Dynamic Release Management: A Market Intensity Approach

Daniel Adelman, Angelo J. Mancini
University of Chicago Booth School of Business, Chicago, IL 60637
{dan.adelman@chicagobooth.edu, amancini@chicagobooth.edu}

We consider a release manager who sequentially releases new versions of her product by drawing from a fixed and non-replenishable finite set of features while facing an exogenous, stochastically evolving marketplace. In the absence of fixed costs, we provide conditions under which there exists a quasi-open-loop optimal policy, i.e., an optimal policy that depends only on the set of available features and not the state of the external market. Computing such a policy amounts to solving a related deterministic release-sequencing problem, and we apply an exchange argument to obtain an index condition necessary for optimality. In the simplest case we consider, a heuristic based on this condition reduces to an optimal ordering of Gittins indices and is equivalent to the “weighted discounted shortest processing time first” stochastic scheduling rule. However, we prove it is suboptimal by an arbitrarily large margin in other settings. We apply approximate dynamic programming (ADP) to address the case with positive fixed costs by making a novel value function approximation motivated by the case without fixed costs. The resulting policy can provably outperform an intuitive certainty-equivalent heuristic by an arbitrarily large margin, and performs within 3.5% of optimality across a range of numerical trials.

Key words: product development; release management; semi-Markov decision processes; approximate dynamic programming

History: First version: May 30, 2014; Latest version: December 30, 2014

1. Introduction  A profit-maximizing firm must regularly release new versions of its product to provide new functionality to consumers. The managers issuing these releases must balance several tradeoffs. Ideally, the firm would instantaneously issue releases with all available functionality. In reality, features take time to implement, and implementation time often increases in the desirability of the feature. Longer development cycles not only postpone the realization of revenues, but also increase the likelihood that changes in the marketplace will diminish the release’s value. In addition, managers must consider the fixed cost associated with issuing a release, e.g., testing, training and marketing expenses. The goal of this paper is to provide release managers with a tactical model that negotiates these tradeoffs. Our motivation comes from the software industry, but extends to any sector in which firms must continually release updated products.

The existing product development literature typically models product quality as a real-valued scalar, multi-dimensional vector, or with a binary “high/low” classification. However, in software development releases consist of sets of discrete features that often can not be easily prioritized. We therefore represent the release manager’s problem as a combinatorial release-sequencing problem embedded within a semi-Markov decision process (SMDP) that accounts for a stochastic marketplace. To the best of our knowledge, no such formulation appears in either the product development or project scheduling literatures.

By casting the release management problem as a special case of the SMDP in Adelman and Mancini (2014), we decompose the problem into the exogenous market dynamics over which the release manager exerts no influence, and the endogenous release-sequencing dynamics entirely within her control. This approach has yet to appear in the literature, and illustrates how to enrich a sequencing problem by accounting for an exogenous marketplace without sacrificing tractability. In
order to obtain this decomposition, we characterize the external marketplace in terms of its overall intensity. As the market becomes more intense over time, the release manager’s product generates lower revenues unless new functionality is added to compensate. Possible drivers of market intensity include changing consumer preferences, increases in the utility of consumers’ best outside option, changing macroeconomic factors, or some combination of these forces. Formally, market intensity evolves as a nondecreasing scalar-valued multiplicative compound Poisson processes that is uninfluenced by the release manager’s product or actions. While the compound Poisson model admittedly oversimplifies market dynamics, its parsimony eases the burden of applying our results. Based on discussions with managers at a large software firm, it is also consistent with the amount of market information normally available to tactically-focused release managers. If market shocks induce geometric decay in the rate at which revenues accumulate to the release manager’s firm, and there are no fixed development costs, Adelman and Mancini (2014) reduce the SMDP to a deterministic dynamic program that accounts for market stochasticity with a modified discount factor. This discount factor depends only on the statistical properties of the external market, and increases in both the average frequency and expected magnitude of market shocks. The solution to this dynamic program is an optimal release sequence that, without loss of optimality, the release manager can implement without monitoring the evolution of the marketplace. In this paper, we leverage the combinatorial structure of the release management problem to analyze the reduced dynamic program furnished by Adelman and Mancini (2014) for the case with no fixed costs. We then apply our insights to the case with positive fixed costs, which lies outside the scope of Adelman and Mancini (2014).

When there are no fixed costs, we show that the endogenous dynamics of the release-sequencing problem and the exogenous dynamics of the stochastic marketplace are coupled via a set of optionality multipliers. For any subset of undeveloped features, the optionality multiplier encodes the value of continuing to issue new releases as opposed to permanently stopping development. We prove that these multipliers decrease in the intensity of the marketplace, and that the multiplier associated with a set of undeveloped features exceeds that of any of its subsets when the incremental value of new features decreases in the amount of functionality already present in the release manager’s product.

We then analyze the optionality multipliers to obtain an index ordering condition necessary for the optimality of a release sequence. This index is of interest even in the absence of an external marketplace, as it exhibits two unconventional properties: path dependence and nesting of feature subsets. Path dependence requires knowledge of the full release history when calculating the index to determine the next release in the sequence. Nesting refers to the fact that once a feature is developed, the release manager can not issue any releases containing that feature in the future. While this index coincides with the Gittins index in a specialized version of our problem, we show that a greedy heuristic based on this index can perform arbitrarily poorly in other settings.

When the release manager incurs positive fixed costs for issuing a release, much of our analysis breaks down. Instead, we take a novel approach to approximate dynamic programming (ADP) and approximate the value function with a decomposition motivated by the optionality multipliers identified in the case without fixed costs. After inserting this approximation into an infinite dimensional linear program representing the SMDP, we reduce the linear program to a finite dimensional linear program by exploiting the same assumptions applied in the case without fixed costs. By solving this linear program, we obtain a policy bound and an approximate policy that performs within 3.5% of optimality across all of our numerical trials. Finally, we also prove that the approximate policy can outperform a certainty-equivalent heuristic by an arbitrarily large margin. This paper is the first to apply the linear programming approach to ADP in the context of product development, and our technique illustrates how to extend the methods from Adelman and Mancini (2014) to sequencing problems with fixed costs.
We begin with a brief literature review in §2. We formulate our model in §3 and present assumptions in §4. Section 5 considers the case with no fixed costs, including the key value function decomposition (Theorem 2) and quasi-open-loop results (Theorem 3), as well as the novel optionality results and index-based analysis of the optimal release sequence. In §6 we apply approximate dynamic programming to the case with fixed costs. Extensions and avenues for further research are discussed in §7.

2. Literature Review The release management problem most closely resembles work in the new product development literature. Table 1 categorizes a sample of ten relevant articles along the following five dimensions: the number of simultaneous product generations allowed in the market, the number and strategic nature of the firm’s rivals, the method for parameterizing product quality, the type of consumer demand model employed by the authors, and whether or not product pricing is exogenous or endogenous. Several of these articles allow for the coexistence of multiple generations of the same product in the marketplace, whereas we omit this possibility. When interpreted as the consumer utility associated with a rival product, our model of market intensity as an exogenous shock process is a compromise between the “one-static” and “one-strategic” formulations displayed in Table 1, and provides a reasonable tradeoff between tractability and realism given the novel combinatorial complexity in our model. Regarding consumer demand and pricing, we require only a generic mapping that calculates the firm’s revenue accrual rate as a function of market intensity and the features in the release manager’s product. Our key departure from the product development literature is that we model the firm’s product as a collection of discrete features.

In the project scheduling literature, a firm aims to schedule a set of projects with either known or random duration times in order to maximize the net present value of payments received upon completion. Examples include Elmaghraby and Herroelen (1990), Herroelen and Gallens (1993), Reyck and Herroelen (1998), and Vanhoucke et al. (2003). In certain cases, project scheduling corresponds to the single machine stochastic scheduling problem with total weighted discounted completion time objective. In fact, the “weighted discounted shortest processing time first” (WDSPT) rule found in the stochastic scheduling literature (Pinedo (2012)) solves a specialized instance of our problem. We contribute to the project scheduling and stochastic scheduling literatures by accounting for an exogenous, stochastically evolving marketplace that impacts the firm’s revenues.

Finally, we have adapted the SMDP formulation from Adelman and Mancini (2014) to the release management problem. The exogeneity assumptions in §4.2 and the multiplicative separability assumption in §4.3 come directly from the more general paper. Those assumptions lead directly to the modified discount factor identified in Proposition 1, and form the foundation for the value function decomposition derived in Theorem 2 and associated quasi-open-loop optimal policy identified in Theorem 3. However, all analysis of the optionality multipliers in §5.1 and the optimal release sequence in §5.2 and §5.3 is novel.

3. Semi-Markov Decision Process Formulation  
In this section, we describe the release management problem, formalize its transition structure, and develop the optimality equations of the associated SMDP.

3.1. Description  
At the beginning of the horizon, let \( F_0 \) denote the features already included in the release manager’s product, \( F_0^A \) the finite set of all features available for development in future releases, and \( x_0 \in \mathbb{R}_+ \) the initial intensity of the marketplace. We define the state space of our SMDP as \( \mathcal{S} := \{(x, F_A) \mid x \in [x_0, \infty), F_A \subseteq F_A^0\} \). For each state \( (x, F_A) \in \mathcal{S} \), the set of actions \( \phi \) available to the release manager consists of all subsets of features that could be included in the next release. We define the set of permissible state-action pairs as \( \mathcal{K} := \{(x, F_A) \mid (x, F_A) \in \mathcal{S}, \phi \subseteq F_A^0\} \), and take as given a deterministic mapping \( \tau : 2^{F_0^A} \rightarrow \mathbb{R}_+ \) that determines the required development time for any collection of features. Our assumption that development times are deterministic eases exposition and does not restrict our analytical results. However, the time required to develop a bundle of features could depend on the set of features already in the existing product. For instance, features that simplify the software’s code may accelerate development of future features. We discuss this extension in §7.

Suppose that the release manager is in state \( (x, F_A) \) and takes the action \( \phi \subseteq F_A \). She immediately incurs a fixed, finite development cost \( c \in \mathbb{R}_+ \) if \( \phi \neq \emptyset \), and otherwise incurs no fixed cost. While the bundle \( \phi \) is under development, market intensity increases stochastically in continuous-time. If \( F_t \) denotes the set of features already included in the release manager’s product, then at any time \( u \in [0, \tau(\phi)) \) after initiating development of the bundle \( \phi \), revenue accumulates at the rate \( e^{-\beta u} g(X(u), F_t) \) for some measurable function \( g : \mathbb{R} \times 2^{F_0^A} \rightarrow \mathbb{R}_+ \) and discount factor \( \beta > 0 \). The next decision epoch occurs when development of the bundle \( \phi \) is complete, at which time the set of available features reduces to \( F_A \setminus \phi \) and the release manager immediately replaces her existing product with the set \( F_t \cup \phi \). The release manager seeks to implement a policy that maximizes her expected discounted profits over an infinite horizon.

We emphasize five key properties of our model. First, the set \( F_A^0 \) contains all features available for development over the entire horizon, and no new features arrive over time. Second, the release manager can not remove previously released features from her product. These two assumptions imply that if the set of available features is \( F_A \subseteq F_A^0 \), then the set of features already included in the release manager’s product must be \( F_t = F_t^0 \cup (F_A^0 \setminus F_A) \). We use the notation \( F_t \) throughout the paper, with its definition clear from context. Similarly, we sometimes take the set \( F_t \) as given and instead infer that \( F_A = F_A^0 \setminus F_t \). Third, the release manager can not pause development and change the set of features slated for the next release. Once she chooses to develop a bundle of features, she makes no further decisions until that bundle is completed and released to the market. In addition, we do not allow multiple generations of the release manager’s product to exist simultaneously in the market. Finally, our model embeds the release manager’s pricing strategy in the instantaneous revenue function \( g \) and implicitly assumes that price is a deterministic function market intensity and the features in the release manager’s product.
3.2. Formal Transition Structure  As much of our analysis depends on the inter-epoch evolution of the marketplace and its interaction with the release manager’s product and decisions, we now formalize the transition structure of the release management SMDP.

Recall that the local index $u$ measures time relative to the most recent decision epoch. We introduce the following objects:

- a probability space $(\Omega, \Sigma, P)$,
- a $\mathbb{R}^+_\infty$-valued stochastic process $\{X (u) : u \geq 0\}$ representing the intensity of the external market,
- a $2^{F^\phi_A}$-valued stochastic process $\{F_A (u) : u \geq 0\}$ representing the set of features available for development,
- a $2^{F^I_\infty \cup F^\phi_0}$-valued stochastic process $\{F_I (u) : u \geq 0\}$ representing the set of features already included in the release manager’s product, and
- a $2^{F^\phi_A}$-valued random variable $A$ representing the release manager’s most recent action.

The space $\Omega$ is local in the sense that if the decision maker takes an action $\phi$ in state $(x, F_A)$, her observations until the next decision epoch consist of a realization of the random variable $\{(X (u), F_A (u)) : 0 \leq u \leq \tau (\phi)\}$ subject to the conditional probability measure $P (\cdot | (X (0), F_A (0)) = (x, F_A), A = \phi)$. The local time index $u$ is reset to zero at each decision epoch, and all objects defined over $\Omega$ are reinitialized. As discussed in §3.1, the release manager’s product only changes at decision epochs. Hence, if the release manager takes the action $\phi$ while in state $(x, F_A)$, then $F_A (u) = F_A$ and $F_I (u) = F_I$ for all $u \in [0, \tau (\phi))$. Furthermore, $F_A (\tau (\phi)) = F_A \setminus \phi$ and $F_I (\tau (\phi)) = F_I \cup \phi$, which represents the completion of the bundle $\phi$ after a development cycle of duration $\tau (\phi)$.

3.3. Optimality Equations We apply the contraction mapping approach in Puterman (2005) to develop the optimality equations for our SMDP and characterize an optimal stationary deterministic policy. We require two technical assumptions. The first prohibits instantaneous transitions between decision epochs, and identifies the action $\phi = \emptyset$ with the decision to wait for a fixed amount of time. The second assumption requires that, in addition to being nonnegative, the instantaneous revenue function $g$ is also bounded.

**Assumption 1.** There exists a scalar $w \in (0, \infty)$ such that $\tau (\emptyset) = w \leq \tau (\phi) < \infty$ for all feature bundles $\phi \subseteq F^0_A$.

**Assumption 2.** The measurable function $g$ is nonnegative on $\mathbb{R} \times 2^{F^0_I \cup F^0_A}$, and there exists a constant $M \in [0, \infty)$ such that $g \leq M$.

We denote the value function for our SMDP by $V^\phi$. For any measurable mapping $d : \mathcal{S} \to 2^{F^\phi_A}$ such that $d (x, F_A) \subseteq F_A$ for all $(x, F_A) \in \mathcal{S}$, we denote the associated stationary deterministic policy by $d^\infty$ and its value by $V^{d^\infty}$. Without loss of generality, we now focus on characterizing an optimal stationary deterministic policy, i.e., a policy $d^\infty$ such that $V^{d^\infty} = V^\star$.

