New Markov Decision Process Formulations and Optimal Policy Structure for Assemble-to-Order and New Product Development Problems

by

EMRE NADAR

Submitted to the Tepper School of Business
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

at

CARNEGIE MELLON UNIVERSITY

26 April 2012

Dissertation Committee:
Professor Alan Scheller-Wolf (Chair)
Assistant Professor Mustafa Akan
Associate Professor Laurens Debo
Assistant Professor Fatma Kılınç-Karzan

Abstract

This thesis examines two complex, dynamic problems by employing the theory of Markov Decision Processes (MDPs). The first two chapters consider assemble-to-order (ATO) inventory systems. An ATO system consists of several components and several products, and assembles products as demand is realized; it is becoming increasingly popular since it provides greater flexibility in manufacturing at a reasonable cost. This work contributes to the ATO research stream by characterizing optimal inventory replenishment and allocation policies. The last chapter examines new product development (NPD) with scarce resources and many projects in parallel, each lasting several periods in the face of uncertainty. This study advances the NPD literature by revealing that optimal project selection and resource allocation decisions are congestion-dependent. Below, we elaborate on the novel optimal policies and structural results we obtain using MDP formulations, which is the overarching theme of each chapter.

In the first chapter, we consider generalized ATO "M-systems" with multiple components and multiple products. These systems involve a single "master" product which uses multiple units from each component, and multiple individual products each of which consumes multiple units from a different component. An example of such a system would be a manufacturer who sells an assembled product as well as individual spare parts. We model these systems as an infinite-horizon MDP under the discounted cost criterion. Each component is produced in batches of fixed size in a make-to-stock fashion; batch sizes are determined by individual product sizes. Production times are independent and exponentially distributed. Demand for each product arrives as an independent Poisson process. If not satisfied immediately upon arrival, these demands are lost. Therefore the state of the system can be described by component inventory levels. A control policy specifies when a batch of components should be produced (i.e., inventory replenishment), and whether an arriving demand for each product should be satisfied (i.e., inventory allocation). Since the convexity property that has been largely used to characterize optimal policies in the MDP literature may fail to hold, we introduce new functional characterizations for submodularity and supermodularity restricted to certain lattices of the state space. The optimal cost function satisfies these new characterizations: The state space of the problem can be partitioned into disjoint lattices such that, on each lattice, (a) it is optimal to produce a batch of a particular component if and only if the state vector is less than a certain threshold associated with that component, and (b) it is optimal to fulfill a demand of a particular product if and only if the state vector is greater than or equal to a certain threshold associated with that product. We refer to this policy as a *lattice-dependent base-stock and lattice-dependent rationing* (LBLR) policy. We also show that if the optimization criterion is modified to the average cost rate, LBLR remains optimal.

The first chapter makes three important contributions. First, this study is the first to characterize the optimal inventory replenishment and allocation policies for M-systems. Second, this study is the first to characterize the optimal policies for any ATO problem when different products may use the same component in different quantities. Third, we introduce new functional characterizations restricted to certain lattices of the state space, giving rise to a lattice-dependent policy.

In the second chapter, we evaluate the use of an LBLR policy for general ATO systems as a heuristic. We numerically compare the globally optimal policy to LBLR and two other heuristics from the literature: a state-dependent base-stock and state-dependent rationing (SBSR) policy, and a fixed base-stock and fixed rationing (FBFR) policy. Taking the average cost rate as our performance criterion, we develop a linear program to find the globally optimal cost, and Mixed Integer Programming formulations to find the optimal cost within each heuristic class. We generate more than 1800 instances for the general ATO problem, violating the M-system product structure. Interestingly, LBLR yields the globally optimal cost in all instances, while SBSR and FBFR provide solutions within 2.7% and 4.8% of the globally optimal cost, respectively. These numerical results also provide several insights into the performance of LBLR relative to other heuristics: LBLR and SBSR perform significantly better than FBFR when replenishment batch sizes imperfectly match the component requirements of the most valuable or most highly demanded product. In addition, LBLR substantially outperforms SBSR if it is crucial to hold a significant amount of inventory that must be rationed.

Based on the numerical findings in the second chapter, future research could investigate the optimality of LBLR for ATO systems with general product structures. However, as we construct counter examples showing that submodularity and supermodularity, which are used to prove the optimality of LBLR in the first chapter, need not hold for general ATO systems, showing optimality of LBLR for general ATO systems will likely require alternate proof techniques.

In the third chapter, we study the problem of project selection and resource allocation in a multi-stage new product development (NPD) process with stage-dependent resource constraints. As in the first two chapters, we model the problem as an infinite-horizon MDP under the discounted cost criterion. Each NPD project undergoes a different experiment in each stage of the NPD process; these experiments generate signals about the true nature of the project. Experimentation times are independent and exponentially distributed. Beliefs about the ultimate outcome of each project are updated after each experiment according to a Bayesian rule. Projects thus become differentiated through their signals, and all available signals for a project determine its category. The state of the system is described by the numbers of projects in each category. A control policy specifies when and at what rate to utilize the resources at each stage, and on which projects.

We characterize the optimal control policy as following a new type of strategy, *state-dependent non-congestive promotion* (SDNCP), for two different special cases of the general problem: (a) when there is a single informative experiment, or (b) when there are multiple uninformative experiments. An SDNCP policy implies that, at each stage, it is optimal to advance a project with the highest expected reward to the next stage if and only if the number of projects in each successor category is less than a state-dependent threshold. Specifically, threshold values decrease in a non-strict sense as a later stage becomes more congested or as an earlier stage becomes less congested; a stage becomes more congested with an increase in the number of projects at this stage or with an increase in the expected reward of any project at this stage. We also report the strong numerical performance of an SDNCP policy as a heuristic for the general problem. Taken together, these findings highlight the importance of congestion in optimal portfolio strategies.

Table of Contents

List of Figures

Chapter 1

Introduction

Markov Decision Processes (MDPs) offer an elegant mathematical framework for modeling arbitrarily complex, sequential decision problems that arise in operations research, management science, finance, and computer science, among many other fields. Fundamentally, MDPs enable researchers to theorize the dynamics of a stochastic process whose transition mechanism is controlled over time: The state of the process provides the decision maker with all the information necessary for choosing a feasible action in that state. The process responds to the chosen action by randomly evolving to a new state, and incurring a cost (or yielding a reward) to the decision maker. But the probability that the process transitions to its new state depends only on the current state and the chosen action; the state transitions of an MDP possess the *memoryless* property, which greatly simplifies the analysis of stochastic processes. (Due to its memoryless property, exponential distribution has been widely used in the MDP literature.)

The strength of MDPs lies in their ability to formulate a recursive value function for the expected future cost to be incurred (or reward to be earned); the optimal action as a function of the current state can be found by calculating the value function. In this context many researchers have studied various techniques, including dynamic programming and linear programming, to compute value functions. However, most computational methods suffer from the curse of dimensionality; their practical applications are limited to cases where the state space is manageably small and/or the value function has a simple analytical form. In an effort to handle computationally nontrivial problems, many other researchers have focused on the abstraction of structural properties from value functions, using inductive proof techniques.

Structural properties provide a powerful methodology for partial or complete characterization of optimal policies, which might have important managerial implications and might improve knowledge about computational methods.

This dissertation analyzes two complex, dynamic problems by employing the theory of MDP and deriving optimal structural results. Chapters 2 and 3 discuss assemble-to-order (ATO) inventory systems under Markovian assumptions on production and demand. An ATO system consists of several components and several products, and assembles products as demand is realized. ATO systems are becoming increasingly popular as they provide greater flexibility in manufacturing at a reasonable cost. This work contributes to the ATO research stream by characterizing optimal inventory replenishment and allocation policies. Chapter 4 examines the MDP representation of a new product development (NPD) process with scarce resources and many projects in parallel, each lasting several periods in the presence of uncertainty. Chapter 4 contributes to the NPD research stream by highlighting the importance of considering the effects of congestion in project selection and resource allocation decisions. In what follows we will elaborate on the novel optimal policies and computational results we obtain using MDP formulations in Chapters 2, 3, and 4.

Chapter 2 considers generalized ATO "M-systems" with multiple components and multiple products. These systems involve a single "master" product which uses multiple units from each component, and multiple "individual" products each of which consumes multiple units from a different component. (An example of such a system would be a manufacturer who sells an assembled product as well as individual spare parts.) We model these systems as an infinite-horizon MDP under the discounted cost criterion. Each component is produced in batches of fixed size in a make-to-stock fashion; batch sizes are determined by individual product sizes. Production times are independent and exponentially distributed. Demand for each product arrives as an independent Poisson process. If not satisfied immediately upon arrival, these demands are lost. Therefore the state of the system can be described by component inventory levels. A control policy specifies when a batch of components should be produced (i.e., inventory replenishment), and whether an arriving demand for each product should be satisfied (i.e., inventory allocation).

The convexity property, being among the most commonly investigated structural properties in the MDP literature, may fail to hold for the optimal cost function (i.e., the value function). To allow us to establish the optimal policy, we introduce new functional characterizations for submodularity and supermodularity restricted to certain lattices of the state space. These new functional characterizations are preserved by the optimal cost function: The state space of the problem can be partitioned into disjoint lattices such that, on each lattice, (a) it is optimal to produce a batch of a particular component if and only if the state vector is less than a certain threshold associated with that component, and (b) it is optimal to fulfill a demand of a particular product if and only if the state vector is greater than or equal to a certain threshold associated with that product. We refer to this policy as a *lattice-dependent base-stock and lattice-dependent rationing* (LBLR) policy. We also extend the optimality of LBLR to the average cost rate.

The main contributions of Chapter 2 follow. To our knowledge, this study is the first attempt to characterize the optimal inventory replenishment and allocation policies for M-systems. This study is also the first to characterize optimal policies for any ATO problem when different products may require the same component in different quantities. More importantly, this work introduces novel functional characterizations restricted to certain lattices of the state space, giving rise to a *lattice-dependent* policy. We believe that our lattice-dependent solution for M-systems could probably be usefully employed in optimal policy characterization for more complex ATO models.

Chapter 3 evaluates the use of an LBLR policy for general ATO systems as a heuristic. We numerically compare the globally optimal policy to LBLR and two other heuristics from the literature: a state-dependent base-stock and state-dependent rationing (SBSR) policy, and a fixed base-stock and fixed rationing (FBFR) policy. Taking the average cost rate as our performance criterion, we develop a linear program to find the globally optimal cost, and Mixed Integer Programming formulations to find the optimal cost within each heuristic class. We generate more than 1800 instances for the general ATO problem, violating the M-system product structure. Interestingly, LBLR yields the globally optimal cost in all instances, while SBSR and FBFR provide solutions within 2.7% and 4.8% of the globally optimal cost, respectively. These numerical results also provide several insights into the performance of LBLR relative to other heuristics: LBLR and SBSR perform significantly better than FBFR when replenishment batch sizes imperfectly match the component requirements of the most valuable or most highly demanded product. In addition, LBLR substantially outperforms SBSR if it is crucial to hold a significant amount of inventory that must be rationed.

Based on the numerical findings in Chapter 3, future research could investigate the optimality of LBLR for ATO systems with general product structures. However, Chapter 3 also constructs counter examples showing that submodularity and supermodularity need not hold for general ATO systems. Therefore, our proof technique in Chapter 2 might be inadequate to prove the optimality of LBLR for the general ATO problem, which calls for a new methodology to obtain such an analytical result.

Chapter 4 deals with the problem of project selection and resource allocation in a multi-stage new product development (NPD) process under stage-dependent resource constraints. As in previous chapters, we model the problem as an infinite-horizon MDP under the discounted cost criterion. Each NPD project undergoes a different experiment in each stage of the NPD process; these experiments generate signals about the true nature of the project. Experimentation times are independent and exponentially distributed. Beliefs about the ultimate outcome of each project are updated after each experiment according to a Bayesian rule. Projects thus become differentiated through their signals, and all available signals for a project determine its category. The state of the system is described by the numbers of projects in each category. A control policy specifies, given the system state, (i) how to utilize the resources at each stage, (ii) which projects to experiment at each stage, and (iii) which projects to terminate.

We characterize the optimal control policy as following a new type of strategy, *state-dependent non-congestive promotion* (SDNCP), for two different special cases of the general problem: (a) when there is a single informative experiment, or (b) when there are multiple uninformative experiments. An SDNCP policy implies that, at each stage, it is optimal to advance a project with the highest expected reward to the next stage if and only if the number of projects in each successor category is less than a state-dependent threshold. We further reveal that threshold values decrease in a non-strict sense as a later stage becomes more congested or as an earlier stage becomes less congested; a stage becomes more congested with an increase in the number of projects at this stage or with an increase in the expected reward of any project at this stage. These analytical findings uncover the role congestion plays in optimal policies. We also evaluate the use of SDNCP as a heuristic for the general NPD problem, comparing it to a fixed non-congestive promotion (FNCP) policy with fixed thresholds and several other heuristics. Our steps to compute the optimal cost within each heuristic class proceed very much in the same way as in Chapter 3. We generate 80 instances of the general NPD problem, and find that SDNCP minimizes the average costs in over half of these instances. (The average distances from optimal cost are 0.05% for SDNCP, and 8.23% for FNCP.) The strong numerical performance of SDNCP demonstrates that promotion decisions at each stage should be made based on a broader monitoring of projects across all categories.

This dissertation widens our knowledge of optimal policies for MDPs by advancing novel structural results for the challenging ATO and NPD problems under Markovian assumptions. Chapter 2 introduces new functional characterizations along with lattice-dependent policies to characterize optimal policies for M-systems. Computational results in Chapter 3 further emphasize the practicality of LBLR policies for the general ATO problem. Chapter 4 establishes the optimality of SDNCP in two special cases of the NPD problem. Chapter 4 also reports the strong numerical performance of SDNCP for the general NPD problem, which might have substantial implications for many industries including, but not limited to, pharmaceutical and IT where R&D plays a vital role. Future work needs to be carried out to establish optimal policies for general ATO and NPD problems. But we are confident that the findings of this dissertation will serve as a basis for future studies on similar and even more complex problems.

Chapter 2

New Functional Characterizations and Optimal Structural Results for Assemble-to-Order M-Systems

2.1 Introduction

Assemble-to-order (ATO) production is a popular strategy for manufacturing firms. ATO allows companies to reduce their response window by stocking components, but gives them the flexibility of postponing final assembly until demand is realized (Benjaafar and ElHafsi 2006). Many high-tech firms, facing shrinking product life cycles and increasing demand for product varieties, use ATO to broaden customized product offerings, lower inventory cost, and mitigate the effect of product obsolescence. Besides manufacturing, ATO systems can be observed in cases where customer orders may include several items in different quantities (Song 2000). However, despite its popularity, little is known about the forms of optimal policies for ATO systems. Much of this is attributable to the fact that there is considerable difficulty in identifying optimal policies, as ATO systems build upon the features of *both* assembly and distribution systems (Song and Zipkin 2003). (An assembly system has only one product and aims to optimally coordinate components. A distribution system has only one component and seeks to optimally allocate the component among different products.) Hence, one needs to address both coordination and allocation issues in an ATO system, making them notoriously

This chapter has been submitted to the journal *Operations Research* with co-authors Mustafa Akan and Alan Scheller-Wolf.

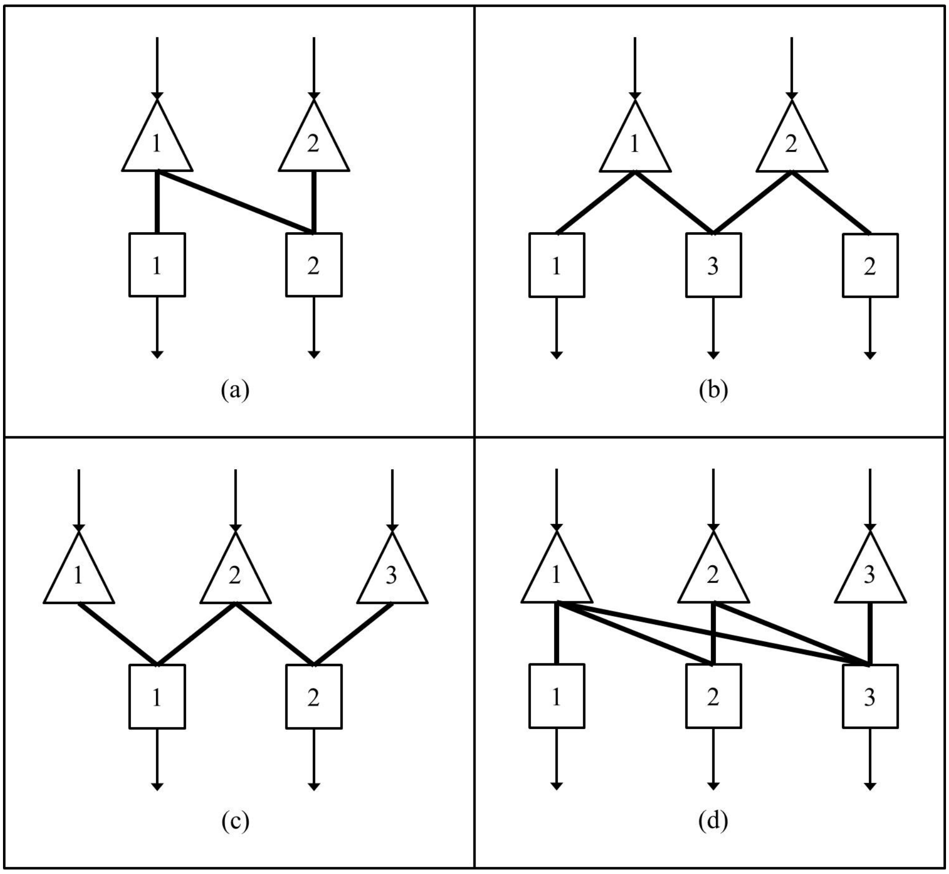

Figure 2.1 Specific types of ATO product structures: (a) N-system, (b) M-system, (c) W-system, and (d) Nested system with three products.

difficult to analyze.

ATO systems can be categorized based on their product structures (Lu et al. 2010). Figure 2.1 depicts the four specific types of ATO product structures: (a) An N-system, the simplest of the ATO product structures, has two components and two products. One product uses both components while the other product uses only one component. (b) An M-system has two components and three products. One product uses both components while the other two products use different components. (c) A W-system has three components and two products. Each product is assembled from one product-specific component and one common component. (d) A nested system has multiple components and products, where the set of components required by one product is a subset of the set of components required by the next larger product. Figure 2.1(d) depicts a nested system with three components. There are papers characterizing the optimal policies for ATO systems with product structures (a), (c), or (d);

for instance, see Dogru et al. (2010) for a W-system; Lu et al. (2010) for an N-system, and a W-system and its generalizations; and ElHafsi et al. (2008) for a nested system.

In this chapter, we consider the inventory control of a continuous time ATO system with multiple products and components structured according to a generalized version of the M-system. Specifically, the system involves a single (master) product which uses multiple units from each component, and multiple (individual) products each of which consumes multiple units from a different component. Our product structure takes the form of M-system when there are three products, cf. Figure 2.1(b).

We formulate the problem as an infinite-horizon Markov decision process (MDP) under the total expected discounted cost criterion. We assume each component is produced in batches of a fixed size in a make-to-stock fashion; production times are independent and exponentially distributed. Demand for each end-product arrives as an independent Poisson process. If not satisfied immediately upon arrival, these demands are lost. A control policy specifies when to produce a batch of any component and, upon arrival of a demand, whether or not to satisfy it from inventory if sufficient inventory exists.

A standard approach for the analysis of optimal policies of MDPs is to explore the first- and/or second-order properties of the optimal cost function (see Koole 2006). In the literature, optimal cost functions are typically shown to be *convex* (or *concave*). However, the existence of counter-examples proves that *convexity* need not hold for our model (see Chapter 3 for counter examples). Taking an alternative approach we define new functional characterizations for *submodularity* and *supermodularity*, restricted to certain subspaces of the state space. (See Topkis (1998) for definitions of *submodularity* and *supermodularity*.)

With these new definitions, we characterize the forms of optimal inventory replenishment and allocation policies under a mild condition: If the batch size for any component equals the number of units of that component needed to make one unit of the individual product using that component only (Assumption 2.4.1), the optimal inventory replenishment is a *lattice-dependent base-stock production policy* and the optimal inventory allocation is a *lattice-dependent rationing policy* (cf. Theorem 2.4.1). This implies that the state space of the problem can be partitioned into disjoint lattices such that, on each lattice, (a) it is optimal to produce a batch of a particular component if and only if the state vector is less than the base-stock level associated with that component, and (b) it is optimal to fulfill a demand of a particular product if and only if the state vector is greater than or equal to the rationing level associated with that product. Furthermore, as the system moves to a different lattice upon replenishment of a particular component, (i) the base-stock level of any other component increases, (ii) the rationing level for any individual product not using that component increases, and (iii) the rationing level for the master product using all components decreases, in a non-strict sense.

Our contributions in the ATO research stream are as follows: First, to our knowledge, our study is the first attempt to characterize the forms of optimal replenishment and allocation policies for the M-system and its generalizations. Second, unlike previous research dealing with the optimal policy characterization for ATO systems under stochastic lead times, we are the first to allow different products to use the same component in *different* quantities. Third, we define new functional characterizations for *submodularity* and *supermodularity*, restricted to certain subspaces of the state space. Fourth, we introduce the notion of a *lattice-dependent* policy, which represents a significant step towards understanding the problem and may enable researchers to develop near-optimal heuristic solutions for general ATO systems.

The rest of this chapter is organized as follows: In Section 2.2 we review the related literature. In Section 2.3 we formulate our model under the discounted cost criterion. In Section 2.4, we introduce the new functional characterizations, establish the optimal replenishment and allocation policies, and extend our structural results to the average cost case. In Section 2.5, we offer some extensions and concluding remarks.

2.2 Literature Review

Literature on ATO systems is extensive; Song and Zipkin (2003) provide a detailed review of the ATO literature. However, there has been little research on optimal policy characterization for ATO systems, which can be classified along the dimensions of system structure (assembly, distribution, or general ATO) and nature of component supply leadtime (deterministic vs. stochastic); see Table 2.1. Our model falls into the last column and last row in Table 2.1. Below, we review the literature shown in Table 2.1. We also discuss several studies on stochastic

Table 2.1 Literature on optimal policy characterization for ATO systems.

	Assembly System	Distribution System	General ATO System
Deterministic Leadtime	Schmidt and Nahmias (1985) Rosling (1989) Chen and Zheng (1994) Janakiraman and Muckstadt (2004)	Topkis (1968) Sobel and Zhang (2001)	Veinott (1965) Gerchak and Henig (1989) Hillier (2000) Van Mieghem and Rudi (2001) Dogru et al. (2010) Lu et al. (2010)
Stochastic Leadtime	Song et al. (2000) Benjaafar and ElHafsi (2006) ElHafsi (2009)	Ha (1997) Ha (2000) de Véricourt et al. (2002) Gayon et al. (2009a) Gayon et al. (2009b)	ElHafsi et al. (2008) Nadar et al. (2012)

leadtime single-item models.

Deterministic Leadtime ATO Models. Schmidt and Nahmias (1985) study a finite horizon inventory model in which an end-product is assembled from two components, which are ordered from an external supplier. They show that the optimal ordering rule for components is a state-dependent base-stock policy, with the order-up-to point in each component nondecreasing in the inventory position of the other component. The optimal assembly policy is a base-stock policy as well. Rosling (1989) considers an assembly system with proportional production and stock holding costs, showing that the optimal policies are equivalent to those of a series system (Clark and Scarf 1960) under an initial condition on stock levels. Chen and Zheng (1994), and Janakiraman and Muckstadt (2004) provide extensions of Rosling's results.

Topkis (1968) considers a single-product model with multiple independent demand classes. He shows that it is optimal to satisfy demand of a class if the stock level is above a certain class-dependent level, independent of the levels of unmet demand of lower or equally important classes. Topkis also proves the optimality of a base-stock ordering policy under certain conditions. Sobel and Zhang (2001) consider an inventory model with two nonstationary demand sources: deterministic demand that must be satisfied immediately and stochastic demand that can be backordered. Assuming zero leadtime and a fixed setup cost, they establish the optimality of a modified (s, S) policy.

Veinott (1965) is the first to consider a nonstationary model with multiple products, multiple demand classes, and zero delivery lag. He develops sufficient conditions ensuring that a base-stock ordering policy is optimal over an infinite horizon. This result extends to positive delivery lag under certain restrictions. Gerchak and Henig (1989) develop an optimal myopic solution procedure for ordering and sales decisions in a stationary system with zero leadtimes, infinite production capacity, and partial backlogging. They state that generalizations to nonstationary and fixed leadtimes are possible. Hillier (2000) studies ATO systems with general demand, backlogging, and zero leadtime. If component commonality is allowed, he develops lower and upper bounds on the optimal solution. Otherwise, the model becomes a multi-period newsvendor problem with a myopic solution. These results can also be generalized to lost sales and nonzero leadtimes. Van Mieghem and Rudi (2001) consider *newsvendor networks* with multiple products, multiple processing and storage points, and independent demand over time. They show that the structure of the optimal policy in a single-period *newsvendor network* carries over to a multi-period setting under certain conditions.

