T(d)$, i.e., the SO policy has a higher target inventory level so will be more likely to produce than the IO policy. We define $S(s)$ as the cost for the sunk orders (those for which items have already been produced) and $W(s)$ as the SO cost-to-go function for the outstanding orders, and we recharacterize $V(s)$ as the sum of the sunk cost and the cost for the outstanding orders: $V(s) = S(s) + W(s)$. That is,

$$S(s) = \frac{1}{\sum_{i=1}^{m} \lambda_i(d_i) + \alpha + \nu} \left\{ g(d_{m+1}) + \nu S(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i)S(R_i(s)) \right\} \quad (6)$$

where $S(d,0) = 0$ and $S(d,-1) = h\Psi(s)$, and

$$W(s) = \frac{1}{\sum_{i=1}^{m} \lambda_i(d_i) + \alpha + \nu + \bar{\mu}} \left[ \nu W(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i)W(R_i(s)) + \min_{\mu \in (0,\bar{\mu})} \{ \mu (h\Psi(s) + W(P(s)) + (\bar{\mu} - \mu)W(s)) \} \right], \quad (7)$$

so $V(s) = S(s) + W(s)$. Since there is no decision variable in $S(s)$, finding the SO policy that minimizes $V(s)$ is equivalent to finding a policy that minimizes $W(s)$.

We also define $Q_{IO}(s)$ to be the expected cost for the remaining orders under the IO policy and let $W_{IO}(s) = M_{IO}(s) + Q_{IO}(s)$, so $W_{IO}(s)$ is the total expected discounted cost-to-go function associated with the IO policy, excluding the sunk cost $S(s)$. In other words, $V_{IO}(s) = W_{IO}(s) + S(s)$. We define $M_{SO}(s)$ and $Q_{SO}(s)$ in a similar way for the SO policy. That is, $M_{SO}(s)$ is the cost for the priority order and $Q_{SO}(s)$ is the cost for the remaining orders under the SO policy, so $W(s) = M_{SO}(s) + Q_{SO}(s)$. Let $Q_{RO}(s)$ be the cost for the remaining orders under the policy that is optimized only for the remaining orders, the RO policy. Once the priority order becomes a sunk order due to the completion of a new unit, the remaining orders become the outstanding orders so
$Q^{RO}(P(s))$ coincides with $W(P(s))$. Therefore,

$$Q^{RO}(s) = \frac{1}{\sum_{i=1}^{m} \lambda_i(d_i) + \alpha + \nu + \bar{\mu}} \left[ \nu Q^{RO}(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i) Q^{RO}(R_i(s)) + \min_{\mu \in \{0, \bar{\mu}\}} \{ \mu W(P(s)) + (\bar{\mu} - \mu) Q^{RO}(s) \} \right]$$

(8)

The following technical Lemmas are used in the proof of Theorem 3.4. Lemma 3.2 is true because $M^{IO}(s)$ is individually optimized for the priority order and $Q^{RO}(s)$ is optimized for the remaining orders.

Lemma 3.2. $M^{SO}(s) \geq M^{IO}(s)$ and $Q^{SO}(s) \geq Q^{RO}(s)$.

Lemma 3.3. When optimized only for the remaining orders, the action is always produce.

Proof. From equation (8), we see that if $W(P(s)) \leq Q^{RO}(s)$ then $P$ is optimal for the remaining orders. Let $\Phi_s$ be the set of remaining orders in state $s$. Notice that both $W(P(s))$ and $Q^{RO}(s)$ are the cost for the same $\Phi_s$ under a policy that is optimized for those orders. However, $W(P(s))$ can make a decision to wait or produce a unit to be used for $\Phi_s$ now while $Q^{RO}(s)$ must wait at least one more transition period to be able to make a unit for $\Phi_s$ because it needs to take care of the priority order first. In other words, having one unit ready for the priority order not only costs nothing for $\Phi_s$ but also allows the option of starting a new production for $\Phi_s$ right away. Therefore, $W(P(s)) \leq Q^{RO}(s)$.

Note that Lemma 3.3 trivially implies transition monotonicity of the RO policy, i.e., the optimal action for the remaining orders is always produce until production completes.

Now we are ready to prove that $T^{IO}(d)$ provides a lower bound for $T(d)$ by showing that when the IO action for a state is produce, then the SO action for that state is also produce. For convenience, when $F \in \{W, M^\pi, Q^\pi\}$ and $\pi \in \{SO, IO, RO\}$, let $F_W$ and $F_P$ denote the expected cost assuming the first action is wait and produce until it completes, respectively, and the relevant optimal policy is followed thereafter.

**Theorem 3.4.** If ‘produce’ is the IO action in state $s$, then it is also the SO action in the same state. Therefore, $T^{IO}(d)$ provides a lower bound for $T(d)$: $T^{IO}(d) \leq T(d)$.

Proof. If, in a certain state, produce is optimal for both the priority order and the remaining orders, it seems clear that produce is the SO action in the state. This can be more rigorously justified as follows. Suppose $M^{IO}(s) = M^{PO}(s)$, and remember that $M^{IO}(s) \leq M^{SO}(s)$, $Q^{RO}(s) \leq Q^{SO}(s)$ and $W(P(s)) \leq Q^{RO}(s)$ from Lemmas 3.2 and 3.3. We prove that $T^{IO}(d) \leq T(d)$ by showing that $W(s) = W_P(s)$ when $M^{IO}(s) = M^{PO}(s)$. From (5), $M^{IO}(s) = M^{PO}(s)$ if and only if $h\Psi(s) \leq M^{IO}(s)$. In order to show that $W(s) = W_P(s)$ when $h\Psi(s) \leq M^{IO}(s)$ (i.e., $M^{IO}(s) = M^{PO}(s)$),
we compare $h\Psi(s) + W(P(s))$ and $W(s)$.

\[
\begin{align*}
   h\Psi(s) + W(P(s)) & \leq M^{IO}(s) + W(P(s)) \\
   & \leq M^{SO}(s) + Q^{RO}(s) \leq M^{SO}(s) + Q^{SO}(s) = W(s).
\end{align*}
\]

That is, $h\Psi(s) + W(P(s)) \leq W(s)$, which implies that $\mu = \bar{\mu}$ in (7), so $W(s) = W_P(s)$. Therefore, $T^{IO}(d) \leq T(d)$. 