Let $\mathcal{V}_b$ denote the space of real-valued, bounded, measurable functions on $\mathcal{S}$ equipped with the supremum norm $\| \cdot \|_{\infty}$. We define the *inter-epoch reward* $r$ as follows: for all $((x, F_A), \phi) \in \mathcal{S}$,

$$
\begin{align*}
  r ((x, F_A), \phi) &:= -c \cdot 1_{\{\phi \neq \emptyset\}} + \\
  &\quad E \left[ \int_0^{\tau (\phi)} e^{-\gamma u} g (X (u), F_I) du \bigg| (X (0), F_A (0)) = (x, F_A), A = \phi \right],
\end{align*}
$$

(1)

with $E$ denoting expectation with respect to $P$ on $\Omega$. Consider the following operator $\mathcal{L}$ defined on $\mathcal{V}_b$: for all $v \in \mathcal{V}_b$ and all $(x, F_A) \in \mathcal{S}$,

$$
\begin{align*}
\mathcal{L} v (x, F_A) &:= \max_{\phi \subseteq F_A} \left\{ r ((x, F_A), \phi) + \\
&\quad e^{-\beta \tau (\phi)} E \left[ v (X (\tau (\phi)), F_A \setminus \phi) \bigg| (X (0), F_A (0)) = (x, F_A), A = \phi \right] \right\}.
\end{align*}
$$

(2)
Our assumption that $F^0_A$ has finite cardinality justifies the use of the max operator, and ensures that $Lv$ is measurable on $\mathcal{S}$ for all $v \in \mathcal{V}_b$ (Proposition D.5 in Hernández-Lerma and Lassere (1996)). Furthermore, $Lv$ is bounded on $\mathcal{S}$ since $c$ is finite and $g$ is bounded by Assumption 2. Hence, the range of $L$ is $\mathcal{V}_b$. Assumption 1 and the fact that $\beta > 0$ therefore imply that $L$ is a contraction operator on $\mathcal{V}_b$. The following standard result thus follows from Puterman (2005):

**Theorem 1.**

(i) The value function $V^*$ is the unique solution in $\mathcal{V}_b$ of the optimality equations $v = Lv$.

(ii) For any $v \in \mathcal{V}_b$, if $\mathcal{L}^{n+1}v = \mathcal{L}(\mathcal{L}^n v)$ for all $n \in \mathbb{N}$, then $\|\mathcal{L}^n v - V^*\|_\infty \to 0$.

(iii) If there exists a mapping $d_*: \mathcal{S} \to 2^{F_\beta}$ such that, for all $(x, F_A) \in \mathcal{S}$,

$$d_*(x, F_A) \in \arg \max \{r((x, F_A), \phi) + e^{-\beta r(\phi)} E[V^*(X(\tau(\phi)), F_A \setminus \phi) | (X(0), F_A(0)) = (x, F_A), A = \phi] \} \tag{3}$$

then the stationary deterministic policy $d^\infty_*$ is optimal.

By part (iii) of Theorem 1, the assumption that $F^0_A$ has finite cardinality immediately implies the existence of a decision rule $d_*$ that satisfies $V^{d^\infty_*}_A = V^*$.

4. Assumptions  The following assumptions hold throughout the paper.

4.1. Initial Parameters and Revenues  The first assumption is self-evident, as a product with no features generates no revenues:

**Assumption 3.**  For all $x \in [x_0, \infty)$, $g(x, \emptyset) = 0$.

In order to avoid division by zero later in the paper, we impose the following condition:

**Assumption 4.**  The release manager’s initial product is nonempty, i.e., $F^0_I \neq \emptyset$.

The next assumption describes the interaction between market intensity and the revenues earned by the release manager’s firm.

**Assumption 5.**  For any pair of sets $F^1_I$ and $F^2_I$ such that $F^1_I \subset F^2_I \subseteq (F^0_I \cup F^0_A)$, the difference $g(\cdot, F^1_I) - g(\cdot, F^2_I)$ is strictly positive and nonincreasing on $[x_0, \infty)$.

This assumption formalizes the intuition that a product with more functionality should generate more revenue than a product with less functionality. However, the marginal value of additional functionality presumably decreases as the marketplace becomes more intense due to improvements in rival products, increased consumer disaffection with the product category, etc. Observe that taking $F^1_I = \emptyset$ in Assumption 5 implies that $g(\cdot, F_I)$ is both strictly positive and nonincreasing on $[x_0, \infty)$ for all nonempty subsets $F_I$ of $F^0_I \cup F^0_A$.

4.2. The Market Intensity Process  We now impose structure on the market intensity process, and describe its relationship to the release manager’s product and actions. Relaxation of these assumptions are discussed in §7.

**Assumption 6.**  There exists a Poisson process $N$ with rate $\lambda \in (0, \infty)$ and a sequence of i.i.d. shocks $\{\xi_i\}_{i \in \mathbb{N}}$, both defined on $(\Omega, \Sigma, P)$, such that $\xi_i : \Omega \to [1, \infty)$ for all $i \in \mathbb{N}$ and

$$X(u) := X(0) \prod_{i=1}^{N_u} \xi_i \quad \forall u \geq 0. \tag{4}$$
The events of the process $N$ correspond to increases in marketplace intensity, and the shocks $\{\xi_i\}_{i \in \mathbb{N}}$ represent the magnitudinal changes in intensity. For instance, if intensity represents the quality of a rival product, a shock of magnitude 1.05 suggests a five percent increase in the consumer utility associated with a new release of that product. Since the range of the shock variables is $[1, \infty)$, we are implicitly assuming that the market only becomes more intense over time. To avoid a trivial scenario, we also impose two additional conditions. The first avoids a market that never evolves, and the second ensures that market intensity is not fixed at zero given the multiplicative structure in (4).

**Assumption 7.** For all $i \in \mathbb{N}$, $P(\xi_i = 1) < 1$.

**Assumption 8.** The initial market intensity level is strictly positive, i.e., $x_0 > 0$.

We now describe the interaction between the shock-time process $N$ and the shocks $\{\xi_i\}_{i \in \mathbb{N}}$.

**Assumption 9.** For all $u \geq 0$ and all $i \in \mathbb{N}$, the following conditions hold on $(\Omega, \Sigma, P)$:

(i.) $N_u$ and $X(0)$ are independent,

(ii.) $\xi_i$ and $X(0)$ are independent, and

(iii.) $\xi_i$ and $N_u$ are independent.

Under Assumption 9, neither the shocks nor the process $N$ can depend on the market intensity at the beginning of a decision epoch. Furthermore, part (iii) implies that the magnitude of a shock can not depend on the length of time between successive shocks.

The next assumption characterizes the interaction between market intensity and the release manager’s state and actions.

**Assumption 10.** For all $i \in \mathbb{N}$ and all $u \geq 0$, the following conditions hold on $(\Omega, \Sigma, P)$:

(i.) $\xi_i$ and $A$ are independent,

(ii.) $\xi_i$ and $F_A(u)$ are independent,

(iii.) $N_u$ and $A$ are independent, and

(iv.) $N_u$ and $F_A(u)$ are independent.

Parts (i) and (iii) imply that market intensity evolves independently of the release manager’s actions. These conditions reflect the fact that the external market is likely not privy to the release manager’s internal product development decisions. However, conditions (ii) and (iv) also require that the market intensity process is not impacted by the release manager’s existing product. We discuss the role of these conditions in §7. Finally, Assumption 10 permits us to condition on only the intensity level at each epoch in both the definition (1) of the inter-epoch reward $r$ and the optimality equations: for all $((x, F_A), \phi) \in \mathcal{X}$,

$$ r((x, F_A), \phi) := -c \cdot 1_{\{\phi \neq \emptyset\}} + E \left[ \int_0^{T(\phi)} e^{-\beta u} g(X(u), F_i) du | X(0) = x \right], $$

and, for all $(x, F_A) \in \mathcal{X}$,

$$ V^*(x, F_A) := \max_{\phi \subseteq F_A} \left\{ r((x, F_A), \phi) + e^{-\beta T(\phi)} E \left[ V^*(X(T(\phi)), F_A \setminus \phi) | X(0) = x \right] \right\}. $$

**4.3. Multiplicative Separability** Motivated by Assumption 6, we propose a multiplicative separability condition on the instantaneous revenue function $g$:

**Assumption 11 (Multiplicative Separability).** There exists some measurable, nonincreasing function $\vartheta^* : [1, \infty) \to (0, 1]$ such that

$$ g \left( x_0 \prod_{i=1}^j \xi_i, F_l \right) = g(x_0, F_l) \prod_{i=1}^j \vartheta^*(\xi_i) \quad \forall F_l \subseteq F_l^0 \cup F_A^0 \quad \forall j \in \mathbb{N} \quad a.e.-P. $$
Under the multiplicative separability assumption, successive increases in market intensity induce geometric decay in the rate at which the release manager’s revenues accumulate. For instance, if intensity increases by 10%, then the release manager observes a \((1 - \vartheta^* (1.1)) \times 100\%\) decline in her instantaneous revenue rate. This approach is parsimonious, and allows managers to conceptualize the impact of market shocks in terms of elasticities. By appropriately specifying the function \(\vartheta^*\), a manager can also model complex relationships between market intensity and revenues. The key limitation of Assumption 11 is that the percentage change in the revenue rate \(g\) in response to a market shock does not depend on the release manager’s product \(F_t\) or the current level of market intensity.

When combined with the independence assumptions in §4.2, multiplicative separability leads to two crucial results. The first result permits a simplification of the inter-epoch reward \(r\) by introducing a modified discount factor that encodes the impact of market intensity on the release manager’s revenues. It follows immediately from Proposition 1 and Theorem 3 in Adelman and Mancini (2014).

**Proposition 1 (Modified Discount Factor).** For all \(F_A \subseteq F_A^0\) and all \(\phi \subseteq F_A\),

\[
r((X(0), F_A), \phi) = -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[1 - \frac{e^{-\gamma^* \tau(\phi)}}{\gamma^*}\right] g(X(0), F_t) \quad a.e.-P,
\]

with \(\gamma^* := \beta + \lambda (1 - \zeta^*)\) and \(\zeta^* := E[\vartheta^* (\xi_t)]\).

Proposition 1 indicates that the expected value of the stochastic revenue stream \(\{e^{-\beta u} g(X(u), F_t) : 0 \leq u < \tau (\phi)\}\) coincides with the value of a deterministic revenue stream \(\{e^{-\gamma^* u} g(X(0), F_t) : 0 \leq u < \tau (\phi)\}\) that depends only on the intensity level at time \(u = 0\) and accounts for market stochasticity with the modified discount factor \(\gamma^*\). Since \(\zeta^* \in (0, 1]\), \(\gamma^* \geq \beta\), implying that intensity induces additional decay. Furthermore, the fact that \(\gamma^*\) increases in the values of \(\lambda\) and \((1 - \zeta^*)\) confirms the following intuition: a market with frequent and/or significant shocks is more intense than one in which shocks are rare and/or minor.

The second implication of multiplicative separability is technical in nature and follows from Assumption 11 by inspection. We formally present it here given its importance throughout our analysis. Let \(X_j := x_0 \prod_{i=1}^{J} \xi_i\) for all \(j \in \mathbb{N}\).

**Proposition 2 (Ratio Invariance).** If \(F_I\) and \(F'_I\) are arbitrary subsets of \(F_I^0 \cup F_A^0\) with \(F'_I \neq \emptyset\), then

\[
\frac{g(X_j, F_I)}{g(X_j, F'_I)} = \frac{g(x_0, F_I)}{g(x_0, F'_I)} \quad a.e.-P \quad \forall j \in \mathbb{N}.
\]

Since \(F'_I \neq \emptyset\), Assumptions 3 and 5 eliminate the possibility of division by zero in Proposition 2. To ease notation, we let

\[
b(F_I, F'_I) := \frac{g(x_0, F_I)}{g(x_0, F'_I)}
\]

for arbitrary subsets \(F_I\) and \(F'_I\) of \(F_I^0 \cup F_A^0\) with \(F'_I \neq \emptyset\).

Before proceeding, we provide an example of a function that satisfies Assumptions 3, 5, and 11. Consider the following specification for \(g\):

\[
g(x, F_I) := \frac{z(F_I)}{x^\alpha} \quad \forall x \in [x_0, \infty) \quad \forall F_I \in F_I^0 \cup F_A^0,
\]

with \(z : 2F_I^0 \cup F_A^0 \rightarrow \mathbb{R}_+\). If \(z(0) = 0\) and \(z\) is strictly increasing on its domain with respect to the set-containment partial ordering, then the specification (9) satisfies Assumptions 3 and 5. Regardless of the structure of the shock random variables \(\{\xi_j \}_{j \in \mathbb{N}}\), Assumption 11 is satisfied with \(\vartheta^*(y) := \frac{1}{y^\alpha}\) for all \(y \in [1, \infty)\). If the distribution of the shocks is known, then other specifications for \(g\) may also satisfy Assumptions 3, 5, and 11 (Adelman and Mancini (2014)).
5. The Case with No Fixed Costs  In this section, we analyze the combinatorial dynamics of the release management problem in the absence of fixed costs. For the remainder of this section, we impose the condition that \( c = 0 \).

5.1. Value Function Decomposition  Suppose that the release manager is in state \((x, F_A)\) at some decision epoch and decides to permanently cease development. Letting \( \tau(\phi) \rightarrow \infty \) in Proposition 1 suggests that the value of this decision is \( \frac{g(x,F_I)}{\gamma^*} \). However, this decision may be suboptimal as it foregoes the potential benefits of future development. The following result indicates that the value of an optimal policy scales this guaranteed revenue stream by an optionality multiplier representing the value of the undeveloped features \( F_A \).

**Theorem 2.** The value function \( V^* \) decomposes as follows:

\[
V^* (X_j, F_A) = \frac{g(X_j,F_I)}{\gamma^*} \Lambda^* (F_A) \quad \text{a.e.-} P \quad \forall F_A \subseteq F_A^0 \quad \forall j \in \mathbb{N},
\]

with the values \( \{\Lambda^* (F_A) : F_A \subseteq F_A^0 \} \) determined by the equations:

\[
\Lambda^* (F_A) = \max_{\phi \subseteq F_A} \left\{ \left( 1 - e^{-\gamma^*\tau(\phi)} \right) + e^{-\gamma^*\tau(\phi)} b (F_I \cup \phi, F_I) \Lambda^* (F_A \setminus \phi) \right\} \quad \forall F_A \subseteq F_A^0.
\]

Since the decomposition (10) holds almost everywhere with respect to the probability measure \( P \), it applies both at each decision epoch, as well as between decision epochs.

**Proof.** By Theorem 4 in Adelman and Mancini (2014),

\[
V^* (X_j, F_A) = \frac{g(X_j,F_I)}{g(x_0,F_I)} V^* (x_0,F_A) \quad \text{a.e.-} P \quad \forall F_A \subseteq F_A^0 \quad \forall j \in \mathbb{N}.
\]

If we define

\[
\Lambda^* (F_A) := \left[ V^* (x_0,F_A) \right] \frac{\gamma^*}{g(x_0,F_I)} \quad \forall F_A \subseteq F_A^0,
\]

then (10) immediately follows.