Two recent papers have managed to characterize optimal policies for more complex, nontrivial special cases of ATO product structures: Dogru et al. (2010) consider an ATO W-system with identical component lead times (see Figure 2.1 for the W-system product structure). Based on stochastic programming, they show the optimality of myopic allocation policies along with a base-stock replenishment rule when the base-stock level of the common component equals the sum of the base-stock levels of the unique parts, or when both products have the same unit inventory cost. Using a sample path argument, Lu et al. (2010) obtain a similar result for W-systems operating under a base-stock policy and nonidentical lead times; no hold-back rules are optimal for the inventory/backlog cost minimization problem when the symmetric cost condition holds. Lu et al. (2010) also extend this optimality result to N-systems (see Figure 2.1 for the N-system product structure) and generalized W-systems.

Stochastic Leadtime Single-Item Models. In his cutting edge work, Kaplan (1970) shows that the multidimensional minimization problem can be reduced to a one-dimensional minimization for single-item models where orders never cross in time and order arrival probabilities depend only on the time since the order was placed. The state reduction is still possible if there is a fixed ordering cost. The ordering policies obtained are functions of the sum of the stock on hand plus stock on order: A state-dependent base stock policy is optimal when there is no fixed cost, otherwise a state-dependent (s, S) policy is optimal. Ehrhardt (1984) extends Kaplan's work by deriving conditions for the optimality of myopic base-stock policies. See also Song and Zipkin (1996) for a generalization of Kaplan's leadtime model.

Zipkin (1986) relaxes Kaplan's leadtime model in a continuous-time setting by allowing the probability of having a certain number of outstanding orders at any period to depend on the number of outstanding orders at a previous period, showing in this case the inventory level has a stationary and limiting distribution. Song and Zipkin (1993) consider a situation in which the demand rate varies with an underlying Markov chain, unmet demand is backordered, orders never cross, and the leadtime history is ignored in placing orders. If the ordering cost is linear in the order quantity, they show the optimality of a state-dependent base-stock policy. If there is a fixed ordering cost, a state-dependent (s, S) policy is optimal. Finally, Hariharan and Zipkin (1995) analyze a setting with Poisson customer orders each of which arrives with a due date and cannot be fulfilled early. If due dates and replenishment leadtimes are fixed, they prove the optimality of a base-stock policy by transforming the system into a conventional inventory model and adapting Veinott's (1965) approach. If the replenishment leadtime is a random variable bounded below and satisfies the assumptions of Kaplan (1970), a base-stock policy remains optimal.

Stochastic Leadtime ATO Models. Song et al. (2000) is the first significant attempt to consider leadtime uncertainty in an assembly problem. In their setting, a one-time demand of a random quantity of the end-item occurs at a known time. Component ordering decisions about the quantity and timing are made simultaneously, at the beginning of the horizon. The objective is to determine how much and when to order each component to minimize the total expected holding, tardiness, overage, and underage costs. They provide several structural results regarding the total cost function. Their numerical results underscore the importance of considering the effects of leadtime uncertainty in an assembly system.

For continuous-time models, most authors assume that component production and demand interarrival times are exponentially distributed. Ha (1997) considers a production system with lost sales and several demand classes which differ in their lost sale costs, showing the optimality of a base-stock production and stock-reservation inventory allocation policy with class-based rationing levels. Ha (2000) extends the findings of Ha (1997) to Erlang production times. de Véricourt et al. (2002) extend the results of Ha (1997) in a backordering environment. In a recent study, Gayon et al. (2009a) allow for both Erlang production times and backordering. Lastly, Gayon et al. (2009b) consider a supplier with multiple customer classes, some of which share tentative advance demand information by announcing their orders ahead of their due date. Customer classes vary in their expected due dates, cancellation probabilities, and shortage costs. They prove in this case the optimal inventory replenishment is a state-dependent base-stock policy, and the optimal inventory allocation is a state-dependent rationing policy.

Benjaafar and ElHafsi (2006) consider an assembly system with a *single* end-product and multiple components. One unit of each component is assembled into the end-product, which is demanded by multiple customer classes. Again under Markovian assumptions on production and demand, a state-dependent base-stock and state-dependent rationing policy is optimal. ElHafsi (2009) extends the results of Benjaafar and ElHafsi (2006) by allowing customer orders to arrive as a compound Poisson process. Our model in this chapter generalizes the model in Benjaafar and ElHafsi (2006) in several directions: (i) Each of our components is required by an individual product as well as the master product, (ii) each of our components may be used by the two products in *different* quantities, and (iii) the master product may use *different* quantities of different components. Furthermore, the state-dependent base-stock and state-dependent rationing (SBSR) policy in Benjaafar and ElHafsi (2006) is a special case of our lattice-dependent base-stock and lattice-dependent rationing (LBLR) policy if lattices are chosen optimally (see Chapter 3). Therefore, LBLR is analytically no worse than SBSR for general ATO systems.

To our knowledge, ElHafsi et al. (2008) is the *only* prior work considering a nontrivial special case of ATO product structures along with random leadtimes. Specifically, they consider a nested system with multiple components under Markovian assumptions (see Figure 2.1 for a nested system with three components), proving the optimality of state-dependent base-stock and state-dependent rationing policies.

All of the papers cited in Table 2.1 assume that products require common components in equal quantities (if they share common components). We significantly relax this assumption; each of our components may be used by individual and master products in unequal quantities. Furthermore, the ATO literature neglects to characterize the forms of optimal policies for generalized M-systems. Our work represents an initial step towards filling this gap as well, by introducing novel functional characterizations that give rise to *lattice-dependent* policies.

2.3 Problem Formulation

We consider an ATO system with n components ($j = 1, 2, .., n$) and $n+1$ products ($i = 1, 2, .., n+1$), where each component j is consumed by products $i = j$ and $i = n+1$ only. Notice that the ATO system we consider reduces to an "M-system" when $n = 2$, cf. Figure 2.1(b). Define $\mathbf{a} = (a_1, a_2, .., a_n)$ as the vector of component requirements for product $n+1$; a_j is the number of component j needed to assemble one unit of product $n+1$. Define $\mathbf{b} = (b_1, b_2, .., b_n)$ as the vector of component requirements for all the other products; b_j is the number of component j required to make one unit of product $i = j$. Each component j is produced in batches of a fixed size q_j in a make-to-stock fashion. Define $\mathbf{q} = (q_1, q_2, .., q_n)$ as the vector of production batch sizes. Production time for component j is independent of the system state and the number of outstanding orders of any type, and exponentially distributed with finite mean $1/\mu_j$. Assembly lead times are negligible so that assembly operations can be postponed until demand is realized. Demand for each product i arrives as an independent Poisson process with finite rate λ_i. Demand for product i can be fulfilled only if all the required components are available; otherwise, the demand is lost, incurring a unit lost sale cost c_i. Demand may also be rejected in the presence of all the necessary components, again incurring a unit lost sale cost.

The state of the system at time t is the vector $\mathbf{X}(t) = (X_1(t), .., X_n(t))$, where $X_j(t)$ is a nonnegative integer denoting the on-hand inventory for component j at time t. Each component held in stock has a holding cost per unit time which is strictly increasing convex in the number of available units of that component. Denote by $h(\mathbf{X}(t)) = \sum_j h_j(X_j(t))$ the inventory holding cost rate at state $\mathbf{X}(t)$. Since all inter-event times are exponentially distributed, the system retains no memory, and decision epochs can be restricted to times when the state changes. Using the memoryless property, we can formulate the problem as an MDP and focus on Markovian policies for which actions at each decision epoch depend solely on the current state. A control policy π specifies for each state $\mathbf{x} = (x_1, .., x_n)$, the action $\mathbf{u}^\pi(\mathbf{x}) = (y_1, .., y_n, z_1, .., z_{n+1})$, $y_j, z_i \in \{0, 1\}$, $\forall i, j$, where $y_j = 1$ means produce component j,

$y_j = 0$ means do not produce component j, $z_i = 1$ means satisfy demand for product i, and $z_i = 0$ means reject demand for product i.

As each ordering decision specifies only whether or not to produce a component, there is at most one outstanding order for each component at any time. Also, as component orders are not part of our system state, these can in effect be cancelled upon transition to a new state. Both of these assumptions are standard in the literature (see Ha 1997, Benjaafar and ElHafsi 2006, and ElHafsi et al. 2008).

Define $0 < \alpha < 1$ as the discount rate. For a given policy π and a starting state $\mathbf{x} \in \mathbb{N}_0^n$ (where $\mathbb{N}_0$ is the set of nonnegative integers and $\mathbb{N}_0^n$ is its n-dimensional cross product), the expected discounted cost over an infinite planning horizon $v^\pi(\mathbf{x})$ can be written as

$$v^\pi(\mathbf{x}) = \mathbb{E}_{\mathbf{x}}^\pi \left[\sum_{j=1}^{n} \int_0^\infty e^{-\alpha t} h_j(X_j(t))dt + \sum_{i=1}^{n+1} \int_0^\infty e^{-\alpha t} c_i dN_i(t) \right]$$

where $N_i(t)$ is the number of demands for product i that have not been fulfilled from on-hand inventory up to time t. Letting β denote the upper bound on transition rates for all system states (i.e., $\beta = \sum_j \mu_j + \sum_i \lambda_i$), we below formulate the optimality equation that holds for the optimal cost function $v^* = v^{\pi^*}$ (see Lippman 1975, and Chapter 5 in Bertsekas 2007 for an explanation of how the continuous-time control problem can be transformed into an equivalent discrete-time control problem):

$$v^*(\mathbf{x}) = \frac{1}{\alpha + \beta} \left\{ h(\mathbf{x}) + \sum_j \mu_j T^{(j)} v^*(\mathbf{x}) + \sum_i \lambda_i T_i v^*(\mathbf{x}) \right\}, \tag{2.3.1}$$

where the operator $T^{(j)}$ for component j is defined as

$$T^{(j)} v(\mathbf{x}) = \min\{v(\mathbf{x} + b_j e_j), v(\mathbf{x})\}, \tag{2.3.2}$$

the operator T_i for individual product $i \le n$ is defined as

$$T_i v(\mathbf{x}) = \begin{cases} \min\{v(\mathbf{x}) + c_i, v(\mathbf{x} - b_i e_i)\} & \text{if } x_i \ge b_i, \\ v(\mathbf{x}) + c_i & \text{otherwise,} \end{cases} \tag{2.3.3}$$

and the operator T_{n+1} for the master product is given by

$$T_{n+1}v(\mathbf{x}) = \begin{cases} \min\{v(\mathbf{x}) + c_{n+1}, v(\mathbf{x} - \mathbf{a})\} & \text{if } \mathbf{x} \geq \mathbf{a}, \\ v(\mathbf{x}) + c_{n+1} & \text{otherwise,} \end{cases} \tag{2.3.4}$$

where e_j is the jth unit vector of dimension n. For a given state $\mathbf{x}$, the operator $T^{(j)}$ specifies whether or not to produce a batch of component j; and the operator T_i specifies, upon arrival of a demand for product i, whether or not to fulfill it from inventory if sufficient inventory exists. In the optimality equation 2.3.1, as it is always possible to redefine the time scale, without loss of generality we assume $\alpha + \beta = 1$.

2.4 Characterization of the Optimal Policy

In this section we first define new second-order functional characterizations, and show how these properties propagate through our optimal cost function. We then use these propagation results to establish the optimality of lattice-dependent base-stock and rationing policies under a mild condition on component batch sizes.

2.4.1 Functional Characterizations

Define f as the class of real-valued functions on the n-dimensional nonnegative orthant, and let $\Delta_{\mathbf{p}} f = f(\mathbf{x} + \mathbf{p}) - f(\mathbf{x})$ where $\mathbf{p} = (p_1, p_2, .., p_n)$ is a vector of nonnegative integers.

We introduce the notion of "submodularity with step size $\mathbf{p}$" for $\mathbf{p} \in \mathbb{N}_0^n$ to describe the class of functions f for which $\Delta_{p_j e_j} f$ is nonincreasing with an increase of p_k in the kth dimension, $\forall j \neq k$. We denote the class of functions satisfying this property by $Sub(\mathbf{p})$.

We also define the concept of "supermodularity with step sizes $\mathbf{r}$ and $\mathbf{p}$" for $\mathbf{r}, \mathbf{p} \in \mathbb{N}_0^n$ to describe the class of functions f with $\Delta_{p_j e_j} f$ nondecreasing with an increase of $\mathbf{r}$ in the domain, $\forall j$. We denote this class of functions by $Super(\mathbf{r}, \mathbf{p})$.

Lastly, we define the notion of "n-dimensional supermodularity with step sizes $\mathbf{r}$ and $\mathbf{p}$" for $\mathbf{r}, \mathbf{p} \in \mathbb{N}_0^n$ to describe the class of functions f with $\Delta_{\mathbf{p}} f$ nondecreasing with an increase of $\mathbf{r}$ in the domain, and denote it by $nSuper(\mathbf{r}, \mathbf{p})$. Note that both $Super(1, 1)$ and $nSuper(1, 1)$ are the class of *convex* functions of one dimension.

Definition 2.4.1 (Second-Order Properties). *Let f be a real-valued function defined on $\mathbb{N}_0^n$. Also let $\mathbf{r}, \mathbf{p} \in \mathbb{N}_0^n$.*

(a) $f \in Sub(\mathbf{p})$, *if* $f(\mathbf{x} + p_j e_j) - f(\mathbf{x}) \geq f(\mathbf{x} + p_j e_j + p_k e_k) - f(\mathbf{x} + p_k e_k)$, $\forall \mathbf{x} \in \mathbb{N}_0^n$, $\forall j$ *and* $\forall k \neq j$.

(b) $f \in Super(\mathbf{r}, \mathbf{p})$, *if* $f(\mathbf{x} + p_j e_j + \mathbf{r}) - f(\mathbf{x} + \mathbf{r}) \geq f(\mathbf{x} + p_j e_j) - f(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{N}_0^n$ *and* $\forall j$.

(c) $f \in nSuper(\mathbf{r}, \mathbf{p})$, *if* $f(\mathbf{x} + \mathbf{p} + \mathbf{r}) - f(\mathbf{x} + \mathbf{r}) \geq f(\mathbf{x} + \mathbf{p}) - f(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{N}_0^n$.

The following lemma shows that the class of $Super(\mathbf{r}, \mathbf{p})$ is a subset of that of $nSuper(\mathbf{r}, \mathbf{p})$:

Lemma 2.4.1. $Super(\mathbf{r}, \mathbf{p}) \subseteq nSuper(\mathbf{r}, \mathbf{p})$, $\forall \mathbf{r}, \mathbf{p} \in \mathbb{N}_0^n$.

Proof. See Appendix A.1. □

2.4.2 Propagation Results

We now proceed to the analysis of our optimal cost function based on the functional characterizations of Section 2.4.1. First notice that our optimal cost function 2.3.1 is a linear function of replenishment control operators (i.e., $T^{(j)}$, $\forall j$), allocation control operators (i.e., T_i, $\forall i$), and holding cost rates (i.e., h_j, $\forall j$). The lemma below shows that (a) each of our replenishment control operators preserves both "submodularity with step size $\mathbf{q}$" and "supermodularity with step sizes $\mathbf{a}$ and $\mathbf{q}$"; (b) each of our allocation control operators preserves both "submodularity with step size $\mathbf{b}$" and "supermodularity with step sizes $\mathbf{a}$ and $\mathbf{b}$"; and (c) our holding cost rate satisfies all these properties:

Lemma 2.4.2. (a) $T^{(j)} : Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q}) \to Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q})$, $\forall j$,

(b) $T_i : Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b}) \to Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b})$, $\forall i$, *and*

(c) $h \in Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q}) \cap Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b})$.

Proof. See Appendix A.2. □

While our second-order properties are preserved by linear transformations, the second-order properties shown to propagate through our replenishment and allocation control operators above differ in their parameters (i.e., $\mathbf{q}$ vs. $\mathbf{b}$), and thus need *not* hold for our optimal cost function: Only for cases with equal parameters ($\mathbf{q} = \mathbf{b}$) are we able to characterize the structure of our cost function. Therefore, we assume the production batch size for each component j equals the number of units of component j required by one unit of product $i = j$:

Assumption 2.4.1. $q_j = b_j$, $\forall j$.

Although we make the above assumption for analytical tractability, this corresponds to systems with component batch sizes which are, reasonably, determined by the individual product sizes. This assumption is consistent with previous treatments of Markovian inventory systems (see, for example, Ha 1997, Benjaafar and ElHafsi 2006, and ElHafsi et al. 2008).

We now define V^* as the set of real-valued functions satisfying the properties of $Sub(\mathbf{b})$, $Super(\mathbf{a}, \mathbf{b})$, and $nSuper(\mathbf{a}, \mathbf{b})$. Then, under Assumption 2.4.1, the lemma below follows from Lemmas 2.4.1 and 2.4.2, and Propositions 3.1.5 and 3.1.6 in Bertsekas (2007):

Lemma 2.4.3. *Under Assumption 2.4.1, if* $v \in V^*$*, then* $Tv \in V^*$ *where* $Tv(\mathbf{x}) = h(\mathbf{x}) + \sum_j \mu_j T^{(j)} v(\mathbf{x}) + \sum_i \lambda_i T_i v(\mathbf{x})$*. Furthermore, the optimal cost function* v^* *is an element of* V^**.*

Proof. See Appendix A.2. □

In the next subsection, we use the second-order properties of our optimal cost function to characterize the forms of optimal inventory replenishment and allocation policies.

2.4.3 Optimal Inventory Replenishment and Allocation

We introduce the notation $\mathbb{L}(\mathbf{p}, \mathbf{r}) = \{\mathbf{p} + k\mathbf{r} : k \in \mathbb{N}_0\}$ to denote an n-dimensional lattice with initial vector $\mathbf{p} \in \mathbb{N}_0^n$ and common difference $\mathbf{r} \in \mathbb{N}_0^n$, where $\exists j$ such that $p_j < r_j$. With this we are now ready to state the main result of this chapter:

Theorem 2.4.1. *Under Assumption 2.4.1, there exists an optimal stationary policy that can be specified as follows.*

(1) *The optimal inventory replenishment policy for each component* j *is a* <u>*lattice-dependent base-stock policy*</u> *with lattice-dependent base-stock levels* $S_j^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{a})$, $\forall \mathbf{p}$*: It is optimal to produce a batch of component* j *if and only if* $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{a})$ *is less than* $S_j^*(\mathbf{p})$*.*

(2) *The optimal inventory allocation policy for each product $i \leq n$ is a lattice-dependent rationing policy with lattice-dependent rationing levels $R_i^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{a})$, $\forall \mathbf{p}$: It is optimal to fulfill a demand for product $i \leq n$ if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{a})$ is greater than or equal to $R_i^*(\mathbf{p})$.*

(3) *The optimal inventory allocation policy for product $n+1$ is a lattice-dependent rationing policy with lattice-dependent rationing levels $R_{n+1}^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{b})$, $\forall \mathbf{p}$: It is optimal to fulfill a demand for product $n+1$ if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{b})$ is greater than or equal to $R_{n+1}^*(\mathbf{p})$.*

The optimal policy has the following additional properties:

i. As the system moves to a difference lattice with an increment of b_k in the inventory level of component k, both the optimal base-stock level of component $j \neq k$ and the optimal rationing level for (individual) product $i \notin \{k, n+1\}$ increase in a non-strict sense, $\forall k$.

ii. As the system moves to a difference lattice with an increment of b_k in the inventory level of component k, the optimal rationing level for (master) product $n+1$ decreases in a non-strict sense, $\forall k$.

iii. It is optimal to fulfill a demand of (master) product $n+1$ if $x_j \geq a_j + b_j \left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$.

Proof. See Appendix A.3. □

Theorem 2.4.1 builds upon the properties of $Super(\mathbf{a}, \mathbf{b})$, $nSuper(\mathbf{a}, \mathbf{b})$, and $Sub(\mathbf{b})$: $Super$ $(\mathbf{a}, \mathbf{b})$ implies that, as the system moves to a higher inventory level on the lattice $\mathbb{L}(\mathbf{p}, \mathbf{a})$, the desirability of producing a batch of any component decreases in a non-strict sense (i.e., optimality of base-stock policies, point 1), and the desirability of satisfying a demand for any product $i \leq n$ increases in a non-strict sense (i.e., optimality of rationing policies for each product $i \leq n$, point 2). $nSuper(\mathbf{a}, \mathbf{b})$ implies that, as the system moves to a higher inventory level on the lattice $\mathbb{L}(\mathbf{p}, \mathbf{b})$, the incentive to fulfill a demand for product $n+1$ increases in a non-strict sense (i.e., optimality of a rationing policy for product $n+1$, point 3).

Notice that the rationing policy for each product $i \leq n$ in point 2 is defined over lattices with common difference $\mathbf{a}$, while the rationing policy for product $n+1$ in point 3 is defined over lattices with common difference $\mathbf{b}$. The intuition behind these results is as follows: Demands of each product $i \leq n$ compete with those of product $n+1$ for the same component. For a given product $i \leq n$, an increment of $\mathbf{a}$ in the inventory level increases the total demand for its competitor product that can be satisfied, thereby mitigating the competition. Hence, the incentive to fulfill a demand of product $i \leq n$ increases in a non-strict sense (point 2). Likewise, for product $n+1$, an increment of $\mathbf{b}$ in the inventory level mitigates the competition as the total demand for each of its competitors that can be satisfied increases. Hence, the incentive to fulfill a demand of product $n+1$ increases in a non-strict sense (point 3). Note that under the rationing policy described in Theorem 2.4.1, for a given product, an increment in the inventory level that does *not* increase the total demand for any of its competitors that can be satisfied, may reduce the incentive to fulfill a demand of this product (in a non-strict sense).

Theorem 2.4.1, using the properties of $Sub(\mathbf{b})$ and $Super(\mathbf{a}, \mathbf{b})$, proves the following additional properties of the optimal policy: Point (i) says that, based on the property of $Sub(\mathbf{b})$, upon replenishment of a batch of a component k, the desirability of producing a batch of component $j \neq k$ increases while the desirability of satisfying a demand for product $i \notin \{k, n+1\}$ decreases, in a non-strict sense. Therefore, both the base-stock level of component $j \neq k$ and the rationing level for product $i \notin \{k, n+1\}$ increase in a non-strict sense. The intuition is that the presence of product $n+1$ requires us to coordinate inventory replenishment and fulfillment decisions across components; it is less beneficial to produce or hold a batch of one component when the inventory level of any other component is significantly smaller. Point (ii) states that, based on the property of $Super(\mathbf{a}, \mathbf{b})$, upon replenishment of a batch of any component, the incentive to fulfill a demand for product $n+1$ increases in a non-strict sense since the total demand for one of its competitors that can be satisfied increases. Lastly, point (iii) shows that it is optimal to fulfill a demand of product $n+1$ as long as the total demand for any other product that can be satisfied stays the same.

To our knowledge, we are the first to introduce the notion of a *lattice-dependent base-stock and rationing* (LBLR) policy. Such a policy differs from *state-dependent base-stock and rationing* (SBSR) policies shown to be optimal in a single-product ATO system by Benjaafar and ElHafsi (2006) in the following ways: There are inventory levels $\mathbf{x}_1 \in \mathbb{L}(\mathbf{p}_1, \mathbf{a})$ and $\mathbf{x}_2 \in$

$\mathbb{L}(\mathbf{p}_2, \mathbf{a})$, $\mathbf{x}_1 \geq \mathbf{x}_2$, $\mathbf{p}_1 \neq \mathbf{p}_2$, such that an LBLR policy allows a particular component to be produced at $\mathbf{x}_1$ even if it is not produced at $\mathbf{x}_2$, but an SBSR policy does *not*. Likewise, there are inventory levels $\mathbf{x}_1 \in \mathbb{L}(\mathbf{p}_1, \mathbf{b})$ and $\mathbf{x}_2 \in \mathbb{L}(\mathbf{p}_2, \mathbf{b})$, $\mathbf{x}_1 \geq \mathbf{x}_2$, $\mathbf{p}_1 \neq \mathbf{p}_2$, such that an LBLR policy allows a demand for product $n+1$ to be rejected at $\mathbf{x}_1$ even if it is satisfied at $\mathbf{x}_2$, but again an SBSR policy does *not*. Conversely, if $\mathbf{a} \neq \sum_j ze_j$ for $z \in \mathbb{N}_0$, then there also may exist inventory levels $\mathbf{x}_1 \geq \mathbf{x}_2$, such that an SBSR policy allows a particular component to be produced at $\mathbf{x}_1$ even if it is not produced at $\mathbf{x}_2$, but an LBLR policy does *not*. But if $\mathbf{a}$ is chosen optimally, then it can be shown that an SBSR policy is a subclass of LBLR policies (see Chapter 3).