Next we show that the converse of Theorem 3.4 is not true.

**Theorem 3.5.** When 'produce' is the SO action in state $s$, it is not necessarily the IO action in the same state.

**Proof.** We give a counter example when $m = 1$. Consider the case in which $s = (6, -5), \alpha = 0.2, \nu = 0.5, \bar{\mu} = 1, \lambda_1 = 0.5, h = 1,$ and $b = 2$. Then, $V_P(6, -5) = 7.42785 < V_W(6, -5) = 7.46948$, so produce is the SO action. However, $M^{IO}_P(6, -5) = 0.912 > M^{IO}_P(6, -5) = 0.910$, so wait is the IO action.

Figure 2 in the previous section illustrates the structure of the IO threshold policy and the fact that the IO threshold function is a lower bound for the SO threshold function.

### 3.2 An algorithm to compute the individually optimal policy

We present a simple and efficient algorithm to compute the individually optimal policy for the model without cancellation, based on Properties P1–P3 for the IO policy. We assume that there is an upper bound on the inventory level, $H$, which is finite but can be arbitrarily large, in order to avoid an infinite state space. The algorithm shows that spanning the state space only once is more than enough to compute the individually optimal policy.

Before we describe the algorithm, let us define some variables. For a fixed state $s = (d, d_{m+1})$, let $I_{max}$ and $I_{min}$ indicate the most downstream and upstream stage in which there is at least one pending order respectively. Let $G(d)$ represent the set of states at which the total number of orders equals to that of $d$, $\sum_{i=1}^{m} d_i$, but the pending orders are in the same or more upstream stages in the demand realization process. That is,

$$
G(d) = \{c : \sum_{i=1}^{m} c_i = \sum_{i=1}^{m} d_i \text{ and } \sum_{i=1}^{l} c_i \geq \sum_{i=1}^{l} d_i \forall l \leq m, l \in \mathbb{Z}^+\}.
$$

Let $\hat{C}(d)$ be the set of states with fewer orders in each stage than $G(d)$, i.e., $\hat{C}(d) = \bigcup_{c \in G(d)} \{c : 0 \leq c \leq \hat{c}\}$. For fixed $d_{m+1}$, we say that the states in $\hat{C}(d)$ are less pressing, because of the following corollary to Theorem 3.1.
Corollary 3.6. If the IO action is wait in state \((d, d_{m+1})\), then it is optimal to wait in \((d', d_{m+1})\) for all \(d' \in \hat{C}(d)\).

The basic idea of the algorithm is to start computing from the most pressing state, i.e., the state in which produce is the most likely to be optimal, according to the properties of Theorem 3.1. This state is \(d_{m+1} = 0, d = (0, 0, \ldots, N)\). Recall that we already know it is optimal to produce when \(d_{m+1} > 0\), so our algorithm only computes the optimal policies for states with \(d_{m+1} \leq 0\). We update the state to less pressing states until the first time that wait is the IO policy. We do the update as follows. For fixed \(d_{m+1}\) and \(\sum d_i\), we move orders backward (upstream) in the demand realization process. If this is not possible (all orders are in the first stage), we decrease the total number of orders by 1 and put all of them back in the last stage of the realization process. If it is not possible to decrease the number of orders by 1 (\(\sum d_i = 0\)), we decrease \(d_{m+1}\) by 1 and set \(d = (0, 0, \ldots, N)\) (until \(-d_{m+1} = H + 1\), when we terminate). We do this updating procedure until the first time wait is the IO policy, in state \((d, d_{m+1}')\), say. Because produce was optimal in state \((d, d_{m+1} + 1)\), we know \(T(d) = -(d_{m+1}' + 1) = -d_{m+1}' - 1\). From the corollary above we also know \(T(d') = -d_{m+1}' - 1\) for all less pressing states \(d' \in \hat{C}(d)\). We now let \(C = \hat{C}(d)\) be the states for which the threshold has been determined; we update the state to \((0, 0, \ldots, N, d_{m+1}' - 1)\); and continue until the first time wait is again optimal, for some state \((d_1, d_{m+1}'\)\), say, where \(d_1 \notin C\). Again, because produce was optimal in state \((d_1, d_{m+1}' + 1)\), we know \(T(d_1) = -(d_{m+1}' + 1) = -d_{m+1}' - 1\), and \(T(d'') = -d_{m+1}'' - 1\) for all less pressing states in which the thresholds have not been determined, \(d'' \in \hat{C}(d_1) \setminus C\). We continue in this manner until we have covered all the states, i.e., until, for some \(d_{m+1}\), wait is the IO when \(d = (0, 0, \ldots, N)\), or \(-d_{m+1}\) has been updated to \(H + 1\). The complete algorithm is in Appendix B.