To verify (11), we combine the optimality equations for our SMDP with the decomposition (10). For arbitrary \( F_A \subseteq F_A^0 \),

\[
V^* (x_0,F_A) = \max_{\phi \subseteq F_A} \left\{ \left[ 1 - e^{-\gamma^*\tau(\phi)} \right] g(x_0,F_I) + e^{-\beta \tau(\phi)} E \left[ V^* (X (\tau(\phi)), F_A \setminus \phi) \bigg| X(0) = x_0 \right] \right\}
\]

by (6) and Proposition 1

\[
= \max_{\phi \subseteq F_A} \left\{ \left[ 1 - e^{-\gamma^*\tau(\phi)} \right] g(x_0,F_I) + e^{-\beta \tau(\phi)} \sum_{j=0}^{\infty} E \left[ V^* (X_j,F_A \setminus \phi) \bigg| N_{\tau(\phi)} = j \right] P \left( N_{\tau(\phi)} = j \right) \right\}
\]

by Assum.'s 6 and 10

\[
= \max_{\phi \subseteq F_A} \left\{ \left[ 1 - e^{-\gamma^*\tau(\phi)} \right] g(x_0,F_I) + e^{-\beta \tau(\phi)} \sum_{j=0}^{\infty} \left[ \frac{g(X_j,F_I \cup \phi)}{\gamma^*} \Lambda^* (F_A \setminus \phi) \bigg| N_{\tau(\phi)} = j \right] P \left( N_{\tau(\phi)} = j \right) \right\}
\]

by (10)

\[
= \max_{\phi \subseteq F_A} \left\{ \left[ 1 - e^{-\gamma^*\tau(\phi)} \right] g(x_0,F_I) + e^{-\beta \tau(\phi)} \frac{g(x_0,F_I \cup \phi)}{\gamma^*} \Lambda^* (F_A \setminus \phi) \sum_{j=0}^{\infty} (\zeta^*)^j P \left( N_{\tau(\phi)} = j \right) \right\}
\]

by Assum.'s 9 and 11

\[
= \max_{\phi \subseteq F_A} \left\{ \left[ 1 - e^{-\gamma^*\tau(\phi)} \right] g(x_0,F_I) + e^{-\gamma^*\tau(\phi)} \frac{g(x_0,F_I \cup \phi)}{\gamma^*} \Lambda^* (F_A \setminus \phi) \right\},
\]
with the final line following from the fact that \( N \) is Poisson by Assumption 6. Dividing both sides by \( \frac{g(x_0,F_I)}{e} \) yields (11). □

As the next result shows, optionality can not make the release manager worse off.

**Proposition 3.** For all \( F_A \subseteq F^0_A \), \( \Lambda^* (F_A) \geq 1 \), with equality if the empty set maximizes (11).

**Proof.** Recall that \( \tau(\emptyset) = w > 0 \). For any subset \( F_A \) of \( F^0_A \), evaluating the argument of the max operator in (11) at \( \phi = \emptyset \) implies that:

\[
\Lambda^* (F_A) \geq \left( 1 - e^{-\gamma^* w} \right) + e^{-\gamma^* w} b(F_I, F_I) \Lambda^* (F_A) = \left( 1 - e^{-\gamma^* w} \right) + e^{-\gamma^* w} \Lambda^* (F_A) \quad \text{by (8)}
\]

\[
\iff \Lambda^* (F_A) \geq 1.
\]

The same derivation trivially holds with equality for any set \( F_A \subseteq F^0_A \) for which the empty set is a maximizer of (11). □

However, increased market intensity diminishes the value of undeveloped functionality.

**Proposition 4.** The option value \( \Lambda^* (F_A) \) of any nonempty subset \( F_A \) of \( F^0_A \) is nonincreasing in \( \gamma^* \).

**Proof.** First, Proposition 3 and equation (11) indicate that for all nonempty subsets \( F_A \) of \( F^0_A \):

\[
\Lambda^* (F_A) = \max \left\{ 1, \max_{\phi \subseteq F_A, \phi \neq \emptyset} \left\{ \left( 1 - e^{-\gamma^* \tau(\phi)} \right) + e^{-\gamma^* \tau(\phi)} b(F_I \cup \phi, F_I) \Lambda^* (F_A \setminus \phi) \right\} \right\}.
\]

(13)

We now proceed by induction on the cardinality of \( F_A \). If \( F_A = \{ f \} \) for some feature \( f \in F^0_A \), then (13) yields:

\[
\Lambda^* (\{ f \}) = \max \left\{ 1, \left( 1 - e^{-\gamma^* \tau(f)} \right) + e^{-\gamma^* \tau(f)} b(F_I \cup \{ f \}, F_I) \Lambda^* (\emptyset) \right\}
\]

\[
= \max \left\{ 1, \left( 1 - e^{-\gamma^* \tau(f)} \right) + e^{-\gamma^* \tau(f)} b(F_I \cup \{ f \}, F_I) \right\} \quad \text{by Prop. 3.}
\]

(14)

Since \( b(F_I \cup \{ f \}, F_I) > 1 \) by Assumption 5, both arguments of the max operator in (14) are non-increasing functions of \( \gamma^* \). The result follows because the pointwise maximum of nonincreasing functions is itself nonincreasing.

Assume that the result holds for all cardinalities \( i \in \{ 1, 2, \ldots, n \} \) and let \( F_A \subseteq F^0_A \) be an arbitrary subset of cardinality \( n + 1 \). Since \( |F_A \setminus \phi| < n + 1 \) for all nonempty subsets \( \phi \) of \( F_A \), our induction hypothesis together with the fact that \( \Lambda^* (F_A \setminus \phi) \geq 1 \) by Proposition 3 implies that each of the arguments of the max operator in (13) is nonincreasing in \( \gamma^* \). The result immediately follows. □

Finally, we examine the intuition that, in some sense, more optionality should be “better” than less.

**Proposition 5.** Let \( h(F_I, \phi) := g(x_0, F_I \cup \phi) \) on \( \left\{ (F_I, \phi) \mid F_I \subseteq F^0_I \cup F^0_A, \phi \subseteq F_A \right\} \), and suppose that \( h \) is submodular on its domain with respect to the set-containment partial ordering. For any \( F_A \subseteq F^0_A \), \( \Lambda^* (F_A) \geq \Lambda^* (F_A') \) for all \( F_A' \subseteq F_A \).

**Proof.** We proceed by induction on the cardinality of \( F_A \). If \( |F_A| = 0 \), the statement trivially follows from Proposition 3 since \( \Lambda^* (\emptyset) = 1 \). Suppose now that the statement holds for all \( F_A \subseteq F^0_A \) with \( |F_A| \leq n \). Let \( F_A \) be an arbitrary subset of \( F^0_A \) of cardinality \( n + 1 \), and let \( F_A' \) be an arbitrary subset of \( F_A \). If the empty set maximizes (11) for \( F_A' \), then Proposition 3 implies that \( 1 = \Lambda^* (F_A') \leq \Lambda^* (F_A) \).

Next, consider the case when the empty set does not maximize (11) for \( F_A' \). Since \( F_A' \subseteq F_A \), it follows that \( F_A' \supseteq F_I \). For any nonempty subset \( \phi \) of \( F_A' \), submodularity of \( h \) thus implies:

\[
g(x_0, F_A' \cup \phi) - g(x_0, F_I') \leq g(x_0, F_I \cup \phi) - g(x_0, F_I).
\]

(15)
Furthermore, Assumption 5 ensures that both sides of (15) are strictly positive and that \( g(x_0, F'_i) \geq g(x_0, F_i) \). Therefore,
\[
\frac{g(x_0, F'_i \cup \phi) - g(x_0, F'_i)}{g(x_0, F'_i)} \leq \frac{g(x_0, F_i \cup \phi) - g(x_0, F_i)}{g(x_0, F_i)} \Rightarrow b(F'_i \cup \phi, F'_i) \leq b(F_i \cup \phi, F_i).
\]

In addition, \(|F_A \setminus \phi| \leq n\) since \( \phi \neq \emptyset \), and so our induction hypothesis implies that \( \Lambda^* (F_A \setminus \phi) \geq \Lambda^* (F'_A \setminus \phi) \). Hence,
\[
\Lambda^* (F_A) \geq \max_{\phi \subseteq F'_A, \phi \neq \emptyset} \left\{ \left( 1 - e^{-\gamma^* \tau(\phi)} \right) + e^{-\gamma^* \tau(\phi)} b(F_i \cup \phi, F_i) \Lambda^* (F_A \setminus \phi) \right\} \quad \text{since } F'_A \subseteq F_A
\]
\[
\geq \max_{\phi \subseteq F'_A, \phi \neq \emptyset} \left\{ \left( 1 - e^{-\gamma^* \tau(\phi)} \right) + e^{-\gamma^* \tau(\phi)} b(F'_i \cup \phi, F'_i) \Lambda^* (F'_A \setminus \phi) \right\}
\]
\[
= \Lambda^* (F'_A).
\]

Since \( F_A \) and \( F'_A \) were arbitrary, we are done. \( \Box \)

The validity of the submodularity requirement depends on the relationship between the set of features available for development and the existing product. If there are significant complementarities, then (15) may be inappropriate since the incremental benefit of adding features that complement existing functionality could exceed the benefit of adding the same functionality to a product lacking complimentary features. In order to preserve the fidelity of our model, we do not require submodularity elsewhere in our analysis.

### 5.2. Quasi-Open-Loop Optimal Policy

We now turn our attention to identifying an optimal policy for the release management problem in the absence of fixed costs. The next result follows immediately from Theorem 4 in Adelman and Mancini (2014), and Theorem 2 in §5.1.

**Theorem 3.** If \( \bar{\pi} = d^\infty \) is a stationary deterministic policy such that \( d(x, F_A) = \bar{d}_{x_0} (F_A) \) on \( \mathcal{S} \) for any function \( \bar{d}_{x_0} : 2^{F_A^0} \rightarrow 2^{F_A^0} \) satisfying the following condition:
\[
\bar{d}_{x_0} (F_A) \in \arg \max_{\phi \subseteq F_A^0} \left\{ \left( 1 - e^{-\gamma^* \tau(\phi)} \right) + e^{-\gamma^* \tau(\phi)} b(F_i \cup \phi, F_i) \Lambda^* (F_A \setminus \phi) \right\} \quad \forall F_A \subseteq F_A^0,
\]
then the policy \( \bar{\pi} \) is optimal with probability one, i.e., \( V^\pi (X_j, F_A) = V^* (X_j, F_A) \) a.e.-\( P \) for all \( F_A \subseteq F_A^0 \) and all \( j \in \mathbb{N} \).

We refer to the policy \( \bar{\pi} \) as quasi-open-loop because the function \( \bar{d}_{x_0} \) ignores the current level of market intensity, but still depends on the set of available features. Computing the policy \( \bar{\pi} \) thus reduces to evaluating the finite set of conditions (16), as opposed to an uncountably infinite set of conditions on the full state space \( \mathcal{S} \). Furthermore, the release manager can implement the policy \( \bar{\pi} \) knowing that, with probability one, she will only transition to states at which the policy is optimal.

Theorem 3 suggests that constructing the policy \( \bar{\pi} \) amounts to identifying a release sequence \( \{ \phi_i^* \}_{i \in \mathbb{N}} \) that solves a deterministic dynamic program. Applying (11) recursively, we obtain:
\[
\phi_i^* := \bar{d}_{x_0} (F_A^0), \quad \phi_{i+1}^* := \bar{d}_{x_0} (F_A^0 \setminus (\bigcup_{n=1}^{i} \phi_n^*)) \quad \forall i \geq 1.
\]

As the next result shows, the release manager has no incentive to wait between releases since Assumption 5 implies that adding functionality to the existing product always leads to strictly higher revenues.

**Proposition 6.** If \( F_A \subseteq F_A^0 \) and \( F_A \neq \emptyset \), then \( \bar{d}_{x_0} (F_A) \neq \emptyset \).
Proof. If \( \tilde{d}_x (F_A) = \emptyset \), then (11) implies that \( \Lambda^* (F_A) = 1 \). However, for any \( f \in F_A \),
\[
\Lambda^* (F_A) \geq \left(1 - e^{-\gamma \tau(f)}\right) + e^{-\gamma \tau(f)} b(F_I \cup \{f\}, F_I) \Lambda^* (F_A \setminus \{f\})
\]
\[
\geq \left(1 - e^{-\gamma \tau(f)}\right) + e^{-\gamma \tau(f)} b(F_I \cup \{f\}, F_I) \text{ since } \Lambda^* (F_A') \geq 1 \text{ by Prop. 3}
\]
\[
> \left(1 - e^{-\gamma \tau(f)}\right) + e^{-\gamma \tau(f)} \text{ since } b(F_I \cup \{f\}, F_I) > 1 \text{ by Assumption 5}
\]
\[= 1 \text{ which is a contradiction. } \]

Proposition 6 indicates that the release manager will only take the waiting action once all features have been implemented. Hence, there must exist some \( j^* \geq 1 \) such that \( \phi_i^* \neq \emptyset \) for all \( i \in \{1, 2, \ldots, j^*\} \), and \( \phi_i^* = \emptyset \) for all \( i > j^* \). Furthermore, \( j^* < \infty \) since \( F_A^0 \) has finite cardinality, and \( \bigcup_{i=1}^{j^*} \phi_i^* = F_A^0 \).

For each \( i \in \{1, 2, \ldots, j^*\} \), the incremental revenue generated by the \( i^{th} \) release in the sequence \( \Phi^* \) equals:
\[
E \left[ \int_0^\infty e^{-\beta u} \left[ g \left( X (u), F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right) - g \left( X (u), F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right) \right] du \right] \bigg| X (0) = x_0 ,
\]
which reduces to the following by the definition (5) of the inter-epoch reward \( r \) and Proposition 1:
\[
g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right) - g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right).
\]

The following result thus suggests that the sequence \( \Phi^* \) optimally manages the tradeoff between the timing and magnitude of the increased revenues resulting from new releases by solving a deterministic sequencing problem with a static market and discount factor \( \gamma^* \). A proof is included in Appendix A.

**Proposition 7.** If \( \Phi^* := \{\phi_i^*\}_{i \in \mathbb{N}} \) is the optimal release sequence defined by (17), then
\[
V^* \left( x_0, F_A^0 \right) = \frac{g \left( x_0, F_I^0 \right)}{\gamma^*} + \sum_{i=1}^{j^*} e^{-\gamma^* \tau(\phi_i^*)} \left[ g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right) - g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*) \right) \right].
\]

Observe that the determinism of (18) does not imply that the release manager can entirely ignore the revenue impact of market intensity. She must note the initial intensity level and account for future changes by adjusting her discount factor.