2.4.4 The Case of Average Cost

In this subsection, as our optimization criterion, we take the average cost per unit time over an infinite planning horizon. Given a policy π, the average cost rate is given by

$$v^\pi(\mathbf{x}) = \lim_{T\to\infty} \sup \frac{1}{T} \left\{ \sum_{j=1}^{n} \int_0^T h_j(X_j(t))dt + \sum_{i=1}^{n+1} \int_0^T c_i dN_i(t) \right\}.$$

The objective is to identify a policy π^* that yields $v^*(\mathbf{x}) = \inf_\pi v^\pi(\mathbf{x})$ for all states $\mathbf{x}$. The following proposition shows that our structural results carry over to the average cost case:

Proposition 2.4.1. *Suppose that Assumption 2.4.1 holds and the Markov chain governing the system is irreducible. There exists a stationary policy that is optimal under the average cost criterion. The policy retains all the properties of the optimal policy under the discounted cost criterion, as introduced in Theorem 2.4.1. Also, the optimal average cost is finite and independent of the initial state; there exists a finite constant v^* such that $v^*(\mathbf{x}) = v^*$, $\forall \mathbf{x}$.*

Proof. See Appendix A.4. □

2.5 Extensions and Concluding Remarks

We have studied the inventory replenishment and allocation problem in an ATO production system with generalized M-system product structure. We extend the existing literature by characterizing the optimal policy while allowing different products to use different quantities of the same component. Assuming component batch sizes are determined by the individual product sizes, we establish the optimality of a lattice-dependent base-stock and lattice-dependent rationing policy for both the discounted cost and average cost cases. We discuss below two extensions to our analysis and several concluding remarks.

First, our analysis can be extended to systems where a nonempty subset of the components is required *only* by product $n+1$. These systems take the form of N-system when there are two components. Define A_1 as the set of components used by product $n+1$ only, and A_2 as the set of components j used by products $i = j$ and $i = n+1$ (i.e., $A_1 = \{1, 2, .., n\} \setminus A_2$). Such systems are a special case of our model in which the demand rate for each product $i \in A_1$ is zero, and therefore an LBLR policy is optimal for these systems. Notice that Assumption 2.4.1 is no longer required in setting the batch sizes for components $i \in A_1$. Since the demand rate for each product $i \in A_1$ is zero, one can choose b_i to be the ideal batch size for component $j = i$, $\forall i \in A_1$.

Second, our model can be extended to allow each product to be requested by multiple demand classes with different lost sale costs. Suppose that there are D^i different demand classes for product i, and let $d^i = 1, 2, .., D^i$. A demand for one unit of product i from class d^i arrives as an independent Poisson process with rate λ_{i,d^i} and has a lost sale cost c_{i,d^i}, $\forall i$. Without loss of generality, we assume $c_{i,1} \geq c_{i,2} \geq \cdots \geq c_{i,D^i}$, $\forall i$. We can revise our optimal cost function by augmenting the allocation control operator T_i to include the index of demand class d^i, $\forall i$. We can then prove the optimality of LBLR under the following modifications: (i) The optimal inventory allocation for demand class d^i of each product $i \leq n$ is a lattice-dependent rationing policy with rationing levels $R^*_{i,d^i}(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{a})$, $\forall \mathbf{p}$, (ii) the optimal inventory allocation for demand class d^{n+1} of product $n+1$ is a lattice-dependent rationing policy with rationing levels $R^*_{n+1,d^{n+1}}(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{b})$, $\forall \mathbf{p}$, and (iii) it is optimal to fulfill a demand of product $n+1$ from class 1 as long as the total demand for any other product that can be satisfied stays the same. Furthermore, $R^*_{i,1}(\mathbf{p}) \leq R^*_{i,2}(\mathbf{p}) \leq \cdots \leq R^*_{i,D^i}(\mathbf{p})$, $\forall \mathbf{p}$, $\forall i$.

In Chapter 3, we conduct numerical experiments to evaluate the use of an LBLR policy as a heuristic for general ATO systems, comparing it with two other heuristics: a state-dependent base-stock and rationing policy (SBSR), and a fixed base-stock and rationing policy (FBFR), both adapted from Benjaafar and ElHafsi (2006). We numerically show, in the average cost case, that LBLR *always* yields the globally optimal cost in over 1800 examples, while SBSR (or FBFR) provides solutions within 2.7% (or 4.8%) of the globally optimal cost. We are also able to analytically show that LBLR outperforms the other heuristics. Based on these results, future research could investigate whether an LBLR policy is indeed optimal for general ATO systems, and if so, how the state space should be partitioned into disjoint lattices. However, one may need to develop a different methodology to prove the optimality of LBLR, because in Chapter 3 we also provide counter-examples showing that the second-order properties of our optimal cost function, which are *sufficient* to ensure the optimality of LBLR, may fail to hold for general ATO systems.

Future extensions of this study could also consider ATO systems with backordering. In this case, we would need to include the number of backordered demands for each product in the state space, and investigate the optimal backorder clearing mechanism upon replenishment of any component. Another direction for future research is to extend our model to phase-type or even general component production and demand interarrival times. Also, it would be more realistic to allow for dependent demand across products and over time. Lastly, extending our model to include nonzero assembly times is an interesting problem to pursue. However, with today's manufacturing technology, assembly times are usually small and our model would likely provide a good approximation.

Chapter 3

Performance Evaluation of Lattice-Dependent Base-Stock and Rationing Policies for ATO Systems

3.1 Introduction

In Chapter 2, we considered the control of a continuous-time assemble-to-order (ATO) generalized "M-system" (see Figure 3.1) with multiple products and components. This system involves a single product which uses multiple units from each component, and multiple individual products each of which uses multiple units from a different component. We model the problem as an infinite-horizon Markov decision process (MDP) under the discounted cost criterion. Each component is produced in batches of fixed size in a make-to-stock fashion; production times are independent and exponentially distributed. Demand for each product arrives as an independent Poisson process. If not satisfied immediately upon arrival, these demands are lost. A control policy specifies when to produce a batch of any component and, upon arrival of a demand, whether or not to satisfy it from inventory if sufficient inventory exists.

In Chapter 2, we prove that if replenishment batch sizes are determined by individual product sizes (as in Figure 3.1), the optimal inventory replenishment policy is a *lattice-dependent base-stock production policy* and the optimal inventory allocation policy is a *lattice-dependent*

This chapter has been submitted to the journal *Production and Operations Management* with co-authors Mustafa Akan and Alan Scheller-Wolf.

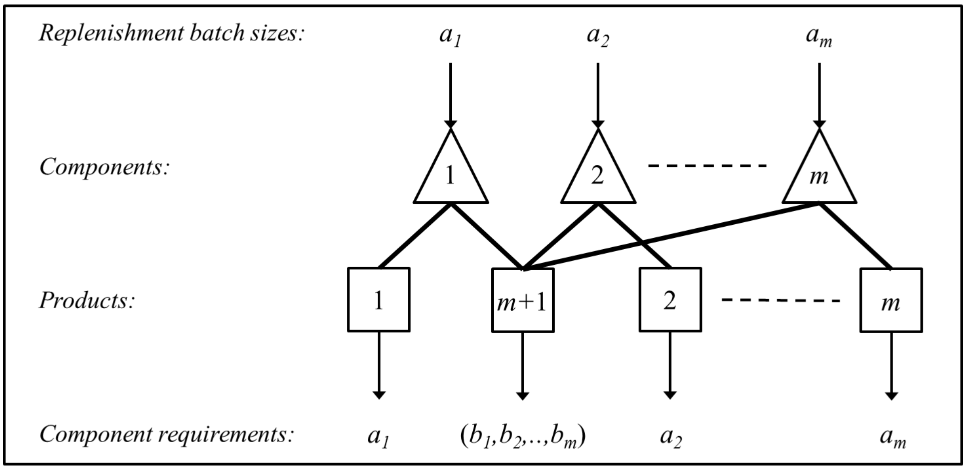

Figure 3.1 An M-system where a_i is the number of units of component i used by one unit of product i, b_i is the number of units of component i needed to assemble one unit of product $m+1$, and the batch size for component i is a_i.

rationing policy. This implies that the state space of the problem can be partitioned into disjoint lattices such that, on each lattice, (a) it is optimal to produce a batch of a particular component if and only if the state vector is less than the base-stock level of that component, and (b) it is optimal to fulfill a demand of a particular product if and only if the state vector is greater than or equal to the rationing level for that product. In Chapter 2, we also show these structural results carry over to the average cost case.

In this chapter, we conduct numerical experiments to evaluate the use of a lattice-dependent base-stock and lattice-dependent rationing (LBLR) policy as a *heuristic* replenishment and allocation policy for ATO systems with general product structures. We also compare the LBLR policy to two other heuristics: a state-dependent base-stock and state-dependent rationing (SBSR) policy inspired by Benjaafar and ElHafsi (2006), and a fixed base-stock and fixed rationing (FBFR) policy as presented in Benjaafar and ElHafsi (2006). We take the average cost rate as our performance criterion, for robustness against the starting state of the system and discount factor.

We develop a Linear Programming (LP) formulation to find the globally optimal cost, and Mixed Integer Programming (MIP) formulations to find the optimal cost within each heuristic class (LBLR, SBSR, and FBFR). Remarkably, we find that LBLR *always* yields the globally optimal cost in over 1800 compiled instances, while SBSR (or FBFR) provides solutions within

2.7% (or 4.8%) of the globally optimal cost. (The average distances from optimal cost are 0.5% and 1.5%, respectively.) We also analytically show that LBLR outperforms the other heuristics. Our numerical results indicate that LBLR and SBSR perform significantly better than FBFR when the component batch sizes imperfectly match the component requirements of the most highly demanded and/or most valuable product. In addition, LBLR has the greatest benefit over SBSR when products are highly differentiated but demand for each product should have a substantial fill rate. This observation is also supported by a regression study.

We then reformulate the ATO problem under the total expected discounted cost criterion, and construct counter examples showing that the properties of *submodularity* and *supermodularity* (which are used to prove the optimality of LBLR for ATO generalized M-systems under this criterion in Chapter 2) may fail to hold for ATO systems with general product structures. Consequently, if LBLR is optimal for general product structures, one may need to develop a different methodology to establish this result.

We proceed as follows: In Section 3.2 we formulate our general model under the average cost criterion, and describe the heuristic policies and their MIP formulations. In Section 3.3 we present and interpret our numerical results for the heuristic policies. In Section 3.4 we reformulate our general model under the discounted cost criterion, and show the functional characterizations that are sufficient to ensure the optimality of LBLR need *not* hold for general ATO systems. In Section 3.5 we offer a summary and a few concluding remarks.

3.2 The Case of Average Cost

In this section, we first describe our general model for an infinite-horizon continuous-time ATO system and develop a linear program to find the globally optimal average cost. We then present the heuristics and develop MIP formulations to find the optimal solution within each heuristic class.

3.2.1 Problem Formulation

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

The state of the system at time t is the vector $\mathbf{X}(t) = (X_1(t), .., X_m(t))$, where $X_i(t)$ is a nonnegative integer denoting the on-hand inventory for component i at time t. Each component held in stock incurs a holding cost per unit time which is convex and strictly increasing in the number of available units of that component. Denote by $h(\mathbf{X}(t)) = \sum_i h_i(X_i(t))$ the inventory holding cost rate at state $\mathbf{X}(t)$. Since all inter-event times are exponentially distributed, the system retains no memory, and decision epochs can be restricted to times when the state changes. Using the memoryless property, we can formulate the problem as an MDP and focus on Markovian policies for which actions at each decision epoch depend solely on the current state. A control policy ℓ specifies for each state $\mathbf{x} = (x_1, .., x_m)$, the action $\mathbf{u}^\ell(\mathbf{x}) = (u^1, .., u^m, u_1, .., u_n)$, $u^i, u_j \in \{0, 1\}$, $\forall i, j$; where $u^i = 1$ means produce component i, and $u^i = 0$ means do not produce component i; $u_j = 1$ means satisfy demand for product j, and $u_j = 0$ means reject demand for product j. Thus there is at most one outstanding order for each component at any time. Also, as component orders are not part of our system state, these can in effect be cancelled upon transition to a new state. Both of these assumptions are standard in the literature (see Ha 1997, and Benjaafar and ElHafsi 2006). (Our numerical results suggest that the latter assumption is acceptable: Orders are cancelled optimally in 54% of the instances in Section 3.3. However, for those instances, if the optimal policy of our model is followed but orders are never cancelled, it increases costs by no more than 0.11%, and the average cost increase is virtually 0%.)

For a given policy ℓ and starting state $\mathbf{x} \in \mathbb{N}_0^m$ (where $\mathbb{N}_0$ is the set of nonnegative integers and $\mathbb{N}_0^m$ is its m-dimensional cross product), the average cost per unit time over an infinite planning horizon $v^\ell(\mathbf{x})$ can be written as follows (see, for example, Chapter 2 and ElHafsi et al. 2008):

$$v^\ell(\mathbf{x}) = \lim_{T\to\infty} \sup \frac{1}{T} \left\{ \sum_{i=1}^{m} \int_0^T h_i(X_i(t))dt + \sum_{j=1}^{n} \int_0^T c_j dN_j(t) \right\},$$

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

We next formulate a linear program to find a global optimal solution to the above problem. First, denote by $\mathbb{U}(\mathbf{x})$ the set of admissible actions at state $\mathbf{x}$. Also, define $\nu_{\mathbf{y}|\mathbf{x},\mathbf{u}}$ as the rate at which the system moves from state $\mathbf{x}$ to state $\mathbf{y}$ if action $\mathbf{u} = (u^1, .., u^m, u_1, .., u_n) \in \mathbb{U}(\mathbf{x})$ is chosen, and $\pi_{\mathbf{x},\mathbf{u}}$ as the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{u} \in \mathbb{U}(\mathbf{x})$ is chosen. Then, the globally optimal average cost Z^* can be found by solving the following linear program (see Puterman 1994, for an explanation of the Linear Programming method to solve MDPs):

$$(\mathcal{LP}) \quad \text{minimize} \quad \sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} h(\mathbf{x})\pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} \sum_{j:u_j=0} \lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

$$\text{subject to} \quad \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{y})} \pi_{\mathbf{y},\mathbf{u}} \sum_{\mathbf{x}\in\mathbb{N}_0^m} \nu_{\mathbf{x}|\mathbf{y},\mathbf{u}} - \sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} \nu_{\mathbf{y}|\mathbf{x},\mathbf{u}} \pi_{\mathbf{x},\mathbf{u}} = 0, \quad \forall \mathbf{y} \in \mathbb{N}_0^m, \tag{3.2.1}$$

$$\sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} \pi_{\mathbf{x},\mathbf{u}} = 1, \tag{3.2.2}$$

$$\pi_{\mathbf{x},\mathbf{u}} \geq 0, \quad \forall \mathbf{x} \in \mathbb{N}_0^m, \quad \forall \mathbf{u} \in \mathbb{U}(\mathbf{x}). \tag{3.2.3}$$

The first term of the objective function corresponds to the time-average inventory holding cost and the second term corresponds to the time-average cost of lost sales. Constraints (3.2.1) and (3.2.2) are the balance equations and yield the limiting probability values.

3.2.2 Formulation of Heuristic Policies

We next describe the heuristic policies for our general model, and develop MIP formulations to compute the optimal average cost within each heuristic class.

Lattice-Dependent Base-Stock and Lattice-Dependent Rationing (LBLR): We introduce the notation $\mathbb{L}(\mathbf{p}, \mathbf{r}) = \{\mathbf{p} + k\mathbf{r} : k \in \mathbb{N}_0\}$ to denote an m-dimensional lattice with initial vector $\mathbf{p} \in \mathbb{N}_0^m$ and common difference $\mathbf{r} \in \mathbb{N}_0^m$, where $\exists i$ such that $p_i < r_i$. We also define $\mathbf{\Delta}^i = (\Delta_1^i, \Delta_2^i, .., \Delta_m^i)$ and $\mathbf{\Delta}_j = (\Delta_{j1}, \Delta_{j2}, .., \Delta_{jm})$ as m-dimensional vectors of nonnegative integers. With these we describe an LBLR policy as follows: (i) Inventory replenishment of each component i follows a lattice-dependent base-stock policy with lattice-dependent base-stock levels $\mathbf{S}_i(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)$ such that a batch of component i is produced if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}^i)$ is less than $\mathbf{S}_i(\mathbf{p})$, and (ii) inventory allocation for each product j follows a lattice-dependent rationing policy with lattice-dependent rationing levels $\mathbf{R}_j(\mathbf{p}) \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j)$ such that a demand for product j is satisfied if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p}, \mathbf{\Delta}_j)$ is greater than or equal to $\mathbf{R}_j(\mathbf{p})$. An illustration of such a policy for a 2-component 2-product system is shown in Figure 3.2.

We could optimize over the vectors $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ to obtain the LBLR policy with the least average cost. But it is both time-consuming and unnecessary to do so, as the following rule of thumb achieves the globally optimal cost in *each* of our numerical instances: Choose any vectors $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ such that (i) $\Delta_i^i = \max_j a_{ij}$, $\forall i$; (ii) $\Delta_k^i = \min_j a_{kj}$, $\forall k \neq i$; and (iii) $\Delta_{ji} = a_{ij^*}$, where $j^* = \arg\max_{k \neq j} c_k$, $\forall i, j$ (see Figure 3.2 for an example). Such a selection of the vectors $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ is also consistent with previously established optimality results for ATO systems (see Chapter 2, and Benjaafar and ElHafsi 2006).

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

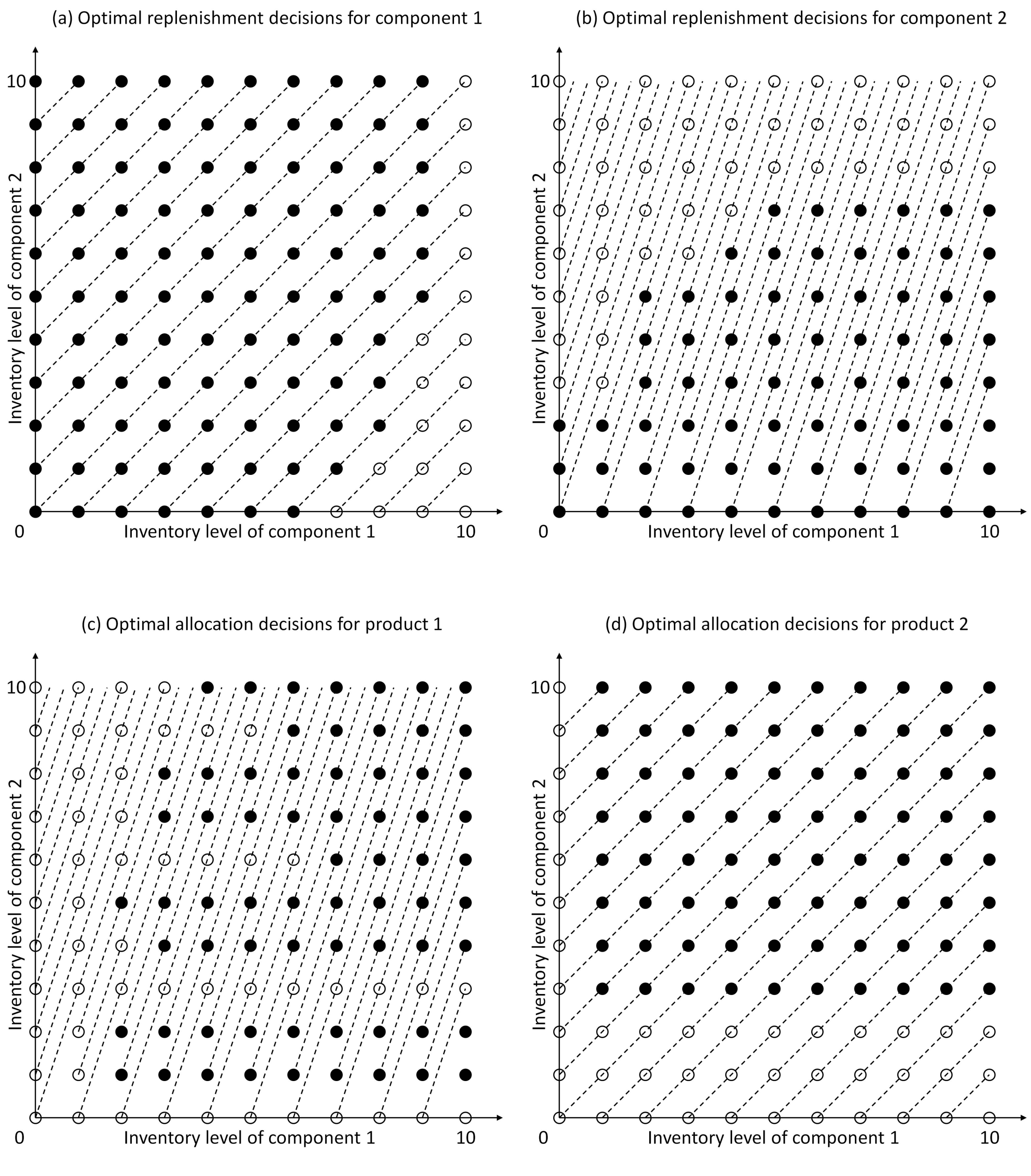

Figure 3.2 The illustration of LBLR for a 2x2 system with $a_{11} = a_{12} = a_{21} = 1$, $a_{22} = 3$, $q_1 = 1$, $q_2 = 3$, $h_1 = 1$, $h_2 = 5$, $\mu_1 = \mu_2 = \lambda_1 = \lambda_2 = 1$, $c_1 = 20$, $c_2 = 100$. In graphs (a) and (b), a filled circle means produce a batch of components at the corresponding inventory levels. In graphs (c) and (d), a filled circle means fulfill the demand at the corresponding inventory levels. In graphs (a)-(d), each dashed line forms a different lattice; its slope is determined by $\mathbf{\Delta}^1 = (1, 1)$, $\mathbf{\Delta}^2 = (1, 3)$, $\mathbf{\Delta}_1 = (1, 3)$, and $\mathbf{\Delta}_2 = (1, 1)$, respectively.

that the system is in state $\mathbf{x}$ and action $\mathbf{u}$ is chosen should be zero when the rationing level for product j equals $\mathbf{b}$ on the lattice with initial vector $\mathbf{p}$. Lastly, define $z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}$ and $z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}$ as binary variables as follows:

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

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

We are now ready to describe the constraints of the MIP problem. First, the optimal solution of the MIP problem should satisfy constraints (3.2.1)-(3.2.3) of the LP formulation of the optimal policy. Also, on each lattice, the optimal solution should select exactly one base-stock level for each component and one rationing level for each product. To this end, we impose the following constraints:

$$\sum_{\mathbf{b} \in \mathbb{L}(\mathbf{p}, \boldsymbol{\Delta}^i)} z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})} = 1, \quad \forall \mathbf{p} \text{ and } \forall i, \tag{3.2.4}$$

$$\sum_{\mathbf{b} \in \mathbb{L}(\mathbf{p}, \boldsymbol{\Delta}_j)} z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})} = 1, \quad \forall \mathbf{p} \text{ and } \forall j. \tag{3.2.5}$$

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

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{S}_i(\mathbf{p},\mathbf{b})} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}, \quad \forall \mathbf{p}, \ \forall \mathbf{b}, \text{ and } \forall i, \tag{3.2.6}$$

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{R}_j(\mathbf{p},\mathbf{b})} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}, \quad \forall \mathbf{p}, \ \forall \mathbf{b}, \text{ and } \forall j. \tag{3.2.7}$$

In constraint (3.2.6), if $z_{\mathbf{b}}^{\mathbf{S}_i(\mathbf{p})}$ equals one, then all limiting probability variables corresponding to the state-action pairs in set $\mathbb{S}_i(\mathbf{p}, \mathbf{b})$ are forced to equal zero. Likewise, in constraint (3.2.7), if $\mathbf{z}_{\mathbf{b}}^{\mathbf{R}_j(\mathbf{p})}$ equals one, then all limiting probability variables corresponding to the state-action pairs in set $\mathbb{R}_j(\mathbf{p}, \mathbf{b})$ are forced to equal zero. Otherwise, these constraints become redundant.

(See Bhandari et al. 2008, for a similar MIP formulation in a different context.)

The optimal average cost of this policy Z_{LBLR} can be found by solving the following MIP problem:

$$\text{(LBLR)} \qquad \text{minimize} \qquad \sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} h(\mathbf{x})\pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x}\in\mathbb{N}_0^m} \sum_{\mathbf{u}\in\mathbb{U}(\mathbf{x})} \sum_{j:u_j=0} \lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

$$\text{subject to} \qquad (3.2.1) - (3.2.7).$$

State-Dependent Base-Stock and State-Dependent Rationing (SBSR): Define $\mathbf{x}_{-i} = (x_1, .., x_{i-1}, x_{i+1}, .., x_m)$ as an $m-1$ dimensional vector of the inventory levels for components $k \neq i$. With this we describe an SBSR policy as follows (as in Theorem 1 of Benjaafar and ElHafsi 2006): (i) Inventory replenishment of each component i follows a state-dependent base-stock policy with state-dependent base-stock levels $S_i(\mathbf{x}_{-i})$ such that a batch of component i is produced if and only if $x_i \leq S_i(\mathbf{x}_{-i})$; and (ii) inventory allocation for demand class j follows a state-dependent rationing policy with state-dependent rationing levels $R_{ij}(\mathbf{x}_{-i})$, $\forall i$, such that a demand from class j is fulfilled if and only if $x_i \geq R_{ij}(\mathbf{x}_{-i})$, $\forall i$. Different demand classes in Benjaafar and ElHafsi (2006) correspond to different products in our model.