### 3.3 The IO policy with order cancellation

With order cancellation, the IO policy is again defined by a threshold function. Also, Lemmas 3.2, 3.3, and Theorem 3.5 hold with the same argument because we can still define the sunk, outstanding, priority, and remaining orders, and the cost for each of them in the same way as in the model without cancellations. Therefore, \(T^{IO}(d)\) provides a lower bound for \(T(d)\): \(T^{IO}(d) \leq T(d)\) even in the case where order cancellation is allowed. However, as with the SO policy, we need condition \(\text{A}\) to show that the IO policy preserves all of the monotonicity properties that hold for the no-cancellation model. Because the proofs are similar to other proofs in the paper, we omit them. When orders can be cancelled, the algorithm for computing the IO policy is more complicated than the one in the previous section, but it is still much simpler than computing the socially optimal policy.
4 Numerical results

In our numerical studies we compare three different policies: the state-dependent SO policy, the state-dependent IO policy and the state-independent base-stock policy (No-MADI policy) for the model with no order cancellation. The SO policy is determined by solving the corresponding dynamic program, the IO policy is obtained by the algorithm described in Section 3, and the optimal state-independent base-stock is computed by applying the result in Gershwin [13]. We use three different values for the new order placement rate: $\nu = 0.3, 0.5, 0.8$, and for of all our examples we assume the demand realization rate is linear in the number of pending orders, that is, $\lambda_j(d_j) = \lambda_j d_j$. As a result, the time that it takes each order to move from stage $j$ to stage $j+1$ is exponentially distributed with rate $\lambda_j$, independent of other orders. For our base case, we set $b = 2$, but we also vary $b$ to investigate how the gain from using MADI changes in backorder cost rate, $b$. Without loss of generality, we set $h = 1$ and $\mu = 1$.

In order to compare costs that do not depend on the initial state, we compute the average cost rather than the discounted cost. Note that all of our results for the discounted cost case are preserved for the average cost case (Theorem 8.4.3. in Puterman [17]). To compute the SO policy we use relative value iteration and truncate the state space to a large finite set.

Figure 4 illustrates the effect on the average cost of the number of stages (1 or 2) and the expected demand leadtime in each stage ($EL_j = 1/\lambda_j$) when $\nu = 0.3$ and $\lambda_1 = \lambda_2 = \lambda$. Note that, for each value of $\lambda$, the expected leadtime for the one stage model is $1/\lambda$ and for the two-stage model it is $2/\lambda$. As expected, the expected cost using MADI decreases with more stages. We also note that the advantage of MADI over the standard base-stock policy is largest for moderate values of $EL$ and is smallest when $EL$ takes on extreme values. Considering how the firm may use advance demand information in its production decisions, this result is quite intuitive. When $EL$ is very short, orders are due shortly after their placement (relative to the production rate $\mu$), so there is not enough time for the updated demand information to be reflected in production decisions. When $EL$ is very long, advance demand information is too noisy to have much of an impact on production decisions. This observation is similar to that of Gayon et al. [9].
Figure 4  The Effect of the Number of Stages $(\nu = 0.3)$.  

Figure 5  The Effect of System Load $\nu/\mu$ $(m = 1, \mu = 1)$.  

Figure 5 shows the effect of system load, $\nu$. Here we show the relative cost reduction rather than the absolute costs because as $\nu$ increases, the costs under all policies will increase. The percentage cost reduction,

$$\frac{(\text{cost under No-MADI policy}) - (\text{cost under the SO using MADI})}{(\text{cost under No-MADI policy})} \times 100,$$

ensures a fair comparison of the reduction obtained by using MADI among the systems with different $\nu$’s. We observe that, independent of the system load, the cost advantage of MADI over the base-stock policy first increases and then decreases in $EL$. Also, the expected demand leadtime at which the cost reduction is most significant tends to increase in the system load. This is consistent with our earlier observation that MADI is most effective when $EL$ is not too short nor too long relative to the traffic intensity. For moderate values of $EL$, the cost advantage of MADI tends to decrease as the load in the system increases. In fact, MADI is most effective when the system is lightly loaded (with 20-50% lower cost than the standard base-stock policy). Thus, advance information is more beneficial when there is sufficient capacity to ramp up production based on that information. In a lightly loaded system ($\nu = 0.3$), the cost reduction from MADI exceeds 40% when $EL$ is 1 or 2 while it is less than 10% in a heavily loaded system ($\nu = 0.8$). Also, even when a heavily loaded system benefits the most from MADI (when $EL$ is relatively large), the cost reduction is still less than 15%.

The ratio of the inventory and the backorder cost affects the cost advantage of MADI in a non-linear way. For fixed $EL$, the cost advantage of MADI first increases then decreases in the
ratio. Thus, using MADI is most effective when $b$ is not far from $h$. When the backorder cost rate is too large (small) relative to the inventory holding cost rate, the optimal policy becomes increasingly insensitive to the current state and demand evolution, and produce (wait) is optimal for the majority of states. Consequently, in these cases advance information adds less value.

Our numerical study indicates that the individually optimal (IO) policy captures the benefit of MADI fairly well, and performs significantly better than the No-MADI policy in situations where MADI is most beneficial (Figure 6). In several instances the two-stage IO policy performs better than the one stage SO policy at moderate values of $EL$ when the system is not heavily loaded. However, when $EL$ is too long or too short, i.e., when the benefit of MADI diminishes, the performance of the IO policy also deteriorates. Similarly, the IO policy performs quite poorly in heavily loaded systems, again, where MADI adds little value. In such cases, the No-MADI policy performs better than the IO policy. In summary, the IO policy performs well when the benefit of MADI is significant, and it is much easier to compute. Otherwise, the state-dependent base-stock (no-MADI) policy is a good heuristic.