### 5.3. Characterizing the Optimal Release Sequence

We now derive structural properties of the optimal release sequence \( \{\phi_i^*\}_{i \in \mathbb{N}} \) defined in (17) by applying an exchange argument motivated by (18). To ease notation, we define the function \( \Upsilon \) as follows: for any release sequence \( \Phi = \{\phi_i\}_{i \in \mathbb{N}} \) such that there exists some \( j \geq 1 \) for which \( \phi_i \neq \emptyset \) for all \( i \in \{1, 2, \ldots, j\} \), and \( \phi_i = \emptyset \) for all \( i > j \), let
\[
\Upsilon (\Phi) := \frac{g \left( x_0, F_I^0 \right)}{\gamma^*} + \sum_{i=1}^{j} e^{-\gamma^* \tau(\phi_i)} \left[ g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n) \right) - g \left( x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n) \right) \right].
\]

The function \( \Upsilon \) calculates the value of any given release sequence.
PROPOSITION 8. If the release sequence \( \{\phi_i^*\}_{i \in \mathbb{N}} \) is optimal and \( 2 \leq j^* < \infty \), then for all \( m \in \{1, 2, \ldots, j^*-1\} \):

\[
\frac{e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) \right]}{1 - e^{-\gamma^* \tau(\phi_m^*)}} \geq \frac{e^{-\gamma^* \tau(\phi_{m+1}^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right]}{1 - e^{-\gamma^* \tau(\phi_{m+1}^*)}}.
\]

(20)

Proof. Let \( m \in \{1, 2, \ldots, j^*-1\} \) be arbitrary and consider the permuted sequence \( \Phi' := \{\phi_1^*, \phi_{m-1}^*, \phi_{m+1}^*, \phi_{m+2}^*, \ldots\} \). The optimality of \( \Phi^* \) implies that \( \Upsilon(\Phi^*) - \Upsilon(\Phi') \geq 0 \), which by (19) requires:

\[
\left\{ e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) \right] \right. + \\
\left. e^{-\gamma^* \tau(\phi_{m+1}^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] \right. - \\
\left\{ e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] \right. + \\
\left. e^{-\gamma^* \tau(\phi_{m+1}^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] \right\} \geq 0.
\]

Multiplying through by \( \gamma^* \), canceling the term \( e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] \), and dividing through by \( e^{-\gamma^* \tau(\phi_{m+1}^*)} \) yields:

\[
\left\{ e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) \right] - \\
e^{-\gamma^* \tau(\phi_{m+1}^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] - \\
e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] + \\
e^{-\gamma^* \tau(\phi_{m+1}^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right] \right\} \geq 0.
\]

By collecting terms, we obtain:

\[
e^{-\gamma^* \tau(\phi_m^*)} \left( 1 - e^{-\gamma^* \tau(\phi_{m+1}^*)} \right) g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) - e^{-\gamma^* \tau(\phi_m^*)} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \geq \\
e^{-\gamma^* \tau(\phi_{m+1}^*)} \left( 1 - e^{-\gamma^* \tau(\phi_m^*)} \right) g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m+1} \phi_n^*)\right) - e^{-\gamma^* \tau(\phi_{m+1}^*)} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right),
\]

which, after dividing both sides by \( \left( 1 - e^{-\gamma^* \tau(\phi_m^*)} \right) \left( 1 - e^{-\gamma^* \tau(\phi_{m+1}^*)} \right) \), further simplifies to:

\[
\frac{e^{-\gamma^* \tau(\phi_m^*)}}{1 - e^{-\gamma^* \tau(\phi_{m+1}^*)}} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) - \frac{e^{-\gamma^* \tau(\phi_{m+1}^*)}}{1 - e^{-\gamma^* \tau(\phi_m^*)}} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) \geq \frac{e^{-\gamma^* \tau(\phi_{m+1}^*)}}{1 - e^{-\gamma^* \tau(\phi_m^*)}} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) - \frac{e^{-\gamma^* \tau(\phi_m^*)}}{1 - e^{-\gamma^* \tau(\phi_{m+1}^*)}} g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right).
\]

(21)

Consider the left-hand side of (21). Adding and subtracting the term

\[
\frac{e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right]}{1 - e^{-\gamma^* \tau(\phi_{m+1}^*)}}
\]

yields:

\[
\frac{e^{-\gamma^* \tau(\phi_m^*)} \left[ g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m-1} \phi_n^*)\right) - g\left(x_0, F_0^0 \cup (\bigcup_{n=1}^{m} \phi_n^*)\right) \right]}{1 - e^{-\gamma^* \tau(\phi_{m+1}^*)}} \leq
\]
showing that the value of the action than the value of the action arbitrary subset of absence of fixed development costs. interact appropriately, and such testing can lead to superadditive development times. As shown by In software development, newly added features are typically tested jointly to ensure that they development times will appear earlier in the release sequence. Condition (20) also resembles an ordering of Gittins indices for a traditional project selection problem in which each nonempty subset of $F_0^0$ is an available project, and the release manager must complete a project before beginning her next project (Gittins et al. (2011)). However, in the release management problem there are dependencies between projects. Once the release manager selects a release $\phi$, all subsets of $F_0^0$ containing any features in $\phi$ are no longer available. In addition, the index in (20) exhibits path dependence, i.e., calculating the index for the $m^{th}$ release requires knowledge of all previous releases.

Without additional conditions, Proposition 8 says nothing about the number of nonempty releases $j^*$ in the optimal release sequence, or the composition of each release. However, consider the case when the development time mapping $\tau$ is superadditive across disjoint subsets:

$$\tau(\phi \cup \phi') \geq \tau(\phi) + \tau(\phi') \quad \forall \phi, \phi' \subseteq F_A^0 \quad \phi \cap \phi' = \emptyset.$$

In software development, newly added features are typically tested jointly to ensure that they interact appropriately, and such testing can lead to superadditive development times. As shown by the next result, this condition imposes significant structure on the optimal release sequence in the absence of fixed development costs.

**Proposition 9.** If $\tau$ is superadditive, then every nonempty release selected by the policy $\pi$ consists of a single feature, i.e., $|d_{x_0}(F_A)| = 1$ for all nonempty $F_A \subseteq F_A^0$.

**Proof.** If $|F_A| = 1$, then by Proposition 6 we are done. Suppose now that $|F_A| > 1$. Let $\phi$ be an arbitrary subset of $F_A$ such that $|\phi| > 1$, and let $f$ be an arbitrary element of $\phi$. We proceed by showing that the value of the action $\{f\}$, as measured by the right-hand side of (11), is no less than the value of the action $\phi$.

Consider the value of the action $\{f\}$:

$$\left(1 - e^{-\gamma^*\tau(f)}\right) + e^{-\gamma^*\tau(f)}b(F_f \cup \{f\}, F_f) \Lambda^* (F_A \setminus \{f\}) \geq \left(1 - e^{-\gamma^*\tau(f)}\right) + e^{-\gamma^*\tau(f)}b(F_f \cup \{f\}, F_f) \left[1 - e^{-\gamma^*\tau(\phi \setminus f))}\right] + e^{-\gamma^*\tau(\phi \setminus f))}b(F_f \cup \phi, F_f \cup \{f\}) \Lambda^* (F_A \setminus \phi) = \left(1 - e^{-\gamma^*\tau(f)}\right) + e^{-\gamma^*\tau(f)}b(F_f \cup \{f\}, F_f) \left(1 - e^{-\gamma^*\tau(\phi \setminus f))}\right) + e^{-\gamma^*\tau(\phi \setminus f))}b(F_f \cup \phi, F_f) \Lambda^* (F_A \setminus \phi) = \left(1 - e^{-\gamma^*\tau(f)}\right) + e^{-\gamma^*\tau(f))} - e^{-\gamma^*\tau(\phi \setminus f))}]b(F_f \cup \{f\}, F_f) + e^{-\gamma^*\tau(\phi \setminus f))} - e^{-\gamma^*\tau(\phi)}b(F_f \cup \phi, F_f) \Lambda^* (F_A \setminus \phi).$$
However,
\[
\left( e^{-\gamma^* \tau(f)} - e^{-\gamma^* \tau(f) + \tau(\phi \setminus f))}] \right) b(F_I \cup \{f\}, F_I) > \left( e^{-\gamma^* \tau(f)} - e^{-\gamma^* \tau(f) + \tau(\phi \setminus f))}] \right)
\]
by Assumption 5 since \( b(F_I \cup \{f\}, F_I) > 1 \), and
\[
\left( e^{-\gamma^* \tau(f)} - e^{-\gamma^* \tau(\phi))}] \right) b(F_I \cup \phi, F_I) \Lambda^* (F_A \setminus \phi) \geq \left( e^{-\gamma^* \tau(f)} - e^{-\gamma^* \tau(\phi))}] \right)
\]
since \( b(F_I \cup \phi, F_I) > 1 \) by Assumption 5, \( \Lambda^* (F_A \setminus \phi) \geq 1 \) by Proposition 3, and
\( (e^{-\gamma^* \tau(f))} + \tau(\phi \setminus f)) - e^{-\gamma^* \tau(\phi))}] \geq 0 \) by superadditivity. Therefore, (24) implies:
\[
\left( 1 - e^{-\gamma^* \tau(f)} \right) + e^{-\gamma^* \tau(f)} b(F_I \cup \{f\}, F_I) \Lambda^* (F_A \setminus \{f\}) > \left( 1 - e^{-\gamma^* \tau(f)} \right) + e^{-\gamma^* \tau(\phi))} b(F_I \cup \phi, F_I) \Lambda^* (F_A \setminus \phi) =
\]
\[
\left( 1 - e^{-\gamma^* \tau(\phi))} \right) + e^{-\gamma^* \tau(\phi))} b(F_I \cup \phi, F_I) \Lambda^* (F_A \setminus \phi) .
\]
(25)

Since the right-hand side of (25) is the value of the action \( \phi \), we are done. □.

When there are no fixed development costs, the release manager only needs to consider the tradeoff between development time and revenues. Proposition 9 thus states that the release manager has no incentive to issue multi-feature releases when \( \tau \) is superadditive, since bundling features together can only lengthen the release cycle. Proposition 9 also simplifies the computation of the policy \( \tilde{\pi} \) as follows: for any set \( F_A \subseteq F^0_A \), it suffices to evaluate (11) over the set of individual features \( f \in F_A \) instead of the collection of all subsets of \( F_A \).

We now combine Propositions 8 and 9 to identify a setting in which the condition (20) is both necessary and sufficient for optimality. Our result stipulates that the function \( g(x_0, \cdot) \) is additive on \( 2^{F^0_A \cup F^0_A} \), i.e.,
\[
g(x_0, F^1_I \cup F^2_I) = g(x_0, F^1_I) + g(x_0, F^2_I) \quad \forall F^1_I, F^2_I \subseteq F^0_I \cup F^0_A \quad \text{s.t.} \quad F^1_I \cap F^2_I = \emptyset.
\]

**Corollary 1.** If \( \tau \) is superadditive and \( g(x_0, \cdot) \) is additive on \( 2^{F^0_A \cup F^0_A} \), then it is optimal to release features individually and in the order \( f_1^*, f_2^*, \ldots, f_j^* \) indicated by the following ranking:
\[
\frac{e^{-\gamma^* \tau(f_1^*)} g(x_0, \{f_1^*\})}{1 - e^{-\gamma^* \tau(f_1^*)}} \geq \frac{e^{-\gamma^* \tau(f_2^*)} g(x_0, \{f_2^*\})}{1 - e^{-\gamma^* \tau(f_2^*)}} \geq \cdots \geq \frac{e^{-\gamma^* \tau(f_j^*)} g(x_0, \{f_j^*\})}{1 - e^{-\gamma^* \tau(f_j^*)}},
\]
with \( j^* = |F^0_A| \).

**Proof.** By Proposition 9, superadditivity of \( \tau \) implies that every nonempty release consists of a single feature. Additivity of \( g(x_0, \cdot) \) thus immediately converts (20) into (26). By construction, any sequence \( \{f_1, f_2, \ldots, f_j^*\} \) satisfying (26) must also satisfy:
\[
f_i \in \arg \max_{f \in (F^0_A \setminus \{f_1, f_2, \ldots, f_{i-1}\})} \frac{e^{-\gamma^* \tau(f)} g(x_0, \{f\})}{1 - e^{-\gamma^* \tau(f)}} \quad \forall i \in \{1, 2, \ldots, j^*\}.
\]
(27)

If there is a unique maximizer in (27) for each \( i \in \{1, 2, \ldots, j^*\} \), then there exists only one sequence satisfying the necessary condition (26). This sequence must therefore be optimal. Alternatively, multiple sequences satisfying (26) exist if multiple maximizers exist in (27) for at least one index \( i \in \{1, 2, \ldots, j^*\} \). Let \( \Phi^* = \{f_1^*, \ldots, f_j^*\} \) and \( \Phi' = \{f_1', \ldots, f_j'\} \) be two sequences satisfying (26). Condition (27) indicates we can convert \( \Phi^* \) into \( \Phi' \) via a finite sequence of pairwise permutations of features that have the same index value. By applying the exchange argument from the proof of Proposition 8, however, it follows that the value of a feature sequence does not change when two features with the same index value are permuted. Applying this logic to each of the permutations required to convert \( \Phi^* \) to \( \Phi' \) indicates that the sequences \( \Phi^* \) and \( \Phi' \) have the same value. □
When all releases consist of a single feature, the fact that implementing any individual feature removes all supersets of that feature from future consideration becomes irrelevant. The superadditivity condition on $\tau$ in Corollary 1 thus eliminates the dependencies between projects that distinguishes the release management problem from the traditional project selection problem. Furthermore, the additivity condition on $g(x_0, \cdot)$ eliminates the path-dependence inherent in the exchange condition (20). As a result, the ordering index in (26) exactly corresponds to the Gittins index for the project selection problem without preemption (Gittins et al. (2011)). It also coincides with the WDSPT rule found in the stochastic scheduling literature (Pinedo (2012)).

In the context of Corollary 1, the release manager can determine the relative ordering of any two features in $F^0_A$ simply by comparing the indices for each feature. We now consider the stability of such pairwise orderings with respect to the value of the modified discount factor $\gamma^*$. The following result takes a feature $f_i$, as given and compares its development time $\tau(f_i)$ and “bang-for-buck” $g(x_0, \{f_i\})/\tau(f_i)$ to that of an alternative feature $f_j$ to identify scenarios in which feature $f_i$ should precede $f_j$ regardless of the value of $\gamma^*$, as well as instances in which the relative ranking is sensitive to the value of $\gamma^*$.