The SBSR policy has the following additional properties (again as in Benjaafar and ElHafsi 2006): (a) The base-stock level of one component is nondecreasing in the inventory level of any other component; (b) a unit increase in the inventory level of one component leads to at most a unit increase in the base-stock level of any other component; (c) the rationing level for any demand class at one component is nonincreasing in the inventory level of any other component; (d) the production of a component is never interrupted once it is initiated; (e) for each component, the rationing level for any demand class is greater than or equal to the rationing level for the demand class with the next higher lost sale cost; and (f) demands with the highest lost sale cost are always satisfied if sufficient inventory exists. Note that properties (e) and (f) are inapplicable to our general model, as our products differ not only in their lost sale costs but also in their component usage rates, and thus we will not enforce properties (e) and (f) in our numerical experiments. We also omit property (d) from SBSR to keep the state space manageable. Nevertheless in our numerical experiments, SBSR performs no worse than it would under property (d).

Benjaafar and ElHafsi (2006) showed that, under Markovian assumptions on production and demand, the SBSR policy is optimal when the system involves a *single* end-product and multiple components. In their setting, one unit of each component is assembled into the end-product, which is demanded by multiple demand classes.

We proceed to the MIP formulation of this heuristic class. Define the set $\mathbb{S}_i(\mathbf{x}_{-i}, b) = \{(\mathbf{y}, \mathbf{u}) : \mathbf{y} \in \mathbb{N}_0^m,\ \mathbf{u} \in \mathbb{U}(\mathbf{y}),\ \mathbf{y}_{-i} = \mathbf{x}_{-i}, \text{ and } \sum_{\mathbf{y},\mathbf{u}} \pi_{\mathbf{y},\mathbf{u}} = 0 \Leftrightarrow S_i(\mathbf{x}_{-i}) = b\}$ for $b \in \mathbb{N}_0$, and the set $\mathbb{R}_{ij}(\mathbf{x}_{-i}, b) = \{(\mathbf{y}, \mathbf{u}) : \mathbf{y} \in \mathbb{N}_0^m,\ \mathbf{u} \in \mathbb{U}(\mathbf{y}),\ \mathbf{y}_{-i} = \mathbf{x}_{-i}, \text{ and } \sum_{\mathbf{y},\mathbf{u}} \pi_{\mathbf{y},\mathbf{u}} = 0 \Leftrightarrow R_{ij}(\mathbf{x}_{-i}) = b\}$ for $b \in \mathbb{N}_0$. Also, define $z_b^{S_i(\mathbf{x}_{-i})}$ and $z_b^{R_{ij}(\mathbf{x}_{-i})}$ as binary variables as follows:

$$z_b^{S_i(\mathbf{x}_{-i})} = \begin{cases} 1 & \text{if } S_i(\mathbf{x}_{-i}) = b, \\ 0 & \text{otherwise.} \end{cases}$$

$$z_b^{R_{ij}(\mathbf{x}_{-i})} = \begin{cases} 1 & \text{if } R_{ij}(\mathbf{x}_{-i}) = b, \\ 0 & \text{otherwise.} \end{cases}$$

We next describe constraints of the MIP problem. Again, the optimal solution of the MIP problem should satisfy constraints (3.2.1)-(3.2.3). In addition, the optimal solution should select exactly one base-stock level for each component and one rationing level for each product at each component, given the inventory levels of all other components:

$$\sum_{b \in \mathbb{N}_0} z_b^{S_i(\mathbf{x}_{-i})} = 1, \quad \forall i \text{ and } \forall \mathbf{x}_{-i}, \tag{3.2.8}$$

$$\sum_{b \in \mathbb{N}_0} z_b^{R_{ij}(\mathbf{x}_{-i})} = 1, \quad \forall i,\ \forall j, \text{ and } \forall \mathbf{x}_{-i}. \tag{3.2.9}$$

The constraint below ensures that (a) the base-stock level of each component is nondecreasing in the inventory levels of other components, and (b) a unit increase in the inventory level of one component leads to at most a unit increase in the base-stock level of another component:

$$z_b^{S_i(\mathbf{x}_{-i})} \leq z_b^{S_i(\mathbf{x}_{-i}+e_k)} + z_{b+1}^{S_i(\mathbf{x}_{-i}+e_k)}, \quad \forall k \neq i,\ \forall \mathbf{x}_{-i}, \text{ and } \forall b \in \mathbb{N}_0. \tag{3.2.10}$$

The constraint below ensures that (c) the rationing level for each product at each component is nonincreasing in the inventory levels of other components:

$$z_b^{R_{ij}(\mathbf{x}_{-i})} \leq \sum_{0 \leq b' \leq b} z_{b'}^{R_{ij}(\mathbf{x}_{-i}+e_k)}, \quad \forall k \neq i, \ \forall j, \ \forall \mathbf{x}_{-i}, \ \text{and} \ \forall b \in \mathbb{N}_0. \tag{3.2.11}$$

The binary variables are linked to the appropriate limiting probability variables as follows:

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{S}_i(\mathbf{x}_{-i},b)} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_b^{S_i(\mathbf{x}_{-i})}, \quad \forall b, \ \forall i, \ \text{and} \ \forall \mathbf{x}_{-i}, \tag{3.2.12}$$

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{R}_{ij}(\mathbf{x}_{-i},b)} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_b^{R_{ij}(\mathbf{x}_{-i})}, \quad \forall b, \ \forall i, \ \forall j, \ \text{and} \ \forall \mathbf{x}_{-i}. \tag{3.2.13}$$

The optimal average cost of this policy Z_{SBSR} can be found by solving the following MIP problem:

$$\text{(SBSR)} \quad \text{minimize} \quad \sum_{\mathbf{x} \in \mathbb{N}_0^m} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x} \in \mathbb{N}_0^m} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} \sum_{j: u_j = 0} \lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

$$\text{subject to} \quad (3.2.1) - (3.2.3) \text{ and } (3.2.8) - (3.2.13).$$

Fixed Base-Stock and Fixed Rationing (FBFR): Lastly, we describe an FBFR policy as follows (as in Benjaafar and ElHafsi 2006): (i) Inventory replenishment of each component i follows a base-stock policy with a fixed base-stock level S_i such that a batch of component i is produced if and only if $x_i \leq S_i$; and (ii) inventory allocation for each product j follows a rationing policy with a vector of fixed rationing levels $\mathbf{R}_j = (R_{1j}, R_{2j}, .., R_{mj})$ such that a demand for product j is satisfied if and only if $x_i \geq R_{ij}$, $\forall i$.

We next describe the MIP problem of this heuristic class. Define the set $\mathbb{S}_i(b) = \{(\mathbf{x}, \mathbf{u}) : \mathbf{x} \in \mathbb{N}_0^m, \ \mathbf{u} \in \mathbb{U}(\mathbf{x}), \ \sum_{\mathbf{x},\mathbf{u}} \pi_{\mathbf{x},\mathbf{u}} = 0 \Leftrightarrow S_i = b\}$ for $b \in \mathbb{N}_0$, and the set $\mathbb{R}_j(\mathbf{b}) = \{(\mathbf{x}, \mathbf{u}) : \mathbf{x} \in \mathbb{N}_0^m, \ \mathbf{u} \in \mathbb{U}(\mathbf{x}), \ \sum_{\mathbf{x},\mathbf{u}} \pi_{\mathbf{x},\mathbf{u}} = 0 \Leftrightarrow \mathbf{R}_j = \mathbf{b}\}$ for $\mathbf{b} \in \mathbb{N}_0^m$. Also, define $z_b^{S_i}$ and $z_{\mathbf{b}}^{\mathbf{R}_j}$ as binary variables as follows:

$$z_b^{S_i} = \begin{cases} 1 & \text{if } S_i = b, \\ 0 & \text{otherwise.} \end{cases}$$

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

Once again, the optimal solution of the MIP problem should satisfy constraints (3.2.1)-(3.2.3). In addition, the optimal solution should select exactly one base-stock level for each component and one rationing level for each product:

$$\sum_{b \in \mathbb{N}_0} z_b^{S_i} = 1, \quad \forall i, \tag{3.2.14}$$

$$\sum_{\mathbf{b} \in \mathbb{N}_0^m} z_{\mathbf{b}}^{\mathbf{R}_j} = 1, \quad \forall j. \tag{3.2.15}$$

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

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{S}_i(b)} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_b^{S_i}, \quad \forall b \text{ and } \forall i, \tag{3.2.16}$$

$$\sum_{(\mathbf{x},\mathbf{u}) \in \mathbb{R}_j(\mathbf{b})} \pi_{\mathbf{x},\mathbf{u}} \leq 1 - z_{\mathbf{b}}^{\mathbf{R}_j}, \quad \forall \mathbf{b} \text{ and } \forall j. \tag{3.2.17}$$

The optimal average cost Z_{FBFR} can be found by solving the following MIP problem:

$$\text{(FBFR)} \qquad \text{minimize} \qquad \sum_{\mathbf{x} \in \mathbb{N}_0^m} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{u}} + \sum_{\mathbf{x} \in \mathbb{N}_0^m} \sum_{\mathbf{u} \in \mathbb{U}(\mathbf{x})} \sum_{j: u_j = 0} \lambda_j c_j \pi_{\mathbf{x},\mathbf{u}}$$

$$\text{subject to} \qquad (3.2.1) - (3.2.3) \text{ and } (3.2.14) - (3.2.17).$$

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

Proposition 3.2.1. $Z^* \leq Z_{LBLR} \leq Z_{SBSR} \leq Z_{FBFR}$

Proof. See Appendix B.1. □

Proposition 3.2.1 establishes that LBLR performs no worse than the other heuristics. In the next section, we conduct numerical experiments to test the significance of this finding and provide insights into the performance of LBLR relative to the other heuristics.

3.3 Numerical Experiments

Our primary goals in this section are to examine the performance of LBLR relative to SBSR, investigate how system parameters affect the cost advantage of LBLR over SBSR, and explain why such an advantage occurs. Our secondary goal is to examine the performance of LBLR relative to FBFR, which is easier to implement. For ease of exposition, we confine our detailed analysis to 2-product 2-component systems in which either (1) one product requires a subset of components used by the other product (nested structure, Section 3.3.1), or (2) neither product requires a set of components that is a subset of those used by the other product (non-nested structure, Section 3.3.2). We believe these two cases capture the essence of the ATO problem. However, after comparing computational efforts in Section 3.3.3, we also report numerical results for selected larger instances in Section 3.3.4.

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

	A	B	C	D
ϕ	1	1	2	1
γ	1	2	1	3

For each of our 2-product 2-component systems, we generate numerical instances by varying values of the related parameters (i.e., q_i, h_i, c_j, and λ_j) under linear holding cost rates (i.e., $h_i(x_i) = h_i x_i$). For each generated instance, we solve the LP and MIP problems to compute the average costs and corresponding product fill rates (denoted by f_j). We compare the heuristic policies in terms of (i) their percentage differences from optimal cost Z^*, calculated as $100 \times \frac{Z_H - Z^*}{Z^*}$ where $H \in$ {LBLR, SBSR, FBFR}; and (ii) their computational times. We coded the LP and MIP formulations in the GAMS programming language, incorporating CPLEX 10.1 optimization subroutines, and used a dual processor WinNT server, with Intel Core i7 2.67 GHz processor and 8 GB of RAM. We restricted the computation time of any instance to be no more than 1000 seconds.

An important note here is that, although our 2-product 2-component systems violate the conditions ensuring the optimality of LBLR (i.e., the generalized M-system product structure,

Table 3.1 Parameters for numerical instances.

Products j, k	$\mathbf{q} = (q_\phi, q_\gamma)$	h_ϕ, h_γ	c_j/c_k	c_k	λ_j, λ_k	Δ^ϕ	Δ^γ	Δ_j	Δ_k
$j = A$, $k = B$	$\{(1,1),(1,2)\}$	$\{1,3,5\}$	$\{.2,.4,.6,.8,1\}$	100	$\{.5,1\}$	(1,1)	(1,2)	(1,2)	(1,1)
$j = A$, $k = D$	$\{(1,1),(1,3)\}$	$\{1,3,5\}$	$\{.2,.4,.6,.8,1\}$	100	$\{.5,1\}$	(1,1)	(1,3)	(1,3)	(1,1)
$j = B$, $k = C$	$\{(1,2),(2,1)\}$	$\{1,3,5\}$	$\{.2,.4,.6,.8,1\}$	100	$\{.5,1\}$	(2,1)	(1,2)	(2,1)	(1,2)
$j = B$, $k = D$	$\{(1,2),(1,3)\}$	$\{1,3,5\}$	$\{.2,.4,.6,.8,1\}$	100	$\{.5,1\}$	(1,2)	(1,3)	(1,3)	(1,2)
$j = C$, $k = D$	$\{(2,1),(1,3)\}$	$\{1,3,5\}$	$\{.2,.4,.6,.8,1\}$	100	$\{.5,1\}$	(2,1)	(1,3)	(1,3)	(2,1)

Notes. $\mu_\phi = \mu_\gamma = 1$ in all instances. The last four columns list optimal $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ values for LBLR policy. $Z^* = Z_{LBLR}$ in all instances.

see Chapter 2), *LBLR yields the globally optimal cost in each of our more than 1800 compiled instances.* Table 3.1 exhibits the parameters for 1800 of these instances; the products that we select to construct our 2-product 2-component systems are listed in the first column, the corresponding ranges of parameters are listed in the intermediate columns, and the vectors $\mathbf{\Delta}^i$ and $\mathbf{\Delta}_j$ used to compute Z_{LBLR} are listed in the last four columns. *LBLR also yields the globally optimal cost in all numerical instances generated in Sections 3.3.1, 3.3.2, and 3.3.4.*

3.3.1 Nested Structure

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

LBLR vs. SBSR. We first analyze the percentage gaps between LBLR and SBSR. We observe from Table 3.2 that, for fixed holding cost rates, the largest two gaps always occur when the ratio of lost sale costs is 0.2 and 0.4: Products become less differentiated when the ratio takes greater values, and therefore they should be treated as if they are almost equally important in stock allocation decisions. Since stock rationing becomes less crucial as products become less differentiated, the benefit of a lattice-dependent rationing policy decreases; hence the gaps between LBLR and SBSR are lower at higher values of this ratio. An important insight here is that product differentiation is driven *both* by differences in lost sale costs and component usage rates. Thus, when the ratio of lost sale costs is sufficiently large but lower than 1 (say 0.6 and 0.8), we expect products A and D to be only slightly differentiated, since product A requires fewer components. But, when the ratio is 1, products again become significantly differentiated, due to the difference in component usage rates. This explains why the fill rates of product D are lower than those of product A when the ratio is 1. However, such differentiation results in relatively small optimal cost gaps.

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

We list computational times for the heuristics in the last three columns of this and subsequent tables. It is clear LBLR has distinct computational advantage over SBSR, and a slight one over FBFR. We discuss computational times in greater detail in Section 3.3.3.

Next, we vary demand rates, as shown in Table 3.3. First, we observe that, for a fixed demand rate of product A, the largest two gaps always occur when the demand rate of product D is 0.5 and 1. When λ_D takes greater values, the cost of rejecting the demand per unit time for product D relative to the cost of rejecting all demands per unit time (i.e., $\frac{\lambda_D c_D}{\lambda_A c_A + \lambda_D c_D}$) is higher. Since product D has a greater impact on total costs, product D dominates product A and the system is close to the one with a single product.

Hence, as product D begins to dominate product A, lattice-dependent rationing becomes equivalent to state-dependent rationing, and the gap decreases. But, when λ_D is 0.5 or 1, since

Table 3.2 Numerical results for nested structure.

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_A/c_D	Avg. cost	f_A	f_D	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	54.974	0.160	0.707	0.000	1.397	1.481	3.18	**1000**	6.02
-	-	0.4	69.827	0.300	0.669	0.000	1.054	1.293	2.57	352.40	5.20
-	-	0.6	83.416	0.340	0.642	0.000	0.495	0.513	2.64	962.11	3.94
-	-	0.8	96.217	0.407	0.582	0.000	0.178	0.312	2.95	144.47	5.65
-	-	1.0	106.280	0.631	0.360	0.000	0.123	0.994	2.41	15.95	6.65
-	3	0.2	63.591	0.213	0.690	0.000	2.000	2.511	2.67	**1000**	15.13
-	-	0.4	78.085	0.316	0.651	0.000	1.550	2.456	2.22	**1000**	6.44
-	-	0.6	91.221	0.381	0.599	0.000	0.852	1.880	2.59	**1000**	6.76
-	-	0.8	102.751	0.474	0.508	0.000	0.364	1.790	2.22	146.89	6.64
-	-	1.0	111.582	0.629	0.356	0.000	0.218	2.273	2.45	6.20	7.29
-	5	0.2	71.364	0.244	0.668	0.000	2.706	3.446	2.25	**1000**	11.46
-	-	0.4	85.140	0.358	0.610	0.000	1.711	3.391	2.05	296.75	7.73
-	-	0.6	97.362	0.423	0.551	0.000	1.014	2.668	2.36	191.44	6.75
-	-	0.8	107.718	0.511	0.466	0.000	0.738	2.288	2.47	48.20	4.37
-	-	1.0	116.091	0.623	0.358	0.000	0.128	3.164	2.16	7.10	5.37
3	1	0.2	61.368	0.112	0.689	0.000	0.917	1.119	2.03	9.14	3.90
-	-	0.4	76.644	0.328	0.632	0.000	0.620	0.680	2.54	80.22	4.20
-	-	0.6	89.403	0.389	0.598	0.000	0.377	0.391	3.04	65.80	4.27
-	-	0.8	101.044	0.451	0.541	0.000	0.214	0.298	2.21	184.04	5.30
-	-	1.0	110.406	0.608	0.385	0.000	0.033	0.605	2.67	25.97	4.81
-	3	0.2	70.509	0.132	0.670	0.000	1.103	2.088	2.43	339.01	9.51
-	-	0.4	85.362	0.337	0.617	0.000	1.023	1.812	2.47	120.75	4.70
-	-	0.6	97.654	0.429	0.555	0.000	0.668	1.588	2.47	624.89	5.40
-	-	0.8	108.447	0.500	0.487	0.000	0.351	1.646	2.46	53.17	7.46
-	-	1.0	116.867	0.615	0.375	0.000	0.436	2.347	2.47	50.95	7.34
-	5	0.2	78.196	0.150	0.652	0.000	1.270	2.136	2.38	67.71	4.04
-	-	0.4	92.564	0.373	0.578	0.000	1.481	2.638	2.63	110.78	10.65
-	-	0.6	104.024	0.466	0.515	0.000	0.710	2.622	2.25	48.93	6.79
-	-	0.8	113.995	0.531	0.453	0.000	0.409	2.685	2.55	8.93	5.88
-	-	1.0	122.202	0.622	0.365	0.000	0.222	2.885	2.74	11.50	5.68
5	1	0.2	65.655	0.123	0.664	0.000	0.786	1.250	2.00	89.78	3.15
-	-	0.4	81.147	0.328	0.607	0.000	0.755	0.925	2.05	68.08	4.27
-	-	0.6	93.924	0.407	0.561	0.000	0.101	0.153	2.28	43.85	5.68
-	-	0.8	105.146	0.483	0.505	0.000	0.444	0.588	2.37	204.44	6.56
-	-	1.0	114.037	0.588	0.406	0.000	0.030	0.348	2.69	4.45	3.28
-	3	0.2	75.148	0.137	0.646	0.000	0.816	2.354	2.18	27.74	5.50
-	-	0.4	90.404	0.336	0.590	0.000	0.907	1.585	2.07	213.12	4.48
-	-	0.6	102.664	0.450	0.518	0.000	0.359	1.256	2.45	7.93	4.83
-	-	0.8	113.290	0.553	0.429	0.000	0.178	1.833	2.72	35.52	12.79
-	-	1.0	121.537	0.606	0.384	0.000	0.308	2.119	2.36	14.06	9.24
-	5	0.2	82.579	0.151	0.612	0.000	0.679	1.672	2.01	113.45	3.79
-	-	0.4	97.557	0.355	0.556	0.000	1.332	2.829	2.37	229.81	8.86
-	-	0.6	109.376	0.478	0.484	0.000	0.434	2.199	2.43	90.09	6.58
-	-	0.8	119.242	0.576	0.397	0.000	0.174	2.485	2.31	15.23	5.58
-	-	1.0	127.367	0.612	0.373	0.000	0.422	2.850	3.19	17.86	6.75

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

Table 3.3 Numerical results for nested structure.

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_A	λ_D	Average cost	f_A	f_D	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	38.387	0.506	0.712	0.000	2.157	2.668	1.93	26.14	8.82
-	1.0	62.807	0.199	0.693	0.000	0.777	1.616	2.46	199.08	6.21
-	1.5	96.090	0.000	0.559	0.000	0.081	1.157	2.43	4.37	2.02
-	2.0	138.053	0.000	0.440	0.000	0.056	1.107	2.18	4.12	2.20
-	2.5	183.980	0.000	0.359	0.000	0.035	1.089	2.06	3.11	2.10
1.0	0.5	44.757	0.544	0.679	0.000	1.562	3.704	2.07	**1000**	12.75
-	1.0	71.364	0.244	0.668	0.000	2.706	3.446	2.39	**1000**	11.70
-	1.5	106.032	0.045	0.552	0.000	0.128	1.102	2.95	8.14	4.35
-	2.0	148.053	0.000	0.440	0.000	0.052	1.032	2.36	5.43	2.22
-	2.5	193.980	0.000	0.359	0.000	0.034	1.033	2.17	2.40	2.33
1.5	0.5	52.369	0.433	0.667	0.000	1.566	4.785	2.52	30.65	11.31
-	1.0	80.498	0.194	0.659	0.000	2.195	4.131	2.72	**1000**	21.45
-	1.5	115.877	0.035	0.551	0.000	0.251	1.143	2.52	99.63	4.81
-	2.0	158.053	0.000	0.440	0.000	0.049	0.967	2.20	2.61	2.10
-	2.5	203.980	0.000	0.359	0.000	0.032	0.982	2.28	2.01	2.52
2.0	0.5	61.127	0.326	0.671	0.000	1.276	4.370	2.53	122.25	10.09
-	1.0	90.017	0.157	0.644	0.000	1.742	4.064	2.51	**1000**	8.69
-	1.5	125.770	0.031	0.550	0.000	0.316	1.138	2.69	19.05	6.16
-	2.0	168.053	0.000	0.440	0.000	0.046	0.909	2.35	5.93	2.06
-	2.5	213.980	0.000	0.359	0.000	0.030	0.936	2.26	2.61	2.39
2.5	0.5	70.416	0.259	0.678	0.000	0.991	3.616	2.21	**1000**	7.10
-	1.0	99.717	0.129	0.640	0.000	1.451	3.745	2.47	**1000**	12.51
-	1.5	135.675	0.031	0.549	0.000	0.363	1.125	2.70	86.33	5.96
-	2.0	178.053	0.000	0.440	0.000	0.043	0.858	2.42	2.00	1.79
-	2.5	223.980	0.000	0.359	0.000	0.029	0.894	2.27	2.68	2.34

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

product D has a higher lost sale cost, the effect of product dominance is less significant and LBLR can outperform SBSR by a couple of percent. Also, we observe that as λ_A increases while λ_D is 0.5, the gap declines (there is a minor exception at $\lambda_A = 1.5$), but as λ_A increases while λ_D is 1, the gap first increases and then decreases. Our explanation is that again, if one product clearly dominates the other, a low percentage gap results. But when one product is only slightly dominant, system performance can be improved by LBLR. Finally, as λ_A increases while λ_D is 1.5, the gap increases; since product D has a higher lost sale cost, when λ_D is 1.5 the dominance of product D is so strong that increasing λ_A up to 2.5 only mitigates this dominance. We expect the gap to fall at higher values of λ_A, since product A will eventually dominate product D.

Another important observation from Table 3.3 is that, as both demand arrival rates go from 0.5 to 2.5, the gap first increases and then declines. When capacity is high relative to demand (i.e., $\lambda_A = \lambda_D = 0.5$), it is optimal to hold less inventory and therefore the benefit of LBLR is lower. When capacity is scarce (i.e., $\lambda_A, \lambda_D \geq 1.5$), the system focuses more on filling the high value item, even under high base-stock levels. Consequently, it is not critical to ration inventory in a sophisticated manner, and again the benefit of LBLR is lower.

