Production Campaign Planning Under Learning and Decay

Abstract

Problem definition. We analyze a catalyst-activated batch-production process with uncertainty in production times, learning about catalyst-productivity characteristics, and decay of catalyst performance across batches. The challenge is to determine the quality level of batches and to decide when to replenish a catalyst so as to minimize average costs, consisting of inventory holding, backlogging, and catalyst switching costs.

Academic / Practical Relevance. This is an important problem in a variety of process industry sectors such as food processing, pharmaceuticals, and specialty chemicals but has not been adequately studied in the academic literature. Our paper also contributes to the stochastic economic lot-sizing literature.

Methodology. We formulate this problem as a Semi-Markov Decision Process (SMDP), and develop a two-level heuristic for it. This heuristic consists of a lower-level problem which determines the quality of batches to meet an average target quality, and a higher-level problem which determines when to replace the costly catalyst as its productivity decays. To evaluate our heuristic, we present a lower bound on the optimal value of the SMDP. This bound accounts for all costs, as well as the randomness and discreteness in the process. We then extend our methods to multiple-product settings: an advanced stochastic economic lot-sizing problem.

Results. We test our proposed solution methodology with data from a leading food processing company and show our methods outperform current practice with average improvements of around 22% in costs. In addition, compared to the stochastic lower bounds, our results show the simple two-level heuristic attains near-optimal performance for the intractable multi-dimensional SMDP.

Managerial Implications. Our results imply three important managerial insights: first, our simulation-based lower bound provides a close approximation to the optimal cost of the SMDP and it is nearly attainable using a relatively simple two-level heuristic. Second, the re-optimization policy used in the lower-level problem adequately captures the value of information and Bayesian learning. Third, in the higher level problem of choosing when to replace a catalyst, the intractable multidimensional state of the system is efficiently summarized by a single statistic: the probability of inventory falling below a specific threshold.

Key Words and Phrases: Campaign planning, Batch production, Stochastic economic lot sizing problem, Bayesian updating, Production learning, Semi-Markov processes, Stochastic dynamic programming
1 Introduction

Several batch production processes in the manufacturing of specialty chemical, food processing and pharmaceutical industries use catalysts to control the characteristics of production. The product is processed in batches of fixed volume in a reactor (or machine), which refines the product using a catalyst and achieves a target attribute level. The effectiveness of the catalyst decays with usage. Every catalyst has an optimal life span; it may be pushed too far or prematurely replaced if a proper strategy is not employed. The producer must choose efficient policies to minimize the expected costs, including inventory-related costs and the price of the catalysts. This is an important problem found in several settings in the process industry sector (Casas-Liza et al. 2005, Liu et al. 2014), but has been modeled as a single-stage deterministic problem, while the problem is dynamic in nature. By considering the dynamic problem, this paper addresses the limitations in the literature and more accurately represents the actual decision process.

The sequence of batches that are exposed to the same catalyst at different stages of decay are referred to as a “campaign” of batches. Batches within a campaign can be mixed to meet the average attribute level target. Mixing batches allows for the optimal exploitation of the potential of a catalyst as it offers more flexibility in determining the duration of batches. The productivity of the catalyst decays as it is used across more batches, until at some point the producer has to switch to a new catalyst and incur the associated switch-over costs. The initial productivity of a new catalyst is a random value drawn from a known distribution, estimated from historical data. We cannot observe the exact productivity of the current catalyst due to random shocks experienced by each batch. The noisy observations on each batch allow us to update our information about the productivity of the catalyst through Bayesian updating. Our goal is to plan the duration of the batches and decide when to replenish a catalyst so as to minimize the expected average cost which includes inventory holding, backlogging, and catalyst-switching costs. We first explore the single product case and then extend the results to multiple products. With multiple products, in addition to batch planning and catalyst replenishment decisions, we must also decide which product to produce next.

We formulate the single-product batch-production planning problem with learning and decay as a semi-Markov, average-cost model. We decompose it into two levels to make it tractable.
The lower-level problem plans the duration of batches within the current campaign to maximize the efficiency of the catalyst, while satisfying the target average attribute level. The higher-level problem is a binary decision after each batch: whether or not to end this campaign and switch to a new catalyst. The lower level is formulated as a stochastic dynamic programming problem, similar to the Bayesian decision model by Mazzola and McCardle (1996), which had a production learning curve (as opposed to our decay curve). Despite the similarities of the models, our problem has an additional constraint (the average attribute level constraint), which adds one more dimension to the state space. Therefore we adopt a re-optimization policy that relies on learning to take care of itself and show that it has near-optimal performance. The higher-level problem is to design a control policy mapping the state space to a binary control variable- changing the catalyst or continuing. The objective is to minimize the average costs (switchover, backlogging and inventory). The state space consists of the current inventory level, current consumption of the catalyst (measured by the total time the current catalyst has actively been used to produce batches), and the current belief regarding the catalyst productivity parameter. Solving for an optimal mapping from the state space to the decision variable is intractable. We propose a heuristic policy to approximately solve this problem.

To evaluate the performance of this two-level heuristic, we obtain a lower bound on the optimal performance of the original integrated decision process. This bound simultaneously accounts for costs, randomness, and the discrete nature of the process. We also compare the performance of our heuristic with the fixed cycle policy that is currently practiced by the leading food-processing company that provided the motivation of this work.

We then extend our model to consider the multi-product case with uncertainty in production times. This can be regarded as a Stochastic Economic Lot Scheduling Problem (SELSP) with fixed batch sizes, learning and decay. Due to the complexity of this problem, we solve this using a dynamic model-based heuristic. Vaughan (2007) compares dynamic vs. cyclical policies for SELSP problems and shows that in many cases cyclical policies perform better than dynamic policies. However, in our context cyclical policies suffer from delayed reaction towards backlogged demand; a dynamic policy is needed to recognize and attend to the critical product that would otherwise cause large backlogging costs. The model-based heuristic is benchmarked with a practitioners heuristic used by a large food processing company. We also develop a lower bound on this problem.
to evaluate the performance of these heuristics.

The literature related to our work can be classified by problem type: single product and multi-product. The single product problem is related to the single item stochastic economic lot sizing problem. Levi and Shi (2013) review a wide selection of the single-item SELSP literature, many of which show that \((s, S)\) policies are optimal under specific problem configurations. However, in our problem \((s, S)\) policies are not feasible because depending upon the current catalyst productivity and the decaying nature of the catalyst, it may not be possible to produce up to \(S\). Another important feature of our problem is that we learn about the current productivity of the catalyst by observing the performance in previous batches. This learning is used to predict catalyst performance and production times of future batches. Levi and Shi (2013) consider a related stochastic lot sizing problem and include the prediction of future demand (got from observations in previous periods) in the current decision process. They propose a randomized cost-balancing policy to decide when and how much to order. However, their problem considers uncapacitated orders and discrete periods, while we consider production that is constrained by the limited productivity of a catalyst, and our decision epochs are randomly determined by batch completion times. The relevant literature for multi-product models include work on the SELSP (Winands et al. 2011) and economic lot sizing models under production decay (Casas-Liza et al. 2005, Liu et al. 2014). To the best of our knowledge, ours is the first paper to consider all these settings together. In terms of solution methods for the SELSP, work that uses dynamic approaches to solve the SELSP includes Rajaram and Karmarkar (2002), Dusonchet and Hongler (2003), and Wang et al. (2012). In terms of application, Rajaram and Karmarkar (2004) also consider a campaign planning problem applied to the food-processing industry. However, none of the methods used in these papers can be used to solve the problem considered in this paper due to batching, learning across batches and decay in performance of the catalyst.

In this context, our paper makes the following contributions. First, to the best of our knowledge, this is the first paper to consider the campaign planning problem under production time uncertainty, learning and decay. As previously discussed, this is an important problem that has not been adequately studied in the academic literature. Second, we formulate this problem as a semi-Markov decision process, incorporating key aspects of this problem which include uncertainty in production time, learning about productivity characteristics and decay in catalyst performance. Third, we
develop efficient, near optimal solution methods to solve this problem. In addition, our approach to find lower bounds by associating a continuous state space dynamic programming problem with a similar but regenerative process can be applied to other stochastic dynamic programming problems. Fourth, we validate our model and solution methods with real data from the process industry. Fifth, we provide several insights which could be useful for practitioners in other industries which have a similar production setting.

The remainder of this paper is organized as follows. We begin by formulating the single product problem as a Semi-Markov Decision Process (SMDP) in §2, followed by lower bounds on the optimal cost of the SMDP in §3. In §4 we present the solution methodology based on a two-level heuristic which involves decomposing the problem into two level of decision making. To benchmark the two-level heuristic, we also provide a practitioner’s heuristic that is currently employed at a large food processing company. We extend our methodology to the multiple product case in §5. In §6 we compare the performance of the heuristics and lower bound on real factory data and present managerial insights. Conclusions and future research directions are provided in §7.

2 Model Formulation

Consider a batch production process in which the present state of the production system is defined by the current inventory level and the state of the catalyst currently in use (to be explained in further detail). Based on this information, the firm must decide whether to replace the current catalyst and start a new campaign, or to produce another batch with the current catalyst and determine the attribute level of the next batch. We denote the current inventory level as $I$ and treat it as a continuous variable. Further, for ease of exposition and without loss of generality we measure inventory as batches and assume that the batch size is equal to 1. That is, inventory replenishes by discrete counts but depletes continuously with a constant per-unit-time demand. Next, we define the following parameters and variables:

Parameters:

$d$: constant demand rate (batches/unit-time).

$C_B$: backlogging cost ($/batch/unit-time)$.

$C_I$: inventory holding cost ($/batch/unit-time)$.
\( C_S \): cost of changing a catalyst ($).
\( t_s \): switch-over time required to change a catalyst (unit-time).

Stochastic variables:

\( b \): inverse productivity parameter of a catalyst

\( z_i \): random shock observed by the catalyst while producing batch \( i \).
\( z = [z_1, z_2, ..., z_N] \): combining the \( z_i \)'s of a campaign into one vector.

Decision variables:

\( N \): number of batches produced by a catalyst, i.e. the length of a campaign.
\( q_i \): The attribute level of the \( i \)'th batch in a campaign (normalized, dimensionless).

The decision variables imply the following intermediary variables:

\( q = [q_1, q_2, ..., q_N] \): vector of attribute levels of all batches produced by a catalyst.
\( Q_i = \sum_{j=1}^{i-1} q_j \): sum of all attribute levels up to the beginning of batch \( i \).
\( t_i \): time spent by batch \( i \) in the reactor (unit-time/batch).
\( T_i = \sum_{j=1}^{i-1} t_j \): total consumption of the catalyst up to batch \( i \) (unit-time).

We define the following functions:

\( \gamma(b) \): a density function representing the current belief distribution over possible values of the parameter \( b \). The prior belief is \( \gamma_0(b) \).

\( \tau(q, b, z) \): time spent on a catalyst with production schedule \( q \) and inverse productivity \( b \). Takes random values based on the realizations of the random shocks \( z_i \). (unit-time)

\( \tau^*(N) \): expected optimal time it would take to produce \( N \) batches in a campaign, given full information of \( b \) and \( z \). (unit-time)

\( g(I, \tau) \): total inventory holding and backlogging cost during time length \( \tau \), where the inventory level starts at \( I \) and ends at \( I - \tau d \). (unit-cost).