**Proposition 10.** Let $\tau_i = \tau(f_i)$ and $g_i = g(x_0, \{f_i\})$ for any feature $f_i$ in $F^0_A$. Suppose $\tau$ is superadditive and $g(x_0, \cdot)$ is additive on $2F^0_A$. For any pair of distinct features $f_i$ and $f_j$ in $F^0_A$, the following two statements hold:

(i.) if $\tau_i \leq \tau_j$ and $\frac{g_i}{\tau_i} \geq \frac{g_j}{\tau_j}$, then it is optimal to release feature $f_i$ before feature $f_j$ for any value $\gamma^* \in (0, \infty)$, and

(ii.) if $\tau_i \leq \tau_j$ and $\frac{g_i}{\tau_i} < \frac{g_j}{\tau_j}$, then there exists a unique value $\hat{\gamma}_{i,j} \in (0, \infty)$ such that it is optimal to release feature $f_j$ before feature $f_i$ if $\gamma^* \in (0, \hat{\gamma}_{i,j})$ and optimal to release feature $f_i$ before feature $f_j$ if $\gamma^* \in [\hat{\gamma}_{i,j}, \infty)$.
Proof. Let features \( f_i \) and \( f_j \) be arbitrary, and define the function \( q_{i,j}(\gamma) \) on \((0, \infty)\) as \( \frac{e^{\gamma \tau_i} - 1}{e^{\gamma \tau_j} - 1} \). By Corollary 1, it is optimal to release feature \( f_i \) before feature \( f_j \) if

\[
\frac{e^{-\gamma \tau_i} g_i}{1 - e^{-\gamma \tau_i}} \geq \frac{e^{-\gamma \tau_j} g_j}{1 - e^{-\gamma \tau_j}} \iff \frac{g_i}{g_j} \geq \frac{e^{\gamma \tau_i} - 1}{e^{\gamma \tau_j} - 1} = q_{i,j}(\gamma^*). \tag{28}
\]

We prove statement (i) by analyzing the function \( q_{i,j} \) to determine when the second inequality in (28) can or cannot hold.

The function \( q_{i,j} \) is clearly continuous on \((0, \infty)\), and L'Hôpital's rule indicates that \( \lim_{\gamma \to 0^+} q_{i,j}(\gamma) = \frac{\tau_i}{\tau_j} \). Differentiation implies that on the domain \((0, \infty)\),

\[
\frac{\partial q_{i,j}}{\partial \gamma} < 0 \iff (e^{\gamma \tau_j} - 1) (\tau_i e^{\gamma \tau_i}) - (e^{\gamma \tau_i} - 1) (\tau_j e^{\gamma \tau_j}) < 0 \iff \frac{1 - e^{-\gamma \tau_j}}{1 - e^{-\gamma \tau_i}} < \frac{\tau_j}{\tau_i}, \tag{29}
\]

with the statement also holding with all inequalities reversed.

Suppose now that \( \tau_i \leq \tau_j \). If \( \tau_i = \tau_j \), then (i) follows trivially from (28). Consider the case when \( \tau_i < \tau_j \). The function \( 1 - e^{-\gamma x} \) is strictly concave in \( x \) on \([0, \infty)\) for any fixed \( \gamma \in (0, \infty) \), and so \( 1 - e^{-\gamma \tau_i} > 0 \cdot \left(1 - \frac{\tau_i}{\tau_j}\right) + (1 - e^{-\gamma \tau_j}) \cdot \left(\frac{\tau_i}{\tau_j}\right) \) because \( \frac{\tau_i}{\tau_j} < 1 \). Hence,

\[
\frac{1 - e^{-\gamma \tau_j}}{1 - e^{-\gamma \tau_i}} < \frac{\tau_j}{\tau_i},
\]

which indicates that (29) holds for all \( \gamma \in (0, \infty) \) when \( \tau_i < \tau_j \). The function \( q_{i,j} \) is therefore monotonically decreasing on \((0, \infty)\) and thus never exceeds its limit \( \frac{\tau_i}{\tau_j} \) at 0. Statement (i) follows from (28) because

\[
\frac{g_i}{\tau_i} \geq \frac{g_j}{\tau_j} \Rightarrow \frac{g_i}{g_j} \geq \frac{\tau_i}{\tau_j} \Rightarrow \frac{g_i}{g_j} > q_{i,j}(\gamma) \ \forall \gamma \in (0, \infty), \tag{30}
\]

with the final inequality following from the fact that \( q_{i,j} \) is monotonically decreasing in \( \gamma \).

Next, we prove statement (ii) by establishing the existence of a value \( \tilde{\gamma}_{i,j} \in (0, \infty) \) that satisfies (28) with equality. We require the following result, obtained from L'Hôpital's rule:

\[
\lim_{\gamma \to \infty} q_{i,j}(\gamma) = \begin{cases} 
\infty, & \tau_i > \tau_j \\
1, & \tau_i = \tau_j \\
0, & \tau_i < \tau_j 
\end{cases}
\]

If \( \tau_i = \tau_j \), then statement (ii) follows trivially by reversing the inequalities in (28). Suppose that \( \tau_i < \tau_j \). As argued earlier, the function \( q_{i,j} \) then approaches \( \frac{\tau_i}{\tau_j} \) at 0 and tends to 0 as \( \gamma \to \infty \). If \( \frac{\tau_i}{\tau_j} < \frac{\tau_i}{\tau_i} \), then \( \frac{g_i}{g_j} < \frac{\tau_i}{\tau_j} \), and so the intermediate value theorem ensures the existence of a value \( \tilde{\gamma}_{i,j} \in (0, \infty) \) such that \( \frac{g_i}{g_j} = q_{i,j}(\tilde{\gamma}_{i,j}) \). Uniqueness follows from the earlier argument that \( q_{i,j} \) is strictly decreasing when \( \tau_i < \tau_j \), as do the following statements:

\[
\frac{g_i}{g_j} < q_{i,j}(\gamma) \ \forall \gamma \in (0, \tilde{\gamma}_{i,j}) \quad \text{and} \quad \frac{g_i}{g_j} \geq q_{i,j}(\gamma) \ \forall \gamma \in [\tilde{\gamma}_{i,j}, \infty). \]

Statement (ii) follows immediately from (28). \( \square \)

Figure 1 illustrates the conclusions of Proposition 10. Given the feature \( f_i \), the first panel plots all development time and bang-for-buck values for an alternate feature \( f_j \) such that \( f_i \) and \( f_j \) have the same index value when \( \gamma^* = \beta \). We refer to these points as the \( \beta \)-indifference curve (\( \beta \)-IC) through \( f_i \), since the release manager has no preference over the sequencing of features on this curve. Features above the \( \beta \)-IC curve have a strictly higher index value than feature \( f_i \), and thus
precede \( f_i \) in any optimal release sequence. Analogously, features below \( \beta \)-IC should appear after \( f_i \) when \( \gamma^* = \beta \). In the second panel in Figure 1, we include a second indifference curve \( \gamma^* \)-IC through \( f_i \) for some value \( \gamma^* > \beta \). Features to the southeast of \( f_i \) take longer to develop and offer lower bang-for-buck than \( f_i \). As indicated by part (i) of Proposition 10, \( f_i \) should precede all such features regardless of the value of \( \gamma^* \). Similarly, features to the northwest of \( f_i \) dominate \( f_i \) for all values of \( \gamma^* > 0 \). These two regions represent “error-free” scenarios in which the release manager can misspecify the market intensity process, or entirely ignore it, and still correctly rank features relative to \( f_i \). As indicated by part (ii), however, more care is required when considering a feature \( f_j \) that compensates for a longer development time than \( f_i \) by offering a higher bang-for-buck. If the release manager ignores marketplace intensity by setting \( \gamma^* = \beta \) instead of using the correct value \( \gamma^* > \beta \), she may undervalue the lower development time of \( f_i \) relative to \( f_j \). Because of this “pessimistic” assessment of \( f_i \), she will develop \( f_j \) before \( f_i \) even though this ordering is actually suboptimal. By ignoring or misspecifying market intensity, the release manager may similarly overvalue the higher bang-for-buck value of \( f_i \) relative to features \( f_j \) with lower development times in the “optimistic” region.

5.4. Evaluating a Greedy Index Heuristic  

The exchange condition (20) suggests a heuristic that greedily constructs a release sequence \( \Phi \) as follows:

\[
\hat{\phi}_1 \in \arg \max_{\phi \subseteq F_A^0} \left\{ \frac{e^{-\gamma^* \tau(\phi)} [g(x_0, F^0_A \cup \phi) - g(x_0, F^0_I)]}{1 - e^{-\gamma^* \tau(\phi)}} \right\}
\]

\[
\hat{\phi}_i \in \arg \max_{\phi \subseteq F_A^0 \setminus \left( \bigcup_{n=1}^{i-1} \hat{\phi}_n \right)} \left\{ \frac{e^{-\gamma^* \tau(\phi)} [g(x_0, F^0_I \cup (\bigcup_{n=1}^{i-1} \hat{\phi}_n) \cup \phi) - g(x_0, F^0_I \cup (\bigcup_{n=1}^{i-1} \hat{\phi}_n))]}{1 - e^{-\gamma^* \tau(\phi)}} \right\} \quad \forall i \geq 2.
\]

By construction, the sequence \( \hat{\Phi} \) satisfies (20) and we denote its value by \( \hat{V}(x_0, F_A^0) := \Upsilon(\hat{\Phi}) \). However, our next result shows that this heuristic can perform arbitrarily poorly when there are no fixed costs.

**Proposition 11.** Assume that \( |F_A^0| \geq 2 \). If there exists some \( f \in F_A^0 \) such that \( \tau(F_A^0 \setminus \{f\}) + \tau(f) \leq \tau(F_A^0) \), then for any \( J \geq 1 \) there exists some \( H \geq 0 \) such that if \( \tau(F_A^0) - [\tau(F_A^0 \setminus \{f\}) + \tau(f)] \geq H \), then

\[
\lim_{g(x_0, F^0_I \cup F_A^0) \to \infty} \frac{\hat{V}^* (x_0, F_A^0)}{\hat{V} (x_0, F_A^0)} \geq J.
\]

**Proof.** Let \( J \geq 1 \) be arbitrary. As \( g(x_0, F^0_I \cup F_A^0) \) tends towards infinity, all of the following conditions will eventually be satisfied:

\[
\left\{ \frac{e^{-\gamma^* \tau(F_A^0)} [g(x_0, F^0_I \cup F_A^0) - g(x_0, F^0_I)]}{1 - e^{-\gamma^* \tau(F_A^0)}} \right\} \geq \left\{ \frac{e^{-\gamma^* \tau(\phi)} [g(x_0, F^0_I \cup \phi) - g(x_0, F^0_I)]}{1 - e^{-\gamma^* \tau(\phi)}} \right\} \quad \forall \phi \subseteq F_A^0.
\]

Hence, in the limit the greedy index heuristic opts to develop the entire set \( F_A^0 \) in a single release. We thus take the value of the sequence \( \hat{\Phi} \) to be:

\[
\hat{V}(x_0, F_A^0) = \Upsilon(\hat{\Phi}) = \frac{g(x_0, F^0_I)}{\gamma^*} + e^{-\gamma^* \tau(F_A^0)} \left[ \frac{g(x_0, F^0_I \cup F_A^0) - g(x_0, F^0_I)}{\gamma^*} \right].
\]

Next, consider the release sequence \( \Phi' \) defined as follows:

\[
\phi'_1 = F_A^0 \setminus \{f\} \quad \phi'_2 = \{f\} \quad \phi'_i = \emptyset \quad \forall i > 2.
\]
Applying (19), we obtain the value of the sequence $\Phi'$:
\[
\Upsilon (\Phi') = \frac{g(x_0, F^0_A) + e^{-\gamma^*\tau(F^0_A(f))} [g(x_0, F^0_f \cup F^0_A \setminus \{f\}) - g(x_0, F^0_f)]}{\gamma^*} + e^{-\gamma^*\tau(F^0_A(f)) + \tau(f)} [g(x_0, F^0_f \cup F^0_A \setminus \{f\}) - g(x_0, F^0_f)] + \frac{e^{-\gamma^*\tau(F^0_A(f)) + \tau(f)} [g(x_0, F^0_f \cup F^0_A \setminus \{f\}) - g(x_0, F^0_f)]}{\gamma^*}.
\] (32)

By inspection of (31) and (32),
\[
\lim_{\Phi(x_0, F^0_f \cup F^0_A) \to \infty} \Upsilon (\Phi') = \frac{e^{-\gamma^*\tau(F^0_A(f)) + \tau(f)}}{e^{-\gamma^*\tau(F^0_A)}}.
\] (33)

Furthermore, basic algebra indicates that
\[
\frac{e^{-\gamma^*\tau(F^0_A(f)) + \tau(f)}}{e^{-\gamma^*\tau(F^0_A)}} \geq J \iff \tau(F^0_A) - [\tau(F^0_A \setminus \{f\}) + \tau(f)] \geq \frac{\ln(J)}{\gamma^*}.
\]

Since $\Upsilon (\Phi') \leq \Upsilon (\Phi^*) = V^*(x_0, F^0_A)$, the result follows from (33) with $H = \frac{\ln(J)}{\gamma^*}$. □

In Proposition 11, letting the value of the entire feature set $F^0_A$ approach infinity blinds the greedy heuristic to the superior two-release sequence. This result stands in stark contrast to Corollary 1, which provides conditions under which the exchange condition (20) leads to an optimal release sequence. The key difference is that when development times are superadditive, we know a priori from Proposition 9 that each nonempty release consists of a single feature. We leveraged this property to obtain Corollary 1, but it does not necessarily apply in the more general setting considered in Proposition 11.

6. The Case with Fixed Costs

Before issuing a new release, software firms typically perform several rounds of testing to ensure that the new functionality is compatible with the existing product. Once the product is ready for release, firms will also need to market the product to inform both existing and potential customers of the new functionality. Depending on the type of product, firms may need to organize and host formal education efforts to teach consumers how to use the product or expedite its implementation. We accommodate all of these scenarios by allowing for a strictly positive fixed cost, i.e., $c > 0$. Even if the the external market does not evolve over time, fixed costs raise the profitability hurdle for new releases and likely lead to more feature bundling compared to the case without fixed costs. Furthermore, the release manager may find herself in a situation in which there is no way to recover the fixed cost of development for a new release given the features available for implementation. As a result, she must “leave features on the table” and terminate development. Proposition 6 indicates that this outcome is impossible when there are no fixed costs. A stochastically deteriorating market environment only exacerbates these tensions.

We can not apply the results of Section 5 when $c > 0$, as Theorem 2 fails when fixed costs are strictly positive. Instead, we approximate the value function of the SMDP with a functional form inspired by the decomposition (10), and employ the linear programming approach to approximate dynamic programming to generate an upper bound on the value function at the initial state $(x_0, F^0_A)$. This technique leads to a policy that can provably outperform a straightforward certainty-equivalence heuristic by an arbitrarily large margin, and also performs well in numerical trials.