Our overall conclusion is that LBLR may substantially outperform SBSR when demands for both products are fulfilled in significant quantities, when products are highly differentiated, or when products differ mainly in their lost sale costs. Thus we predict that the gap between LBLR and SBSR will increase as the fill rates of both products increase, as the difference of fill rates increases, or as the ratio of lost sale costs decreases. To test these predictions we use the data in Tables 3.2 and 3.3 in a regression model for the percentage gap between SBSR and LBLR with the following independent variables: (i) f_A, (ii) $f_D - f_A$, and (iii) c_A/c_D. As we report in the upper left panel of Table 3.4, variables (i)-(iii) have the predicted sign and are statistically significant at $p = 0.001$. The results continue to hold when stepwise regression is used by including all the candidate variables (i.e., system parameters) in the model and eliminating those that are statistically insignificant.

We also test the above prediction in Examples (b) and (c), and find that it remains true in Example (b), but not in Example (c) (see the upper right and lower left panels of Table 3.4). The ambiguity in Example (c) arises because SBSR performs so well that the gaps are insignificant for many data points. An explanation of this performance is that the lower batch sizes in Example (c) require less flexibility in inventory control decisions, making LBLR less important.

LBLR vs. FBFR. As expected, the percentage gaps between LBLR and FBFR are higher than the ones between LBLR and SBSR. In Table 3.2, in contrast to the comparison of LBLR and SBSR, we observe significant gaps between LBLR and FBFR when products differ only in their component usage rates (i.e. when $c_A/c_D = 1$). This benefit comes from the coordination of the components achieved by LBLR and SBSR but not FBFR: Since batch sizes for components ϕ and γ are 1 and 3, respectively, it is easier to match supply with the demand of product D (using 1 and 3 units of components ϕ and γ), compared to product A (using 1 unit of each component). Hence, it becomes more crucial to coordinate inventory decisions when product A becomes more important, as is the case when $c_A/c_D = 1$. Likewise, in Table 3.3 we

Table 3.4 Regression results.

3.1(a). Products A and D, $q_\phi = 1$, and $q_\gamma = 3$

Variable	Estimate	SE	t-stat.	p-value
Intercept	−0.5191	0.3977	−1.3053	0.1963
c_A/c_D	−2.0499	0.4612	−4.4442	0.0000*
f_A	5.2332	0.5169	10.1246	0.0000*
$f_D - f_A$	2.4549	0.6277	3.9108	0.0002*

$N = 70$, $R^2 = 67.30\%$, and adjusted-$R^2 = 65.82\%$.

3.1(b). Products A and B, $q_\phi = 1$, and $q_\gamma = 2$

Variable	Estimate	SE	t-stat.	p-value
Intercept	−0.2461	0.2209	−1.1141	0.2693
c_A/c_B	−1.8331	0.2853	−6.4260	0.0000*
f_A	4.0617	0.3363	12.0794	0.0000*
$f_B - f_A$	1.7025	0.3092	5.5060	0.0000*

$N = 70$, $R^2 = 76.41\%$, and adjusted-$R^2 = 75.34\%$.

3.1(c). Products A and B, $q_\phi = 1$, and $q_\gamma = 1$

Variable	Estimate	SE	t-stat.	p-value
Intercept	0.5720	0.1665	3.4354	0.0010*
c_A/c_B	−0.6958	0.2528	−2.7518	0.0076
f_A	−0.2652	0.4360	−0.6081	0.5452
$f_B - f_A$	−0.6361	0.3528	−1.8028	0.0760

$N = 70$, $R^2 = 11.94\%$, and adjusted-$R^2 = 7.94\%$.

3.2. Products B and C, $q_\phi = 2$, and $q_\gamma = 2$

Variable	Estimate	SE	t-stat.	p-value
Intercept	−1.6590	0.2330	−7.1200	0.0000*
f_B	3.3140	0.3770	8.7898	0.0000*
$f_C - f_B$	3.5845	0.3618	9.9076	0.0000*

$N = 70$, $R^2 = 59.46\%$, and adjusted-$R^2 = 58.25\%$.

Notes. SE stands for standard error. Starred p-values indicate that the corresponding variables are statistically significant at probability of 0.001.

see that the gaps between FBFR and the other heuristics are noticeably higher when product A is more highly demanded (especially when $\lambda_D \leq 1 \leq \lambda_A$). These observations underscore the importance of the coordinated inventory decisions when the component batch sizes imperfectly match the component usage rates of the most valuable and/or mostly demanded product.

3.3.2 Non-Nested Structure

We next consider an ATO system with products B and C, and $q_\phi = q_\gamma = 2$, in Tables 3.5 and 3.6.

LBLR vs. SBSR. We observe from Table 3.5 that, for fixed holding costs, LBLR provides the least savings when c_B/c_C is 0.6 (there is a minor exception when $h_\phi = 5$ and $h_\gamma = 3$). For smaller values of c_B/c_C, products are highly differentiated and therefore lattice-dependent rationing greatly improves the system performance. For higher values of c_B/c_C, products are almost equally important since the total numbers of components they require are equal, and product fill rates are quite close to each other. Nevertheless, when c_B/c_C is greater than 0.6, there are cases where the optimal cost gaps between LBLR and SBSR are comparatively large. To understand why this happens, we examined the optimal solutions when c_B/c_C is 1: If inventory levels are equal and sufficiently great to satisfy any demand, it is optimal to

Table 3.5 Numerical results for non-nested structure.

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_B/c_C	Average cost	f_B	f_C	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	45.970	0.135	0.800	0.000	1.416	2.293	2.71	**1000**	5.12
-	-	0.4	61.586	0.313	0.743	0.000	0.671	1.416	2.93	**1000**	7.31
-	-	0.6	72.943	0.505	0.654	0.000	0.090	0.106	2.93	35.29	4.13
-	-	0.8	82.243	0.560	0.615	0.000	0.554	0.556	2.76	43.79	6.37
-	-	1.0	90.722	0.589	0.589	0.000	0.257	0.257	2.91	91.85	5.16
-	3	0.2	52.497	0.138	0.778	0.000	1.158	2.103	2.56	**1000**	3.67
-	-	0.4	68.437	0.294	0.727	0.000	0.576	2.586	2.56	159.70	5.15
-	-	0.6	80.874	0.456	0.653	0.000	0.117	0.904	2.77	17.38	4.48
-	-	0.8	90.622	0.565	0.595	0.000	0.556	0.818	2.73	**1000**	6.55
-	-	1.0	98.944	0.605	0.567	0.000	0.609	0.658	2.97	205.89	6.19
-	5	0.2	57.452	0.122	0.764	0.000	0.782	2.258	2.78	640.69	3.17
-	-	0.4	73.412	0.255	0.726	0.000	0.646	2.919	2.42	155.57	4.40
-	-	0.6	86.437	0.411	0.660	0.000	0.174	1.459	2.31	18.05	5.58
-	-	0.8	97.158	0.516	0.604	0.000	0.888	1.609	2.21	31.80	8.90
-	-	1.0	106.030	0.587	0.555	0.000	0.277	0.705	2.59	22.99	6.78
3	1	0.2	54.913	0.199	0.783	0.000	1.535	2.511	3.15	**1000**	8.69
-	-	0.4	69.340	0.346	0.733	0.000	0.798	1.196	2.97	**1000**	5.02
-	-	0.6	80.220	0.499	0.658	0.000	0.131	0.142	2.95	38.12	5.34
-	-	0.8	90.026	0.539	0.629	0.000	0.190	0.232	3.00	24.12	5.24
-	-	1.0	98.944	0.567	0.605	0.000	0.609	0.658	3.07	131.37	10.68
-	3	0.2	61.973	0.183	0.763	0.000	1.272	3.134	2.95	**1000**	5.28
-	-	0.4	76.955	0.306	0.725	0.000	0.717	2.196	3.08	98.93	5.58
-	-	0.6	88.944	0.473	0.651	0.000	0.111	0.767	3.28	42.94	4.99
-	-	0.8	98.931	0.554	0.607	0.000	0.171	0.621	2.94	83.36	7.46
-	-	1.0	107.503	0.586	0.586	0.000	0.643	0.680	2.77	887.82	6.66
-	5	0.2	67.262	0.165	0.748	0.000	0.990	2.797	2.26	**1000**	3.94
-	-	0.4	82.624	0.272	0.721	0.000	0.736	2.583	2.77	237.54	5.63
-	-	0.6	95.092	0.426	0.656	0.000	0.157	1.287	2.72	16.89	5.04
-	-	0.8	105.871	0.493	0.627	0.000	0.397	1.268	2.96	126.21	6.67
-	-	1.0	115.073	0.570	0.584	0.000	0.655	1.122	3.22	101.65	9.55
5	1	0.2	62.755	0.227	0.750	0.000	1.613	2.680	2.50	**1000**	14.89
-	-	0.4	76.537	0.366	0.708	0.000	0.784	1.303	2.90	296.68	6.32
-	-	0.6	87.039	0.500	0.642	0.000	0.221	0.511	2.55	37.55	4.84
-	-	0.8	96.899	0.517	0.621	0.000	0.071	0.517	2.58	10.89	4.78
-	-	1.0	106.030	0.555	0.587	0.000	0.277	0.705	2.94	41.38	6.41
-	3	0.2	69.789	0.173	0.724	0.000	1.550	2.016	2.69	**1000**	3.99
-	-	0.4	84.698	0.312	0.703	0.000	0.702	1.975	2.53	**1000**	5.56
-	-	0.6	96.246	0.481	0.630	0.000	0.062	1.009	2.93	35.87	5.31
-	-	0.8	106.112	0.516	0.613	0.000	0.098	0.812	2.68	19.00	6.18
-	-	1.0	115.073	0.584	0.570	0.000	0.655	1.122	2.73	66.69	10.55
-	5	0.2	74.924	0.144	0.703	0.000	0.814	2.156	2.22	480.08	4.65
-	-	0.4	90.553	0.284	0.694	0.000	0.550	2.291	2.38	285.22	6.61
-	-	0.6	102.748	0.446	0.621	0.000	0.169	1.372	2.40	26.05	4.71
-	-	0.8	113.464	0.490	0.608	0.000	0.292	1.385	2.96	33.92	5.19
-	-	1.0	123.004	0.564	0.564	0.000	0.598	1.729	2.63	79.46	17.09

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

satisfy demands of both products. However, if the inventory level of component γ (or ϕ) is much greater than that of component ϕ (or γ), it may be optimal to reject demand of product C (or B) which uses a greater number of component ϕ (or γ). SBSR may not induce this kind of structure, but LBLR does.

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

We now vary demand arrival rates, as shown in Table 3.6. Our conclusions from the nested structure remain valid: As one product grows more dominant, it becomes less critical to ration inventory, and the gap between LBLR and SBSR decreases. Likewise, when capacity becomes scarce or high relative to demand, it is not critical to ration inventory in a sophisticated manner, and therefore the gap shrinks. Also, notice that the gap between LBLR and SBSR is significant even when λ_B is 2.5 and λ_C is 0.5, due to the lower lost sale cost of product B.

Based on the previous findings, we again predict that the gap between LBLR and SBSR increases as the product fill rates or difference of fill rates increase. To test this prediction, we use the data in Tables 3.5 and 3.6, and develop a regression model with two independent variables: (i) f_B and (ii) $f_C - f_B$. In contrast to the nested case, we excluded c_B/c_C from the regression model due to its nonmonotonic relationship with our dependent variable, the percentage gap between LBLR and SBSR. All the variables have the predicted sign and are statistically significant (see lower right panel of Table 3.4).

LBLR vs. FBFR. FBFR performs, on average, better than in the nested structure. Our explanation is that as component usage rates of both products are closer to component batch sizes in the current case, it is easier to match supply with demand, and thus coordination of inventory decisions is less crucial. Furthermore, no matter which product is more valuable or dominant, the degree of difficulty of inventory coordination remains the same. Hence, the

Table 3.6 Numerical results for non-nested structure.

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_B	λ_C	Average cost	f_B	f_C	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	32.743	0.506	0.794	0.000	1.426	2.147	2.18	187.57	4.63
-	1.0	53.926	0.259	0.758	0.000	1.643	1.899	2.39	**1000**	7.04
-	1.5	84.862	0.026	0.626	0.000	0.015	0.515	3.09	13.85	3.04
-	2.0	126.450	0.000	0.488	0.000	0.000	0.275	2.59	6.53	2.19
-	2.5	172.774	0.000	0.395	0.000	0.000	0.149	2.59	4.05	2.16
1.0	0.5	39.922	0.456	0.803	0.000	1.265	2.169	2.32	**1000**	11.77
-	1.0	62.755	0.227	0.750	0.000	1.613	2.680	2.45	**1000**	14.51
-	1.5	94.745	0.034	0.622	0.000	0.137	0.585	3.03	440.08	2.56
-	2.0	136.450	0.000	0.488	0.000	0.000	0.255	2.41	4.89	2.13
-	2.5	182.774	0.000	0.395	0.000	0.000	0.141	2.44	10.04	2.30
1.5	0.5	47.885	0.380	0.794	0.000	1.134	2.383	2.95	**1000**	14.59
-	1.0	72.092	0.176	0.745	0.000	1.364	2.286	2.97	**1000**	13.09
-	1.5	104.645	0.029	0.621	0.000	0.220	0.626	3.63	696.38	4.42
-	2.0	146.450	0.000	0.488	0.000	0.000	0.238	2.77	3.63	1.95
-	2.5	192.774	0.000	0.395	0.000	0.000	0.134	2.67	3.09	2.32
2.0	0.5	56.723	0.310	0.778	0.000	0.883	1.849	2.80	532.94	4.44
-	1.0	81.721	0.132	0.751	0.000	1.224	1.947	2.75	**1000**	9.88
-	1.5	114.577	0.024	0.620	0.000	0.260	0.630	2.72	316.97	3.69
-	2.0	156.450	0.000	0.488	0.000	0.000	0.222	2.38	3.30	2.16
-	2.5	202.774	0.000	0.395	0.000	0.000	0.127	2.71	5.98	2.13
2.5	0.5	66.026	0.261	0.773	0.000	0.729	1.558	2.59	103.17	4.48
-	1.0	91.469	0.109	0.748	0.000	1.092	1.707	2.78	**1000**	7.52
-	1.5	124.528	0.021	0.620	0.000	0.279	0.619	2.88	192.57	3.65
-	2.0	166.450	0.000	0.488	0.000	0.000	0.209	2.58	11.82	2.20
-	2.5	212.774	0.000	0.395	0.000	0.000	0.121	2.80	7.67	2.35

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

performances of SBSR and FBFR are closer, although SBSR again significantly outperforms FBFR in many instances.

3.3.3 Computational Efforts

In Table 3.7, we report four statistics of computation times for the numerical instances introduced in Sections 3.3.1 and 3.3.2: the average, standard deviation, minimum, and maximum computation time within each heuristic class and each example. We omit the statistics of computation times for the global optimal solution, as it is instantaneous. We observe from Table 3.7 that LBLR outperforms the other heuristics in terms of average computation times. Although one would intuitively expect the computation times for SBSR and LBLR to be similar since the feasible solution sets for the MIP problems of SBSR and LBLR are comparable, we find that the average computation times are significantly shorter for LBLR (by up to two

Table 3.7 Computation times (in seconds).

3.1(a). Products A and D, $q_\phi = 1$, and $q_\gamma = 3$	LBLR	SBSR	FBFR	3.1(b). Products A and B, $q_\phi = 1$, and $q_\gamma = 2$	LBLR	SBSR	FBFR
Average	2.42	239.67	6.32	Average	2.46	368.30	6.03
Std. Deviation	0.27	361.99	3.52	Std. Deviation	0.25	436.79	2.22
Minimum	1.93	2.00	1.79	Minimum	1.76	2.21	2.27
Maximum	3.19	1000.00	21.45	Maximum	3.11	1000.00	11.02
3.1(c). Products A and B, $q_\phi = 1$, and $q_\gamma = 1$	**LBLR**	**SBSR**	**FBFR**	**3.2. Products B and C, $q_\phi = 2$, and $q_\gamma = 2$**	**LBLR**	**SBSR**	**FBFR**
Average	2.19	89.12	6.28	Average	2.73	359.56	5.96
Std. Deviation	0.33	194.76	3.40	Std. Deviation	0.28	415.98	3.31
Minimum	1.57	1.56	1.93	Minimum	2.18	3.09	1.95
Maximum	3.10	1000.00	17.45	Maximum	3.63	1000.00	17.09

orders of magnitude). Such a computational advantage of LBLR over SBSR arises because a lattice-dependent structure closely fits the globally optimal structure, and our rule of thumb discussed in Section 3.2.2 enables the MIP solver to start from good initial solutions. In addition, the range of LBLR computation times is lower within each example, implying that the computation time for LBLR is more robust to parameter change in our instances.

3.3.4 Selected Larger Instances

We next generate several instances with more components and/or products to determine the maximum problem size that can be solved within a reasonable time for each heuristic. To construct such instances we use the following resource-consumption matrix and parameter values:

Components / Products	A	B	C	D	E	F	G	H	I	J	K	L	q_i	h_i	μ_i
ϕ	1	1	2	1	2	3	2	3	1	1	2	1	2	1	1
γ	1	2	1	3	2	1	3	2	1	2	1	3	2	1	1
η	1	1	2	2	1	1	2	2					2	1	1
θ	2	2											2	1	1
c_j	30	50	40	70	60	50	80	70	25	45	35	65			
λ_j	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25			

Table 3.8 exhibits our numerical results; the components and products that we select to construct our instances are shown in the first two columns. We restrict the computation time

Table 3.8 Numerical results for selected larger instances.

		Optimal solution		Heuristic solutions			Computation times		
Components	Products	Avg. cost	Time	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
ϕ, γ	*A*, *B*	8.086	0.29	8.086	8.086	8.123	12.41	3.27	2.28
-	*A*, *B*, *C*	11.470	0.34	11.470	11.470	11.504	18.57	6.65	5.48
-	*A-D*	18.093	0.41	18.093	18.093	18.218	23.20	241.88	14.75
-	*A-E*	26.037	0.57	26.037	26.076	26.243	36.70	**18000**	35.93
-	*A-F*	34.692	0.75	34.692	34.730	34.768	42.14	**18000**	148.10
-	*A-G*	51.078	1.56	51.078	51.219	51.265	58.46	**18000**	1006.77
-	*A-H*	66.784	3.33	66.784	*	70.807	87.73	5666.93	**18000**
-	*A-I*	72.779	5.57	72.779	110.519	*	142.86	**18000**	202.08
-	*A-J*	83.284	13.68	83.284	*	*	274.26	291.24	180.00
-	*A-K*	91.686	39.86	91.686	*	*	611.96	268.64	307.00
-	*A-L*	107.106	140.42	*	*	*	973.96	454.00	635.00
ϕ, γ, η	*A*, *B*	10.732	5.04	10.732	20.000	10.796	6499.90	**18000**	871.46
-	*A*, *B*, *C*	15.912	8.49	15.912	30.000	*	7862.30	**18000**	714.04
-	*A-D*	24.555	14.75	24.555	47.500	*	9234.14	**18000**	209.00
-	*A-E*	32.782	28.21	32.782	62.500	*	12984.66	**18000**	216.00
-	*A-F*	41.565	51.84	41.565	75.000	*	15796.73	**18000**	252.00
-	*A-G*	58.123	180.53	58.123	*	*	**18000**	398.89	283.00
-	*A-H*	73.924	242.60	**	*	*	**18000**	430.00	320.00
ϕ, γ, η, θ	*A*, *B*	14.591	618.03	**	**	**	**18000**	**18000**	**18000**

Notes. A single star (*) indicates that the MIP solver fails to report a feasible solution as it runs out of memory. Two stars (**) indicate that the MIP solver fails to report a feasible solution within 5 hours. Computation times equal to 18000 seconds indicate termination of the algorithm.

of each instance to be no more than 5 hours (i.e., 18000 seconds).

Table 3.8 indicates that *LBLR again yields the globally optimal cost in all instances.* (For instances where the LBLR solution is unavailable, the global optimal solution adheres to the structure of LBLR.) We also observe that computation times for each of our heuristics considerably increase with the number of components and/or products. Relatively speaking, an increment in the number of components increases computation times more than an increase in the number of products, since both the state space and action space rapidly grow with the number of components. For LBLR, we could solve instances with two components and eleven products, or three components and six products, within 5 hours. For SBSR, we could solve an instance with two components and four products within 5 hours. For FBFR, we could solve instances with two components and seven products, or three components and two products, within 5 hours. Consequently, given the time limit, we are able to solve larger instances for LBLR. (We also observe in the numerical model that the majority of the computation time for LBLR tends to come from generating the MIP model rather than finding its solution.)

3.3.5 Summary of Insights

Our numerical study shows that LBLR outperforms SBSR substantially if it is crucial to hold a significant amount of inventory that must be rationed. This is the case, for example, when (i) demands for both products should be satisfied in significant quantities, (ii) products are highly differentiated mainly through lost sale costs, (iii) both holding cost rates are small, (iv) capacity and demand are comparable, implying inventory should be rationed, or (v) demand rates of the products are similar. Our regression results indicate that conditions (i) and (ii) are statistically significant. (Due to an insufficient number of data points, we are unable to show the significance of other conditions using a regression model.) Another important insight is that in the nested structure, LBLR performs better than SBSR if there is an increase in the holding cost of the component used by the different products in different quantities. In the non-nested structure, if there is a decrease in the holding cost of the component used more by a less valuable product, the gap between LBLR and SBSR increases.

FBFR performs substantially worse than both LBLR and SBSR when component batch sizes do not match the component requirements of the most valuable or most highly demanded product. This is because both *state-dependent* and *lattice-dependent* structures enable us to coordinate inventory decisions across components, and therefore adjust supply levels for the most important product. LBLR achieves such a coordination more economically than SBSR in terms of both average costs and computation times.

3.4 The Case of Discounted Cost

In this section we first reformulate our general model under the total expected discounted cost criterion. We then present several counter examples showing that the properties of *submodularity* and *supermodularity*, which are used to ensure the optimality of LBLR in Chapter 2, may fail to hold for general ATO systems under the discounted cost criterion. We execute this reformulation, as we used these properties in Chapter 2 to prove the optimality of LBLR for the discounted cost case, and then extended this optimality result to the average cost case.

3.4.1 Problem Formulation

Define $0 < \alpha < 1$ as the discount rate. For a given policy ℓ and a starting state $\mathbf{x} \in \mathbb{N}_0^m$, the expected discounted cost over an infinite planning horizon $v^\ell(\mathbf{x})$ can be written as

$$v^\ell(\mathbf{x}) = \mathbb{E}_{\mathbf{x}}^\ell \left[\sum_{i=1}^m \int_0^\infty e^{-\alpha t} h_i(X_i(t))dt + \sum_{j=1}^n \int_0^\infty e^{-\alpha t} c_j dN_j(t) \right].$$

Letting β denote the upper bound on transition rates for all system states (i.e., $\beta = \sum_i \mu_i + \sum_j \lambda_j$), we below formulate the optimality equation that holds for the optimal cost function $v^* = v^{\ell^*}$:

$$v^*(\mathbf{x}) = \frac{1}{\alpha + \beta} \left\{ h(\mathbf{x}) + \sum_i \mu_i T^{(i)} v^*(\mathbf{x}) + \sum_j \lambda_j T_j v^*(\mathbf{x}) \right\}, \tag{3.4.1}$$

where the operator $T^{(i)}$ for component i is defined as

$$T^{(i)} v(\mathbf{x}) = \min\{v(\mathbf{x} + q_i e_i), v(\mathbf{x})\}, \tag{3.4.2}$$

and the operator T_j for product j is defined as

$$T_j v(\mathbf{x}) = \begin{cases} \min\{v(\mathbf{x}) + c_j, v(\mathbf{x} - \mathbf{a}_j)\} & \text{if } \mathbf{x} \geq \mathbf{a}_j, \\ v(\mathbf{x}) + c_j & \text{otherwise,} \end{cases} \tag{3.4.3}$$

where e_i is the ith unit vector of dimension m. For a given state $\mathbf{x}$, the operator $T^{(i)}$ specifies whether or not to produce a batch of component i; and the operator T_j specifies, upon arrival of a demand for product j, whether or not to fulfill it from inventory if it is feasible.

3.4.2 Counter Examples

Define f as the class of real-valued functions on $\mathbb{N}_0^m$, and let $\delta_{\mathbf{p}} f = f(\mathbf{x}+\mathbf{p}) - f(\mathbf{x})$ for $\mathbf{p} \in \mathbb{N}_0^m$. In Chapter 2, we introduced the notion of "submodularity with step size $\mathbf{p}$" for $\mathbf{p} \in \mathbb{N}_0^m$ to describe the class of functions f with $\delta_{p_i e_i} f$ nonincreasing with an increase of p_k in the kth dimension, $\forall i \neq k$. Denote this class of functions by $Sub(\mathbf{p})$. In Chapter 2, we also introduced the notion of "supermodularity with step sizes $\mathbf{r}$ and $\mathbf{p}$" for $\mathbf{r}, \mathbf{p} \in \mathbb{N}_0^m$ to describe the class of functions f with $\delta_{p_i e_i} f$ nondecreasing with an increase of $\mathbf{r}$ in the domain, $\forall i$. Denote this class of functions by $Super(\mathbf{r}, \mathbf{p})$. Thus:

Definition 3.4.1. *Let f be a real-valued function defined on $\mathbb{N}_0^m$. Also let $\mathbf{r}, \mathbf{p} \in \mathbb{N}_0^m$.*

(a) $f \in Sub(\mathbf{p})$, *if* $f(\mathbf{x} + p_i e_i) - f(\mathbf{x}) \geq f(\mathbf{x} + p_i e_i + p_k e_k) - f(\mathbf{x} + p_k e_k)$, $\forall \mathbf{x} \in \mathbb{N}_0^m$, $\forall i$ *and* $\forall k \neq i$.