A batch that enters the reactor begins at an initial attribute level. This attribute level declines as the batch is refined. The time it takes to process batch \( i \) is denoted by \( t_i \) and is determined by the consumption of the catalyst \( T_i \), inverse productivity of the catalyst and the random shock observed \( (b + z_i) \), and the target attribute level \( q_i \). The three elements have separable effect on the
duration of the batch, thus $t_i$ is related to these elements via:

$$t_i = k(T_i)(b + z_i)f(q_i),$$  \hfill (1)

where $k$ is a monotone increasing function that defines the dependancy of the processing time of the $i$th batch on the total consumption of the catalyst up to the starting time of this batch. The inverse productivity parameter $b$ comes from a known distribution with mean $\mu_b$ and variance $\sigma_b^2$, and it only takes positive values. It varies by catalyst but is fixed across batches produced by a particular catalyst. The shocks $z_i$ are iid variables with mean zero and a known distribution. We assume $b + z_i$ is always positive. Finally, $f(q_i)$ is a convex decreasing function to capture the increases in time it takes to purify batch $i$ and reduce attribute $q_{i} \geq 0$ from its initial starting level. This function is convex because of the process characteristic: the lower the attribute level $q_i$, the longer it takes for further reductions from this level.

Equivalently, to show the attribute level of a batch as a function of the time it spends in the machine, we have:

$$q_i = f^{-1}\left(\frac{t_i}{k(T_i)(b + z_i)}\right).$$ \hfill (2)

For example, if $k(T) = T + 1$ and $f(q) = -\ln(q)$, the relation between $q$ and $t$ would take the form:

$$q_i = \exp\left(-\frac{t_i}{(T_i + 1)(b + z_i)}\right).$$ \hfill (3)

The decision to be made prior to placing the next batch inside the reactor is to either choose a target attribute level $q_{n+1}$ for the next batch, or to end the current campaign and replace the catalyst. We do not allow the option of removing a batch before meeting the target attribute level.

The belief distribution $\gamma(b)$ is updated by observing the pair $(t_i, q_i)|T_i$ after each batch. At the end of the campaign and after replacing the catalyst, the new inventory level will be:

$$I' = I - (\tau(q, b, z) + t_s)d + N. \hfill (4)$$

Due to the randomness of $b$ and $z_i$, the new state ($I'$) will be a random function of the old state ($I$) and the vector of actions $q_i$ summarized in $q$. The campaign time ($\tau(q, b, z)$) is also a random function of $q$. The cost of this transition is the inventory and backlogging cost during the
time \( \tau(q, b, z) + t_s \) plus the cost of changing the catalyst. Due to the pooling strategy, in which
the batches in a campaign are mixed by the producer, the batches produced by a catalyst are not
available in the sales inventory until the catalyst is changed. We make two assumptions regarding
the inventory and backlogging costs:

(i) Backlogging costs occur if the total demand during the catalyst lifespan (i.e. \( (\tau(q, b, z) + t_s)d \))
exceeds the initial inventory level \( I \).

(ii) The batches that have not been pooled and prepared for sale do not induce inventory costs.

Let the function \( g(I, \tau) \) represent the total inventory and backlogging costs during a time span
of \( \tau \) when the starting inventory is \( I \) and no batches are added to the inventory during \( \tau \). To derive
\( g(I, \tau) \), we need to consider three cases. First, if we do not run out of inventory during time \( \tau \),
we incur only inventory holding costs which are proportional to the average inventory \( (I - \tau d/2) \)
multiplied by the duration of the time horizon \( (\tau) \). Second, if we start from a positive inventory
level but run out during time \( \tau \), we incur both inventory and backlogging costs. The length of time
with positive inventory is \( I/d \) and the average inventory level during this time is \( I/2 \). The final
inventory is \( I - \tau d \), hence the length of time with backlogging is \( (I - \tau d)/d \) and average backlogging
during this time is \( (I - \tau d)/2 \). Finally, if we start from a negative inventory, the only cost during
\( \tau \) will come from backlogging and is computed similarly to the case where we only incur inventory
holding cost. Thus \( g(I, \tau) \) is computed as:

\[
g(I, \tau) \triangleq \begin{cases} 
\tau(I - \tau d/2)C_I & \text{if } I - \tau d \geq 0 \\
I^2/2d \times C_I - C_B(I - \tau d)^2/2d & \text{if } I \geq 0 \& I - \tau d < 0 \\
-\tau(I - \tau d/2)C_B & \text{if } I < 0.
\end{cases}
\]  

(5)

In order to define the objective function, we first define the term “cycle”. A cycle refers to the
length of time between the ending of two subsequent campaigns. A cycle of duration \( \tau_j \) consists of
a campaign with length \( \tau_j^c = \tau(q_j, b_j, z_j) + t_s \) and an idle time \( \tau_j^0 \) before setting up the campaign.
During a cycle no batches are added to the inventory, hence the realized cost during cycle \( j \) is
\( g(I_j, \tau_j) \), where the cycle starts at inventory \( I_j \). This problem can be formulated as a semi-Markov
average-cost problem with transition cost \( g(I_j, \tau_j) \). The cost function for the average-cost problem
is:
\[
\lim_{R \to \infty} \frac{1}{\sum_{j=1}^{R} \tau_j} E\{\sum_{j=1}^{R} [C_S + g(I_j, \tau_j)]\}.
\] (6)

We formulate the problem of minimizing (6) as a Bayesian stochastic dynamic program. A control policy maps the state space to a decision of either choosing a target attribute level \(q_{n+1}\) for the next batch, or ending the current campaign. The state space consists of the current inventory level \((I)\), number of batches produced so far in the current campaign \((n)\), current belief distribution \(\gamma(b)\), the total consumption of the current catalyst \((T)\), and the cumulative attribute level of the \(n\) batches produced in the current campaign \((Q)\). For simplicity and conformance to practice, we allow idle time periods only immediately prior to setting up campaigns. We consider two types of states; the first is when a campaign is in process while the second is when a campaign has finished and the next campaign has not yet been set up. Denote the differential costs of the first state by \(h(I, n, Q, T, \gamma(b))\) and the differential costs of the second states by \(w(I)\). The optimal average cost of the problem is denoted by \(\lambda^*\), which is treated as a variable in the Bellman equation. The SMDP is formalized as follows.

\[ (SMDP) \quad h(I, n, Q, T, \gamma(b)) = \min \{ C_s + w(I + n), \]
\[
\min_{q_{n+1}} E_{t_{n+1}|q_{n+1}} [h(I - t_{n+1}d, n + 1, Q + q_{n+1}, T + t_{n+1}, \gamma'(b)) + g(I, t_{n+1}) - \lambda^* t_{n+1}] \}
\]
\[
w(I) = \min_{t \geq t_s} \{ h(I - td, 0, 0, 0, \gamma_0(b)) - \lambda^* t \}. \] (7)

Here \(\gamma_0(b)\) is the prior belief distribution on \(b\) (the distribution from which \(b\) is drawn), \(\gamma'(b)\) is the belief distribution over \(b\) after observing the next batch. The time for the next batch, denoted by \(t_{n+1}\), is random and depends on \(q_{n+1}\). The first term in the minimization represents the decision to switch to a new catalyst, and the second term represents the decision to produce another batch with the current catalyst, in which case the next target attribute level \(q_{n+1}\) must also be chosen.

As a consequence of the curse of dimensionality, this problem is too complex to be approached directly. Therefore we construct a two-level heuristic described in §4 to solve this problem. To evaluate the performance of this heuristic, we next present a procedure to compute lower bounds on the optimal average cost \(\lambda^*\) of the SMDP. Some results from the lower bounds will be used to develop the two-level heuristic.
3 Lower Bounds

To compute a lower bound on the optimal average cost of the SMDP, we first consider a deterministic version of the problem and compute an associated lower bound. This lower bound does not consider the cost of randomness and discrete production and is usually a loose bound. However, we consider this bound for two reasons: (i) the resulting (loose) lower bound is used to construct a tighter stochastic lower bound, and (ii) this tractable model structure and the respective insights are used to construct our heuristic solution in §4. We then present a stochastic lower bound which accounts for discrete production and randomness of the process. This bound resolves the inadequacies of the deterministic bound which result from ignoring discreteness and randomness, but still ignores the uncertainty on production parameters and assumes perfect knowledge of the (randomly) realized catalyst productivity of each batch; in other words it assumes that we can optimally exploit the productivity of a catalyst as if we had full information. Such clairvoyant bounds have been used in the stochastic programming literature (Ciocan and Farias 2012). Clairvoyant bounds underestimate optimal costs because they assume more accurate learning than is actually possible (Brown and Smith 2013). However, we found in our computational analysis that this bound performed quite well in our problem context, as we have considered randomness and discrete production of the process in computing this bound.

3.1 Deterministic Lower Bound

The deterministic version of the problem is formed by assuming that a campaign with \( N \) batches always takes a deterministic amount of time equal to \( \tau^*(N) \), where \( \tau^*(N) \) is the expected optimal time it would take to produce \( N \) batches in a campaign, given full information of \( b \) and \( z \), or formally:

\[
\tau^*(N) = E_{b,z} \min_q [\tau(q, b, z)].
\]  

(8)

To relax the integer constraint on \( N \) and allow continuous production, we define \( \tau^*(N) \) for non-integer values of \( N \) by a weighted average of the production time of \([N]\) and \([N]\), the two closest integers to \( N \).

\[
\tau^*(N) \triangleq ([N] - N)\tau^*([N]) + (N - [N])\tau^*([N]) \quad N \text{ non-integer.}
\]  

(9)
To see the reasoning behind equation (9), note that one way to produce \(N\) batches per campaign on average is to produce \(\lfloor N \rfloor\) batches in \((\lfloor N \rfloor - N)\) fraction of the campaigns and \(\lceil N \rceil\) batches in \((N - \lfloor N \rfloor)\) fraction of the campaigns, leading to an average time of \((\lfloor N \rfloor - N)\tau^*(\lfloor N \rfloor) + (N - \lfloor N \rfloor)\tau^*(\lceil N \rceil)\) per campaign. The function \(\tau^*(N)\) is piecewise linear and convex. It is convex because for an integer \(N\), the following inequality holds as a result of decaying productivity:

\[
\tau^*(N + 1) - \tau^*(N) \geq \tau^*(N) - \tau^*(N - 1). \tag{10}
\]

Let \(T_{cyc}\) be the total length of a cycle, including the idle time and the campaign time.

**Proposition 1** The following Economic Production Quantity (EPQ) problem provides a lower bound on \(\lambda^*\), the optimal average cost of the SMDP.

\[
[\text{EPQ}] \quad \lambda_{EPQ} = \min_{T_{cyc}, I} \frac{C_s + g(I, T_{cyc})}{T_{cyc}} \quad \text{s.t. } \tau^*(T_{cyc}d) + t_s \leq T_{cyc}. \tag{11}
\]

All proofs are provided in the online supplement §A. Here, the objective of the EPQ is to minimize the average cost during a fixed cycle. The decision variables \(T_{cyc}\) and \(I\) denote the length and the starting inventory of the cycle, respectively. The total cost during a cycle is equal to a one-time switching cost, plus inventory holding and backlogging costs during the cycle \((g(I, T_{cyc}))\). A total of \(T_{cyc}d\) batches are produced in a campaign to match the total demand during the length of the cycle. The constraint ensures that the total time required to produce \(T_{cyc}d\) batches (i.e. \(\tau^*(T_{cyc}d)\)) plus the setup time is less than or equal to the length of the cycle. We define the following parameters based on the solution to (11):

\(T_{cyc}^*: \) optimal cycle length in (11).