![Figure 6: IO vs. SO ($\nu = 0.3$).](image)

### 5 Conclusion and Future Research

We have shown that multistage advance demand information (MADI) is beneficial and the socially optimal (SO) policy incorporating MADI is a state-dependent base-stock policy. The base-stock (threshold) level is monotone in the number of orders as well as in the progress of the demand realization process. We also introduced the individually optimal (IO) policy that only considers
the priority order and has the same monotonicity properties as the socially optimal (SO) policy. The IO policy is much easier to compute, and numerical study shows that it performs significantly better than the optimal base-stock policy without MADI in moderately and lightly loaded systems with moderately long expected demand leadtime, which are also the cases when MADI is most beneficial. In heavy traffic, the value of MADI is less significant and the usual base-stock policy without MADI is a good heuristic policy.

A promising direction for future research is extending the model to a multi-echelon supply chain system by including multiple stages in the production process as well as in the demand process. Inventory control for multi-echelon supply chains with no or fixed demand leadtime has been extensively studied for a long time. It would be interesting to see how things will change once the multistage stochastic demand leadtime is introduced into the model. Assembly systems with two production streams merging or joining at some point can also fit in this framework. Considering a more general capacitated production system with MADI will be a challenging but interesting extension. If we can produce multiple units at a time with an upper limit on the number of units and/or a time limit on the next production (e.g., once we produce any item, we have to wait three days before the next production), we will need to determine the lot size as well as the timing of production. We suspect that there will still be a monotonic policy for this model, but it will be a lot more complicated.

Acknowledgment

We would like to thank the referee for a thorough review and many helpful suggestions.

References


Appendix A: Proof of Lemma 2.5

We use induction on a sequence of finite-horizon problems. Let \( V_t(s) \) be the optimal discounted expected cost function when there are \( t \geq 1 \) periods to go and the starting state is \( s \):

\[
V_i(s) = g(d_{m+1}) + \nu V_{i-1}(E_1(s)) + \sum_{i=1}^{m} \lambda_i(d_i)V_{i-1}(R_i(s)) \\
+ (\lambda_{\text{max}} - \sum_{i=1}^{m} \lambda_i(d_i))V_{i-1}(s) + \min_{\mu \in \{0, \mu\}} (\mu V_{t-1}(P(s)) + (\bar{\mu} - \mu)V_{i-1}(s))
\]  

where \( V_0(s) = 0 \) for all \( s \). Since we consider a discounted cost with a finite action space, the cost \( V_t(s) \to V(s) \) as \( t \to \infty \) from Theorem 6.10.2. in Puterman [17]. For \( t = 1 \), \( P5-P8 \) hold trivially, because \( V_0(s) = 0 \). Assume that \( P5-P8 \) hold for \( t = 2, \cdots, n \) for all \( s \in S_N \). We now show they hold for \( t = n + 1 \).

**Proof of P5**

In order to show convexity in \( d_{m+1} \), we need the following technical result.

**Lemma 5.1.** (Lippman, 1975). If \( f(d_{m+1}) \) is convex in \( d_{m+1} \) for all \( d_{m+1} \in \mathbb{Z} \), then \( q(d_{m+1}) = \min \{ f(d_{m+1}), f(d_{m+1} - 1) \} \) is also convex in \( d_{m+1} \) for all \( d_{m+1} \in \mathbb{Z} \).

By Lemma 5.1 and the induction hypothesis, \( \min \{ V_n(s), V_n(P(s)) \} \) is convex in \( d_{m+1} \). Then, in (9) each of \( g(d_{m+1}), V_n(E_1(s)), V_n(R_j(s)), V_n(s) \) and \( \min_{\mu \in \{0, \mu\}} (\mu V_{t-1}(P(s)) + (\bar{\mu} - \mu)V_{i-1}(s)) = \min \{ V_n(s), V_n(P(s)) \} \) is convex in \( d_{m+1} \). Therefore, \( V_{n+1}(s) \) is convex in \( d_{m+1} \) and letting \( n \to \infty \) \( P5 \) holds.

**Proof of P6**

Before we start the proof, we discuss the commutativity of the operators (e.g., \( P \circ E_1(s) = E_1 \circ P(s) \)). All are commutative when \( \sum_{i=1}^{m} d_i \neq 0 \) and \( \sum_{i=1}^{m} d_i < N \). The operator representing the completion of a production, \( P \), is commutative with any other operator in any state. When \( \sum_{i=1}^{m} d_i \neq 0 \) or \( \sum_{i=1}^{m} d_i = N \), some operators do not commute with others, so these cases will be handled separately.

From the induction hypotheses for \( P6 \) and \( P7 \), we have, for \( j \) fixed,

\[
V_n(s) - V_n(P(s)) - V_n(E_j(s)) + V_n(P \circ E_j(s)) \leq 0 \text{ for } s \in S_N, \ j = 1, \cdots, m 
\]  

(HS6)

and

\[
V_n(s) - V_n(P(s)) - V_n(R_j(s)) + V_n(P \circ R_j(s)) \leq 0 \text{ for } s \in S_N, \ j = 1, \cdots, m 
\]  

(HS7)

We will show that

\[
V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(E_j(s)) + V_{n+1}(P \circ E_j(s))
\]  

(10)
is negative for \( s \in S_N, \, j = 1, \cdots, m \).