(b) $f \in Super(\mathbf{r}, \mathbf{p})$, *if* $f(\mathbf{x} + p_i e_i + \mathbf{r}) - f(\mathbf{x} + \mathbf{r}) \geq f(\mathbf{x} + p_i e_i) - f(\mathbf{x})$, $\forall \mathbf{x} \in \mathbb{N}_0^m$ *and* $\forall i$.

In Chapter 2 we proved that, for the M-system product structure, the optimal cost function satisfies the properties of both $Sub(\mathbf{b})$ and $Super(\mathbf{a}, \mathbf{b})$, where $\mathbf{a}$ is the vector of component requirements for the master product, and $\mathbf{b}$ is the vector of component requirements for individual products. We then used this result to establish the optimality of LBLR. As the model presented in this chapter allows for general product structures, it is not apparent whether the results in Chapter 2 will extend to our setting. To answer this question we construct counter examples that do not satisfy the M-system product structure for which the above functional characterizations *fail* to hold. Thus showing optimality of LBLR policies, if this indeed holds, will likely require an alternate proof technique.

We restrict our attention to a 1-component 3-product system, which violates the M-system product structure since more than two products consume the same component. We select products A, B, and D, and component γ, to construct our system (recall products A, B, and D use 1, 2, and 3 units of component γ, respectively). We then generate several counter examples by varying values of the related parameters for this system, and using the value iteration method to determine the optimal cost function evaluated at different initial inventory levels (see Chapter 1 in Bertsekas 2007 for an explanation of the value iteration method).

As the concept of submodularity is inapplicable to single-component systems, we only check whether the optimal cost function satisfies the property of $Super(r, p)$, at various step sizes $r \in \mathbb{N}_0$ and $p \in \mathbb{N}_0$. Note that $f \in Super(r, p)$ if $f(x + p + r) - f(x + r) \geq f(x + p) - f(x)$, $\forall x \in \mathbb{N}_0$. We consider various step sizes as we want to generalize the following structural result, which is valid in a 1-component 2-product system (see Chapter 2): The property of

Table 3.9 Counter examples for supermodularity with various step sizes.

Parameters for counter examples									Results for various step sizes					
α	q_γ	μ_γ	c_A	c_B	c_D	λ_A	λ_B	λ_D	(1,2)	(1,3)	(1,5)	(2,3)	(2,4)	(3,3)
0.3	2	0.15	10	80	120	0.025	0.80	0.025	N	N	N	Y	Y	N
0.4	2	0.15	10	60	120	0.050	0.35	0.450	N	N	N	Y	N	Y
0.3	3	0.05	1	2	40	0.150	0.72	0.080	N	N	N	N	Y	Y

Notes. $h_\gamma = 1$ in all examples. The last six columns show whether $Super(r, p)$ holds depending on step sizes r and p: "Y" means that $Super(r, p)$ holds for the corresponding example, and "N" means that $Super(r, p)$ fails to hold.

$Super(r, p)$ holds when step sizes r and p are chosen to be component requirements of the two products. (But if r and p are chosen differently, $Super(r, p)$ may fail to hold, even for M-systems. Consider a 2-product system with products A and B, and component γ. Suppose that $q_\gamma = 1$, $\mu_\gamma = 1$, $\lambda_A = 1$, $\lambda_B = 10$, $c_A = 10$, $c_B = 100$, $h_\gamma = 60$, and $\alpha = 0.5$: $Super(1, 1)$ fails to hold in this example, while $Super(1, 2)$ holds.)

The intuition behind this positive result in Chapter 2 is as follows: Demands of one product compete with those of the other product for the same component, and therefore the incentive to satisfy a demand for one product increases as the competition becomes less severe due to a supply increase equal to the component requirement of the competitor product. Following the same intuition, this structural result might be foreseeably extended to our 3-product system in different ways: (i) r and p may be the numbers of components used by different products (e.g., $r = 1$ and $p = 2$, in our 3-product system), or (ii) r may be the number of components used by one product while p is the sum of the numbers of components used by the other products (e.g., $r = 1$ and $p = 5$). Hence, for our 1-component 3-product system, there are six possible pairs of r and p.

We report our results for three counter examples in Table 3.9. The left panel of Table 3.9 lists the parameters for our counter examples, while the right panel shows whether the property of $Super(r, p)$ holds for each r and p. The existence of counter examples for *each* pair of step sizes proves that the property of $Super(r, p)$ need *not* hold for 1-component 3-product systems, and thus for general ATO systems. Hence, one may need to develop a different methodology from that introduced in Chapter 2 to prove the optimality of LBLR for general ATO systems.

3.5 Conclusions

We have studied the lattice-dependent base-stock and lattice-dependent rationing (LBLR) policy introduced in Chapter 2 as a heuristic for general ATO systems. In the average cost case, we compare it to two other heuristics from the literature: state-dependent base-stock and state-dependent rationing (SBSR), and fixed base-stock and fixed rationing (FBFR). We numerically show that an LBLR policy minimizes the average costs in each of the more than 1800 instances of general ATO problems we tested. Our numerical experiments also demonstrate that LBLR performs significantly better than SBSR (by up to 2.7% of the optimal cost) when products are highly differentiated and it is optimal to fulfill a significant fraction of the demand for each product. FBFR performs worse than the other two heuristics (by up to 4.8% of the optimal cost), since it lacks the coordination of inventory decisions across different components. We also analytically show that LBLR performs no worse than the other heuristics.

Based on our numerical results, future research could investigate the optimality of LBLR for ATO systems with general product structures. However, our counter examples for the discounted cost case show that the functional characterizations that are sufficient to ensure the optimality of LBLR need *not* hold for general product structures. Thus, if LBLR is indeed optimal, it may be necessary to develop a new method to prove this. Another direction for future research is to study the performance of LBLR in ATO systems with backordering and/or general component production and demand interarrival times. Lastly, assuming LBLR, future research could develop solution procedures for the optimization of base-stock and rationing levels in high-dimensional ATO problems for which even solving the linear program formulation to optimality might prove problematic.

Chapter 4

Optimal Portfolio Strategies for New Product Development

4.1 Introduction

In today's competitive environment, many large-scale manufacturers, facing rapid technology innovations and changing customer preferences, devote significant efforts to continually develop new products and launch them successfully into the market. These firms generally pursue many new product development (NPD) projects in parallel to achieve broader product lines and higher market share (Ulrich and Eppinger 2004). Concurrent projects often place competing demands on scarce resources (e.g., testing equipment or specialists with unique areas of expertise), adding complexity to the NPD portfolio management (Kavadias and Chao 2008). If not managed properly, scarce resources may lead to significant delays in project completion times reducing the firm's profitability (Adler et al. 1995).

Managing scarce resources to optimize their use becomes even more difficult in the face of uncertainty regarding project outcomes. Aiming to reduce uncertainty and adjust development efforts accordingly, project managers generally divide the NPD process into a series of distinct experimental stages and review the evolution of projects at each stage (Cooper 2008). Hence, managers sequentially gather additional information and use it to update their prior beliefs about project outcomes (Artmann 2009). This approach therefore allows managers to focus and redeploy available resources on high value projects as they are able to eliminate projects

This chapter presents joint work undertaken with Mustafa Akan, Laurens Debo, and Alan Scheller-Wolf.

with little promise before testing at expensive downstream stage (Thomke 2008). (An example of a sequential NPD process is a new drug development process, which typically involves three stages: (i) safety trials, (ii) efficacy trials, and (iii) large-scale trials; see DiMasi et al. 2003.)

The above treatment of scarce resources potentially improves the firm's profitability. However, information generated through experimental outcomes is imperfect and experimentation times are typically variable (Sommer et al. 2008). Furthermore, NPD projects at different stages may demand the use of different specialized resources, requiring project managers to make concurrent resource allocation decisions across stages at a time. To our knowledge, the NPD literature has *not* yet developed a comprehensive modeling framework that explicitly captures these aspects of the problem. In this chapter we take the first step towards filling this gap, and study the problem of *project selection* and *resource allocation* in a multi-stage NPD process with imperfect information across stages, exponential experimentation times, and stage-dependent resource constraints.

We model the problem as an infinite-horizon Markov decision process (MDP) under the total expected discounted cost criterion. Each NPD project undergoes a different experiment at each stage of the NPD process; experiments generate signals about the true nature of a project. Beliefs about the true nature of a project are updated after each experiment according to a Bayesian rule. Projects thus become differentiated through their signals, and all available signals for a project determine its *category*. The state of the system is described by the numbers of projects in each *category*. Given the system state, a control policy specifies what fraction of resources should be allocated to each project at each stage and which projects should be terminated. At each stage, experimentation rate for a project is proportional to the fraction of resources allocated to that project. The existence of binding stage-dependent resource constraints implies the total experimentation rate at each stage is bounded above. Each experiment incurs a variable cost upon completion, which is concave and weakly increasing in the utilized fraction of resources. But the returns of a project, which are determined by posterior beliefs at the last stage, are earned only after the project is complete.

We characterize the optimal control policy as following a *new* type of policy, *state-dependent non-congestive promotion* (SDNCP), for two different special cases of the problem: (a) when there is a single informative experiment, or (b) when there are multiple uninformative experiments. An SDNCP policy implies that, at each stage, it is optimal to advance a project with highest expected reward to the next stage if and only if the number of projects in each successor category is less than a congestion-dependent threshold. Specifically, threshold values decrease in a non-strict sense as a later stage becomes more congested or as an earlier stage becomes less congested. A stage becomes more congested with an increase in the number of projects at this stage or with an increase in the expected reward of any project at this stage. These findings further our knowledge of the NPD problem, revealing the impact of congestion on the optimal policy.

We also conduct numerical experiments to evaluate the use of an SDNCP policy as a heuristic for the general NPD problem, comparing it to a fixed non-congestive promotion policy with fixed thresholds (FNCP-Ch), and several other heuristics. Taking the average cost rate as our performance criterion, we formulate a linear program to find the globally optimal cost, and mixed integer programs to find the optimal cost within each heuristic class. We then generate 80 instances of the general NPD problem: Remarkably, SDNCP minimizes the average costs in over half of these instances. (The average distances from optimal cost are 0.05% for SDNCP, and 8.23% for FNCP-Ch.) Our numerical results also indicate that SDNCP has a greater benefit over FNCP-Ch (i) when the NPD process slows down at downstream stages, (ii) when project holding and/or experimentation costs are higher, or (iii) when experiments are less informative.

The rest of this chapter is organized as follows: In Section 4.2 we offer a brief literature review. In Section 4.3 we formulate our general model under the discounted cost criterion. In Section 4.4 we establish the optimal control policy when there is a single informative experiment. In Section 4.5 we establish the optimal control policy when there are multiple uninformative experiments. In Section 4.6 we present our heuristic policies for the general model along with numerical results. In Section 4.7 we offer a summary and conclusions.

4.2 Literature Review

Research in this area comprises two major streams: (i) Resource allocation, and (ii) dynamic scheduling. Kavadias and Loch (2004) provide a comprehensive literature review, which we summarize below.

Resource Allocation. Several authors in this stream studied project selection models with binary decision variables, additive present values of projects, and budget limitations over time (e.g., Lorie and Savage 1955). These models have also been extended to dependent present values and continuous decision variables. There is also a significant body of literature around the dynamic and stochastic knapsack problem, in which each request (project) arrives in time as a stochastic process and has a demand for a limited resource (e.g., Papastavrou et al. 1996, and Kleywegt and Papastavrou 2001). The demands and their rewards are random, and become known upon arrival. If a demand is accepted, the reward is received, otherwise a penalty is paid. The objective is to maximize the expected reward in a given timeline. Another group of studies consider the resource-constrained scheduling problem (e.g., Brucker et al. 1999, and Neumann et al. 2002). The objective is to optimally schedule project activities subject to due dates, precedence relations, and resource constraints. This research stream neglects to consider intermediate project reviews.

Dynamic Scheduling. This stream of research can further be divided into three groups. The first group considers the optimal sequential selection problem in which an NPD project passes through several distinct stages and its status is inspected at each stage (e.g., Roberts and Weitzman 1981). A project at any stage is either terminated or promoted to the next stage based on the information available. However, these models are *not* subject to resource constraints. The second group studies the multi-armed bandit problem in which projects compete for access to a specialized resource which can be utilized by only one project at any point in time. A project utilizing the resource undergoes Markovian transitions and returns an immediate state- and time-dependent reward. Gittins and Jones (1972) introduced the Gittins Index, a number that can be assigned to each project at any time; it is always optimal to work on the project with the highest index. These problems have also been generalized by allowing the state of passive projects to evolve over time and influence the reward of the active project. The last group approaches the project prioritization problem as a multiclass queueing model with stochastic completion times for each job class. The $c\mu$ rule, giving priority to the job with the highest delay cost divided by the expected processing time, is proven optimal for various settings (e.g., Wein 1992, and Ha 1997). Unlike the last two groups, we allow projects at different stages to require different resources.

4.3 Problem Formulation

We consider the problem of project selection and resource allocation in a continuous-time NPD process (e.g., a new drug development process, cf. Figure 4.1). Each NPD project passes through a finite number of experimental stages (e.g., safety, efficacy, and general tests) before the resulting product (e.g., "Lipitor") is placed on the market. Define $\mathcal{M} = \{1, 2, .., m\}$ as the set of experimental stages, and i as the index for the stage (e.g., $m = 3$). The true ultimate *nature* of a project falls into one of a number of states (e.g., "success" or "failure"), and initial expectations about the *nature* are the same across all projects. Each experiment generates a piece of new information (e.g., "good" or "bad" signal) about the *nature* of the project; uncertainty pertaining to the ultimate outcome of the project is further resolved at each stage. Define K as the number of possible signals that can be generated at each stage for each project, and k as the index for the signal (e.g., $K = 2$). There exists a one-to-one correspondence between the set of signals at each stage and the set of states for the true *nature*. Both sets consist of integers from 1 to K such that a lower integer indicates a project with higher expected return (e.g., $k = 1$ means a "good" signal and a "success", and $k = 2$ means a "bad" signal and a "failure").

All available signals for a project determine its *category*; projects become differentiated through their *categories*. Define $\mathcal{N} = \{0, 1, .., n\}$ as the set of project categories, and j as the index for the category. Note that K^i is the number of categories at stage i, and $n = K + K^2 + ... + K^m$ (e.g., $n = 14$). Different stages of the NPD process require different resources, which are limited. Define $\mathcal{W}_i$ as the set of project categories waiting for access to resources of stage i for experimentation (e.g., $\mathcal{W}_1 = \{0\}$, $\mathcal{W}_2 = \{1, 2\}$, and $\mathcal{W}_3 = \{3, 4, 5, 6\}$). Also, define $\mathcal{W}_{m+1}$ as the set of project categories that have completed all stages except the product launch stage $m + 1$ (e.g., $\mathcal{W}_4 = \{7, 8, .., 14\}$).

Experiments imperfectly reveal the true nature of a project throughout the NPD process. Define $\mathbf{\Phi}^i$ as an $K \times K$ informativeness probability matrix at stage i; $\phi^{(i)}_{k,k'}$ is the probability that the experiment at stage i generates signal k' for projects with true nature k. Notice that if

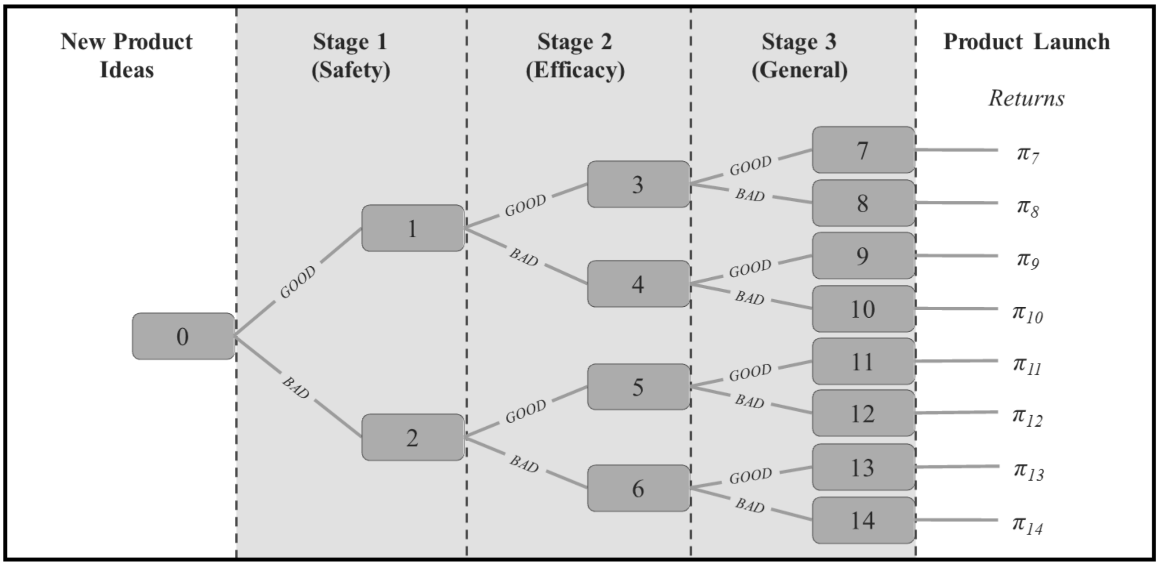

Figure 4.1 A new drug development process. Each small box represents a different project category, and each line at a given stage represents a different signal.

$\phi^{(i)}_{k,k} = 1, \forall k$, then the experiment at stage i perfectly reveals the true nature. Thus we assume $\phi^{(i)}_{k,k} < 1, \forall i, k$. Beliefs about the true nature of a project undergo Bayesian updating after each experiment. Define $\mathbf{p}_j = (p_{j,1}, .., p_{j,K})$ as the probability distribution for the true nature of a project in category $j \in \mathcal{W}_i$. Suppose that a project in category $j \in \mathcal{W}_i$ becomes category $j' \in \mathcal{W}_{i+1}$, returning signal k' at stage i. Then the posterior probability mass function for the true nature of the project is calculated as

$$p_{j',k} = \frac{p_{j,k} \times \phi^{(i)}_{k,k'}}{f_{j \to j'}}, \ \forall k,$$

where $f_{j \to j'}$ is the probability that a project in category $j \in \mathcal{W}_i$ falls into category $j' \in \mathcal{W}_{i+1}$, returning signal k' at stage i, i.e.,

$$f_{j \to j'} = \sum_{1 \le k \le K} p_{j,k} \times \phi^{(i)}_{k,k'}.$$

The expected reward for a project in category j is calculated by $\rho_j = \mathrm{E}[\mathbf{r} \cdot \mathbf{p}_j]$, where $\mathbf{r}$ is a K dimensional nonnegative vector whose elements are in descending order; r_k denotes the reward for a project with ultimate outcome k. Notice that $\rho_j \geq 0, \forall j$. The returns of a project are earned only after the project is complete.

Both experimentation and product launch times are independent and exponentially distributed. Define μ_i as the maximum possible experimentation rate at stage i, and μ_{m+1} as the maximum possible product launch rate. Managerial control of resource allocation at each stage affects experimentation rates; $\mu_i y_j$ is the experimentation rate at stage i for a project in category $j \in \mathcal{W}_i$, which is proportional to the fraction of resources utilized by that project, $y_j \in [0, 1]$. The same is also true for the product launch rate. Once a project in category j completes the experiment at stage i, the system incurs a variable cost, $c_{ij}(y_j)$. But there are no costs associated with interrupted experiments. (This assumption is not restrictive in two special cases of the model introduced in Sections 4.4 and 4.5, as it is never optimal to interrupt any experiment once it has been initiated in those cases.) We assume $c_{ij}(y_j)$ is concave and weakly increasing in y_j, and equals zero when $y_j = 0$, $\forall i$. Also, experimentation costs at a given stage i and rate y are the same across different categories; $c_i(y) = c_{ij}(y)$, $\forall j \in \mathcal{W}_i$. Projects may be terminated at no cost; termination time for any project in the NPD process is exponentially distributed with a fixed rate λ, which can be chosen arbitrarily large.

The state of the system at time t is the vector $\mathbf{X}(t) = (X_1(t), .., X_n(t))$, where $X_j(t)$ is a nonnegative integer denoting the number of projects in category j at time t. Projects held in the NPD process incur an aggregate holding cost per unit time which is convex and strictly increasing in the total number of projects. Denote by $h(\mathbf{X}(t)) = h'(\sum_{j>0} X_j(t))$ the holding cost rate at state $\mathbf{X}(t)$. Since all inter-event times are exponentially distributed, the system retains no memory, and decision epochs can be restricted to times when the state changes. Using the memoryless property, we can formulate the problem as an MDP and focus on Markovian policies for which actions at each decision epoch depend solely on the current state. A control policy u specifies for each state $\mathbf{x} = (x_1, .., x_n)$, the action $\mathbf{a}^u(\mathbf{x}) = (y_0, .., y_n, z)$, $y_j \in [0, 1]$, $\forall j$, and $z \in \{0, 1, .., n\}$, where y_j denotes the fraction of resources of stage i (or stage $m + 1$) allocated to a project in category $j \in \mathcal{W}_i$ (or $j \in \mathcal{W}_{m+1}$), and z denotes the category from which a project is terminated ($z = 0$ means do not terminate any project). The action $a^u(\mathbf{x}) = (y_0, .., y_n, z)$ must satisfy the following conditions: (a) $y_j = 0$ and $z \neq j$ if $x_j = 0$, $\forall j > 0$; (b) $\sum_{j \in \mathcal{W}_i} y_j \leq 1$, $\forall i$; and (c) $\sum_{j \in \mathcal{W}_{m+1}} y_j \leq 1$.

Define $0 < \alpha < 1$ as the discount rate. Also, define β as the upper bound on transition rates for all system states (i.e., $\beta = \lambda + \sum_i \mu_i + \mu_{m+1}$). We below formulate the optimality equation that holds for the optimal cost function $v^* = v^{u^*}$:

$$v^*(\mathbf{x}) = \frac{1}{\alpha + \beta} \left(h(\mathbf{x}) + \lambda T_A v^*(\mathbf{x}) + \sum_{1 \le i \le m} \mu_i T_{B,i} v^*(\mathbf{x}) + \mu_{m+1} T_C v^*(\mathbf{x}) \right), \tag{4.3.1}$$

where the operator T_A is defined as

$$T_A v(\mathbf{x}) = \min_{\substack{0 \le j \le n \\ s.t.\ \mathbf{x} \ge e_j}} v\left(\mathbf{x} - e_j\right), \tag{4.3.2}$$

the operator $T_{B,i}$ for each stage $i \in \{1, 2, .., m\}$ is given by

$$T_{B,i} v(\mathbf{x}) = \min_{\substack{y_j,\ j \in \mathcal{W}_i \\ s.t.\ \sum_{j \in \mathcal{W}_i} y_j \le 1}} \left[\sum_{j \in \mathcal{W}_i} \left(c_{ij}(y_j) + y_j \sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'} \right) + \left(1 - \sum_{j \in \mathcal{W}_i} y_j \right) v\left(\mathbf{x}\right) \right], \tag{4.3.3}$$

and the operator T_C is defined as

$$T_C v(\mathbf{x}) = \min_{\substack{y_j,\ j \in \mathcal{W}_{m+1} \\ s.t.\ \sum_{j \in \mathcal{W}_{m+1}} y_j \le 1}} \left[\sum_{j \in \mathcal{W}_{m+1}} y_j \left(v\left(\mathbf{x} - e_j\right) - \rho_j \right) + \left(1 - \sum_{j \in \mathcal{W}_{m+1}} y_j \right) v\left(\mathbf{x}\right) \right], \tag{4.3.4}$$

where e_0 is a zero vector of dimension n, and e_j is the jth unit vector of dimension n for $1 \le j \le n$. For a given state $\mathbf{x}$, (a) the operator T_A specifies whether or not to terminate a project, and which project to select if a project is to be terminated; (b) the operator $T_{B,i}$ specifies what fraction of resources of stage i should be allocated to a project in each category $j \in \mathcal{W}_i$ for experimentation; and (c) the operator T_C specifies what fraction of resources of stage $m+1$ should be allocated to a project in each category $j \in \mathcal{W}_{m+1}$ for product launch. However, without loss of generality, the action space of the operators $T_{B,i}$ and T_C can be reduced to the set of binary variables:

Lemma 4.3.1. *There is no loss of generality in assuming that $y_j \in \{0, 1\}$, $\forall j$.*

Proof. See Appendix C.1. □

Following Lemma 4.3.1, we assume $y_j \in \{0,1\}$, $\forall j$. This implies that the operator $T_{B,i}$ (or T_C) specifies when to *fully* utilize the resources of stage i (or stage $m+1$), and on which project. For notational convenience, we replace $c_{ij}(1)$ with c_i, $\forall i$ and $\forall j \in \mathcal{W}_i$ (recall $c_i(y_j) = c_{ij}(y_j)$, $\forall i$ and $\forall j \in \mathcal{W}_i$). Also, without loss of generality, we assume $\alpha + \beta = 1$ as it is always possible to redefine the time scale. Then the optimality equation takes the following form:

$$v^*(\mathbf{x}) = h(\mathbf{x}) + \lambda T_A v^*(\mathbf{x}) + \sum_{1 \le i \le m} \mu_i T_{B,i} v^*(\mathbf{x}) + \mu_{m+1} T_C v^*(\mathbf{x}) \tag{4.3.1$'$}$$

where the operator T_A stays the same as

$$T_A v(\mathbf{x}) = \min_{\substack{0 \le j \le n \\ s.t.\ \mathbf{x} \ge e_j}} v\left(\mathbf{x} - e_j\right), \tag{4.3.2$'$}$$

but the operators $T_{B,i}$ and T_C are modified as

$$T_{B,i} v(\mathbf{x}) = \min \left\{ v\left(\mathbf{x}\right), \min_{\substack{j \in \mathcal{W}_i \\ s.t.\ \mathbf{x} \ge e_j}} \left[\sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'} + c_i \right] \right\} \tag{4.3.3$'$}$$

and

$$T_C v(\mathbf{x}) = \min \left\{ v\left(\mathbf{x}\right), \min_{\substack{j \in \mathcal{W}_{m+1} \\ s.t.\ \mathbf{x} \ge e_j}} \left[v\left(\mathbf{x} - e_j\right) - \rho_j \right] \right\}. \tag{4.3.4$'$}$$

Although we considerably simplified the optimal cost function, its analysis is still difficult; the system manager should handle concurrent resource allocation decisions across stages in the face of uncertainty around the ultimate outcome of each project as well as project completion times. Only for two special cases of our model are we able to derive structural results for the optimal cost function: (i) when there is a single informative experiment (Section 4.4) or (ii) when there are multiple uninformative experiments (Section 4.5).