\(T^M: \) largest cycle length satisfying the production constraint in (11).

\(T^m: \) smallest cycle length satisfying the production constraint in (11).

\(N^* \triangleq T_{cyc}^*d: \) total demand during \(T_{cyc}^*\). It is similar to the optimal order quantity \(Q^*\) in an unconstrained EOQ model.
\( T \): optimal inventory at the beginning of each cycle (i.e. after adding to inventory the batches produced in the previous campaign).

\( I \triangleq T - N^* \): lowest inventory reached at the end of a cycle, just before the newly produced batches are added.

\( C_{IB} \triangleq \frac{C_I C_B}{C_I + C_B} \): “balanced” inventory holding and backlogging cost per unit time in an EPQ with backlogging (will be discussed shortly).

\( C_s^0 \): the supremum value of all \( C_s \) such that the constraint in (11) is not binding at the optimal solution (will be discussed shortly).

Given a cycle length \( T_{cyc} \), inventory will be positive for a fraction of the cycle, and for the remaining time inventory will be negative. We show that the optimal fraction of time where inventory is positive and where it is negative are proportional to \( C_B \) and \( C_I \) respectively.

**Proposition 2** The optimal \( T \) can be derived as a function of \( T_{cyc} \) and replaced in the objective function. The resulting objective function is convex in \( T_{cyc} \).

**Corollary 2.1** Problem EPQ becomes a convex optimization problem in the single variable \( T_{cyc} \).

In the proof of Proposition 2 (in the online supplement) we see that the objective over \( T_{cyc} \) becomes \( \frac{C_s}{T_{cyc}} + \left( \frac{C_I C_B}{C_I + C_B} \right) \frac{T_{cyc} d}{2} \), which is similar to an EPQ without backlogging in which \( C_{IB} \triangleq \frac{C_I C_B}{C_I + C_B} \) has replaced the inventory holding cost. The parameter \( C_{IB} \) is interpreted as the balanced inventory cost (holding and backlogging) per unit time when the cycle length is optimally allocated between positive and negative inventory. The optimal production quantity in the unconstrained EPQ problem is

\[
N^u = \sqrt{\frac{2C_S d}{C_{IB}}}. \tag{12}
\]

If \( T^u_{cyc} = N^u / d \) satisfies the production constraint, then the constraint is not binding and \( T^u_{cyc} \) is optimal for problem (11). In this case the following relations would hold:

\[
N^* = N^u = \sqrt{\frac{2C_S d}{C_{IB}}},
\]

\[
I = \frac{C_I}{C_I + C_B} N^*,
\]

\[
\lambda_{EPQ} = \sqrt{2C_{IB} C_S d} = IC_B = N^* C_{IB}. \tag{13}
\]
In order for $T_{ucyc}$ to be feasible (hence optimal) we must have $T^m \leq T_{ucyc} \leq T^M$. If $T_{ucyc}$ does not satisfy the constraint, then the constraint is binding at the optimal $T_{ucyc}^*$ because by Proposition 2 the objective function is convex in $T_{cyc}$. If $T_{ucyc} < T^m$, then $T_{cyc}^* = T^m$, and if $T_{ucyc} > T^M$, then $T_{cyc}^* = T^M$.

3.2 Stochastic Lower Bound

The solution to $EPQ$ gives a lower bound for the original problem, but it does not account for the cost of randomness and discrete production. To obtain a tighter bound, we use the $EPQ$ and define a stochastic process that is similar to the actual production process, but is regenerative and hence more tractable. The optimal cost of this regenerative process is a tighter lower bound than the $EPQ$ bound for the original problem.

**Proposition 3** The differential cost function $w(I)$ defined in (7) has a global minimizer $I^*$. 

According to Proposition 3, the level $I^*$ is the ideal inventory to have at the beginning of a cycle. However, it is not necessarily optimal to immediately start the next campaign at $I^*$ and we might allow some idle time. This is shown in Proposition 4.

**Proposition 4** There exists an inventory level $I_0^* \leq I^*$, such that if the inventory level at the beginning of a cycle is $I^*$, it is optimal to delay the campaign setup till inventory falls to $I_0^*$.

We now define a regenerative process by modifying the original process as follows:

I) Production always starts at $I_0^*$. 

II) If the inventory level $I$ at the end of a campaign is less than $I_0^*$, the inventory level is instantly raised to $I_0^*$ at a cost of $\lambda_{EPQ} t_{idl} - g(I_0^*, t_{idl})$, where $\lambda_{EPQ}$ is the optimal cost of problem (11) and $t_{idl} = (I_0^* - I)/d$.

III) If the inventory level $I$ at the end of a campaign is greater than $I_0^*$, the inventory level is instantly dropped to a level $I'$ of choice ($I \geq I' \geq I_0^*$) and then production is idled until inventory reaches $I_0^*$. 

12
Proposition 5 There exists a regenerative process satisfying I-III that has a lower average cost than the optimal cost of the original process. As a result, the optimal average cost of the regenerative process is less than the optimal average cost of the original process.

In the regenerative process, all campaigns start at $I_0^*$, hence it is more tractable than the original process. However, computing the optimal cost of the regenerative process is still not straightforward. In the Appendix we present an algorithm to compute a lower bound on the optimal performance of the regenerative process given $I_0^*$. A lower bound on the optimal cost of the original process can then be obtained by a line search over values of $I_0^*$.

4 Heuristics and Upper Bounds

In this section, we present a two-level heuristic to solve the SMDP. This provides an upper bound on the value of the SMDP. To benchmark the performance of this heuristic, we also describe a practitioners heuristic currently employed at a large food processing company.

4.1 Two-level Heuristic

In the two-level heuristic, we decompose the campaign planning problem with learning and decay represented by the SMDP into two levels: a lower level problem at the batch level, determining the duration of batches in a campaign, and a higher level problem at the campaign level, determining when to end a campaign. These levels are defined as:

- Level 1: Batch planning- Choose the attribute level $q_{n+1}$ of the next batch.
- Level 2: Catalyst switching- While batch $n$ is inside the reactor, use a control policy to decide whether to change the catalyst after this batch, or to move on to batch $n + 1$.

We next describe the solution method for each level.

Level-1: Batch Planning

In batch planning, we focus on minimizing the campaign duration for a given number of batches $N$. We minimize the campaign duration because it offers more buffer time and hence more flexibility. The objective is to minimize $T_N + t_N$ such that the average attribute level constraint is satisfied.
The number of batches $N$ is tentatively chosen as the expected number of batches in the campaign which depends on the higher-level policy discussed in the next subsection. For ease of exposition and without loss of generality, the attribute level is normalized such that the required average attribute level is less than or equal to 1. Hence the average attribute level constraint becomes:

$$\sum_{i=1}^{N} q_i \leq N. \quad (14)$$

With this constraint, the state space should include the cumulative attribute level. After completion of batch $i-1$, a decision $q_i$ is made for the $i^{th}$ batch, given the current state of the campaign $[Q_i, T_i, \gamma(b)]$. The resulting stochastic DP is represented by the following Bellman equation:

$$v_i(Q_i, T_i, \gamma(b)) = \min_{q_i} \{E_{z_i,b} \{ (b + z_i)k(T_i)f(q_i) + v_{i+1}(Q_i + q_i, T_i + (b + z_i)k(T_i)f(q_i), \gamma'(b)) \} \}, \quad (15)$$

where $\gamma(b)$ is the current belief distribution on the parameter $b$ and $\gamma'(b)$ is the updated belief after observing the random outcome of $b + z_i$. We use a standard Bayesian updating procedure to get $\gamma'(b)$: each period based on our observation of the pair $(q_i, t_i)$, we observe the implied productivity $b_i \triangleq b + z_i$ through

$$t_i = b_i k(T_i)f(q_i) \Rightarrow b_i = t_i / k(T_i)f(q_i).$$

We use this observation along with the known density of $z_i$ (i.e. $\kappa(z_i)$) to update $\gamma(b)$ according to $\gamma'(b) \propto \gamma(b)\kappa(b_i - b)$. Here the current belief $\gamma(b)$ acts as the prior belief, and the new probability density for $b$, according to Bayes’ rule, is proportional to the previous density multiplied by the likelihood of observing $b_i$ conditional on $b$ (i.e. $\kappa(b_i - b)$).

An analytical closed form solution to this problem is not available in general, and the continuous multidimensional state space of this stochastic Bayesian DP problem makes it intractable to numerically find an optimal policy (Mazzola and McCardle 1996). Hence we approach this problem with the following re-optimization heuristic: first we find the vector $q$ that minimizes the expected campaign length $E_{b,z} \{ \tau(q,b,z) \}$, ignoring the fact that $q$ can be adjusted in the future. After the first batch $q_1$ is completed, we incorporate learning by updating $\gamma(b)$ and re-optimizing the remaining batches 2 through $N$ of $q$. We then implement the revised second batch, and repeat.

The re-optimization policy does not directly anticipate the value of information in its decision.
process. In the online supplement §B we set up a tractable 3-batch example and compare the re-optimization policy with the optimal policy. We make three important observations:

i) In all problem settings considered, the choice of \( q_1 \) under the re-optimization policy is very close to the optimal choice of \( q_1 \).

ii) The resulting average campaign time is almost identical under the two policies (they are within 1% of each-other in all considered problem settings).

iii) Both policies attain near-optimal exploitation of the catalyst, as if full information on \( b \) were available apriori (in every problem setting, the expected campaign duration under the two policies is within 2% of the minimum campaign duration).

In addition, our computational results in §6 show that our two-level heuristic achieves near-optimal costs, implying that the re-optimization policy adequately captures the dynamic value of information in this problem.

With the re-optimization policy in place, we are interested in understanding how the chosen \( q \) will differ from the myopic policy of having every batch meet the target attribute level (i.e. \( q = [1, 1, ..., 1] \)). Note that a smaller \( q_i \) implies that batch \( i \) has a bigger contribution to satisfying the average attribute level constraint (14). Therefore, we interpret assigning a smaller \( q_i \) to batch \( i \) as placing a higher “load” on batch \( i \). We are interested in whether it is better to place higher load (i.e. smaller \( q_i \)) on the production of the first batches (while the productivity of the catalyst is still high) and leave a lower load on the concluding batches, or conversely place a lighter load on the initial batches to avoid an overly decayed catalyst when it reaches the final batches. As shown in Proposition 6, this depends on the form of \( k(T_i) \), the function which defines the dependency of the productivity decay rate (equivalently, the rate of increase in processing time) on the total consumption. Denote \( q^* = [q^*_1, ..., q^*_N] \) as the solution to \( \arg \min_q E_{b,z} \{ \tau(q, b, z) \} \).