Consider any state \( s = (d, d_{m+1}) \in S_N \). When \( \sum_{i=1}^{m} d_i = N \), the claim holds trivially because \( E_j(s) = s \), so suppose \( \sum_{i=1}^{m} d_i < N \). Let us substitute for each term in (10) using (9) and consider the terms involving, \( \nu, \lambda_j, \mu \), and \( \lambda_i \), with \( i \) not equal to \( j \), separately. Let \( K = \lambda_{\text{max}} - \sum_{i=1}^{m} \lambda_i(d_i) \) and \( \delta_j(d_j) = \lambda_j(d_j) - \lambda_j(d_j - 1) \) where \( j = 1, \cdots, m \). Note that \( K \) is always positive because \( \lambda_{\text{max}} = \max_{1 \leq i \leq m} \lambda_i(N) \) and \( \delta_j(d_j) \) is always positive because \( \lambda_j(d_j) \) is increasing in \( d_j \). Also, let \( f_t(s) = \min_{\mu \in \{0, \bar{\mu}\}} \left[ \mu V_t(P(s)) + (\bar{\mu} - \mu)V_t(s) \right] \) for \( t = 0, \cdots n \). Then, (10) is

\[
V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(E_j(s)) + V_{n+1}(P \circ E_j(s)) \\
= \nu\{V_n(E_1(s)) - V_n(E_1 \circ P(s)) - V_n(E_1 \circ E_j(s)) + V_n(E_1 \circ P \circ E_j(s))\} \tag{S6-\nu}
\]

\[
+ \sum_{i=1, i \neq j}^{m} \lambda_i(d_i) \{V_n(R_i(s)) - V_n(R_i \circ P(s)) - V_n(R_i \circ E_j(s)) + V_n(R_i \circ P \circ E_j(s))\} \tag{S6-\lambda_i}
\]

\[
+ \lambda_j(d_j) \{V_n(R_j(s)) - V_n(R_j \circ P(s)) - V_n(R_j \circ E_j(s)) + V_n(R_j \circ P \circ E_j(s))\} \tag{S6-\lambda_j}
\]

\[
+ K \{V_n(s) - V_n(P(s)) - V_n(E_j(s)) + V_n(P \circ E_j(s))\} \tag{S6-K}
\]

\[
+ \delta_{j+1}(d_{j+1} + 1) \{V_n(E_j(s)) - V_n(P \circ E_j(s)) - V_n(R_j \circ E_j(s)) + V_n(R_j \circ P \circ E_j(s))\} \tag{S6-\delta}
\]

\[
+ f_n(s) - f_n(P(s)) - f_n(E_j(s)) + f_n(P \circ E_j(s)) \tag{S6-f}
\]

In order to show that (10) is negative, we need the following technical result. This will also be used in the proof of P7. Let us define a general operator \( \Gamma \in \{R_j, E_j\} \) for \( j = 1, \cdots m \).

**Lemma 5.2.** Suppose \( V \) is \((P, s)\)-multimodular, that is,

\[
V(s) - V(P(s)) - V(\Gamma(s)) + V(P \circ \Gamma(s)) \leq 0 \text{ for } s, P(s), \Gamma(s), P \circ \Gamma(s) \in S_N. \tag{11}
\]

Let

\[
f(s) = \min_{\mu \in \{0, \bar{\mu}\}} [\mu V(P(s)) + (\bar{\mu} - \mu)V(s)]. \tag{12}
\]

Then \( f \) is also \((P, s)\)-multimodular, that is (11) also holds when \( V \) is replaced by \( f \).

**Proof.** Let \( \Delta = f(s) - f(P(s)) - f(\Gamma(s)) + f(P \circ \Gamma(s)) \). We need to show that \( \Delta \leq 0 \). Let \( \mu^P \) and \( \mu^\Gamma \) achieve the minimum in (12) with \( s \) replaced by \( P(s) \) and \( \Gamma(s) \), respectively. Using \( \mu^\Gamma \) in
state \( s \) and \( \mu^P \) in state \( P(\Gamma(s)) \), we can get an upper bound on \( \Delta \) as follows.

\[
f(s) \leq \mu^TV(P(s)) + (\bar{\mu} - \mu^V)V(s)
\]

\[
f(P(s)) = \mu^P V(P \circ P(s)) + (\bar{\mu} - \mu^P)V(P(s))
\]

\[
f(\Gamma(s)) = \mu^V (P \circ \Gamma(s)) + (\bar{\mu} - \mu^V)V(\Gamma(s))
\]

\[
f(P \circ \Gamma(s)) \leq \mu^P V(P \circ P \circ \Gamma(s)) + (\bar{\mu} - \mu^P)V(P \circ \Gamma(s))
\]

So, \( \Delta \leq (\bar{\mu} - \mu^V)\{V(s) - V(P(s)) - V(\Gamma(s)) + V(P \circ \Gamma(s))\} + \mu^P\{V(P(s)) - V(P \circ P(s)) - V(P \circ \Gamma(s)) + V(P \circ P \circ \Gamma(s))\} \), and each term in the braces is negative from (11), so \( \Delta \leq 0 \). \( \square \)

Letting \( \Gamma = E_j \), Lemma 5.2 guarantees that (S6-f) is negative.

When \( \sum_{i=1}^m d_i < N - 1 \), \( E_1 \circ E_j(s) = E_j \circ E_1(s) \). Therefore, using the commutativity of \( P \), (S6-\( \nu \)) can be rewritten as

\[
\nu\{V_n(E_1(s)) - V_n(P \circ E_1(s)) - V_n(E_j \circ E_1(s)) + V_n(P \circ E_j \circ E_1(s))\}.
\]

Replacing \( s \) with \( E_1(s) \) in the induction hypothesis of P6 (HS6), this is obviously negative. When \( \sum_{i=1}^m d_i = N - 1 \), \( E_1 \circ E_j(s) = E_j(s) \), (S6-\( \nu \)) can be rewritten as

\[
\nu\{V_n(E_1(s)) - V_n(P \circ E_1(s)) - V_n(E_j(s)) + V_n(P \circ E_j(s))\}.
\]

This is also negative because

\[
V_n(E_1(s)) - V_n(P \circ E_1(s)) - V_n(E_j(s)) + V_n(P \circ E_j(s))
\]

\[
= V_n(E_1(s)) - V_n(P \circ E_1(s)) - V_n(R_{j-1} \circ R_{j-2} \ldots R_1 \circ E_1(s)) + V_n(P \circ R_{j-1} \circ R_{j-2} \ldots R_1 \circ E_1(s))
\]

and this is negative from repeated application of the induction hypothesis of P7 (HS7).