4.4 Informative Single-Experiment Model

In this section, we assume that the NPD process involves (i) an experimental stage that generates one out of the K signals about the true nature of each project, and (ii) a product launch stage:

Assumption 4.4.1. $m = 1$.

Notice that, under Assumption 4.4.1, $n = K$. We also assume that projects are never terminated:

Assumption 4.4.2. $\lambda = 0$.

Under Assumptions 4.4.1 and 4.4.2, the optimality equation can be written as follows:

$$v^*(\mathbf{x}) = h(\mathbf{x}) + \mu_1 T_B v^*(\mathbf{x}) + \mu_2 T_C v^*(\mathbf{x}), \tag{4.4.1}$$

where the operators T_B and T_C are defined as

$$T_B v(\mathbf{x}) = \min \left\{ v(\mathbf{x}), \sum_{1 \le j \le n} v(\mathbf{x} + e_j) f_{0 \to j} + c \right\} \tag{4.4.2}$$

and

$$T_C v(\mathbf{x}) = \min \left\{ v(\mathbf{x}), \min_{\substack{1 \le j \le n \\ s.t.\ \mathbf{x} \ge e_j}} [v(\mathbf{x} - e_j) - \rho_j] \right\} \tag{4.4.3}$$

where c is the experimentation cost. The operator T_B specifies when to utilize resources of stage 1 for a project in category 0, i.e., when to test a new product idea. The operator T_C specifies when to place a new product on the market, and which project to select when a new product is to be introduced.

We proceed to characterize the structure of the optimal cost function (4.4.1). We first introduce additional auxiliary indices d, l, q, and w for the category; the alphabetical order $d \to l \to q \to w$ corresponds to a decrease in the expected reward. We then define $\widehat{V}$ as the set of real-valued functions g on $\mathbb{N}_0^n$ that satisfy the following properties:

(P.1) $g(\mathbf{x} + e_w) \ge g(\mathbf{x} + e_q)$, $\forall \mathbf{x}$, $\forall q, w \in \{1, 2, .., n\}$ where $q < w$,

(P.2) $g(\mathbf{x}+e_w) \geq g(\mathbf{x}) - \rho_w$, $\forall \mathbf{x}$, $\forall w \in \{1,2,..,n\}$,

(P.3) $g(\mathbf{x}+e_q) - \rho_w \geq g(\mathbf{x}+e_w) - \rho_q$, $\forall \mathbf{x}$, $\forall q,w \in \{1,2,..,n\}$ where $q < w$,

(P.4) $g(\mathbf{x}+e_q+e_l) - g(\mathbf{x}+e_q+e_d) \geq g(\mathbf{x}+e_l) - g(\mathbf{x}+e_d)$, $\forall \mathbf{x}$, $\forall d,l,q \in \{1,2,..,n\}$ where $d \leq l \leq q$,

(P.5) $g(\mathbf{x}+e_q+e_l) - g(\mathbf{x}+e_q+e_d) \geq g(\mathbf{x}+e_w+e_l) - g(\mathbf{x}+e_w+e_d)$, $\forall \mathbf{x}$, $\forall d,l,q,w \in \{1,2,..,n\}$ where $d \leq l \leq q \leq w$,

(P.6) $\sum_{1\leq j\leq l} g(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{l<j\leq n} g(\mathbf{x}+e_q+e_j) f_{0\to j} - g(\mathbf{x}+e_q)$ $\geq \sum_{1\leq j\leq l} g(\mathbf{x}+e_l) f_{0\to j} + \sum_{l<j\leq n} g(\mathbf{x}+e_j) f_{0\to j} - g(\mathbf{x})$, $\forall \mathbf{x}$, $\forall l,q \in \{1,2,..,n\}$ where $l \leq q$, and

(P.7) $\sum_{1\leq j\leq l} g(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{l<j\leq n} g(\mathbf{x}+e_q+e_j) f_{0\to j} - g(\mathbf{x}+e_q)$ $\geq \sum_{1\leq j\leq l} g(\mathbf{x}+e_w+e_l) f_{0\to j} + \sum_{l<j\leq n} g(\mathbf{x}+e_w+e_j) f_{0\to j} - g(\mathbf{x}+e_w)$, $\forall \mathbf{x}$, $\forall l,q,w \in \{1,2,..,n\}$ where $l \leq q < w$.

The lemma below shows that the optimal cost function satisfies Properties 1–7:

Lemma 4.4.1. *Under Assumptions 4.4.1 and 4.4.2, if* $v \in \widehat{V}$*, then* $Tv \in \widehat{V}$*, where* $Tv(\mathbf{x}) = h(\mathbf{x}) + \mu_1 T_B v(\mathbf{x}) + \mu_2 T_C v(\mathbf{x})$*. Furthermore, the optimal cost function* v^* *is an element of* $\widehat{V}$*.*

Proof. See Appendix C.2. □

We next consider the implications of Properties 1–7: Property 1 states that the optimal cost function weakly decreases as the number of projects with higher expected reward increases, keeping the total number of projects fixed. Property 2 shows that placing a new product on the market is beneficial no matter what category is chosen, as long as it is feasible (recall that $\rho_j \geq 0$, $\forall j$). However, Property 3 says that choosing the category with higher expected reward is more desirable for product launch (in a non-strict sense). Property 4 states that the incentive to trade a project in category l with one having higher expected reward weakly increases as the number of projects with expected rewards lower than ρ_l increases. Property 5 states that the incentive to trade a project in category l with one having higher expected reward is greater in a non-strict sense when a project with expected reward less than ρ_l is traded with one having higher expected reward (but lower than ρ_l). Property 6 implies that the desirability of testing a new product idea weakly increases as the number of projects in any category decreases. Property 7 implies that the desirability of testing a new product idea weakly increases as the number of projects with lower expected reward increases, keeping the total number of projects fixed.

The intuition behind Properties 1–3 is straightforward: It is more desirable to have a project with higher expected reward, which will reasonably take priority over those with lower expected rewards in the allocation of resources for product launch. More importantly, Properties 4–7 enable us to uncover the role congestion plays in promotion decisions for new product ideas. As the number of projects in category l increases, the system becomes more congested from the perspective of a new product idea: If a new product idea is tested at the experimental stage, it might become a project with expected reward less than ρ_l, taking a lower priority than all projects in category l in the queue for the second stage. Such a project waits longer for access to resources of the second stage when there are more projects in category l. Consequently, since any delay in the project completion time is costly, it is less desirable to test a new product idea (Property 6). Likewise, when a project is replaced with one having higher expected reward, the system again becomes more congested from the same perspective: A new product idea, once tested, is more likely to see a greater number of high priority projects in the queue for the second stage if a low value project is replaced with a high value project. Thus it is less desirable to test a new product idea (Property 7).

As the system becomes more congested due to an increase in the number of low value projects, the system anticipates a lower throughput rate at the experimental stage in the future (due to Property 6), and eventually a small number of high value projects. To hedge against future scarcity of high value projects, the system tends to trade a project in category l with one having higher expected reward. But such a trade becomes more desirable as the number of projects with expected rewards *less* than ρ_l increases (Property 4). Likewise, as the system becomes more congested due to a rise in the expected reward of a low value project, the system tends to trade a project in category l with one having higher expected reward in anticipation of a small number of high value projects (due to Property 7). It is more desirable to do so when a project with expected reward *less* than ρ_l is traded with one having higher expected reward (but *less* than ρ_l) (Property 5).

Using Lemma 4.4.1, we below show the optimality of a state-dependent noncongestive-promotion (SDNCP) policy under Assumptions 4.4.1 and 4.4.2:

Theorem 4.4.1. *Under Assumptions 4.4.1 and 4.4.2, the optimal portfolio strategy is a state-dependent noncongestive-promotion policy with state-dependent promote-up-to levels* $S_j^*(\mathbf{x}_{-j})$*: It is optimal to allocate resources of the experimental stage to a new product idea if and only if* $x_j < S_j^*(\mathbf{x}_{-j})$, $\forall j$, *where* $\mathbf{x}_{-j} = (x_1, .., x_{j-1}, x_{j+1}, .., x_n)$ *is an* $n-1$ *dimensional vector of the numbers of projects in categories* $k \neq j$*. The optimal policy has the following additional properties:*

i. The optimal promote-up-to level $S_j^*(\mathbf{x}_{-j})$ *weakly decreases as the number of projects in category* $k \neq j$ *increases,* $\forall j$*.*

ii. The optimal promote-up-to level $S_j^*(\mathbf{x}_{-j})$ *weakly decreases as the expected reward of a project in category* $k \neq j$ *increases,* $\forall j$*.*

iii. It is always optimal to launch a new product if there are projects available for the product launch stage.

iv. It is always optimal to allocate resources of the product launch stage to a project with highest expected reward.

v. It is never optimal to interrupt any experiment.

Proof. See Appendix C.2. □

Theorem 4.4.1, using Property 6, establishes the optimality of a *new* type of policy, i.e., SD-NCP. Such a policy protects the system against congestion, restricting the number of projects that can be held in each category. Points (i) and (ii) prove that the promote-up-to levels weakly decrease as the system becomes more congested with an increase in the number of projects in any category (due to Property 6), or with an increase in the expected reward of any project in the system (due to Property 7). Points (iii) and (iv) reveal that it is always optimal to launch a new product if it is feasible (due to Property 2), and it is optimal to choose a project with highest expected reward for product launch (due to Property 3). Lastly, point (v) states it is never optimal to interrupt an experiment once it has been initiated.

To our knowledge, we are the first to characterize the optimal resource allocation and project selection for NPD when projects must pass through an informative experimental stage and a product launch stage, which require different resources. Also, we significantly extend the existing literature by showing that optimal promotion decisions depend on the number of projects as well as the breakdown of projects into categories. There are optimality results in the literature for two variations of the problem: (i) Suppose that experimental and product launch stages share the same resources, implying that the system manager makes only one resource allocation decision at a time. Then it can be shown that an *index rule* is optimal (see Chapter 1 of Bertsekas 2007 for an explanation of the index rule). (ii) Suppose that the experimental stage is uninformative. Then the problem bears a close resemblance to the single-item inventory model introduced by Ha (1997). A non-congestive promotion policy remains optimal in this special case of our model; it is optimal to promote a new product idea if and only if the number of projects in the system is less than a fixed promote-up-to level. This concurs well with the base-stock policy that is optimal in Ha (1997); it is optimal to order an item if and only if the inventory level is less than a fixed base-stock level.

4.5 Uninformative Multi-Experiment Model

In this section, we no longer impose Assumptions 4.4.1 and 4.4.2. But we assume that experimental stages do *not* provide any information about the true nature of any NPD project:

Assumption 4.5.1. $K = 1$.

Under Assumption 4.5.1, $n = m$ and the optimality equation can be written as follows:

$$v^*(\mathbf{x}) = h(\mathbf{x}) + \lambda T_A v^*(\mathbf{x}) + \sum_{1 \le i \le m} \mu_i T_{B,i} v^*(\mathbf{x}) + \mu_{m+1} T_C v^*(\mathbf{x}), \tag{4.5.1}$$

where the operators T_A, $T_{B,i}$, and T_C are defined as

$$T_A v(\mathbf{x}) = \min_{\substack{0 \le i \le m \\ s.t.\ \mathbf{x} \ge e_i}} v(\mathbf{x} - e_i), \tag{4.5.2}$$

$$T_{B,i}v(\mathbf{x}) = \begin{cases} \min\{v(\mathbf{x}), v(\mathbf{x} - e_{i-1} + e_i) + c_i\} & \text{if } \mathbf{x} \geq e_{i-1}, \\ v(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{4.5.3}$$

and

$$T_C v(\mathbf{x}) = \begin{cases} \min\{v(\mathbf{x}), v(\mathbf{x} - e_m) - \rho\} & \text{if } \mathbf{x} \geq e_m, \\ v(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{4.5.4}$$

where c_i is the experimentation cost at stage i and $\rho \geq 0$ is the reward for a new product. The operator T_A specifies when to terminate a project, and which project to select when a project is to be terminated. The operator $T_{B,i}$ specifies when to utilize resources of stage i for a project at stage $i-1$, i.e., when to promote a project from stage $i-1$ to stage i. The operator T_C specifies when to place a new product on the market.

We proceed to characterize the structure of the optimal cost function (4.5.1). We will use three of the auxiliary indices introduced in Section 4.4: l, q, and w are the indices for the stage, and the alphabetical order $l \to q \to w$ corresponds to a progress from earlier to later stages. Also, define e_{m+1} as a zero vector of dimension n. We then define $\widetilde{V}$ as the set of real-valued functions g on $\mathbb{N}_0^n$ that satisfy the following properties:

(P.8) $g(\mathbf{x} + e_q) \geq g(\mathbf{x} + e_w)$, $\forall \mathbf{x}$, $\forall q, w \in \{1, 2, .., m\}$ where $q < w$,

(P.9) $g(\mathbf{x} + e_m) \geq g(\mathbf{x}) - \rho$, $\forall \mathbf{x}$,

(P.10) $g(\mathbf{x} + e_q + e_{w-1}) - g(\mathbf{x} + e_{q-1} + e_{w-1}) \geq g(\mathbf{x} + e_q + e_w) - g(\mathbf{x} + e_{q-1} + e_w)$, $\forall \mathbf{x}$, $\forall q, w \in \{1, 2, .., m+1\}$ where $q \neq w$,

(P.11) $g(\mathbf{x} + e_l + e_q) - g(\mathbf{x} + e_{l-1} + e_q) \geq g(\mathbf{x} + e_l + e_w) - g(\mathbf{x} + e_{l-1} + e_w)$, $\forall \mathbf{x}$, $\forall l, q, w \in \{1, 2, .., m+1\}$ where $l \leq q < w \leq m+1$,

(P.12) $g(\mathbf{x} + e_w + e_l) - g(\mathbf{x} + e_{w-1} + e_l) \geq g(\mathbf{x} + e_w + e_q) - g(\mathbf{x} + e_{w-1} + e_q)$, $\forall \mathbf{x}$, $\forall l, q, w \in \{0, 1, .., m+1\}$ where $0 \leq l < q \leq w-1$, and

(P.13) $g(\mathbf{x} + 2e_l) - g(\mathbf{x} + e_{l-1} + e_l) \geq g(\mathbf{x} + e_l + e_{l-1}) - g(\mathbf{x} + 2e_{l-1})$, $\forall \mathbf{x}$, $\forall l \in \{1, 2, .., m\}$.

The lemma below shows that the optimal cost function satisfies Properties 8–13:

Lemma 4.5.1. *Under Assumption 4.5.1, if $v \in \widetilde{V}$, then $Tv \in \widetilde{V}$, where $Tv(\mathbf{x}) = h(\mathbf{x}) + \lambda T_A v(\mathbf{x}) + \sum_{1 \leq i \leq m} \mu_i T_{B,i} v(\mathbf{x}) + \mu_{m+1} T_C v(\mathbf{x})$. Furthermore, the optimal cost function v^* is an element of $\widetilde{V}$.*

Proof. See Appendix C.3. □

We now consider the implications of Properties 8–13: Property 8 says that the optimal cost function weakly decreases as projects move from one stage to the next stage. Note that the optimal cost function may increase as projects leave the system from the last stage. However, even if this is the case, Property 9 shows that introducing a new product is always beneficial as long as it is feasible (recall that $\rho \geq 0$). Property 10 states that the incentive to promote a project from one stage to the next stage weakly increases if any project at another stage gets promoted. Property 11 implies that promoting a project from stage $l-1$ to stage l is more desirable in a non-strict sense when a project at stage $q \geq l$ is traded with another project at stage $w > q$. Furthermore, Property 12 implies that promoting a project from stage $w-1$ to stage w is more desirable in a non-strict sense when a project at stage $l < q$ is traded with another project at stage $q \leq w-1$. Conversely, Property 13 shows that the incentive to promote a project from one stage to the next stage weakly decreases if another project at the same stage gets promoted.

The intuition behind Properties 8 and 9 is as follows: Since delays in project completion times are costly, it is more desirable to have projects that are closer to the product launch stage, as well as to launch a new product immediately if it is feasible. More importantly, Properties 10–13 enable us to uncover the role congestion plays in promotion decisions at each stage. From the perspective of a project at stage $l-1$, a project at stage $q \geq l$ causes more congestion than a project at stage $w > q$: A project at stage $l-1$, if promoted along the NPD process, is more likely to catch the project at stage q than at stage w. Hence, if a project at stage w is replaced with a project at stage q, it becomes more likely that projects accumulate at the same stage and create a bottleneck, leading to investments with a lower rate of return for projects. Thus it becomes less desirable to promote the project at stage $l-1$ (Property 11). Conversely, there is a greater benefit in promoting the project at stage w if a project at stage $l < q$ is replaced with a project at stage $q < w$ (Property 12). The bottleneck of the

NPD process is more likely to occur at stage w when projects at stage $l < w$ get closer to stage w; promoting a project at stage w might help us avoid such an occurrence of the bottleneck. To sum up, it becomes more desirable to promote projects as projects at later or earlier stages get promoted. However, it becomes less desirable to promote a project as another project at the same stage gets promoted (Property 13); the risk of creating a bottleneck at a further stage is reduced by not promoting the project.

Using Lemma 4.5.1, we below show the optimality of an SDNCP policy under Assumption 4.5.1:

Theorem 4.5.1. *Under Assumption 4.5.1, the optimal portfolio strategy at each stage i is a state-dependent noncongestive-promotion policy with state-dependent promote-up-to levels $S_i^*(\mathbf{x}_{-i})$: It is optimal to promote a project to stage i if and only if $x_i < S_i^*(\mathbf{x}_{-i})$, where $\mathbf{x}_{-i} = (x_1, .., x_{i-1}, x_{i+1}, .., x_m)$ is an $m-1$ dimensional vector of the numbers of projects at stages $k \neq i$. The optimal policy has the following additional properties:*

i. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ weakly increases as the number of projects at stage $j > i$ decreases.

ii. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ weakly increases as the number of projects at stage $j < i$ increases.

iii. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ weakly increases as projects at stage $j \neq i-1$ move along the process.

iv. It is always optimal to launch a new product if there are projects available for the product launch stage.

v. It is never optimal to interrupt any experiment.

vi. It is never optimal to terminate any project.

Proof. See Appendix C.3. □

Theorem 4.5.1, using Property 11, again establishes the optimality of an SDNCP policy; each stage of the NPD process is protected against congestion through promote-up-to levels.

Points (i) and (ii) prove that the promote-up-to level at a given stage weakly increases as a later stage becomes less congested with a decline in the number of projects at that stage (due to Property 11), or as an earlier stage becomes more congested with an increase in the number of projects at that stage (due to Property 12). Points (iii) shows that the promote-up-to level at stage i weakly increases as projects at stage $j \neq i-1$ get promoted to the next stage (due to Property 10). Point (iv) states that it is always optimal to launch a new product if it is feasible (due to Property 9). Point (v) states it is never optimal to interrupt an experiment once it has been initiated. Lastly, point (vi) says that it is never optimal to terminate any project during the NPD process.

As far as we aware, Theorem 4.5.1 is the first attempt to characterize the optimal resource allocation in an NPD process with stage-dependent resources. Furthermore, we are the first to reveal the impacts of both upstream and downstream projects on optimal promotion decisions. Our model in this section shares similarities with the inventory model proposed by Benjaafar et al. (2011), although neither subsumes the other. Specifically, Benjaafar et al. (2011) consider an assembly system with multiple stages, each of which produces a different item in batches of variable sizes. They show that the optimal production policy for each item is a state-dependent base-stock policy with the base-stock level nonincreasing in the inventory level of items that are downstream and nondecreasing in the inventory level of items that are upstream. These features of the optimal policy in Benjaafar et al. (2011) match our findings in points (i) and (ii) of Theorem 4.5.1; the optimal promotion policy at each stage is a state-dependent non-congestive promotion policy with the promote-up-to level nonincreasing in the number of projects that are downstream and nondecreasing in the number of projects that are upstream.

4.6 Numerical Study

In this section, we formulate a linear program to find a global optimal solution for our general NPD model in Section 4.3 but under the average cost criterion. Based on our analytical findings in Sections 4.4 and 4.5, we then develop several heuristics for resource allocation, formulating mixed integer programs to find the optimal average cost within each heuristic class. Lastly, we numerically compare the globally optimal policy to each of our heuristics, including a naive resource allocation policy that always promotes projects if it is feasible.

4.6.1 Formulation of the Linear Program

Modifying our performance criterion to the average cost rate over an infinite-horizon planning horizon, we formulate a linear program to find a global optimal solution to the model we introduced in Section 4.3. First, denote by $\mathbb{A}(\mathbf{x})$ the set of admissible actions at state $\mathbf{x}$. Also, define $\nu_{\mathbf{x}'|\mathbf{x},\mathbf{a}}$ as the rate at which the system moves from state $\mathbf{x}$ to state $\mathbf{x}'$ if action $\mathbf{a} = (y_0, .., y_n, z) \in \mathbb{A}(\mathbf{x})$ is chosen, and $\pi_{\mathbf{x},\mathbf{a}}$ as the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{a} \in \mathbb{A}(\mathbf{x})$ is chosen. Then, the globally optimal average cost Z^* can be found by solving the following linear program:

$$\text{minimize} \quad \sum_{\mathbf{x}\in\mathbb{N}_0^n} \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} \sum_{1\le i\le m} \sum_{j\in\mathcal{W}_i} \mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x}\in\mathbb{N}_0^n} \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} \sum_{j\in\mathcal{W}_{m+1}} \mu_{m+1}\rho_j y_j \pi_{\mathbf{x},\mathbf{a}}$$

$$+ \sum_{\mathbf{x}\in\mathbb{N}_0^n} \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} h(\mathbf{x})\pi_{\mathbf{x},\mathbf{a}}$$

$$\text{subject to} \quad \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x}')} \pi_{\mathbf{x}',\mathbf{a}} \sum_{\mathbf{x}\in\mathbb{N}_0^n} \nu_{\mathbf{x}|\mathbf{x}',\mathbf{a}} - \sum_{\mathbf{x}\in\mathbb{N}_0^n} \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} \nu_{\mathbf{x}'|\mathbf{x},\mathbf{a}}\pi_{\mathbf{x},\mathbf{a}} = 0, \quad \forall \mathbf{x}' \in \mathbb{N}_0^n, \tag{4.6.1}$$

$$\sum_{\mathbf{x}\in\mathbb{N}_0^n} \sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} \pi_{\mathbf{x},\mathbf{a}} = 1, \tag{4.6.2}$$

$$\pi_{\mathbf{x},\mathbf{a}} \ge 0, \quad \forall \mathbf{x} \in \mathbb{N}_0^n, \quad \forall \mathbf{a} \in \mathbb{A}(\mathbf{x}). \tag{4.6.3}$$

The first term of the objective function corresponds to time-average experimentation costs, the second term corresponds to time-average rewards of completed projects, and the last term corresponds to time-average project holding costs. Constraints (4.6.1) and (4.6.2) are the balance equations that yield the limiting probability values.