**Proposition 6** (i) If \( k(T) \) is convex in \( T \), then there exists at least one optimal solution where \( q^*_i \geq q^*_{i+1} \). Further, if \( k(T) \) is strictly convex, there are no optimal solutions where \( q^*_i \leq q^*_{i+1} \).

(ii) If \( k(T) \) is concave in \( T \), then there exists at least one optimal solution where \( q^*_i \leq q^*_{i+1} \). Further, if \( k(T) \) is strictly concave, there are no optimal solutions where \( q^*_i \geq q^*_{i+1} \).
(iii) If \( k(T) \) is an affine function, then \( q_i^* = q_{i+1}^* \) for all \( i \).

Proposition 6 establishes that a higher load should be allocated to the finishing batches if \( k(T) \) is a convex function and to the beginning batches if \( k(T) \) is a concave function. To see the intuition behind this finding, consider a two-batch example. When \( k(T) \) is convex, the decay rate is increasing in total consumption, thus if a high load is placed on the first batch, the second batch will face an overly decayed catalyst, increasing the total time required to process the two batches. The reverse is true for a concave \( k(T) \).

![Figure 1: Duration of batches in a reactor for an optimal campaign in comparison to a strict campaign.](image-url)

Figure 1 compares \( q^* = \arg \min_q E_{b,z} \{\tau(q,b,z)\} \) with \( q = [1, \ldots, 1] \) for a convex \( k(T) \). Each downward slope represents one batch, where the attribute level is reduced from an initial level. Each upward jump represents removing a batch and placing another batch inside the reactor. In this example the optimal campaign is 20% faster than the campaign that strictly attains the desired attribute level in all batches.

**Level-2: Catalyst Switching**

In catalyst switching, we develop a policy by which to decide when to switch the catalyst. A control policy maps the state space to a binary decision: switch or don’t switch. The state space consists
of the current inventory level \((I)\), number of batches produced so far in the current campaign \((n)\), current belief distribution of \(b \ (\gamma(b))\), and the total consumption of the current catalyst \((T)\). The Bellman equation for this average-cost problem is an approximation to the SMDP, where the \(Q\) dimension is removed from the state space and the decision variable is binary (instead of being a continuous choice of \(q_n\), which is now handled by the lower-level problem).

\[
\begin{align*}
  h(I, n, T, \gamma(b)) &= \min \{C_s + w(I + n), \\
  &\ E_{t_{n+1}}[h(I - t_{n+1}d, n + 1, T + t_{n+1}, \gamma'(b)) + g(I, t_{n+1}) - \lambda^*t_{n+1}] \} \\
  w(I) &= \min_{t \geq t_s} \{h(I - td, 0, 0, \gamma_0(b)) - \lambda^*t\}.
\end{align*}
\]  

(16)

The state space is multidimensional and has continuous elements. Even if we remove the elements of learning and decay, the state space is still too large to obtain exact solutions (Loehndorf and Minner 2013), even numerically. Hence we propose an approximate policy to efficiently summarize the large state of the system and apply it in a simple decision rule.

The intuition behind our heuristic is to follow as closely as possible the optimal policy of the deterministic process suggested by the solution to the EPQ. Let \(\bar{T}\) and \(T_{cyc}^*\) be the optimal solution to (11). Define \(\bar{I} \triangleq T - T_{cyc}^*d\) and \(I_0 \triangleq \bar{I} + [\tau^*(T_{cyc}^*d) + t_s]d\). In the absence of randomness and discreteness, we would like the inventory level at the beginning of a cycle to be \(\bar{T}\), idle the process until inventory reaches \(I_0\), at which time we set up a campaign and produce batches until inventory is at \(\bar{I}\). The batches produced return inventory to \(\bar{T}\). However, in the discrete-production stochastic process this does not necessarily happen. At some point we expect that if we produce another batch, the inventory level at the end of that batch will be lower than \(\bar{I}\), which would result in excess backlogging costs. On the other hand, if we do not produce another batch and switch the catalyst before reaching the optimal level of \(\bar{I}\), it would result in higher average switching costs, and possibly higher inventory costs in the next cycle.

To choose between these options, we propose to switch before inventory gets to \(\bar{I}\) only if the probability of falling below inventory \(\bar{I}\) after the next batch is greater than some threshold \(\Psi\). The probability \(P[I_{n+1} < \bar{I}]\) is calculated using the distribution of \(z_n\) and the current belief over \(b\):
\[ P[I_{n+1} < I] = P[I - t_{n+1}d < I] = P[t_{n+1}d > I - I] \]
\[ = P[(b + z_{n+1})k(T_{n+1})f(q_{n+1})d > I - I] \]
\[ = P[b + z_{n+1} > \frac{I - I}{k(T_{n+1})f(q_{n+1})d}]. \] 

(17)

To illustrate the implications of this policy, note that the boundary case of \( \Psi = 1 \) translates to a policy where the catalyst is switched only after inventory falls below \( I \). Our numerical results show that if the threshold \( \Psi \) is correctly specified, this policy is nearly optimal for many problem instances. The optimal threshold \( \Psi \) depends on the cost parameters \( (C_I, C_B, C_S) \) and is not analytically computable as it would require solving a Bellman equation almost as big as the original problem (16). Therefore, we choose \( \Psi \) by simulating the process and doing a line search over \( \Psi \).

Denote \( I_n \) for the inventory level after the production of batch \( n \). Given the threshold, our heuristic separates into the following two cases, based on the initial inventory level after the previous campaign \((I)\).

**Case 1:** \( I \geq I_0 \). Idle the process until inventory reaches \( I_0 \), then set up and start the next campaign. Produce batches \( 1, \ldots, n \) until \( P[I_{n+1} \leq I] \geq \Psi \). For \( i \geq n \), end the campaign if and only if (i) \( I_i + i \geq I_0 \) or (ii) \( I_i + i \geq E[I_{i+1}] + i + 1 \).

**Case 2:** \( I < I_0 \). Set up and start the next campaign with zero idle time. Find the smallest \( n \) such that \( P[I_{n+1} \leq I] \geq \Psi \) and either \( E[I_n] + n \geq I_0 \) or \( E[I_n] + n \geq E[I_{n+1}] + n + 1 \). For this \( n \), if the inequality \( E[I_{n+1}] + n \geq I_0 \) holds, run the campaign as in Case 1. Otherwise, produce \( N \) batches such that \( N \) maximizes \( N/(\tau^*(N) + t_s) \); this \( N \) maximizes the production rate.

The additional conditions in Case 1 enhance the threshold policy by ensuring that the process remains stable after the switch. Condition (i) implies that the next campaign will start above inventory \( I_0 \). If condition (ii) holds, we expect that if we switch now, we start the next campaign with a higher inventory compared to switching after the next batch. In Case 2, we use the same policy as in Case 1 only if we expect that we will be able to start the next campaign from inventory.
above \( I_0 \). Otherwise, we produce at the maximum production rate to bring the process back up to stable conditions.

### 4.2 Practitioner’s Heuristic

To benchmark the two-level heuristic, we compare it with a practitioner’s heuristic that was implemented as part of a broader project described in Rajaram et al. (1999). The decision variables are \( t^* \), how long to leave each batch in the reactor, and \( N \), the maximum number of batches in a campaign. The optimal \( t^* \) and \( N \) (not necessarily unique) solve the following optimization problem:

\[
\min_{N,t^*} \left[ C_I \frac{N}{2} + C_S \frac{d}{N} \right] \quad (18a)
\]

\[
\text{s.t.} \quad \frac{N}{Nt^* + t_s} > d \quad (18b)
\]

\[
\sum_{i=1}^{N} f^{-1}\left( \frac{t^*}{k(it^*)\mu_b} \right) \leq N \quad (18c)
\]

In (18a), the average inventory during a cycle is \( \frac{N}{2} \), hence the average inventory holding cost is equal to \( C_I \frac{N}{2} \). The average length of a cycle is \( \frac{N}{4} \), hence the average switching cost per unit time is equal to \( C_S \frac{d}{N} \). The constraint (18b) ensures that the average production rate exceeds the demand rate. This follows as the choice of \( \{N, t^*\} \) implies a production rate of \( N \) batches per \( Nt^* + t_s \) units of time, hence for feasibility \( \frac{N}{Nt^* + t_s} \) has to be greater than the demand rate \( d \). Constraint (18c) is the attribute level constraint, approximating \( b + z_n \) by the expected value \( \mu_b \).

The optimal \( N \) and \( t^* \) are found by discretizing \( t \) and performing a grid search. However, \( t^* \) will not be unique because the objective function of (18) depends only on the variable \( N \). We choose the smallest feasible \( t^* \) to increase the production rate, which increases the idle time between campaigns and allows a larger buffer in case of a bad catalyst outcome.

Once the optimal \( t^* \) and \( N \) are found, the practitioner’s heuristic is implemented as follows: Set \( t_i = t^* \) for all \( i \) and observe the values of the resulting \( q_i \)'s. From the observed \( q_i \)'s, update the belief distribution \( \gamma(b) \). After batch \( n \), decide whether the catalyst has the potential to produce...
another batch with $t_{n+1} = t^*$ while preserving the average attribute-level constraint:

$$\sum_{i=1}^{n+1} f^{-1}(\frac{t^*}{k(i^*)}(b + z_i)) \leq n + 1. \quad (19)$$

Given the current information, use the expected value of $b + z_n$ (i.e. $E[b|\gamma(b)]$) to evaluate (19); produce another batch if and only if $\sum_{i=1}^{n+1} f^{-1}(\frac{t^*}{k(i^*)}E[b|\gamma(b)]) \leq n + 1$ and $n + 1 \leq N$.

If the prediction is wrong and the next batch violates the constraint, stop the current campaign and incur a rework cost. Once the campaign is ended, the catalyst is replaced and the new batches are released to inventory.

Observe that this heuristic does not allow for deliberate backlogging and is designed for the settings where $C_B \gg C_I$; it does not provide a fair benchmark when $C_B$ and $C_I$ are comparable. To enhance this heuristic and provide a fair benchmark in all settings, we replace the cost term $C_I$ in (18) by $C_{IB} = C_I C_B / (C_I + C_B)$ which represents the optimal balance between inventory holding and backlogging costs (see §3.1). To optimally balance these costs, instead of starting and ending the cycles at $I = N$ and $I = 0$, we start and end the cycles at $I = N \frac{C_B}{C_I + C_B}$ and $I = \frac{C_I}{C_I + C_B}$ respectively. Finally, since $t^*$ is chosen as the smallest feasible value for the chosen $N$, the production rate will be higher than the demand rate. Therefore, idle times are chosen such that the cycles start and finish at these inventory levels.

5 Multiple Products

We now consider a setting where multiple products are produced on a single reactor. Each product has its own fixed demand rate, backlogging costs, and inventory holding costs. This problem is now a stochastic economic lot sizing problem with switching costs, batch production, learning and decay.

During the production of a given campaign, after each batch is produced, a decision must be made: continue this campaign and produce another batch of the current product, or finish this campaign, add the produced batches to inventory, and start a new campaign. Note that only one type of product can be produced in a campaign. After a campaign is finished and the produced batches are added to inventory, the next decision is which product to produce next and how much idle time, if any, to allow. These decisions are based not only on the number of batches produced
the total consumption of the catalyst $T$, and the current belief distribution $\gamma(b)$ over the inverse productivity parameter $b$ for the current product, but also on the inventory level $I$ of all the products. We extend the ideas discussed in the single product setting to obtain a lower bound and a heuristic for the multiple product setting.