6.1. Infinite Dimensional Linear Program

Consider the following linear program:

\[
(LP_{(x_0, F^0_A)}) \inf_{v \in \mathcal{V}_b} v(x_0, F^0_A) \quad \text{s.t.} \quad v(x, F_A) \geq r((x, F_A), \phi) + e^{-\beta\tau(\phi)} E \left[ v(X(\tau(\phi), F_A \setminus \phi) \mid X(0) = x) \right] \quad \forall ((x, F_A), \phi) \in \mathcal{X},
\]
with \( r \) defined as in (1), and optimal value \( LP_{(x_0,F_0^0)}^* \). The following standard result relates the linear program to the optimality equations presented in Theorem 1(i):

**Proposition 12.** The value function \( V^* \) is an optimal solution to \( LP_{(x_0,F_0^0)} \).

**Proof.** Theorem 1(i) and the definition (2) of \( L \) indicate that \( V^* \) is a feasible solution to \( LP_{(x_0,F_0^0)} \), and so \( V^* (x_0,F_0^0) \geq LP_{(x_0,F_0^0)}^* \). However, for any arbitrary feasible \( v \), the constraints of the linear program require that \( v \geq Lv \). It follows from Theorem 6.2.2 in Puterman (2005) that \( v \geq V^* \) on \( \mathcal{F} \), implying that \( v(x_0,F_0^0) \geq V^* (x_0,F_0^0) \). Since \( v \) was arbitrary, \( LP_{(x_0,F_0^0)} \geq V^* (x_0,F_0^0) \), which completes the proof. \( \square \)

We aim to exploit the modified discount factor and ratio invariance properties discussed in §4.3 to simplify the constraints in \( LP_{(x_0,F_0^0)} \). Recall, however, that under multiplicative separability these properties only apply with probability one at every stage market intensity process, not necessarily on the entire set \([x_0,\infty)\). To address this issue, we introduce the following definitions:

**Definition 1.** Define the sets \( \mathcal{X}_{\vartheta^*} \) and \( \mathcal{R}_{x_0} \) as follows:

\[
\mathcal{X}_{\vartheta^*} := \left\{ x \in [x_0,\infty) \mid g \left( x, \prod_{i=1}^{j} \xi_i, F_j \right) = g(x,F_1) \prod_{i=1}^{j} \vartheta^* (\xi_i) \quad \forall F_1 \subseteq F_0^0 \cup F_0^0 \quad \forall j \in N \quad \text{a.e.-} \mathcal{P} \right\},
\]
\[
\mathcal{R}_{x_0} := \left\{ x \in [x_0,\infty) \mid \frac{g(x,F_1)}{g(x_0,F_1)} = g(x,F_1) \quad \forall F_1,F'_1 \subseteq F_0^0 \cup F_0^0 \quad \text{s.t.} \quad F'_1 \neq \emptyset \right\}.
\]

We assume that both of these sets are measurable. The set \( \mathcal{X}_{\vartheta^*} \) consists of all values \( x \in [x_0,\infty) \) where multiplicative separability holds relative to the function \( \vartheta^* \), and Assumption 11 implies that \( x_0 \in \mathcal{X}_{\vartheta^*} \). Similarly, the set \( \mathcal{R}_{x_0} \) contains all intensity levels for which the function \( g \) exhibits ratio invariance with respect to \( x_0 \), and it follows trivially that \( x_0 \in \mathcal{R}_{x_0} \). In light of Definition 1, we turn our attention to the following relaxation of \( LP_{(x_0,F_0^0)} \):

\[
\left( \overline{LP}_{(x_0,F_0^0)} \right) \inf_{v \in \mathcal{V}_0} v(x_0,F_0^0)
\]
\[
\text{s.t.} \quad v(x,F_1) \geq r ((x,F_1),\varphi) + e^{-\beta r(\varphi)} E \left[ v(X(\tau(\varphi)), F_A | \varphi) \mid X(0) = x \right] \quad \forall ((x,F_1),\varphi) \in \mathcal{K} \quad \text{s.t.} \quad x \in \mathcal{X}_{\vartheta^*} \cap \mathcal{R}_{x_0},
\]

with optimal value \( \overline{LP}_{(x_0,F_0^0)}^* \). By leveraging the fact that the set \( \mathcal{X}_{\vartheta^*} \cap \mathcal{R}_{x_0} \) is closed under transitions in the market intensity process (Adelman and Mancini (2014)), the next result indicates that this relaxation is without loss of optimality. A proof is included in Appendix B.

**Proposition 13.** \( LP_{(x_0,F_0^0)}^* = \overline{LP}_{(x_0,F_0^0)}^* \).

It follows immediately from Propositions 12 and 13 that any feasible solution to \( \overline{LP}_{(x_0,F_0^0)} \) generates an upper bound on \( V^* (x_0,F_0^0) \).

Before proceeding, we further simplify \( \overline{LP}_{(x_0,F_0^0)} \). The following result extends the closed-form expression (7) for the inter-epoch reward \( r \) to all intensity levels in the set \( \mathcal{X}_{\vartheta^*} \). The proof follows from the definition of the set \( \mathcal{X}_{\vartheta^*} \) and Theorem 3 in Adelman and Mancini (2014).

**Proposition 14.** For all \( F_A \subseteq F_0^0 \) and all \( \varphi \subseteq F_A \),

\[
r ((x,F_A),\varphi) = -c \cdot 1_{\{\varphi \neq \emptyset\}} + \left[ \frac{1 - e^{-\gamma^* \tau(\varphi)}}{\gamma^*} \right] g(x,F_1) \quad \forall x \in \mathcal{X}_{\vartheta^*}.
\]
Therefore, the linear program $\overline{LP}_{(x_0,F_A^0)}$ becomes:

$$\left(\overline{LP}_{(x_0,F_A^0)}\right) \quad \inf_{v \in V_B} v(x_0, F_A^0)$$

subject to:

$$v(x, F_A) \geq -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[\frac{1 - e^{-\gamma^* \tau(\phi)}}{\gamma^*} \right] g(x, F_I) + e^{-\beta \tau(\phi)} E \left[ v\left(X(\tau(\phi)), F_A \setminus \phi\right) | X(0) = x \right] \forall ((x, F_A), \phi) \in \mathcal{X} \text{ s.t. } x \in \mathcal{X}_x \cap \mathcal{R}_{x_0}.$$ (34)

Inserting the approximation (34) into $\overline{LP}_{(x_0,F_A^0)}$, we obtain:

$$\left(\overline{ALP}_{(x_0,F_A^0)}\right) \quad \inf_{\Theta} \frac{g(x_0, F_A^0)}{\gamma^*} \Theta(F_A^0)$$

subject to:

$$\frac{g(x, F_I)}{\gamma^*} \Theta(F_A) \geq -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[\frac{1 - e^{-\gamma^* \tau(\phi)}}{\gamma^*} \right] g(x, F_I) + e^{-\beta \tau(\phi)} E \left[ \frac{g(X(\tau(\phi)), F_I \cup \phi)}{\tau(\phi)} \Theta(F_A \setminus \phi) | X(0) = x \right] \forall ((x, F_A), \phi) \in \mathcal{X} \text{ s.t. } x \in \mathcal{X}_x \cap \mathcal{R}_{x_0}.$$ (35)

with optimal value $\overline{ALP}_{(x_0,F_A^0)}^\ast$. By Propositions 12 and 13, the multipliers $\{\Theta^*(F_A) : F_A \subseteq F_A^0\}$ obtained by solving this linear program provide the tightest upper bound on $V^*(x_0, F_A^0)$ among all approximations of the form (34). We proceed by showing that the semi-infinite program $\overline{ALP}_{(x_0,F_A^0)}$ reduces to the following finite linear program:

$$\left(\overline{ALP}_{(x_0,F_A^0)}\right) \quad \inf_{\Theta} \frac{g(x_0, F_A^0)}{\gamma^*} \Theta(F_A^0)$$

subject to:

$$\frac{g(x, F_I)}{\gamma^*} \Theta(F_A) \geq -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[\frac{1 - e^{-\gamma^* \tau(\phi)}}{\gamma^*} \right] g(x, F_I) + e^{-\gamma^* \tau(\phi)} \frac{g(x_0, F_I \cup \phi)}{\gamma^*} \Theta(F_A \setminus \phi) \forall F_A \subseteq F_A^0 \forall \phi \subseteq F_A,$$ (36)

with optimal value $\overline{ALP}_{(x_0,F_A^0)}^\ast$. Unlike $\overline{ALP}_{(x_0,F_A^0)}$, the linear program $\overline{ALP}_{(x_0,F_A^0)}$ only has constraints corresponding to the initial intensity level $x_0$.

**Proposition 15.** There exists an optimal solution $\{\Theta^*(F_A) : F_A \subseteq F_A^0\}$ to $\overline{ALP}_{(x_0,F_A^0)}$, and $\overline{ALP}_{(x_0,F_A^0)}^\ast = ALP_{(x_0,F_A^0)}^\ast$.

**Proof.** By Theorem 2, the optionality multipliers $\{\Lambda^*(F_A) : F_A \subseteq F_A^0\}$ are a feasible solution to $\overline{ALP}_{(x_0,F_A^0)}$ because $c > 0$. Furthermore, Assumptions 4 and 5 indicate that $g(x_0, F_I) > 0$ for all $F_A \subseteq F_A^0$. The constraints corresponding to the set $\{(F_A, \emptyset) : F_A \subseteq F_A^0\}$ thus indicate that for any feasible solution $\Theta$, $\Theta(F_A) \geq 1$ for all $F_A \subseteq F_A^0$. The optimal value $\overline{ALP}_{(x_0,F_A^0)}^\ast$ is therefore finite.
Hence, there exists an optimal solution to the linear program $\tilde{ALP}_{(x_0, F_A^0)}$ since it is finite, feasible, and bounded.

The process $N$ is Poisson by Assumption 6, and so the definition of the set $X_{\theta^*}$ together with the proof of Theorem 2 indicates:

$$
e^{-\beta(\phi)} E \left[ \frac{g(X(\tau(\phi)), F_I \cup \phi)}{\gamma^*} \Theta(F_A \setminus \phi) \bigg| X(0) = x \right] = e^{-\gamma^* \tau(\phi)} g(x, F_I \cup \phi) \Theta(F_A \setminus \phi) \quad \forall ((x, F_A), \phi) \in \mathcal{X} \quad \text{s.t.} \quad x \in X_{\theta^*} \cap \mathcal{R}_{x_0},$$

which converts the constraints (35) into: for all $((x, F_A), \phi) \in \mathcal{X}$ such that $x \in X_{\theta^*} \cap \mathcal{R}_{x_0}$,

$$\frac{g(x, F_I)}{\gamma^*} \Theta(F_A) \geq -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[ 1 - e^{-\gamma^* \tau(\phi)} \right] g(x, F_I) + e^{-\gamma^* \tau(\phi)} g(x, F_I \cup \phi) \Theta(F_A \setminus \phi). \quad (37)$$

Dividing both sides by $\frac{g(x, F_I)}{\gamma^*}$, and employing the definition of $\mathcal{R}_{x_0}$ yields the equivalent set of constraints:

$$\Theta(F_A) \geq \frac{-c \gamma^*}{g(x, F_I)} \cdot 1_{\{\phi \neq \emptyset\}} + \left( 1 - e^{-\gamma^* \tau(\phi)} \right) + e^{-\gamma^* \tau(\phi)} b(F_I \cup \phi, F_I) \Theta(F_A \setminus \phi) \quad \forall ((x, F_A), \phi) \in \mathcal{X} \quad \text{s.t.} \quad x \in X_{\theta^*} \cap \mathcal{R}_{x_0}. \quad (38)$$

However, for any $F_A$ and $\phi \subseteq F_A$, all constraints in (38) with $x > x_0$ are redundant since $-c \leq 0$ and $g(x, F_I)$ is nonincreasing in $x$ by Assumption 5. Hence, to solve $\tilde{ALP}_{(x_0, F_A^0)}$ it suffices to solve the following linear program:

$$\inf_{\theta} \frac{g(x_0, F_I)}{\gamma^*} \Theta(F_A^0)$$

$$\text{s.t.} \quad \frac{g(x_0, F_I)}{\gamma^*} \Theta(F_A) \geq -c \cdot 1_{\{\phi \neq \emptyset\}} + \left[ 1 - e^{-\gamma^* \tau(\phi)} \right] g(x_0, F_I) + e^{-\gamma^* \tau(\phi)} g(x_0, F_I \cup \phi) \Theta(F_A \setminus \phi) \quad \forall F_A \subseteq F_A^0 \quad \forall \phi \subseteq F_A,$$

which we obtain from $\tilde{ALP}_{(x_0, F_A^0)}$ by retaining only those constraints in (37) corresponding to the intensity level $x_0$. The result follows as this linear program is identical to $\tilde{ALP}_{(x_0, F_A^0)}$. □

The linear program $\tilde{ALP}_{(x_0, F_A^0)}$ has $2\|F_A\|$ variables and $3\|F_A\|$ constraints. As we show in §6.4, when $F_A$ is sufficiently small we can solve $\tilde{ALP}_{(x_0, F_A^0)}$ numerically to obtain the minimizing values $\Theta^*$ and an upper bound on $V^*(x_0, F_A^0)$.

For any $(x, F_A) \in \mathcal{J}$, let $\{\Theta^*(F_A'; (x, F_A)) : F_A' \subseteq F_A\}$ denote the optimal solution to the linear program $ALP_{(x, F_A)}$ constructed by substituting $(x, F_A)$ for $(x_0, F_A^0)$ in the definition of $\tilde{ALP}_{(x_0, F_A^0)}$. Our approximation technique thus gives rise to the following decision rule:

$$d_{\theta^*}(x, F_A) \in \arg \max_{\phi \subseteq F_A} \left\{ -c \cdot 1_{\{\phi \neq \emptyset\}} + \left( 1 - e^{-\gamma^* \tau(\phi)} \right) g(x, F_I) + e^{-\gamma^* \tau(\phi)} g(x, F_I \cup \phi) \Theta^*(F_A \setminus \phi; (x, F_A)) \right\} \quad \forall (x, F_A) \in \mathcal{J}, \quad (39)$$

which we evaluate at each decision epoch by re-solving $\tilde{ALP}$. For the remainder of the paper, we evaluate the performance of the stationary deterministic policy $d_{\theta^*}$. We denote the value of this policy by $V_{\theta^*}$. 
Throughout the following analysis, we assume that $z$ While the certainty-equivalent dynamic program (CE-DP) may be easier to solve than our SMDP, with Jensen’s inequality yields transitions along its expected path. The release manager’s inter-epoch reward function becomes heuristic corresponding to the solution of a deterministic dynamic program in which the market

$$r_{CE}((x, F_A), \phi) := -c \cdot 1_{\{\phi \neq \emptyset\}} + \int_0^{\tau(\phi)} e^{-\beta u} g \left( E \left[ X(u) \left| X(0) = x \right. \right], F_I \right) du \quad \forall ((x, F_A), \phi) \in \mathcal{X}, \tag{40}$$

and she constructs a certainty-equivalent heuristic by solving the following optimality equations:

$$v_{CE}^*(x, F_A) = \max_{\phi \subseteq F_A} \left\{ r_{CE}((x, F_A), \phi) + e^{-\beta \tau(\phi)} v_{CE}^* \left( E \left[ X(\tau(\phi)) \left| X(0) = x \right. \right], F_A \setminus \phi \right) \right\} \quad \forall (x, F_A) \in \mathcal{X}. \tag{41}$$

While the certainty-equivalent dynamic program (CE-DP) may be easier to solve than our SMDP, we now show that it can perform arbitrarily poorly compared to the approximation heuristic $d_{\Theta}^\infty$. Throughout the following analysis, we assume that $g(x, F_I) = \frac{z(F_I)}{x^\alpha}$ is defined as in (9) with $\alpha = 1$.