Using commutativity of \( P \), (S6-\( \nu \)), and the fact that \( R_i \circ E_j(s) = E_j \circ R_i(s) \) when \( \sum_{i=1}^m d_i < N \), (S6-\( \lambda_i \)) becomes

\[
\sum_{i=1, i \neq j}^m \lambda_i(d_i)\{V_n(R_i(s)) - V_n(P \circ R_i(s)) - V_n(E_j \circ R_i(s)) + V_n(P \circ E_j \circ R_i(s))\}.
\]

Replacing \( s \) with \( R_i(s) \) in the induction hypothesis (HS6), this is negative.

Now consider the terms involving \( \lambda_j \), (S6-\( \lambda_j \))-(S6-\( \delta \)). When \( d_j = 0 \), (S6-\( \lambda_j \)) becomes 0. When \( d_j \geq 1 \) (also remember that \( \sum_{i=1}^m d_i < N \), \( R_j \) and \( E_j \) commute with each other, therefore, (S6-\( \lambda_j \)) can be rewritten as

\[
\lambda_j(d_j)\{V_n(R_j(s)) - V_n(P \circ R_j(s)) - V_n(E_j \circ R_j(s)) + V_n(P \circ E_j \circ R_j(s))\}.
\]
Replacing $s$ with $R_j(s)$ in the induction hypothesis (HS6), this is negative. (S6-K) is negative also from (HS6) and the fact that $K$ is always positive.

Using the commutativity of $P$, (S6-$\delta$) can be rewritten as

$$\delta_{j+1}(d_{j+1} + 1)\{V_n(E_j(s)) - V_n(P \circ E_j(s)) - V_n(R_j \circ E_j(s)) + V_n(P \circ R_j \circ E_j(s))\}.$$ 

Replacing $s$ with $E_j(s)$ in the induction hypothesis of $P7$ this is negative because $\delta_j(d_j)$ is positive for all $j = 1, \ldots, m$.

Since each of (S6-$\nu$)-(S6-$f$) is negative, (10) is negative. Letting $n \to \infty$, we have shown that $V(s) - V(P(s)) - V(E_j(s)) + V(P \circ E_j(s)) \leq 0$ for $s \in S_N$, $j = 1, \ldots, m$. \hfill \Box

**Proof of P7**

Using a similar argument as in the proof of $P6$, we will show that

$$V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(R_j(s)) + V_{n+1}(P \circ R_j(s))$$

is negative for $s \in S_N$, $j = 1, \ldots, m$. Note that (HS6) and (HS7) will be used again as our induction hypotheses.

Consider any state $s = (d, d_{m+1}) \in S_N$. When $d_j = 0$, the claim holds trivially because $R_j(s) = s$, so suppose $d_j \geq 1$.

Let us substitute for each term in (13) using (9) and consider the terms involving $\nu$, $\lambda_j$, $\lambda_{j+1}$, $\mu$, and $\lambda_i$ with $i$ is not equal to $j$ or $j + 1$ separately. Then, (13) is

$$V_{n+1}(s) - V_{n+1}(P(s)) - V_{n+1}(R_j(s)) + V_{n+1}(P \circ R_j(s))$$

$$= \nu\{V_n(E_1(s)) - V_n(E_1 \circ P(s)) - V_n(E_1 \circ R_j(s)) + V_n(E_1 \circ P \circ R_j(s))\}$$

$$(S7-\nu)$$

$$+ \sum_{i=1, i \neq j, j+1}^{m} \lambda_i(d_i)\{V_n(R_i(s)) - V_n(R_i \circ P(s)) - V_n(R_i \circ R_j(s)) + V_n(R_i \circ P \circ R_j(s))\}$$

$$(S7-\lambda_i)$$

$$+ \lambda_j(d_j - 1)\{V_n(R_j(s)) - V_n(R_j \circ P(s)) - V_n(R_j \circ R_j(s)) + V_n(R_j \circ P \circ R_j(s))\}$$

$$(S7-\lambda_j)$$

$$+ \lambda_{j+1}(d_{j+1})\{V_n(R_{j+1}(s)) - V_n(R_{j+1} \circ P(s)) - V_n(R_{j+1} \circ R_j(s)) + V_n(R_{j+1} \circ P \circ R_j(s))\}$$

$$(S7-\lambda_{j+1})$$

$$+ K\{V_n(s) - V_n(P(s)) - V_n(R_j(s)) + V_n(P \circ R_j(s))\}$$

$$(S7-K)$$

$$+ \delta_{j+1}(d_{j+1} + 1)\{V_n(R_j(s)) - V_n(P \circ R_j(s)) - V_n(R_{j+1} \circ R_j(s)) + V_n(R_{j+1} \circ P \circ R_j(s))\}$$

$$(S7-\delta_{j+1})$$

$$+ f_n(s) - f_n(P(s)) - f_n(R_j(s)) + f_n(P \circ R_j(s))$$

$$(S7-f)$$

28
From Lemma 5.2, letting $\Gamma = R_j$, (S7-f) is negative.

We separate the rest of the proof into two parts; when $j \neq m$ and $j = m$ because $R_m$ operates in a somewhat different way from the other $R_j$’s when $\sum_{i=1}^{m} d_i = N$. For example, when $\sum_{i=1}^{m} d_i = N$, $E_1 \circ R_j(s) = R_j \circ E_1(s) = R_j(s)$ if $j \neq m$, but $E_1 \circ R_m(s) \neq R_m \circ E_1(s) = R_m(s)$. We first consider the case when $j \neq m$.