5.1 Lower Bound

Let $R$ be the total number of products. We modify the previously introduced parameters and variables by adding indices $r$ (or superscripts, for inventory variables) to denote the product type. A deterministic lower bound on the optimal performance of the stochastic system is to assume that each product undergoes an EPQ process independent of the others, except that the sum of fractions of time that the machine is busy cannot be greater than 1.

$$
\text{MEPQ} \triangleq \lambda_{ELSP} \equiv \min_{\{N_r, T\}} \sum_{r=1}^{R} \frac{C_s + g(T^r, N_r/d_r)}{N_r/d_r} \\
\text{s.t. } \sum_{r=1}^{R} \frac{\tau^*_r(N_r) + t_s N_r}{N_r/d_r} \leq 1. \tag{20}
$$

We can relax the constraint in (20) using a Lagrange multiplier $\delta$. This leads to:

$$
F(\delta) \equiv \min_{\{N_r, T\}} \sum_{r=1}^{R} \frac{C_s + g(T^r, N_r/d_r) + (\tau^*_r(N_r) + t_s)\delta N_r}{N_r/d_r} - \delta. \tag{21}
$$

The minimization problem decomposes into $R$ separate problems.

$$
\min_{N_r, T} \frac{C_s + g(T^r, N_r/d_r) + (\tau^*_r(N_r) + t_s)\delta N_r}{N_r/d_r}, \tag{22}
$$

Each of these $R$ problems can be transformed into a single variable problem over $N_r$. This is done by noting that the optimal fractional allocation of a cycle between positive and negative inventory is fixed, and hence the cycle length $N_r/d$ uniquely determines $T$. We solve these $R$ single-variable problems separately:

$$
\min_{N_r} Z(N_r) = \frac{C_s + C_{IBr} N_r^2 / 2 d_r + (\tau^*_r(N_r) + t_s)\delta N_r}{N_r/d_r}, \tag{23}
$$
where $C_{IrBr} \triangleq \frac{C_{Ir}C_{Br}}{C_{lr}+C_{Br}}$ (this can be shown using a similar logic to the proof of Proposition 2).

Because $\tau^*_r(N_r)$ is convex, the numerator is convex. Further, it is well known that if $f(x)$ is convex then $f(x)/x$ is quasiconvex; thus, $Z(N_r)$ is quasiconvex in $N_r$. We optimize over $N_r$ by using simple numerical methods, so $F(\delta)$ is easy to evaluate. We maximize the concave function $F(\delta)$ using a golden section search, obtaining a lower bound on the average cost of the deterministic relaxation of the original stochastic problem.

**Proposition 7** Strong duality holds in problem $MEPQ$.

Based on Proposition 7, the $N_r$’s obtained by the above procedure are feasible for problem $MEPQ$. If $N_r/d_r$ is the same for all products, the $R$ products could be functioning as if they were independent EPQ systems. The sum of the average costs of these $R$ systems forms a lower bound on the optimal average cost of the original problem. To improve this lower bound, we use the same procedure presented in section 3.2 by constructing a separate regenerative process to individually improve the lower bound for each product and compute the sum of the improved costs.

5.2 Heuristics

5.2.1 Multi-product Two-level Heuristic

Similar to the two-level heuristic for the single-product case, this heuristic uses the $\bar{T}^*$’s and $N_r$’s that solve the lower-bound problem $MEPQ$. Define $\bar{I}' = \bar{T} - N_r$ and $I_0' = \bar{I}' + (\tau^*_r(N_r) + t_s)d_r$.

Ideally, we would like to start a campaign of product $r$ when its inventory level reaches $I_0'$ and produce $N_r$ batches until its inventory level reaches $I'$. During this time we do not want the inventory level of any other product $r'$ to go below its respective $I_0'$. This may not be possible as in the midst of a campaign of product $r$, the inventory level of some other product $r'$ would (in expectation) go below its $I_0'$ if another batch of product $r$ is produced. We must trade off producing fewer batches of product $r$ with starting the next campaign with less inventory of product $r'$.

Similar to the single product case, we use a probability threshold $\Psi$, chosen by a line search over choices of $\Psi \in [0,1]$. Our heuristic dynamically makes decisions by monitoring the inventory level of all products and the changes in the belief over the catalyst productivity $\gamma(b)$. We end the campaign if for some $r'$ the probability of dropping below inventory $I_0''$ during the next batch
is greater than the threshold. We add a few conditions to ensure that the process is stable (i.e. inventory does not arbitrarily increase or decrease).

At the end of a campaign, we need to choose the next product $r^c$ to produce. For this purpose, we try to choose the $r^c$ that would otherwise induce the largest backlogging cost. Note that if $r^c$ is not produced in the next campaign, it will not be replenished for at least the duration of the next two campaigns. We approximate the duration of the next two campaigns by $\hat{\tau} \triangleq \min_r \tau^*(N^r) + \max_r \tau^*(N^r)$, and choose $r^c$ as the product with the largest backlogging cost during this time, starting at its current inventory $I^r$ and ending at $I^r - \hat{\tau}d^r$. With this choice of $r^c$, we describe the proposed policy for the next campaign.

Let $I^r$ be the inventory level of product $r$ at the end of a campaign, and $I_n^r$ be the inventory level of product $r$ after producing batch $n$ in the current campaign. For each product $r$, define $N^r_m \triangleq \arg \max_N \{N/\tau^*_r(N) + t_s\}$. We split the proposed policy into the following three scenarios, depending on the inventory vector $I$ after the end of the previous campaign.

**Case 1.** $I^r \leq I^r_0$ for more than one $r$. This implies that the current inventory of more than one product is below its optimal starting value, indicating that a shortage might occur by the end of the next campaign. Set up a campaign for product $r^c$, and produce exactly $N^c_m$ batches.

**Case 2:** $I^{r^c} \leq I^{r^c}_0$, but $I^r > I^r_0$ for all $r \neq r^c$. Unlike case 1, the only imminent shortage is $r^c$. Produce $n$ batches of product $r^c$ until either (i) $I^{r^c}_n + n \geq \mathcal{T}^{r^c}$, (ii) producing batch $n + 1$ would exceed the time allocated to this campaign $(T_{n+1} + E[t_{n+1}] \geq \tau^*_r(N_{r^c}))$, or (iii) $n \geq N^{r^c}_m$ and for some product $r'$ we have $P[I^{r'}_{n+1} < I^{r'}] \geq \Psi$.

**Case 3:** $I^r > I^r_0$ for all $r$. Let $r$ be the product with the lowest $(I^r - I^r_0)/d_r$ (excluding $r^c$). Compute $J \triangleq I^r_0 + \tau^*_r(N_r)d_r$. If $J \geq I^{r'}$, set up a campaign of product $r^c$ without any idle time. If $J < I^{r'}$, allow enough idle time such that either $I^{r'}$ drops to $J$, or $I^{r^c}$ drops to $I^{r^c}_0$ (whichever happens first) and then set up a campaign of product $r^c$ and follow the procedure in Case 2.

### 5.2.2 Practitioner’s Heuristic

Similar to the single product practitioner’s heuristic, the multiple product practitioner’s heuristic is part of the implementation described in Rajaram et al. (1999). In their heuristic, they use a fixed cycle length with one campaign of each product. Since switchover costs are not product
dependent, the sequence of the products inside the cycle is not considered. Similar to the single product setting, a fixed batch-operation time $t_r^*$ and a target number of batches per campaign $N_r$ is chosen for each product $r$. The policy determining when to end a campaign of product $r$ is the same as in the single-product case. The initial problem solved to determine the cycle length $L$, batch durations $t_r^*$, and target number of batches $N_r$ is:

$$\text{(PH)} \min_{\{N_r, t_r^*\}, L} \left\{ \frac{\sum_{r=1}^{R} \left( \frac{N_r C_{Ir}}{2} + C_S \right)}{L} \right\},$$

s.t. \hspace{1cm} N_r \geq Ld_r \hspace{0.5cm} \forall r; \hspace{1cm} (24a)

$$\sum_{r=1}^{R} (N_r t_r^* + t_s) \leq L, \hspace{1cm} (24b)$$

$$\sum_{n=1}^{N_r} f^{-1}\left( \frac{t_r^*}{k(nt_r^*)\mu_{br}} \right) \leq N_r \hspace{0.5cm} \forall r, \hspace{0.5cm} N_r \in \{1, 2, \ldots\}. \hspace{1cm} (24c)$$

The objective is to minimize the total average inventory holding costs and switching costs during the cycle of length $L$. Constraint (24a) ensures that the number of batches that are planned to be produced should not be lower than the demand of product $r$ during the cycle. Constraint (24b) enforces that the sum of campaign times does not exceed the cycle length. Constraint (24c) ensures that the mixture of batches for each product meets the attribute level constraint. The algorithm used to approximately solve PH is provided in the online supplement §C.

As noted for the single-product case, the practitioner’s heuristic requires enhancements to provide a fair benchmark in all problem settings. These include enhancing the solution by replacing all $C_{Ir}$ in (24) by $C_{IBr} = C_{Ir}C_{Br}/(C_{Ir} + C_{Br})$. In addition, after a campaign of product $r$ is completed, we delay releasing the prepared batches such that the replenished inventory begins at $N^*C_{Br}/(C_{Ir} + C_{Br})$.

### 6 Computational Results

To evaluate our method for the single and multi-product problems, we compare it with the practitioner’s heuristic and with the appropriate lower bound. We first consider the single-product case. Here, we were provided data from a sorbital production process in a large food-processing company. To test our heuristic under a wide range of parameter settings and to capture settings for other industries, we varied these parameters to obtain new problem instances, by considering: (i) concave
$k(T)$ vs. convex $k(T)$, (ii) high traffic vs. low traffic, (iii) $C_B = \alpha C_I$, $\alpha \in [0.2, 0.5, 1, 2, 5]$, and (iv) low $C_S$, medium $C_S$, high $C_S$.

For concave and convex decay functions, we respectively use $k_{\text{conc}}(T) = (1+T)^{0.7}$ and $k_{\text{conv}}(T) = \frac{1}{2}(1+T)^{1.2}$. In all our experiments we use $f(q) = -\log(q/2)$.

Define the capacity utilization as the minimum fraction of time that the machine would be busy (i.e. not idle) to meet demand. In Table 1, low traffic refers to a capacity utilization of 30%, while high traffic refers to a capacity utilization of 75%. In most problem instances 75% is the highest capacity utilization for which the practitioner’s heuristic is feasible.

The results for the single-product case are shown in Table 1. The results are expressed as a percentage gap, defined as the difference between the value of the appropriate heuristic and the lower-bound solution as a percentage of the lower bound for a particular setting defined by the first entry in the row. The results shown for the row are from the problem instances got by varying the remaining parameters one at a time. For example, consider the first row in Table 1 with the problem setting convex $k(T)$. If we vary the remaining parameters one at a time we will have $1 \times 2 \times 5 \times 3 = 30$ problem instances. Similarly, consider the fifth row with problem setting low $C_S$. Here, we will now have $2 \times 2 \times 5 \times 1 = 20$ problem instances. In this manner, we compile all the results in Table 1.