First, we simplify the inter-epoch reward function (40). Assumption 6 indicates $E \left[ X(u) \left| X(0) = x \right. \right] = xe^{\lambda \Gamma[\xi]^{-1}u}$ for all $x \in [x_0, \infty)$. The functional form of $g$ from (9) thus implies

$$r_{CE}((x, F_A), \phi) = -c \cdot 1_{\{\phi \neq \emptyset\}} + \int_0^{\tau(\phi)} e^{-\beta u} \frac{z(F_I)}{xe^{\lambda \Gamma[\xi]^{-1}u}} du$$

$$= -c \cdot 1_{\{\phi \neq \emptyset\}} + \frac{z(F_I)}{\gamma'x} \left( 1 - e^{-\gamma' \tau(\phi)} \right) \quad \forall ((x, F_A), \phi) \in \mathcal{X}, \tag{42}$$

with $\gamma' := \beta + \lambda (E[\xi] - 1)$. As in our SMDP, the certainty-equivalent dynamic program also utilizes a modified discount factor to account for market intensity. However, the next result shows that the certainty-equivalent discount factor $\gamma^*$ is conservative relative to the modified discount factor $\gamma'$.

**Proposition 16.** The discount factor $\gamma^*$ is strictly less than the discount factor $\gamma'$.

**Proof.** It follows from the definitions of $\gamma^*$ and $\gamma'$ that

$$\gamma^* < \gamma' \iff \left( 1 - E \left[ \frac{1}{\xi} \right] \right) < (E[\xi] - 1). \tag{43}$$

Jensen’s inequality yields

$$1 - E \left[ \frac{1}{\xi} \right] < 1 - \frac{1}{E[\xi]},$$

and so it suffices to prove that $1 - \frac{1}{E[\xi]} \leq (E[\xi] - 1)$. However, (43) thus follows immediately since

$$1 - \frac{1}{E[\xi]} \leq E[\xi] - 1 \iff 0 \leq (E[\xi])^2 - 2E[\xi] + 1 = (E[\xi] - 1)^2$$

and $(E[\xi] - 1)^2 \geq 0$. \(\Box\)

We now exploit the discrepancy in the two discount factors to expose the performance gap between the certainty-equivalent heuristic and our approximation policy.

**Proposition 17.** Assume that $c > 0$, $F_0^\alpha = \{f\}$, and that $g(x, F_I) = \frac{z(F_I)}{x^\alpha}$ is defined as in (9) with $\alpha = 1$. For any $J \geq 1$, there exists a value $H \geq 0$ and intensity level $\tilde{x}$ such that if $z(F_I^0 \cup \{f\}) - z(F_I^0) \geq H$, then

$$\frac{V_{\Theta^*}(\tilde{x}, \{f\})}{V_{CE}(\tilde{x}, \{f\})} \geq J.$$

**Proof.**
Proof. First, we compute the decision rule \( d_{\Theta^*} \). For arbitrary \( x \in [x_0, \infty) \), it follows by inspection that \( \Theta^*(\emptyset; (x, \{ f \})) = 1 \) for the linear program \( \widetilde{ALP}_{(x, \{ f \})} \). The definition \((39)\) thus implies that \( d_{\Theta^*}(x, \{ f \}) = \{ f \} \) if and only if
\[
-c + \left(1 - \frac{\gamma' - \tau(f)}{\gamma^*} \right) \frac{z(F_0)}{x} + \frac{\gamma' - \tau(f)}{\gamma^*} \frac{z(F_0 \cup \{ f \})}{x} \geq \frac{z(F_0)}{\gamma^* x} \]
\[
\iff -c + \frac{\gamma' - \tau(f)}{\gamma^*} \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right] \geq 0
\]
\[
\iff x \leq \pi_{\Theta^*},
\]
with
\[
\pi_{\Theta^*} := \sup \left\{ x \in \mathbb{R} \mid -c + \frac{\gamma' - \tau(f)}{\gamma^*} \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right] \geq 0 \right\}
\]
\[
= \left( \frac{\gamma' - \tau(f)}{c \gamma^*} \right) \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right] \quad \text{since} \ c > 0.
\]
By Assumption 5, \( z(F_0 \cup \{ f \}) - z(F_0) > 0 \) and so \( \pi_{\Theta^*} \) must be positive. For simplicity, we assume that \( \pi_{\Theta^*} > x_0 \). It follows that
\[
V_{\Theta^*}(x, \{ f \}) = \begin{cases} 
-c + \frac{z(F_0)}{\gamma^* x} + \frac{\gamma' - \tau(f)}{\gamma^*} \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right], & x_0 \leq x \leq \pi_{\Theta^*} \\
\frac{z(F_0)}{\gamma^* x}, & x \in (\pi_{\Theta^*}, \infty)
\end{cases}
\]
\[
V_{\Theta^*}(x, \emptyset) = \frac{z(F_0 \cup \{ f \})}{\gamma^* x} \quad \forall x \in [x_0, \infty).
\]

In Appendix C, we show that the certainty-equivalent heuristic selects the waiting action if market intensity exceeds the threshold
\[
\pi_{\text{CE}} := \sup \left\{ x \in \mathbb{R} \mid -c + \frac{\gamma' - \tau(f)}{\gamma^*} \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right] \geq 0 \right\}
\]
\[
= \left( \frac{\gamma' - \tau(f)}{c \gamma^*} \right) \left[ \frac{z(F_0 \cup \{ f \})}{x} - \frac{z(F_0)}{x} \right] \quad \text{because} \ c > 0.
\]
For simplicity, we assume that \( \pi_{\text{CE}} > x_0 \). If \( x > \pi_{\text{CE}} \), the value of the CE heuristic in the context of our SMDP is thus
\[
V_{\text{CE}}(x, \{ f \}) = \frac{z(F_0)}{\gamma^* x} \quad \forall x \in (\pi_{\text{CE}}, \infty).
\]
because the heuristic selects the waiting action at every decision epoch.

Comparing \((44)\) to \((45)\) and applying Proposition 16 immediately indicates that \( \pi_{\text{CE}} < \pi_{\Theta^*} \). Therefore, for any \( x \in (\pi_{\text{CE}}, \pi_{\Theta^*}) \) the approximate policy opts to develop the feature \( f \) while the CE heuristic decides to wait. We now exploit this observation by considering a specific intensity level between the two thresholds.

Let \( J \geq 1 \) be arbitrary and define \( \hat{x} := (\pi_{\text{CE}} \pi_{\Theta^*})^{1/2} \), the geometric mean of the two thresholds. Since \( \pi_{\text{CE}} < \hat{x} < \pi_{\Theta^*} \), we have:
\[
\frac{V_{\Theta^*}(\hat{x}, \{ f \})}{V_{\text{CE}}(\hat{x}, \{ f \})} = -c + \frac{1 - \frac{\gamma' - \tau(f)}{\gamma^*}}{\gamma^*} \frac{z(F_0)}{x} + \frac{\gamma' - \tau(f)}{\gamma^*} \frac{z(F_0 \cup \{ f \})}{x} \frac{z(F_0 \cup \{ f \})}{\gamma^* x}
\]
The performance gap between the two policies tends to increase in optimality and consistently beats the CE-DP heuristic. The second plot shows that the performance average solution time for coded in AMPL and solved with Gurobi 5.0 (6 core Intel Xeon at 3.33 GHz, 4 GB of RAM). The 12 releases per year, and (x) $z$ assumed: (i) the maximizing action ($d^*$) is exact when $H$ since $\gamma^* < \gamma'$ by Proposition 16.

In summary, the certainty-equivalent heuristic performs poorly in this case because the associated discount factor $\gamma'$ induces an overly conservative termination threshold $\pi_{CE}$. The penalty for this conservatism increases without bound as the incremental value of the feature $f$ increases.

### 6.4. Numerical Performance

We evaluated the performance of the approximation-based policy $d^\infty_{\tilde{A}}$, as follows: (i) solve $ALP_{(x_0, F_{\tilde{A}})}$ to obtain an upper bound on $V^* (x_0, F_{\tilde{A}})$, (ii) simulate $d^\infty_{\tilde{A}}$ along 1,000 randomly generated paths, re-solving $ALP$ at each decision epoch and identifying the maximizing action (39) via enumeration, and (iii) compare the average performance of the policy to the upper bound calculated in (i) to obtain an upper bound on the optimality gap of $d^\infty_{\tilde{A}}$. In addition, we also simulated the CE-DP heuristic along the same set of paths and compared its average performance to that of the approximation-based policy. Unless otherwise noted, we assumed: (i) $g(x, F_t) = \frac{z(F_t)}{\lambda} e^{-\gamma^* r(f)}$ as in (9) with $\alpha = 1$, (ii) $|F_{\tilde{A}}| = 10$, (iii) $c = 10$, (iv) $x_0 = 10$, (v) $z(F_t) = 10$, (vi) $\xi \sim \text{Unif}[1, 1.1]$, (vii) $\tau$ is linear, (viii) $z$ is linear, (ix) $\lambda$ corresponds to an expected 12 releases per year, and (x) $\beta$ corresponds to a 10% annual interest rate. All linear programs were coded in AMPL and solved with Gurobi 5.0 (6 core Intel Xeon at 3.33 GHz, 4 GB of RAM). The average solution time for $ALP_{(x_0, F_{\tilde{A}})}$ was 1.69 seconds, with a standard deviation of 0.15 seconds.

The statistics for solving CE-DP via linear programming were nearly identical.

By construction, the policy $d^\infty_{\tilde{A}}$ is exact when $c = 0$. The first plot in Figure 2 indicates that the approximation-based policy also performs well for strictly positive fixed costs. Even though the policy’s performance degrades slightly for higher values of $c$, it always performs within 3.5% of optimality and consistently beats the CE-DP heuristic. The second plot shows that the performance gap between the two policies tends to increase in $c$, with a maximum observed gap of 5.5%.

Our second set of trials evaluates the approximation-based policy against markets with varying shock frequency and magnitude. Fixed costs were held constant at the default value of 10. Figure 3 displays the results for the case of uniformly distributed shocks. Lognormal shocks were also considered, with nearly identical results. As indicated by the first plot, the approximation-based policy performed within 2% of optimality in all scenarios regardless of the market shock frequency. In some cases, the expected value of our policy slightly exceeds the upper bound obtained from $ALP_{(x_0, F_{\tilde{A}})}$. This behavior typically indicates that the policy is high-performing, and a much larger number of paths is required to obtain a more accurate estimate of the policy’s value. As in the first set of trials, the second plot shows that our policy consistently outperforms the certainty-equivalent heuristic. Furthermore, the performance gap increases as the market worsens, i.e., exhibits higher average shock frequency and expected shock magnitude.

Finally, we discuss an additional advantage of our approximation-based policy relative to the CE-DP heuristic. Even though the CE-DP has an uncountable state space, the fact that its transitions...
are deterministic means we can solve it over a *finite* state space by tracking the set of available features as well as the expected intensity level given the amount of time that has elapsed since the beginning of the horizon. Suppose the release manager has the set $F_A \subseteq F_0^A$ of features available for development. If $\tau$ is additive, then, *regardless* of the order in which previous releases were implemented, the total time required for all earlier releases must be $\tau(F_0^A \setminus F_A)$. However, if $\tau$ is
not additive, then we must account for all possible order-dependent release sequences that result in the current set of available features $F_A$. The memory requirements for storing the expanded state space quickly become prohibitive. For instance, with only ten features the number of states in the expanded CE-DP is roughly 660 times greater than the number of states required when $\tau$ is additive. Our approximation-based policy does not suffer from this issue. Furthermore, when we performed trials with non-additive specifications for $\tau$, the results and solution times for our approximation-based policy were very similar to those from the additive case discussed earlier.

7. Conclusion By adapting the assumptions from Adelman and Mancini (2014) to the release management problem, we obtained two of the key ingredients for our analysis: (i) the existence of a quasi-open-loop optimal policy, and (ii) the ability to study this policy by analyzing a reduced dynamic program that utilizes a modified discount factor. While we can relax some of the assumptions presented here without invalidating our approach, doing so would complicate our analysis without leading to additional insights. For example, we retain the existence of a quasi-open-loop optimal policy even if the shock-time process $N$ is a generic counting process, so long as it still satisfies the exogeneity assumptions in §4. However, there no longer exists a modified discount factor that encodes the deflationary impact of market intensity on revenues. Similarly, Assumption 6, part (iii) of Assumption 9, and Assumption 11 together suffice for the existence of a modified discount factor akin to the one identified in Proposition 1. However, parts (i) and (ii) of Assumption 9, as well as Assumption 10, ease our analysis by ensuring that this modified discount factor does not depend on the intensity level, the release manager’s actions, or the release manager’s product. Hence, the same modified discount factor applies at every epoch.

The third crucial component of our analysis is the assumption that development times are deterministic and depend only on the features being developed. The determinism of $\tau$ makes the reduced dynamic program used to analyze the quasi-open-loop optimal policy deterministic and, therefore, more tractable. However, if the development time mapping $\tau$ is a strictly positive random variable parameterized by the release manager’s action $\phi$ and independent of the release manager’s existing product $F_I$ and market intensity, then we essentially retain all of our analytical results by taking expectations. For instance, the single-feature index from Corollary 1 becomes

$$\frac{E\left[e^{-\gamma^*\tau(f)}\right]g(x_0,\{f\})}{1 - E\left[e^{-\gamma^*\tau(f)}\right]},$$

which corresponds to the “weighted discounted shortest expected processing time first” rule from the stochastic scheduling literature (Pinedo (2012)). A second extension is to allow development times to depend on the set of features in the release manager’s product. Theorems 2 and 3 remain valid in this case with only minor adjustments, as does the ADP implementation discussed in §6. However, this extension requires modifications to the exchange condition in Proposition 8 that obfuscate any underlying insights (Appendix D). Finally, we note that a third consequence of our assumption on development times is that the release manager can not react to a market shock until the current iteration of her product is complete. In fact, we can model $\tau$ as a randomized stopping time of the shock-time process $N$ without sacrificing the existence of a quasi-open-loop optimal policy (Adelman and Mancini (2014)). However, we lose the tractability afforded by the modified discount factor from Proposition 1.