When $\sum_{i=1}^{m} d_i < N - 1$, (S7-\nu) can be rewritten as

$$\nu\{V_n(E_1(s)) - V_n(P \circ E_1(s)) - V_n(R_j \circ E_1(s)) + V_n(P \circ R_j \circ E_1(s))\},$$

because the operators commute with each other. Replacing $s$ with $E_1(s)$ in the induction hypothesis of $P_7$(HS7), we have that (S7-\nu) is negative. When $\sum_{i=1}^{m} d_i = N$, $E_1(s) = s$ and $E_1 \circ R_j(s) = R_j(s)$ so (S7-\nu) can be rewritten as $\nu\{V_n(s) - V_n(P(s)) - V_n(R_j(s)) + V_n(P \circ R_j(s))\}$ and this is again negative from the same induction hypothesis.

Consider (S7-\lambda_j). When $d_i = 0$, $\lambda_i(d_i) = 0$, so consider only the terms in which $d_i \geq 1$. When $d_i \geq 1$ (also remember that we assume $d_j \geq 1$), $R_i$ and $R_j$ commute each other. Therefore, (S7-\lambda_i) can be rewritten as

$$\sum_{i=1, i \neq j, j+1, d_i \neq 0}^{m} \lambda_i(d_i)\{V_n(R_i(s)) - V_n(P \circ R_i(s)) - V_n(R_j \circ R_i(s)) + V_n(P \circ R_j \circ R_i(s))\}$$

Replacing $s$ with $R_i(s)$ in (HS7), this is negative.

Using the commutativity of $P$ operator, (S7-\lambda_j) can be rewritten as

$$\lambda_j(d_j - 1)\{V_n(R_j(s)) - V_n(P \circ R_j(s)) - V_n(R_j \circ R_j(s)) + V_n(P \circ R_j \circ R_j(s))\}$$

and replacing $s$ with $R_j(s)$ in (HS7), this is negative.

When $d_{j+1} = 0$, (S7-\lambda_{j+1}) becomes 0. When $d_{j+1} \geq 1$ (also remember that we assume $d_j \geq 1$) $R_j$ and $R_{j+1}$ are commutative, therefore, (S7-\lambda_{j+1}) can be rewritten as

$$\lambda_{j+1}(d_{j+1})\{V_n(R_{j+1}(s)) - V_n(P \circ R_{j+1}(s)) - V_n(R_j \circ R_{j+1}(s)) + V_n(P \circ R_j \circ R_{j+1}(s))\}$$

Replacing $s$ with $R_{j+1}(s)$ in (HS7), this is negative. (S7-K) is negative from (HS7) because $K$ is always positive. (S7-\delta_{j+1}) can be rewritten as

$$\delta_{j+1}(d_{j+1} + 1)\{V_n(R_j(s)) - V_n(P \circ R_j(s)) - V_n(R_{j+1} \circ R_j(s)) + V_n(P \circ R_{j+1} \circ R_j(s))\}$$

using the commutativity of the $P$ operator. Replacing $s$ with $R_j(s)$ in (HS7), since $\delta_{j+1}(d_{j+1} + 1)$ is positive, this is also negative.
So far, we have shown that (S7-$\nu$)-(S7-$\delta_{j+1}$) are negative assuming $j < m$. Now suppose $j = m$. Again, we assume that $d_m \geq 1$ since (13) is 0 when $d_m = 0$. When $\sum_{i=1}^{m} d_i < N$, the argument is the same as in the case of $j < m$. Therefore, it remains to show that (13) is negative when $\sum_{i=1}^{m} d_i = N$ and $j = m$.

(S7-$\nu$) is negative because

$$
\nu\{V_n(E_1(s)) - V_n(E_1 \circ P(s)) - V_n(E_1 \circ R_m(s)) + V_n(E_1 \circ P \circ R_m(s))\}
$$

$$
= \nu\{V_n(s) - V_n(P(s)) - V_n(E_1 \circ R_m(s)) + V_n(P \circ E_1 \circ R_m(s))\}
$$

$$
\leq \nu\{V_n(s) - V_n(P(s)) - V_n(R_m(s)) + V_n(P \circ R_m(s))\}
$$

where the inequality holds from the induction hypothesis of P7.

In (S7-$\lambda_i$), when $d_i = 0$, $\lambda_i(d_i) = 0$, so consider only the terms in which $d_i \geq 1$. When $d_i \geq 1$ (also remember that we assume $d_m \geq 1$), $R_i$ and $R_m$ commute each other. Therefore, (S7-$\lambda_i$) can be rewritten as

$$
\sum_{i=1, d_i \neq 0}^{m-1} \lambda_i(d_i)\{V_n(R_i(s)) - V_n(P \circ R_i(s)) - V_n(R_m \circ R_i(s)) + V_n(P \circ R_m \circ R_i(s))\}
$$

Replacing $s$ with $R_i(s)$ in (HS7), terms in the brace are negative, therefore, this is negative.

S7-$\lambda_j$ can be then rewritten as

$$
\lambda_m(d_m - 1)\{V_n(R_m(s)) - V_n(P \circ R_m(s)) - V_n(R_m \circ R_m(s)) + V_n(P \circ R_m \circ R_m(s))\}.
$$

Replacing $s$ with $R_m(s)$ in (HS7), this is negative. (S7-$K$) is also negative from (HS7) because $K$ is always positive. (S7-$\lambda_{j+1}$) and (S7-$\delta_{j+1}$) disappear when $j = m$. Therefore, (13) is negative when $j = m$.