<table>
<thead>
<tr>
<th>Problem Setting</th>
<th>% gap from lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practitioner’s heuristic</td>
</tr>
<tr>
<td></td>
<td>min</td>
</tr>
<tr>
<td>convex $k(T)$</td>
<td>14.3</td>
</tr>
<tr>
<td>concave $k(T)$</td>
<td>4.81</td>
</tr>
<tr>
<td>high traffic</td>
<td>6.22</td>
</tr>
<tr>
<td>low traffic</td>
<td>4.81</td>
</tr>
<tr>
<td>low $C_S$</td>
<td>4.81</td>
</tr>
<tr>
<td>medium $C_S$</td>
<td>5.43</td>
</tr>
<tr>
<td>high $C_S$</td>
<td>6.85</td>
</tr>
<tr>
<td>$C_B = C_I$</td>
<td>4.81</td>
</tr>
<tr>
<td>$C_B = 2C_I$</td>
<td>5.94</td>
</tr>
<tr>
<td>$C_B = 5C_I$</td>
<td>6.82</td>
</tr>
<tr>
<td>$C_B = 0.5C_I$</td>
<td>6.49</td>
</tr>
<tr>
<td>$C_B = 0.2C_I$</td>
<td>5.06</td>
</tr>
</tbody>
</table>

Table 1: Percentage gaps of heuristics for the single product problem ($C_B \geq C_I$)
Based on our computational analysis, we make the following observations:

• In all problem instances the two-level heuristic significantly outperforms the practitioner’s heuristic. In particular, as seen in Table 1, the average cost of the two-level heuristic was 11% lower than the practitioner’s heuristic and this cost improvement ranged from 4% to 25%. In addition, the two-level heuristic achieves very low gaps with the lower bound in spite of the stochastic nature of the problem. This is because reoptimization with learning captured most of the value of information, and the probability threshold policy for catalyst switching provided a near-optimal decision rule.

• The superior performance of the two-level heuristic is explained by two reasons: first, the two-level heuristic exploits the productivity of the catalyst and is able to produce a fixed number of batches in a smaller time span. Consequently, the two-level heuristic can produce more batches in a cycle and meet demand at a faster rate. This advantage is especially pronounced when there is greater uncertainty in catalyst performance and the constraint in EPQ is binding. Second, the practitioner’s heuristic uses expected parameter values in its decision process and thus it does not make efficient use of the available information (i.e. probability distributions on parameters, etc.) and is more prone to making sub-optimal decisions.

• We observe that the optimal probability threshold $\Psi$ decreases in $C_B$ and increases in $C_S$. A lower $\Psi$ results in a lower risk of backlogging, compensating for the higher $C_B$. On the other hand, a low $\Psi$ may result in switching more frequently than desired, thus $\Psi$ is increasing in $C_S$.

• The heuristic is robust to the choice of $\Psi$. A deviation of $\pm0.2$ from the optimal $\Psi$ increases the average cost by less than 5% in all problem instances. A possible explanation for this finding is the well-known insensitivity of the basic EPQ model to small deviations from the optimal cycle length (Schwarz 2008).

The main managerial insight obtained from the simulations is that for the single-product problem, near optimal costs can be achieved by using a simple policy. This policy only requires computing the probability of inventory falling below $I_*$ as defined in (17). Such a policy effectively makes use of all the information we have about cost, current efficacy of the catalyst and inventory level.
For the multiple product case a leading food-processing company provided us data on real parameter settings for a modified starch process with five products. Here again, we varied these parameters to obtain new problem instances and capture settings for other industries, similar to those in the single product problem, except for the relation between \( C_B \) and \( C_I \). Here:

(i) \( C_B \gg C_I \) is defined as \( C_B \geq 5C_I \) for all products,

(ii) \( C_B \sim C_I \) is defined as \( 0.2C_I < C_B < 5C_I \) for all products and \( 0.5C_I \leq C_B \leq 2C_I \) for at least two products, and

(iii) \( C_B \ll C_I \) is defined as \( C_B \leq 0.2C_I \) for all products.

In addition, we considered the setting where \( d_r, C_I, \) and \( C_B \) are similar across products vs. where they are highly varied. When these parameters are similar across products, it’s possible for a rotation production cycle to be optimal, whereas when these parameters are highly varied, rotation cycle policies become less acceptable. The results for the multiple product simulation tests are shown in Table 2.

<table>
<thead>
<tr>
<th>Problem Setting</th>
<th>% gap from lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practitioner’s heuristic</td>
</tr>
<tr>
<td></td>
<td>min</td>
</tr>
<tr>
<td>convex ( k(T) )</td>
<td>19.20</td>
</tr>
<tr>
<td>concave ( k(T) )</td>
<td>9.9</td>
</tr>
<tr>
<td>high traffic</td>
<td>9.9</td>
</tr>
<tr>
<td>low traffic</td>
<td>13.4</td>
</tr>
<tr>
<td>medium ( C_S )</td>
<td>9.9</td>
</tr>
<tr>
<td>high ( C_S )</td>
<td>14.8</td>
</tr>
<tr>
<td>( C_B \gg C_I )</td>
<td>9.9</td>
</tr>
<tr>
<td>( C_B \sim C_I )</td>
<td>13.0</td>
</tr>
<tr>
<td>( C_B \ll C_I )</td>
<td>11.2</td>
</tr>
<tr>
<td>similar costs</td>
<td>9.9</td>
</tr>
<tr>
<td>varied costs</td>
<td>19.6</td>
</tr>
</tbody>
</table>

Table 2: Percentage gaps of heuristics for the multiple product problem (\( C_B \geq C_I \))

It is evident in Table 2 that the proposed two-level heuristic can significantly improve upon the practitioner’s heuristic in all problem settings. The percentage gap of the two-level heuristic with the lower bound is less than 10% in all cases, and equal to 3.58% on average across multiple product settings. These results are encouraging given the complexity of the problem. Additionally, for the actual problem parameters, the simulated average cost of our two-level heuristic was 22.07% lower.
than the practitioner’s heuristic. Thus, this approach if used, has the potential to significantly reduce operational costs. The following additional observations can be drawn from the multi-product case:

- Under the practitioner’s heuristic, the average cost of operation is significantly higher when the cost and demand parameters are varied across products relative to when the parameters are similar across products. This is because the practitioner’s heuristic uses a rotation cycle in which the cycle length of all the products are the same. As the product parameters become more diverse, it becomes less sensible to have the same cycle length for all products. In contrast, the two-level heuristic is a dynamic policy designed specifically to take such variation in product parameters into account. However, even with the two-level heuristic, costs are slightly higher when products are more diverse because it becomes harder to reach a stable production pattern. Nevertheless, the gaps of the two-level heuristic with the lower bound are quite low for all problem settings.

- Similar to the single product case, the percentage gaps of the practitioner’s heuristic significantly increase when \( C_S \) is high. However, the two-level heuristic performs well because it dynamically controls all products to follow their optimal EPQ cycles as closely as possible. The EPQ cycle trades off \( C_S \) with \( C_I \) and \( C_B \) and it is robust to changes in the “production quantity”, which explains why slight deviations do not significantly increase the average costs as long as a good policy is in place to make the switching decisions.

The following managerial insights can be drawn from the computational analysis. These could also be useful for practitioners in similar industries:

1. Our relatively simple two-level heuristic nearly attains the optimal cost of the intractable stochastic decision process. The optimal cost is closely approximated by our simulation-based stochastic lower bound.

2. For minimizing the duration of a campaign of a fixed number of batches, the value of information, Bayesian learning, and dynamic decision making can be adequately captured by employing a re-optimization policy in conjunction with observing and learning.
3. When deciding on the next product to produce, we can choose the product with the greatest expected backlogging cost during the next two campaigns. This is somewhat similar to employing a one-step look-ahead policy, and our numerical results support its effectiveness.

4. When deciding whether or not to switch, the probability that the inventory of each product \( r \) falls below its respective threshold \( I^r \) provides an efficient summary of the intractable multidimensional state of the system. This can be used to make the important decision of when to change the catalyst and switch to the next campaign.

Note that our methods still require repeated Bayesian updating of the belief distribution on the catalyst parameter (this is needed for more accurate computations of \( E[t_{n+1}] \) and \( P[I_{n+1} < I] \)). However, such mathematical procedures seem amenable to implementation, given the ready availability of data from the process control system and tools from standard commercially available statistical software.

7 Conclusions

The problem of production campaign planning with uncertainty in production times, learning about production characteristics, and decay in catalyst performance is a challenging but important problem in a variety of process industry sectors such as food processing, pharmaceuticals, and specialty chemicals. We first considered the single product case and formulated it as an SMDP. To solve this problem, we developed a two-level heuristic. This heuristic splits the problem into a batch-planning problem and a catalyst-switching policy. The batch planning problem determines the production duration of each batch, given the fact that the quality of future batches depends on the decision made on earlier batches. Catalyst-switching defines a policy of when to stop the campaign and switch the catalyst. We present a practitioner’s heuristic, used at a large food-processing company, to benchmark the two-level heuristic. To assess the quality of this heuristic, we develop a lower bound to the SMDP by associating this process with another stochastic process with an infinitely recurrent state. The proposed framework of associating a continuous state space dynamic programming problem with a similar but regenerative process is applicable to other stochastic dynamic programming problems. We then considered the multiple product case. We modeled the relaxed deterministic approximation as a constrained economic lot sizing model, and used a
Lagrange relaxation to solve it. We were able to directly extend all the results and techniques of the single product problem to the multiple product setting.

The computational results for the single and multiple product problems show that our heuristics achieve low percentage gaps with the lower bound on the optimal average costs. In addition, our approach significantly outperforms the practitioner’s heuristic currently employed by a leading food-processing company. Furthermore, our heuristics are robust to changes in the cost parameters. This allows us to further simplify the proposed dynamic policy and present general and easy to implement operational guidelines for practitioners.

This paper opens up several avenues for future research. First, our model could be extended to the case with multiple reactors. Second, one could consider alternate quality models that may be required for meeting attribute quality levels. Third, these techniques could be applied to other industries that would require incorporating different types of production constraints. All these extensions might require significant modifications to the methods presented in this paper and could be fruitful areas for future work.

References


Appendix: Stochastic Lower Bound Algorithm

We refer to the original process as a “Type A” process and to the regenerative process as a “Type B” process. This simulation based algorithm is designed to compute a lower bound on the minimum average cost of a type B process given $I_0^*$. 

1. Set $\lambda$ to $\lambda_{EPQ}$ (the optimal cost of problem (11)).

2. Repeat the following simulation procedure until the simulated average cost seems to have converged:

   set up a campaign at inventory level $I_0^*$. Simulate a $b$ and a sequence of $z_i$s, then with full information on the outcome of $b$ and $z$, plan the campaign to minimize the differential cost of the current cycle. The decision variables are the number of batches $N$ and the attribute levels $q_i$, summarized in the vector $q$. Let $I(q) = I_0^* - \tau(q, b, z)d$ denote the inventory level after the campaign. The differential cost of the cycle is evaluated as follows:
(i) If $I(q) + N \leq I_0^*$, the differential cost is:

$$g(I_0^*, t_{I_0 I(q)}) - \lambda t_{I_0 I(q)} - g(I_0^*, t_{I_0 I'}) + \lambda_{EPQ} t_{I_0 I'}$$

where $t_{I_0 I(q)} = (I_0^* - I(q))/d$ is the time required for inventory to go from $I_0^*$ to $I(q)$. The term $-g(I_0^*, t_{I_0 I'}) + \lambda_{EPQ} t_{I_0 I'}$ is the cost of instantly raising the inventory level from $I(q)$ to $I_0^*$, from the definition of a type B process, and $t_{I_0 I'} = (I_0 - I(q) - N)/d$.