Our results suggest several avenues for further research. First, one could consider embedding our tactical model in a framework in which a strategic analysis of rival products directs the research and development efforts that ultimately furnish the features available to the release manager. Similarly, strategic investment decisions would influence the release manager’s development capabilities, and therefore define the $\tau$ mapping. Second, our SMDP does not allow for new features to arrive over time. While a release manager could accommodate random arrivals by updating the set of available
features at each decision epoch when implementing the approximate policy discussed in §6, our results would no longer provide an upper bound on the value function with which to evaluate the performance of the approximate policy. Recovering an upper bound entails the difficult task of extending the SMDP state space to account for the entire universe of possible features, and solving the associated infinite dimensional linear program. Third, we believe that exploring the relationship between multiplicative separability and primitive models of consumer choice would further inform managers of the applicability of our results. Fourth, our model assumes that new releases have an instantaneous impact on revenues. Allowing consumer decisions to take place over time necessitates a generalization of the crucial modified discount factor result (Proposition 1). Similarly, we do not consider the switching costs consumers incur when transitioning to a new release, or the impact of limited market size on the release manager’s decisions.

From a computational perspective, the numerical performance of the approximate policy in §6 justifies research into the scalability of the underlying linear program \( \widetilde{ALP} \). Since the size of this linear program grows exponentially in the number of available features, we likely can not capture large-scale instances of the release management problem. However, discussions with managers at a major software firm indicate that it is typical for a small number of high priority features to dominate the feature selection process, and so our existing approach still provides value. Nevertheless, a scalable algorithm for our approximate linear program would make the model more applicable.

To the best of our knowledge, our model is the first to consider release management from an operational perspective in a dynamic, combinatorial setting. We provided significant managerial insights in the case with no fixed costs, and built on this analysis to construct a high-performing policy for release managers facing positive fixed costs. Our hope is that this work serves as the foundation for a fruitful discussion of the combinatorial issues involved in dynamic release management.

Appendix A: Proof of Proposition 7 Combining the relation (17) with a recursive application of (11) yields:

\[
\Lambda^* (F_A^0) = \left(1 - e^{-\gamma^* \tau(\phi_i^*)}\right) + e^{-\gamma^* \tau(\phi_i^*)} b(F_I^0 \cup \phi_i^*, F_I^0) \Lambda^* (F_A^0 \setminus \phi_i^*)
\]

\[
= \left(1 - e^{-\gamma^* \tau(\phi_i^*)}\right) + e^{-\gamma^* \tau(\phi_i^*)} b(F_I^0 \cup \phi_i^*, F_I^0) \cdot \left[ \left(1 - e^{-\gamma^* \tau(\phi_2^*)}\right) + e^{-\gamma^* \tau(\phi_2^*)} b(F_I^0 \cup \phi_i^*, \phi_2^*, F_I^0 \cup \phi_1^*) \Lambda^* (F_A^0 \setminus (\phi_i^* \cup \phi_2^*)) \right] \quad \text{by (11)}
\]

\[
= \left(1 - e^{-\gamma^* \tau(\phi_i^*)}\right) + \left(1 - e^{-\gamma^* \tau(\phi_2^*)}\right) e^{-\gamma^* \tau(\phi_1^*)} b(F_I^0 \cup \phi_i^*, F_I^0) + e^{-\gamma^* \tau(\phi_1^*)} e^{-\gamma^* \tau(\phi_2^*)} b(F_I^0 \cup \phi_i^*, \phi_2^*, F_I^0) \Lambda^* (F_A^0 \setminus (\phi_i^* \cup \phi_2^*)) \quad \text{by defn. (8) of } b
\]

\[
\vdots
\]

\[
= \sum_{i=1}^{j^*} \left(1 - e^{-\gamma^* \tau(\phi_i^*)}\right) e^{-\gamma^* \left[\sum_{n=1}^{i-1} \tau(\phi_n^*)\right]} b(F_I^0 \cup \left(\bigcup_{n=1}^{i-1} \phi_n^* \right), F_I^0) + e^{-\gamma^* \left[\sum_{n=1}^{i-1} \tau(\phi_n^*)\right]} b(F_I^0 \cup F_A^0, F_I^0).
\]

(46)

Substituting the definition (12) into (46) and applying the definition (8) of \( b \), we obtain:

\[
V^*(x_0, F_A^0) = \frac{\sum_{i=1}^{j^*} \left(1 - e^{-\gamma^* \tau(\phi_i^*)}\right) e^{-\gamma^* \left[\sum_{n=1}^{i-1} \tau(\phi_n^*)\right]} g(x_0, F_I^0 \cup (\bigcup_{n=1}^{i-1} \phi_n^*))}{e^{-\gamma^* \left[\sum_{n=1}^{j^*} \tau(\phi_n^*)\right]} g(x_0, F_I^0 \cup F_A^0)} + \frac{e^{-\gamma^* \left[\sum_{n=1}^{j^*} \tau(\phi_n^*)\right]} g(x_0, F_I^0 \cup F_A^0)}{\gamma^*}.
\]
\begin{align*}
&= \sum_{i=1}^{j^*} e^{-\gamma^*[\sum_{n=1}^{i-1} r(\phi_n^*)]} \frac{g(x_0, F^0_I \cup (\bigcup_{n=1}^{i-1} \phi_n^*))}{\gamma^*} \\
&\quad + \sum_{i=1}^{j^*} e^{-\gamma^*[\sum_{n=1}^{i} r(\phi_n^*)]} \frac{g(x_0, F^0_I \cup (\bigcup_{n=1}^{i} \phi_n^*)) + e^{-\gamma^*[\sum_{n=1}^{i} r(\phi_n^*)]} g(x_0, F^0_I \cup F^0_A)}{\gamma^*} \\
&= \frac{g(x_0, F^0_I)}{\gamma^*} + \sum_{i=1}^{j^*} e^{-\gamma^*[\sum_{n=1}^{i} r(\phi_n^*)]} \left[ g(x_0, F^0_I \cup (\bigcup_{n=1}^{i} \phi_n^*)) - g(x_0, F^0_I \cup (\bigcup_{n=1}^{i-1} \phi_n^*)) \right],
\end{align*}

which completes the proof. \qed

**Appendix B: Proof of Proposition 13** By construction, \(LP^*_{(x_0,F^0_A)} \geq \overline{LP}^*_{(x_0,F^0_A)}\). Suppose now that \(\tilde{v}\) is an arbitrary feasible solution to \(\overline{LP}^*_{(x_0,F^0_A)}\), and for all \(F_A \subseteq F_A^0\) let

\[\tilde{v}(x, F_A) := \begin{cases} 
\tilde{v}(x, F_A), & \forall x \in X_{\phi^*} \cap R_{x_0} \\
C, & \forall x \in \{x \in [x_0,\infty) | x \notin X_{\phi^*} \cap R_{x_0}\}
\end{cases}\]

for some \(C \in [0,\infty)\). By construction, \(\tilde{v} \in V_b\). We proceed by showing that for sufficiently large values of \(C\), the function \(\tilde{v}\) is a feasible solution to \(LP^*_{(x_0,F^0_A)}\). Hence, \(LP^*_{(x_0,F^0_A)} \leq \overline{LP}^*_{(x_0,F^0_A)}\) and so the result follows.

By Lemma 2 in Adelman and Mancini (2014), if \(x \in X_{\phi^*} \cap R_{x_0}\), then

\[P\left(X(u) \in (X_{\phi^*} \cap R_{x_0}) \bigg| X(0) = x\right) = 1\]

for all \(u \in [0,\infty)\). The constraints of the linear program \(LP^*_{(x_0,F^0_A)}\) evaluated at \(\tilde{v}\) thus become:

\[\tilde{v}(x, F_A) \geq r\left((x, F_A), \phi\right) + e^{-\beta r(\phi)} E\left[\tilde{v}(X(\tau(\phi)), F_A \setminus \phi) \bigg| X(0) = x\right] \quad \forall ((x, F_A), \phi) \in \mathcal{X} \text{ s.t. } x \in X_{\phi^*} \cap R_{x_0}\]

\[C \geq r\left((x, F_A), \phi\right) + e^{-\beta r(\phi)} \left\{ E\left[\tilde{v}(X(\tau(\phi)), F_A \setminus \phi) \bigg| X(\tau(\phi)) \in (X_{\phi^*} \cap R_{x_0}) \right] \right\} \cdot P\left(X(\tau(\phi)) \notin (X_{\phi^*} \cap R_{x_0}) \right) + C \cdot P\left(X(\tau(\phi)) \notin (X_{\phi^*} \cap R_{x_0}) \right) \forall ((x, F_A), \phi) \in \mathcal{X} \text{ s.t. } x \notin X_{\phi^*} \cap R_{x_0}\]

The first set of constraints are satisfied since \(\tilde{v}\) is a feasible solution of \(\overline{LP}^*_{(x_0,F^0_A)}\). Furthermore, the following inequality implies that, for sufficiently large values of \(C\), the second set of inequalities is also satisfied:

\[e^{-\beta r(\phi)} P\left(X(\tau(\phi)) \notin (X_{\phi^*} \cap R_{x_0}) \right) \leq 1 \quad \forall \phi \subseteq F_A^0 \forall x \geq 0. \]

**Appendix C: Supplement to Proof of Proposition 17** We start by showing that

\[v_{CE}^*(x, \{f\}) \geq \frac{z(F^0)}{\gamma x} \quad \forall x \in [x_0, \infty), \quad (47)\]
with the CE-DP value function $v_{CE}^*$ defined as in (41).

Recall that $\tau (0) = w > 0$. The CE-DP optimality equations imply

$$v_{CE}^*(x, \{f\}) \geq \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) + e^{-\beta w} v_{CE}^*(xe^{\lambda(E[F]-1)w}, \{f\}) \quad \forall x \in [x_0, \infty).$$

(48)

Fixing a value $x \in [x_0, \infty)$ and recursively applying (48), we obtain:

$$v_{CE}^*(x, \{f\}) \geq \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) + e^{-\beta w} v_{CE}^*(xe^{\lambda(E[F]-1)w}, \{f\})$$

$$\geq \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) + e^{-\beta w} \left[\frac{z (F_0^0)}{\gamma' xe^{\lambda(E[F]-1)w}} \left(1 - e^{-\gamma' w}\right)\right]$$

$$= \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) \left[1 + e^{-\gamma' w}\right] + e^{-\beta w} \left[\frac{z (F_0^0)}{\gamma' xe^{\lambda(E[F]-1)(2w)}} \left(1 - e^{-\gamma' w}\right)\right]$$

$$\geq \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) \left[1 + e^{-\gamma' w} + e^{-\gamma' (2w)}\right] + e^{-\beta (3w)} v_{CE}^* \left(xe^{\lambda(E[F]-1)(3w)}, \{f\}\right)$$

$$\vdots$$

$$\geq \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' w}\right) \left(\sum_{i=0}^{n} e^{-\gamma' (iw)}\right) + e^{-\beta ((n+1)w)} v_{CE}^* \left(xe^{\lambda(E[F]-1)(n+1)w}, \{f\}\right)$$

\forall n \in \mathbb{N}.

Since $\beta > 0$ and $v_{CE}^*$ is bounded, letting $n \to \infty$ yields (49).

Next, a similar recursion indicates that

$$v_{CE}^*(x, \emptyset) = \frac{z (F_0^0 \cup \{f\})}{\gamma' x} \quad \forall x \in [x_0, \infty)$$

because the release manager must terminate development once the only available feature $f$ has been implemented.

Finally, let $x$ be an arbitrary intensity level that exceeds the certainty-equivalent threshold, i.e.,

$$x > \bar{x}_{CE} = \sup \left\{ x \in \mathbb{R} \mid -c + \frac{e^{-\gamma' \tau(f)}}{\gamma'} \left[\frac{z (F_0^0 \cup \{f\}) - z (F_0^0)}{x}\right] \geq 0 \right\}. \quad (50)$$

Applying (47), we obtain:

$$v_{CE}^*(x, \{f\}) \geq \frac{z (F_0^0)}{\gamma' x}$$

$$\geq \frac{z (F_0^0)}{\gamma' x} + \left\{-c + \frac{e^{-\gamma' \tau(f)}}{\gamma'} \left[\frac{z (F_0^0 \cup \{f\}) - z (F_0^0)}{x}\right]\right\} \quad \text{by (50)}$$

$$= -c + \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' \tau(f)}\right) + e^{-\gamma' \tau(f)} \frac{z (F_0^0 \cup \{f\})}{\gamma' x}$$

$$= -c + \frac{z (F_0^0)}{\gamma' x} \left(1 - e^{-\gamma' \tau(f)}\right) + e^{-\beta \tau(f)} v_{CE}^* \left(xe^{\lambda(E[F]-1)\tau(f)}, \emptyset\right) \quad \text{by (49)}.$$

The final inequality indicates that developing feature $f$ in state $(x, \{f\})$ is suboptimal, and so the waiting action is optimal. \qed
Appendix D: Product-Dependent Development Times Suppose that the development time mapping $\tau$ is deterministic, but depends on the set of features currently in the release manager’s product. Letting $\tau(\phi; \phi')$ denote the development time for the bundle $\phi$ when the existing product consists of the features $F_0^i \cup \phi'$, the exchange argument applied in Proposition 8 yields: for all $m \in \{1, 2, \ldots, j^* - 1\}$,

$$
e^{-\gamma^* \tau(\phi_m; \cup_{n=1}^{m-1} \phi_n)} \left[ g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n)) - g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n^*) \right] 1 - e^{-\gamma^* \tau(\phi_m; \cup_{n=1}^{m-1} \phi_n \cup m+1)} + (51)$$

$$
\frac{e^{-\gamma^* \tau(\phi_m; \cup_{n=1}^{m-1} \phi_n^*)} \left[ g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n) \cup \phi_{m+1}^*) - g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n^*) \right]} {1 - e^{-\gamma^* \tau(\phi_m; \cup_{n=1}^{m-1} \phi_n^*)}} \geq (52)
$$

$$
\frac{e^{-\gamma^* \tau(\phi_{m+1}; \cup_{n=1}^{m-1} \phi_n^*)} \left[ g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n) \cup \phi_{m+1}^*) - g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n^*) \right]} {1 - e^{-\gamma^* \tau(\phi_{m+1}; \cup_{n=1}^{m-1} \phi_n^*)}} + (53)
$$

$$
\frac{e^{-\gamma^* \tau(\phi_{m+1}; \cup_{n=1}^{m-1} \phi_n^*)} \left[ g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n) \cup \phi_{m+1}^*) - g(x_0, F_0^i \cup (\cup_{n=1}^{m-1} \phi_n^*) \right]} {1 - e^{-\gamma^* \tau(\phi_{m+1}; \cup_{n=1}^{m-1} \phi_n^*)}} \geq (54)
$$

When development times do not depend on existing functionality, terms (51) and (53) reduce to those from Proposition 8. Furthermore, the terms (52) and (54) cancel because it takes the same amount of time to develop $\phi_m^*$ followed by $\phi_{m+1}^*$ as it does to develop the bundles in reverse order. If development times are sensitive to the release manager’s product, however, (52) and (54) reflect the possibility that swapping $\phi_m^*$ and $\phi_{m+1}^*$ impacts the time at which both bundles are completed.

Acknowledgements. Both authors are grateful for the financial support provided by the University of Chicago Booth School of Business.

References