Since each term, involving $\nu$, $\lambda_j$, $\lambda_{j+1}$, $\mu$, and $\lambda_i$ with $i$ is not equal to $j$ or $j + 1$ in (13) is negative, and letting $n \to \infty$, we have shown that $V(s) - V(P(s)) - V(R_i(s)) + V(P \circ R_i(s)) \leq 0$ for $s \in S_N, i = 1, \ldots, m$.

**Proof of P8** The argument is similar to the proof of P6 and P7.
Appendix B: The IO Algorithm

BEGIN IO ALGORITHM
For all $d \in D_N$, set $\Psi(d, 1) := 0$; $M^{IO}(d, 1) := \frac{b}{\mu + \alpha}$; $C := \emptyset$; $x := 0$; $\Rightarrow x = -d_{m+1}$
While $x \leq H$ {
    Set $k := N$;
    While $k \geq 1$ {
        Set $d := (0, \ldots, 0, k)$; $s := (d, -x)$; $I_{\text{min}} := m$; $I_{\text{max}} := m$, $\pi := P$;
        \\ Start with all $k$ orders at the last stage before realization
        While $I_{\text{max}} > 1$ {
            Call COMPUTE $M^{IO}_p, M^{IO}_w(s)$; \\ subroutine below
            If $M^{IO}_p(s) < M^{IO}_w(s)$ {
                Call UPDATE $(d, I_{\text{max}}, I_{\text{min}})$; \\ subroutine below
            } Else {
                $\pi := W$;
                $G(d) := \{c : \sum_{i=1}^{m} c_i = \sum_{i=1}^{m} d_i$ and $\sum_{i=1}^{l} c_i \geq \sum_{i=1}^{l} d_i \forall l \leq m, l \in \mathbb{Z}^+\}$;
                $\hat{C}(d) := \cup_{c \in G(d)} \{c : 0 \leq c \leq \bar{c}\}$; $T(d') := x - 1$ for all $d' \in \hat{C}(d) \setminus C$;
                If $k := N$ and $I_{\text{min}} := m$ {
                    Stop;
                } Else {
                    $C := \hat{C}(d)$; $I_{\text{max}} := 1$;
                }
            } Else {
                $C := \hat{C}(d)$; $I_{\text{max}} := 1$;
            }
            end while $I_{\text{max}} > 1$, so $I_{\text{max}} := 1$
        } End while $I_{\text{max}} > 1$, so $I_{\text{max}} := 1$
        If $M^{IO}_p(s) \geq M^{IO}_w(s)$ {
            $\hat{C}(d) := \{c : c \leq d\}$; $T(d') := x - 1$ for all $d' \in \hat{C}(d) \setminus C$;
            $C := \hat{C}(d)$;
            $k := k - 1$;
        } End while $k \geq 1$, so $k := 0$
        $x := x + 1$;
    } End while $x \leq H$
    If $\pi := P$ {
        $T(d) := H$ for all $d \in D_N$;
    }
END
COMPUTE $M^p_w, M^o_w(s)$  
If $k = N$ and $I_{\text{min}} = m$ (i.e., $d = (0, \ldots, 0, N)$)  
\begin{align*}
\Psi(s) &= \frac{1}{\alpha + \sum_{i=1}^{m} \lambda_i(d_i)} \left[ 1 + \sum_{i=1}^{m} \lambda_i(d_i) \Psi_n(R_i(s)) \right] \\
M^p_w(s) &= \frac{1}{\alpha + \sum_{i=1}^{m} \lambda_i(d_i) + \mu} \left[ \sum_{i=1}^{m} \lambda_i(d_i) M^o(R_i(s)) + \mu h(s) \right] \\
M^o_w(s) &= \frac{1}{\alpha + \sum_{i=1}^{m} \lambda_i(d_i)} \left[ \sum_{i=1}^{m} \lambda_i(d_i) M^o(R_i(s)) \right] \\
\end{align*}
}Else  
\begin{align*}
\Psi(s) &= \frac{1}{\alpha + \nu + \sum_{i=1}^{m} \lambda_i(d_i)} \left[ 1 + \nu \Psi_n(A(s)) + \sum_{i=1}^{m} \lambda_i(d_i) \Psi_n(R_i(s)) \right] \\
M^p_w(s) &= \frac{1}{\alpha + \nu + \sum_{i=1}^{m} \lambda_i(d_i) + \mu} \left[ \nu M^o(A(s)) + \sum_{i=1}^{m} \lambda_i(d_i) M^o(R_i(s)) + \mu h(s) \right] \\
M^o_w(s) &= \frac{1}{\alpha + \nu + \sum_{i=1}^{m} \lambda_i(d_i)} \left[ \nu M^o(A(s)) + \sum_{i=1}^{m} \lambda_i(d_i) M^o(R_i(s)) \right] \\
\end{align*}
};

UPDATE $(d, I_{\text{max}}, I_{\text{min}})$  
If $I_{\text{min}} > 1$  
\begin{align*}
d &:= d - e_{I_{\text{min}}(d)} + e_{I_{\text{min}}(d) - 1}; \quad \text{move an order backward (upstream)} \\
\end{align*}
}Else  
\begin{align*}
y &:= d_1; \quad I := \arg \min_{2 \leq i \leq m} \{ i : c_i > 0 \}; \\
d &:= d - ye_1 - e_I + (y + 1)e_{I-1}; \quad \text{there is at least one order in stages > 1} \\
\end{align*}
\begin{align*}
I_{\text{min}} &:= \arg \min_{1 \leq i \leq m} \{ i : d_i > 0 \}; \quad I_{\text{max}} := \arg \max_{1 \leq i \leq m} \{ i : d_i > 0 \}; \\
\end{align*}
};