(ii) If $I(q) + N \geq I_0^*$, the differential cost is:

$$g(I_0^*, t_{I_0 I(q)}) - \lambda t_{I_0 I(q)} + \min_{I_0^* \leq I \leq I(q) + N} \{g(I, t_{II I_0}) - \lambda t_{II I_0}\}$$

where the term $\min\{g(I, t_{II I_0}) - \lambda t_{II I_0}\}$ comes from the definition of process B. It allows the inventory to drop instantly from $I(q) + N$ to any $I$ such that $I_0^* \leq I \leq I(q) + N$. In the optimal policy for process B, the chosen $I$ is one that minimizes the differential cost to return to $I_0^*$. It is easy to check that the differential cost increases with $N$ after $I(q^*) + N < 0$. Hence, the largest $N$ that needs to be considered is the first $N$ that satisfies $I(q^*) + N < 0$.

Once $q$ is chosen, record the cycle cost and cycle time (respectively equal to $g(I_0^*, t_{I_0 I(q)}) - g(I_0^*, t_{I_0 I'}) + \lambda_{EPQ} t_{I_0 I'}$ and $t_{I_0 I(q)}$ for case (i), and equal to $g(I_0^*, t_{I_0 I(q)}) + \min\{g(I, t_{II I_0}) - \lambda t_{II I_0}\}$ and $\lambda t_{I_0 I(q)}$ for case (ii)) to enable computing of the average total cost after each iteration and checking for convergence.

3. Update the value of $\lambda$ to the average cost computed in step 2, and return to step 2. Repeat this process until $\lambda$ converges. The resulting $\lambda$ is the optimal cost of a type B process.

To prove that this algorithm converges to a lower bound, note that the simulated process is an analog of process B, where the decisions are made with full information on $b$ and $z$. Therefore the optimal cost of this process is a lower bound on the optimal cost of process B. Additionally, this process is also regenerative, as the inventory level $I_0^*$ before campaign setup is a recurrent state, hence the value iteration algorithm converges to the optimal cost (Bertsekas (1995, vol. 2)). This algorithm is also effectively a value iteration algorithm, where the $\lambda$ at each iteration is evaluated using simulation.
A Proof of Propositions

Proposition 1. To prove that the constrained EPQ (11) provides a lower bound on the original SMDP (7), we show that a feasible solution to (11) exists with equal or less cost than the optimal average cost of the original process.

A cycle is defined from the end of one campaign to the end of the next campaign and includes any idle time. The average total costs of inventory holding, backlogging, and catalyst switching can be re-written as:

$$
\lambda = \lim_{T \to \infty} \frac{\int_0^T (I(w)^+ C_I + I(w)^- C_B + \delta_s(w)C_s)dw}{T} = T^+ \theta^+ C_I + T^- \theta^- C_B + \frac{C_S}{T_{cyc}}
$$

where $$I(w)$$ is the inventory level at time $$w$$, $$I(w)^\pm \triangleq \max[0, I(w)]$$, and $$I(w)^\mp \triangleq \max[0, -I(w)]$$. The function $$\delta_s(w)$$ has a unit impulse at every time $$w$$ where a switching occurs, and is zero for all other $$w$$. Further simplification results in the righthand side of (1), where $$T^+$$ ($$T^-$$) is the inventory level averaged over all instances where $$I(w) \geq 0$$ ($$I(w) < 0$$), and $$\theta^+$$ ($$\theta^-$$) is the fraction of total time where $$I(w) \geq 0$$ ($$I(w) < 0$$), and $$T_{cyc}$$ is the average cycle time.

Let the superscript * represent the optimal production strategy. We show that a fixed cycle strategy (following the EPQ formulation (11)) exists for which the optimal average cost is equal to or less than $$\lambda^* = \overline{T}^+ \theta^+ C_I + \overline{T}^- \theta^- C_B + \frac{C_S}{\overline{T}_{cyc}}$$. For this fixed cycle strategy, let $$\theta^+ = \theta^{+\ast}$$, $$\theta^- = \theta^{-\ast}$$, and $$\overline{T}_{cyc} = \overline{T}_{cyc}^\ast$$; i.e. the cycle always starts from inventory $$\overline{T}_{cyc}^\ast \theta^+ d$$ and finishes at inventory $$-\overline{T}_{cyc}^\ast \theta^- d$$, and a total of $$\overline{T}_{cyc}^\ast d$$ batches are produced in each campaign. The feasibility of the cycle length $$\overline{T}_{cyc}$$ in the deterministic constraint of problem (11) (i.e. $$\tau^*(T_{cyc} d) + t_s \leq \overline{T}_{cyc}$$) follows from the convexity of $$\tau^*(N)$$, Jenson’s inequality, and the fact that $$\tau^*(N)$$ is a lower bound on the expected campaign length for $$N$$ batches. It suffices to show that for this fixed cycle strategy $$\overline{T}^+ \leq \overline{T}^\ast$$ and $$\overline{T}^- \leq \overline{T}^{\ast \ast}$$.

For the optimal production strategy, assume a total of $$M$$ cycles and let $$M \to \infty$$. The total production time is $$M\overline{T}_{cyc}^\ast$$ and the total time spent in negative inventory levels is $$\theta^- M\overline{T}_{cyc}^\ast$$. Let the number of cycles that reach negative inventory be $$M' \leq M$$. Hence the average time spent in
negative inventory levels per negative turn is:

\[
T^- \triangleq \lim_{M \to \infty} \frac{\theta^* MT^*_cyc}{M'} \geq \theta^* T_{cyc}^- \tag{2}
\]

Assume that in the optimal strategy, all cycles start with a positive inventory level (the proofs for alternative cases follow with similar reasoning). The average negative inventory level per negative turn is \( \frac{T^- d}{2} \). This is not the average inventory level over the total time span of negative inventory; rather it is the average over the number of cycles that reach negative values. To obtain the per-time average, we must compute a weighted average over the cycles: the average negative inventory level of each cycle must be weighted proportional to the total amount of time that specific cycle spends in negative inventory. Notice that in this case, the cycles that reach larger (absolute) negative inventory levels will be weighted more heavily (because they must spend a longer time in negative inventory to reach that level) and cycles that end at smaller (absolute) negative inventory levels will be weighted less. The resulting \( \bar{T}^- \) will be greater than \( \frac{T^- d}{2} \) and thus \( \bar{T}^- \geq \frac{\theta^* T_{cyc} d}{2} \). Notice that for the fixed cycle strategy previously defined, \( T^- = \frac{\theta^* T_{cyc} d}{2} \). Hence, \( T^- \leq \bar{T}^- \). It follows with similar reasoning that \( \bar{T}^+ \leq T^+ \), which completes the proof.

**Proposition 2.** Let \( a \triangleq \frac{T}{T_{cyc} d} \) denote the proportion of \( T_{cyc} \) where the inventory level is positive. The maximum inventory during \( T_{cyc} \) is \( aT_{cyc} d \) and the average is \( aT_{cyc} d/2 \). The proportion of \( T_{cyc} \) where the inventory is negative is \( 1 - a \), with a maximum of \( (1 - a)T_{cyc} d \) and an average of \( (1 - a)T_{cyc} d/2 \) during this time. Hence, the average cost during the cycle time \( T_{cyc} \) becomes:

\[
\frac{C_s + (aT_{cyc})(aT_{cyc} d/2)C_I + [(1 - a)T_{cyc}][(1 - a)T_{cyc} d/2]C_B}{T_{cyc}}.
\]

Minimizing with respect to \( a \) gives \( a^* = \frac{C_B}{C_I + C_B} \). Substituting for \( a \) and arranging the terms, the objective function becomes

\[
\frac{C_s}{T_{cyc}} + \left( \frac{C_I C_B}{C_I + C_B} \right) \frac{T_{cyc} d}{2},
\]

which is a convex function of \( T_{cyc} \).

**Proposition 3.** We prove the existence of \( T = \arg\min\{w(I)\} \), where \( w(I) \) was defined in (7).
Observe that

(i) \( w(I) \) grows unboundedly as \( I \to \pm \infty \), and

(ii) \( w(I) \) is bounded below.

To prove that \( w(I) \) is bounded below, assume that for some \( I = I_{-\infty} \) we have \( w(I_{-\infty}) = -\infty \).

Because we assume that the highest achievable average production rate exceeds the demand rate, we are able to reach \( I_{-\infty} \) from any initial state with probability 1 in finite time by producing at the highest rate until inventory goes over \( I_{-\infty} \), then allowing idle time until inventory drops to \( I_{-\infty} \).

For all states \( I \) we would have \( w(I) = -\infty \), which is a contradiction.

\[ \square \]

**Proposition 4.** If the process reaches state \( T^* \) and production has stopped (the next campaign has not yet been set up), the only state variable determining the decision will be the inventory level \( T^* \). The optimal decision is either to set up the next campaign immediately (in which case \( I_0^* = T^* \)) or to idle the process until the inventory level reaches a certain value which we call \( I_0^* \). \( I_0^* \) is constant because the decision at inventory level \( T^* \) when the process is idle is independent of the history of the process.

\[ \square \]

**Proposition 5.** For the purpose of this proof, we refer to the original process as a “Type A” process and to the regenerative process as a “Type B” process. Define a process of type B that during a campaign, follows the same batch planning and catalyst switching decisions as in the optimal policy of process A. Once the campaign is ended, if the inventory \( I \) after a campaign is below \( I_0^* \), raise it instantly to \( I_0^* \) at a cost of \( -g(I_0^*, I_{-\infty}) + \lambda_{EPQ} I_{-\infty} \). If \( I_0^* \leq I \leq T^* \), idle the process till it reaches inventory \( I_0^* \) and then set up the next campaign. If \( I > T^* \), instantly decrease \( I \) to \( T^* \) and idle the process till it reaches \( I_0^* \). We show that the process B defined here has a lower average cost than the optimal process A.

Let \( w^* \) denote the differential costs of the optimal policy of process A (as defined in (7)), and let \( \lambda^* \) be the optimal cost of process A.

(i) Assume that in the optimal process A, inventory level reaches \( I \leq I_0^* \) after a campaign. Noting that the optimal policy would not idle the process from \( I_0^* \) to \( I \) (by the definition of \( I_0^* \)), we
have:

\[ w^*(I_0^*) \leq g(I_0^*, t_{I_0}) + w^*(I) - \lambda^* t_{I_0} \]
\[ \Rightarrow w^*(I) \geq w^*(I_0^*) - g(I_0^*, t_{I_0}) + \lambda^* t_{I_0} \]
\[ \Rightarrow w^*(I) \geq w^*(I_0^*) - g(I_0^*, t_{I_0}) + \lambda_{EPQ} t_{I_0} \]

where the last inequality follows from the fact that \( \lambda_{EPQ} \) is a lower bound to the optimal cost. The left-side of this inequality is the optimal ongoing differential cost of process A and the right side is the ongoing differential cost of the defined type B process.

(ii) Alternatively assume that \( I > I_0^* \). If \( I \leq I_0^* \), then the optimal decision in process A is to idle the process to reach inventory level \( I_0^* \) and set up the next campaign at \( I_0^* \), same as in the defined process B. If \( I > I_0^* \), then by definition of \( I_0^* \) we know that \( w^*(I) \geq w^*(I_0^*) \). Since the type B process instantly decreases from \( I \) to \( I_0^* \), its ongoing differential cost will be less than or equal to that of process A.

Hence, regardless of where the process begins, the ongoing differential cost of the defined process B is less than or equal to that of the optimal process of type A. 

**Proposition 6.** The objective is to minimize \( E_b[\tau(q, b)] \) over \( q \) of known length \( N \). For \( i \in \{2, 3, ..., N\} \), let \( Q_{i+1}^* \) be the value in the optimal \( q \). Note that \( Q_{N+1}^* = N \). Define \( R \) as the total attribute level that must be met by \( q_i \) and \( q_{i-1} \).

\[ R \triangleq Q_{i+1}^* - Q_{i-1}^* = q_i + q_{i-1}. \]  

(3)

The optimal allocation of \( R \) between \( q_i \) and \( q_{i-1} \) solves:

\[
\begin{align*}
\min & \quad \psi(q_{i-1}, q_i) = E_{b,z_1,z_2}[(b+z_1)f(q_{i-1})k(T_{i-1}) + (b+z_2)f(q_i)k(T_i)] \\
\text{s.t.} & \quad q_i + q_{i-1} = R \\
& \quad T_i = T_{i-1} + (b+z_i)f(q_{i-1})k(T_{i-1}).
\end{align*}
\]  

(4)

For notational simplicity let \( b_i \triangleq b + z_i \) (which implies \( E_{z_i}(b_i) = b \)). Let \( q > q' > 0 \) such that \( q + q' = R \), and let \( \Delta T \triangleq b_1 k(T)f(q) \) and \( \Delta T' \triangleq b_1 k(T)f(q') \); then \( \Delta T' > \Delta T \). For any \( b \geq 0 \) we
have:

\[ \psi(q, q') \leq \psi(q', q) \]

\[ \iff E_z[b_1k(T)f(q) + b_2k(T + b_1k(T)f(q))f(q')] \leq E_z[b_1k(T)f(q') + b_2k(T + b_1k(T)f(q')))f(q)] \]

\[ \iff b_f(q')(E_{b_1}[k(T + \Delta T') - k(T)]) \leq b_f(q)(E_{b_1}[k(T + \Delta T') - k(T)]) \]

\[ \iff \frac{E_{b_1}[k(T + \Delta T')] - k(T)}{b_f(q)} \leq \frac{E_{b_1}[k(T + \Delta T') - k(T)]}{b_f(q')} \] \quad (5)

Convexity of \(k(T)\) is sufficient for the last inequality to hold, because

\[ \Delta T' > \Delta T \text{ and } k(T) \text{ is convex} \implies \frac{k(T + \Delta T) - k(T)}{\Delta T} \leq \frac{k(T + \Delta T') - k(T)}{\Delta T'} \quad \forall b_1 \geq 0 \]

\[ \implies \frac{k(T + \Delta T) - k(T)}{b_1k(T)f(q)} \leq \frac{k(T + \Delta T') - k(T)}{b_1k(T)f(q')} \quad \forall b_1 \geq 0 \]

\[ \implies \frac{k(T + \Delta T) - k(T)}{b_f(q)} \leq \frac{k(T + \Delta T') - k(T)}{b_f(q')} \quad \forall b_1 \geq 0 \]

\[ \implies \frac{E_{b_1}[k(T + \Delta T)] - k(T)}{b_f(q)} \leq \frac{E_{b_1}[k(T + \Delta T') - k(T)]}{b_f(q')} \] \quad (6)

Hence

\[ k(T) \text{ convex and } q > q' \implies \psi(q, q') \leq \psi(q', q) \] \quad (7)

Assume that in the optimal solution to (4) for convex \(k(T), q_{i-1} > q_i\). Then there exists an equally good or better solution by exchanging \(q_{i-1}\) and \(q_i\). Hence if \(k(T)\) is convex, there is always an optimal solution in which \(q_{i-1} \leq q_i\). If \(k(T)\) is strictly convex, the inequalities in (5) become strict inequalities, which proves there are no optimal solutions where \(q_{i-1} > q_i\). This proves part (i).

Similarly, for concave \(k(T)\), the direction of the inequalities in (5) are reversed which proves part (ii). Part (iii) is straightforward: replace \(k(T)\) by \(a + cT\) and take the derivative w.r.t. \(q\).

**Proposition 7.** By using a change of variables \(N_r^{-1} = 1/N_r\), we obtain a convex optimization problem over the variables \(N_r^{-1}\) (note that if \(f(x)\) is convex, then \(xf(x^{-1})\) is convex over positive values of \(x\)). The convex set defined by the constraint has an interior point because the maximum achievable production rate is strictly greater than the demand rate. Hence Slater’s condition holds for the problem over \(N_r^{-1}\) and strong duality holds. The solutions to the primal and dual problems
are unaffected by the change of variables from $N_r$ to $N_r^{-1}$, hence strong duality also holds for the original problem (20) over $N_r$ ■

B 3-Batch Example for Sufficiency of Re-optimization Policy

We consider a 3-batch campaign ($N = 3$) and sequentiall choose $q_1$, $q_2$, and $q_3$ to minimize the total duration of the campaign. We consider two planning scenarios: in the first scenario, a policy $q = [q_1, q_2, q_3]$ is computed as $q = \arg \min_{q} E_{b,z} \{\tau(q, b, z)\}$, of which $q_1$ is executed, then $[q_2, q_3]$ is re-optimized after observing $t_1$. In the second scenario, $q_1$, $q_2$, and $q_3$ are sequentially chosen as the optimal solutions to the dynamic program (15). In the first scenario, in choosing $q_1$ we do not anticipate the fact that we are able to choose $[q_2, q_3]$ based on more accurate information, whereas in the second scenario we solve a dynamic program that accounts for all outcomes of $t_1$. To evaluate the most extreme-case difference between the two scenarios and to enable tractability of the dynamic program, we assume that full information becomes available once we observe $t_1$. That is, we assume that $b$ is initially unknown, but is learned through $f^{-1}(t_1/k(0))$ once $t_1$ is observed (there are no random shocks to hinder the learning process). With these settings, the dynamic program (15) converts to a two-stage formulation as follows:

$$v_2(q_1, t_1) = \min_{q_2} [bk(t_1)f(q_i) + bk(t_1 + t_2)f(3 - q_1 - q_2)],$$

where $b = f^{-1}(t_1/k(0))$ and $t_2 = bk(t_1)f(q_i)$,

$$v_1(0, 0) = \min_{q_1} [E_b(bk(0)f(q_1) + v_2(q_1, bk(0)f(q_1)))].$$

A numerical solution to (8) is computed by discretizing and enumerating $q_1$, $q_2$, and $b$, which evaluates the expected campaign length under scenario 2 and also allows computing $q_1^*$. The first scenario is also evaluated by enumerating $q_1$, $q_2$, and $b$, but here instead of $q_1$ being the optimal solution to (8), it is the first element in $q = \arg \min_{q} E_{b} \{\tau(q, b)\}$ (denoted by $q_1^*$). Additionally, the expected optimal campaign duration under apriori knowledge of $b$ (i.e. a clairvoyant decision maker) is evaluated using $E_b(\min_{q}\{\tau(q, b)\})$ (also using enumeration).

We may now compare the two policies by computational experiments. We let $f(q) = -\log(q/2)$, and in order to compare the policies under a wide range of settings, we consider three classes of functions for the catalyst decay function $k(T)$:
(i) \(k(T) = a(T + 1)^p\),
(ii) \(k(T) = a \log(T + 2)\),
(iii) \(k(T) = a \exp(T)\).

For class (i), we alter the scale parameter \(a\) within the range \([0.1, 100]\) and the power parameter \(p\) within the range \([0.2, 3]\). For classes (ii) and (iii), we alter the scale parameter \(a\) within the ranges \([1, 100]\) and \([0.1, 2]\) respectively. In all our test problems, we assume a normal prior on \(b\) with mean 1 and standard deviation 0.2. We consider 220 problem settings in total, and make the following observations, which are also summarized in §4.1 and used to justify the re-optimization policy.

1. Considering the fact that we enumerate \(q_1\) in increments of 0.02 in the range \([0.02, 2]\), \(q_1^*\) and \(q_1^s\) coincide in 209 out of 220 problem instances, and have at most 0.04 difference in the remaining 11 instances.

2. In all 220 problem instances the expected campaign time under the re-optimization policy is within 1% of the expected campaign time under the optimal policy.

3. In all 220 problem instances the expected campaign time under both policies is within 2% of the expected optimal campaign time under apriori knowledge of \(b\).

C Algorithm for solving PH

In the practitioner’s heuristic for the multi-product case, the problem PH is approximately solved by iterating over \(t_r^*\) of one product. Choose any one of the \(R\) products and index it by \(\hat{r}\).

Algorithm.
1. Initiate \(\hat{t} = t_0\) and \(\delta := \text{step size}\).
2. Let \(t_{\hat{r}}^* = \hat{t}\).
3. Let \(N_{\hat{r}} = \max[N] \ s.t. \sum_{n=1}^{N} f^{-1}(\frac{t_{\hat{r}}^*}{k(nt_{\hat{r}}^*)\mu_{\hat{r}}}) \leq N\).
4. Let \(L = N_{\hat{r}}/d_{\hat{r}}\).
5. Let \(N_r = \lceil Ld_r \rceil\) for all other \(r\).
6. Let \(t_r^* = \min[t_r] \ s.t. \sum_{n=1}^{N} f^{-1}(\frac{t_r^*}{k(nt_{\hat{r}}^*)\mu_{r}}) \leq N_r\) for \(r \neq \hat{r}\).
7. If the variables chosen in steps 2-6 meet the constraints (24a)-(24c), evaluate \(\hat{t}\) by \(V(\hat{t}) = \frac{\sum_{r=1}^{R} (N_r C_{Ir} + C_S)}{L} \). Otherwise, \(V(\hat{t}) = \infty\).
8. \(\hat{t} + \delta \to \hat{t}\). If \(\hat{t} > \frac{1}{d_{\hat{r}}}\), go to 9; otherwise go to 2.
9. Let $t^*_i = \arg \min V(\hat{t})$, choose the rest of the variables by the procedure in steps 3-6, and terminate.

To understand the assignments in steps 4 and 5 of the Algorithm, observe that in the optimal solution to PH, the equations $L = \min \frac{N_r}{d_r}$ and $N_r = \lceil L d_r \rceil$ will hold: if $L < \min \frac{N_r}{d_r}$, $L$ can increase without violating the constraints, reducing the objective function. If $N_r > \lceil L d_r \rceil$, $N_r$ can decrease to reduce the objective function. Both cases contradict optimality. Therefore, the assignment in steps 4 and 5 satisfies the mentioned equations.