4.6.2 Formulation of Heuristic Policies

We next describe the heuristic policies for our general model, and develop Mixed Integer Programming (MIP) formulations to find the optimal average cost within each heuristic class.

State-Dependent Non-Congestive Promotion (SDNCP): We first define $\mathbf{x}_{-j} = (x_1, .., x_{j-1}, x_{j+1}, .., x_n)$ as an $n-1$ dimensional vector of the numbers of projects in categories $k \neq j$. With this we describe an SDNCP policy as follows: (i) Resource allocation at each experimental stage follows a non-congestive promotion policy with state-dependent promote-up-to levels such that a project in category $j \in \mathcal{W}_i$ is promoted to stage i if and only if $x_{j'} < S_{j'}(\mathbf{x}_{-j'})$, $\forall j' \in \mathcal{W}_{i+1}$ where $f_{j \to j'} > 0$, and $x_k = 0$, $\forall k \in \mathcal{W}_i$ where $\rho_k > \rho_j$. (ii) A new product is always placed on the market if it is feasible; a project in category $j \in \mathcal{W}_{m+1}$ is selected for product launch if and only if $x_k = 0$, $\forall k \in \mathcal{W}_{m+1}$ where $\rho_k > \rho_j$.

The SDNCP policy has the following additional properties: (a) The promote-up-to level at one category is nondecreasing in the number of projects in any other category at an earlier stage; (b) the promote-up-to level at one category is nondecreasing in the expected reward of a project in any other category at an earlier stage; (c) the promote-up-to level at one category is nonincreasing in the number of projects in any other category at the same stage or a later stage; and (d) the promote-up-to level at one category is nonincreasing in the expected reward of a project in any other category at the same stage or a later stage.

We proceed to the MIP formulation of this heuristic class. First, define the set $\mathbb{S}_j(\mathbf{x}_{-j}, b) = \{(\mathbf{x}', \mathbf{a}) : \mathbf{x}' \in \mathbb{N}_0^n,\ \mathbf{a} \in \mathbb{A}(\mathbf{x}'),\ \mathbf{x}'_{-j} = \mathbf{x}_{-j}, \text{ and } \sum_{\mathbf{x}',\mathbf{a}} \pi_{\mathbf{x}',\mathbf{a}} = 0 \Leftrightarrow S_j(\mathbf{x}_{-j}) = b\}$ for $b \in \mathbb{N}_0$. The elements of the set $\mathbb{S}_j(\mathbf{x}_{-j}, b)$ are state-action pairs $(\mathbf{x}, \mathbf{a})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{a}$ is chosen should be zero when the promote-up-to level at category j equals $\mathbf{b}$ (given $\mathbf{x}_{-j}$). Also, define $z_b^{S_j(\mathbf{x}_{-j})}$ as a binary variable as follows:

$$z_b^{S_j(\mathbf{x}_{-j})} = \begin{cases} 1 & \text{if } S_j(\mathbf{x}_{-j}) = b, \\ 0 & \text{otherwise.} \end{cases}$$

We are now ready to describe the constraints of the MIP problem. First, the optimal solution of the MIP problem should satisfy constraints (4.6.1)-(4.6.3) of the LP formulation of the optimal policy. Also, the optimal solution should select exactly one promote-up-to level at each category, given the numbers of projects in all other categories. To this end, we impose the following constraint:

$$\sum_{b \in \mathbb{N}_0} z_b^{S_j(\mathbf{x}_{-j})} = 1, \quad \forall j \text{ and } \forall \mathbf{x}_{-j}. \tag{4.6.4}$$

The constraint below ensures that (a) the promote-up-to level at one category is nondecreasing in the number of projects in any other category at an earlier stage:

$$z_b^{S_j(\mathbf{x}_{-j})} \leq \sum_{b \leq b'} z_{b'}^{S_j(\mathbf{x}_{-j}+e_k)}, \ \forall j \in \mathcal{W}_i, \ \forall k \in \mathcal{W}_{i'} \text{ s.t. } 2 \leq i' < i, \ \forall \mathbf{x}_{-j}, \text{ and } \forall b \in \mathbb{N}_0. \tag{4.6.5}$$

The constraint below ensures that (b) the promote-up-to level at one category is nondecreasing in the expected reward of a project in any other category at an earlier stage:

$$z_b^{S_j(\mathbf{x}_{-j})} \leq \sum_{b \leq b'} z_{b'}^{S_j(\mathbf{x}_{-j}-e_k+e_{k'})}, \ \forall j \in \mathcal{W}_i, \ \forall k, k' \in \mathcal{W}_{i'} \text{ s.t. } 2 \leq i' < i \text{ and } \rho_{k'} > \rho_k, \ \forall \mathbf{x}_{-j}, \forall b. \tag{4.6.6}$$

The constraint below ensures that (c) the promote-up-to level at one category is nonincreasing in the number of projects in any other category at the same stage or a later stage:

$$z_b^{S_j(\mathbf{x}_{-j})} \leq \sum_{b' \leq b} z_{b'}^{S_j(\mathbf{x}_{-j}+e_k)}, \ \forall j \in \mathcal{W}_i, \ \forall k \in \mathcal{W}_{i'} \setminus \{j\} \text{ s.t. } i \leq i', \ \forall \mathbf{x}_{-j}, \text{ and } \forall b \in \mathbb{N}_0. \tag{4.6.7}$$

The constraint below ensures that (d) the promote-up-to level at one category is nonincreasing in the expected reward of a project in any other category at the same stage or a later stage:

$$z_b^{S_j(\mathbf{x}_{-j})} \leq \sum_{b' \leq b} z_{b'}^{S_j(\mathbf{x}_{-j}-e_k+e_{k'})}, \ \forall j \in \mathcal{W}_i, \ \forall k, k' \in \mathcal{W}_{i'} \setminus \{j\} \text{ s.t. } i \leq i' \text{ and } \rho_{k'} > \rho_k, \ \forall \mathbf{x}_{-j}, \ \forall b. \tag{4.6.8}$$

The constraint below links our binary variables to the appropriate limiting probability variables:

$$\sum_{(\mathbf{x},\mathbf{a}) \in \mathbb{S}_j(\mathbf{x}_{-j}, b)} \pi_{\mathbf{x},\mathbf{a}} \leq 1 - z_b^{S_j(\mathbf{x}_{-j})}, \ \forall b, \ \forall j, \text{ and } \forall \mathbf{x}_{-j}. \tag{4.6.9}$$

In constraint (4.6.9), if $z_b^{S_j(\mathbf{x}_{-j})}$ equals one, then all limiting probability variables corresponding to the state-action pairs in set $\mathbb{S}_j(\mathbf{x}_{-j}, b)$ are forced to equal zero. Otherwise, this constraint becomes redundant.

We next impose the following constraint to ensure that a new product is always placed on the market if it is feasible:

$$\pi_{\mathbf{x},\mathbf{a}} = 0, \ \forall(\mathbf{x}, \mathbf{a}) \ \text{ s.t. } \sum_{j \in \mathcal{W}_{m+1}} y_j = 0 \text{ and } \sum_{j \in \mathcal{W}_{m+1}} x_j > 0. \tag{4.6.10}$$

Lastly, the following constraint implies that, if a project at a given stage is to be promoted, it should be selected from the most valuable category:

$$\pi_{\mathbf{x},\mathbf{a}} = 0, \ \forall(\mathbf{x}, \mathbf{a}) \ \text{ where } \exists i, \ \exists j, j' \in \mathcal{W}_i \ \text{ s.t. } x_{j'} \geq y_j = 1 \text{ and } \rho_{j'} > \rho_j. \tag{4.6.11}$$

The optimal average cost of this policy Z_{SDNCP} can be found by solving the following MIP problem:

$$\begin{aligned}
\text{(SDNCP)} \quad \text{minimize} \quad & \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{1 \leq i \leq m} \sum_{j \in \mathcal{W}_i} \mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{j \in \mathcal{W}_{m+1}} \mu_{m+1} \rho_j y_j \pi_{\mathbf{x},\mathbf{a}} \\
& + \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{a}} \\
\text{subject to} \quad & (4.6.1) - (4.6.11).
\end{aligned}$$

Fixed Non-Congestive Promotion with Promote-up-to Levels across Categories (FNCP-C): (i) Resource allocation at each experimental stage follows a non-congestive promotion policy with fixed promote-up-to levels such that a project in category $j \in \mathcal{W}_i$ is promoted to stage i if and only if $x_{j'} < S_{j'}, \forall j' \in \mathcal{W}_{i+1}$, and $x_k = 0, \forall k \in \mathcal{W}_i$ where $\rho_k > \rho_j$. (The number of projects in *each* category at stage i affects promotion of a project at stage $i-1$.) (ii) A new product is always placed on the market if it is feasible; a project in category $j \in \mathcal{W}_{m+1}$ is selected for product launch if and only if $x_k = 0, \forall k \in \mathcal{W}_{m+1}$ where $\rho_k > \rho_j$.

We proceed to the MIP formulation of this heuristic class. First, define $\mathbf{S}_i$ as the vector of fixed promote-up-to levels at stage i. (Notice that the number of promote-up-to levels at stage i

equals K^i.) Also, define the set $\mathbb{S}_i(\mathbf{b}_i) = \{(\mathbf{x}, \mathbf{a}) : \mathbf{x} \in \mathbb{N}_0^n,\ \mathbf{a} \in \mathbb{A}(\mathbf{x}),\ \sum_{\mathbf{x},\mathbf{a}} \pi_{\mathbf{x},\mathbf{a}} = 0 \Leftrightarrow \mathbf{S}_i = \mathbf{b}_i\}$ for $\mathbf{b}_i \in \mathbb{N}_0^{K^i}$. The elements of the set $\mathbb{S}_i(\mathbf{b}_i)$ are state-action pairs $(\mathbf{x}, \mathbf{a})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{a}$ is chosen should be zero when the vector of promote-up-to levels at stage i equals $\mathbf{b}_i$. Lastly, define $z_{\mathbf{b}_i}^{\mathbf{S}_i}$ as a binary variable as follows:

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

We now describe the constraints of the MIP problem. First, the optimal solution of the MIP problem should satisfy constraints (4.6.1)-(4.6.3) of the LP formulation of the optimal policy, and constraints (4.6.10) and (4.6.11) of the MIP formulation of the SDNCP policy. Also, the optimal solution should select exactly one promote-up-to level at each category:

$$\sum_{\mathbf{b}_i \in \mathbb{N}_0^{K^i}} z_{\mathbf{b}_i}^{\mathbf{S}_i} = 1,\ \forall i. \tag{4.6.12}$$

The constraint below links our binary variables to the appropriate limiting probability variables:

$$\sum_{(\mathbf{x},\mathbf{a}) \in \mathbb{S}_i(\mathbf{b}_i)} \pi_{\mathbf{x},\mathbf{a}} \le 1 - z_{\mathbf{b}_i}^{\mathbf{S}_i},\ \forall \mathbf{b}_i,\ \forall i. \tag{4.6.13}$$

The optimal average cost of this policy Z_{FNCP-C} can be found by solving the following MIP problem:

$$\text{(FNCP-C)} \quad \text{minimize} \quad \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{1 \le i \le m} \sum_{j \in \mathbb{W}_i} \mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{j \in \mathbb{W}_{m+1}} \mu_{m+1} \rho_j y_j \pi_{\mathbf{x},\mathbf{a}}$$
$$+ \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{a}}$$
$$\text{subject to} \quad (4.6.1), (4.6.2), (4.6.3), (4.6.10), (4.6.11), (4.6.12), (4.6.13).$$

Fixed Non-Congestive Promotion with Promote-up-to Levels across Children (FNCP-Ch): (i) Resource allocation at each experimental stage follows a non-congestive promotion policy with fixed promote-up-to levels such that a project in category $j \in \mathcal{W}_i$ is promoted to stage i if and only if $x_{j'} < S_{j'}$, $\forall j' \in \mathcal{W}_{i+1}$ where $f_{j \to j'} > 0$, and $x_k = 0$, $\forall k \in \mathcal{W}_i$ where $\rho_k > \rho_j$. (ii) A new product is always placed on the market if it is feasible; a project in category $j \in \mathcal{W}_{m+1}$ is selected for product launch if and only if $x_k = 0$, $\forall k \in \mathcal{W}_{m+1}$ where $\rho_k > \rho_j$.

Notice that in this heuristic class the numbers of projects in *child* categories affect promotion of a *parent* project. Therefore, FNCP-Ch is a subclass of SDNCP; SDNCP becomes FNCP-Ch if all promote-up-to levels are constant across system states.

We proceed to the MIP formulation of this heuristic class. First, define $\mathbf{S}_j$ as the vector of fixed promote-up-to levels associated with children of category j. (Notice that the vector $\mathbf{S}_j$ has a dimension of K, $\forall j$.) Also, define the set $\mathbb{S}_j(\mathbf{b}) = \{(\mathbf{x}, \mathbf{a}) : \mathbf{x} \in \mathbb{N}_0^n,\ \mathbf{a} \in \mathbb{A}(\mathbf{x}),\ \sum_{\mathbf{x},\mathbf{a}} \pi_{\mathbf{x},\mathbf{a}} = 0 \Leftrightarrow \mathbf{S}_j = \mathbf{b}\}$ for $\mathbf{b} \in \mathbb{N}_0^K$. The elements of the set $\mathbb{S}_j(\mathbf{b})$ are state-action pairs $(\mathbf{x}, \mathbf{a})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{a}$ is chosen should be zero when the vector of promote-up-to levels associated with children of category j equals $\mathbf{b}$. Lastly, define $z_{\mathbf{b}}^{\mathbf{S}_j}$ as a binary variable as follows:

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

We next describe the constraints of the MIP problem. Again, the optimal solution of the MIP problem should satisfy constraints (4.6.1)-(4.6.3) of the LP formulation of the optimal policy, and constraints (4.6.10) and (4.6.11) of the MIP formulation of the SDNCP policy. Also, the optimal solution should select exactly one promote-up-to level at each category:

$$\sum_{\mathbf{b} \in \mathbb{N}_0^K} z_{\mathbf{b}}^{\mathbf{S}_j} = 1,\ \forall j. \tag{4.6.14}$$

The constraint below links our binary variables to the appropriate limiting probability variables:

$$\sum_{(\mathbf{x},\mathbf{a}) \in \mathbb{S}_j(\mathbf{b})} \pi_{\mathbf{x},\mathbf{a}} \leq 1 - z_{\mathbf{b}}^{\mathbf{S}_j},\ \forall \mathbf{b},\ \forall j. \tag{4.6.15}$$

The optimal average cost of this policy $Z_{FNCP-Ch}$ can be found by solving the following MIP problem:

$$\text{(FNCP-Ch)} \quad \text{minimize} \quad \sum_{\mathbf{x}\in\mathbb{N}_0^n}\sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})}\sum_{1\le i\le m}\sum_{j\in\mathcal{W}_i}\mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x}\in\mathbb{N}_0^n}\sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})}\sum_{j\in\mathcal{W}_{m+1}}\mu_{m+1}\rho_j y_j \pi_{\mathbf{x},\mathbf{a}}$$

$$+\sum_{\mathbf{x}\in\mathbb{N}_0^n}\sum_{\mathbf{a}\in\mathbb{A}(\mathbf{x})} h(\mathbf{x})\pi_{\mathbf{x},\mathbf{a}}$$

$$\text{subject to} \quad (4.6.1), (4.6.2), (4.6.3), (4.6.10), (4.6.11), (4.6.14), (4.6.15).$$

Fixed Non-Congestive Promotion with Promote-up-to Levels across Stages (FNCP-S): (i) Resource allocation at each experimental stage follows a non-congestive promotion policy with fixed promote-up-to levels such that a project in category $j \in \mathcal{W}_i$ is promoted to stage i if and only if $\sum_{j'\in\mathcal{W}_{i+1}} x_{j'} < S_i$, and $x_k = 0$, $\forall k \in \mathcal{W}_i$ where $\rho_k > \rho_j$. (Total number of projects at stage i affects promotion of a project at stage $i-1$.) (ii) A new product is always placed on the market if it is feasible; a project in category $j \in \mathcal{W}_{m+1}$ is selected for product launch if and only if $x_k = 0$, $\forall k \in \mathcal{W}_{m+1}$ where $\rho_k > \rho_j$.

We proceed to the MIP formulation of this heuristic class. First, define the set $\mathbb{S}_i(b) = \{(\mathbf{x},\mathbf{a}) : \mathbf{x} \in \mathbb{N}_0^n,\ \mathbf{a} \in \mathbb{A}(\mathbf{x}),\ \sum_{\mathbf{x},\mathbf{a}} \pi_{\mathbf{x},\mathbf{a}} = 0 \Leftrightarrow S_i = b\}$ for $b \in \mathbb{N}_0$. The elements of the set $\mathbb{S}_i(b)$ are state-action pairs $(\mathbf{x},\mathbf{a})$ such that the limiting probability that the system is in state $\mathbf{x}$ and action $\mathbf{a}$ is chosen should be zero when the promote-up-to level at stage i equals b. Lastly, define $z_b^{S_i}$ as a binary variable as follows:

$$z_b^{S_i} = \begin{cases} 1 & \text{if } S_i = b, \\ 0 & \text{otherwise.} \end{cases}$$

Once again, the optimal solution of the MIP problem should satisfy constraints (4.6.1)-(4.6.3) of the LP formulation of the optimal policy, and constraints (4.6.10) and (4.6.11) of the MIP formulation of the SDNCP policy. Also, the optimal solution should select exactly one promote-up-to level at each stage:

$$\sum_{b \in \mathbb{N}_0} z_b^{S_i} = 1, \ \forall i. \tag{4.6.16}$$

The constraint below links our binary variables to the appropriate limiting probability variables:

$$\sum_{(\mathbf{x},\mathbf{a}) \in \mathbb{S}_i(b)} \pi_{\mathbf{x},\mathbf{a}} \leq 1 - z_b^{S_i}, \ \forall b, \ \forall i. \tag{4.6.17}$$

The optimal average cost of this policy Z_{FNCP-S} can be found by solving the following MIP problem:

$$\begin{aligned} \text{(FNCP-S)} \quad \text{minimize} \quad & \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{1 \leq i \leq m} \sum_{j \in \mathcal{W}_i} \mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{j \in \mathcal{W}_{m+1}} \mu_{m+1} \rho_j y_j \pi_{\mathbf{x},\mathbf{a}} \\ & + \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{a}} \\ \text{subject to} \quad & (4.6.1), (4.6.2), (4.6.3), (4.6.10), (4.6.11), (4.6.16), (4.6.17). \end{aligned}$$

Naive Promotion (NP): This heuristic always promotes projects if it is feasible. Thus we enforce the following constraint:

$$\pi_{\mathbf{x},\mathbf{a}} = 0, \ \forall (\mathbf{x}, \mathbf{a}) \ \text{ where } \exists i \text{ s.t. } \sum_{j \in \mathcal{W}_i} y_j = 0 \text{ and } \sum_{j \in \mathcal{W}_i} x_j > 0. \tag{4.6.18}$$

The optimal average cost of this policy Z_{NP} can be found by solving the following linear program:

$$\begin{aligned} \text{(NP)} \quad \text{minimize} \quad & \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{1 \leq i \leq m} \sum_{j \in \mathcal{W}_i} \mu_i c_i y_j \pi_{\mathbf{x},\mathbf{a}} - \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} \sum_{j \in \mathcal{W}_{m+1}} \mu_{m+1} \rho_j y_j \pi_{\mathbf{x},\mathbf{a}} \\ & + \sum_{\mathbf{x} \in \mathbb{N}_0^n} \sum_{\mathbf{a} \in \mathbb{A}(\mathbf{x})} h(\mathbf{x}) \pi_{\mathbf{x},\mathbf{a}} \\ \text{subject to} \quad & (4.6.1), (4.6.2), (4.6.3), (4.6.10), (4.6.11), (4.6.18). \end{aligned}$$

4.6.3 Numerical Experiments

We proceed to numerically compare the globally optimal policy to our heuristic policies, especially investigating how system parameters affect the cost advantage of SDNCP over others. For ease of exposition, we confine our analysis to NPD processes that involve (i) two experimental stages each of which generates one out of two signals, and (ii) a product launch stage (i.e., $m = 3$ and $K = 2$). We then construct our numerical instances by varying values of the related parameters (i.e., h, c_1, c_2, μ_1, μ_2, μ_3, $\phi^{(1)}$, and $\phi^{(2)}$). We assume in all numerical instances that holding cost rates are linear (i.e., $h(\mathbf{x}) = h\sum_j x_j$), initial beliefs for a new product idea are evenly distributed among two possible states of the true nature (i.e., $p_{0,1} = p_{0,2} = 0.5$), and informativeness probabilities at each experimental stage are independent from the true nature (i.e., $\phi^{(1)}_{1,1} = \phi^{(1)}_{2,2} = \phi^{(1)}$ and $\phi^{(2)}_{1,1} = \phi^{(2)}_{2,2} = \phi^{(2)}$).

For each numerical instance, we solve the LP problem to compute the globally optimal average cost, and the MIP problems to compute the optimal average cost within each heuristic class. We compare the heuristic policies in terms of their percentage differences from optimal cost Z^*, calculated as $100 \times \frac{Z_H - Z^*}{Z^*}$ where $H \in \{\text{SDNCP, FNCP-C, FNCP-Ch, FNCP-S, NP}\}$. We coded the LP and MIP formulations in the GAMS programming language, incorporating CPLEX 10.1 optimization subroutines, and used a dual processor WinNT server, with Intel Core i7 2.67 GHz processor and 8 GB of RAM. We restricted the computation time of any instance to be no more than two hours.

We first vary experimentation and product launch rates, all else being equal; see Table 4.1. Remarkably, SDNCP yields the globally optimal cost in *all* instances we could solve without exceeding the time limit. We also observe from Table 4.1 that fixed non-congestive promotion policies (i.e., FNCP-C, FNCP-Ch, and FNCP-S) show similar performances; the average distances from optimal cost are 1.59%, 1.56%, and 1.53%, respectively. For fixed non-congestive promotion policies, the largest percentage gaps occur when the experimentation rate at stage 1 is equal to 1.5 and the product launch rate is equal to 0.5. Our explanation is that, if the NPD process slows down at further stages, it is more crucial to protect the system against congestion in a sophisticated manner, which can be achieved by SDNCP but not the other heuristics. Therefore, the optimality gaps are higher under (monotonically) decreasing

Table 4.1 Numerical results for various rates of experimentation and product launch.

				Percentage difference from optimal cost				
μ_1	μ_2	μ_3	Optimal Cost	SDNCP	FNCP-C	FNCP-Ch	FNCP-S	NP
0.5	0.5	0.5	-1.486	0.000	0.417	0.417	0.572	25.485
0.5	0.5	1.0	-2.196	0.000†	0.000	0.000	0.023	8.488
0.5	0.5	1.5	-2.425	*	0.000	0.008	0.054	6.449
0.5	1.0	0.5	-2.150	0.000	2.493	2.446	2.498	5.246
0.5	1.0	1.0	-3.327	0.000†	0.000	0.000	0.000	0.021
0.5	1.0	1.5	-3.723	0.000†	0.000	0.000	0.000	0.000
0.5	1.5	0.5	-2.366	0.000	2.265	2.223	2.278	3.128
0.5	1.5	1.0	-3.712	0.000†	0.000	0.000	0.000	0.000
0.5	1.5	1.5	-4.171	0.000†	0.000	0.000	0.000	0.000
1.0	0.5	0.5	-1.996	*	2.400	2.385	2.655	91.970
1.0	0.5	1.0	-3.092	0.000	0.058	0.058	0.071	42.196
1.0	0.5	1.5	-3.480	0.000†	0.000	0.000	0.049	38.121
1.0	1.0	0.5	-2.936	*	4.427	4.339	4.233	33.686
1.0	1.0	1.0	-5.397	*	0.020	0.020	0.026	1.219
1.0	1.0	1.5	-6.329	0.000†	0.000	0.000	0.000	0.269
1.0	1.5	0.5	-3.318	0.000	5.741	5.566	5.081	24.507
1.0	1.5	1.0	-6.269	*	0.030	0.030	0.030	0.078
1.0	1.5	1.5	-7.449	0.000†	0.000	0.000	0.000	0.000
1.5	0.5	0.5	-2.233	*	6.687	6.642	6.794	142.088
1.5	0.5	1.0	-3.467	*	0.061	0.063	0.337	72.667
1.5	0.5	1.5	-3.920	0.000†	0.000	0.000	0.051	65.740
1.5	1.0	0.5	-3.313	0.000	8.204	8.032	7.386	64.889
1.5	1.0	1.0	-6.468	*	0.077	0.077	0.173	5.753
1.5	1.0	1.5	-7.898	*	0.003	0.004	0.020	2.621
1.5	1.5	0.5	-3.753	0.000	9.721	9.417	8.698	51.362
1.5	1.5	1.0	-7.727	*	0.362	0.360	0.366	1.185
1.5	1.5	1.5	-9.859	*	0.003	0.003	0.006	0.048
Average				0.000	1.591	1.559	1.533	25.452

Notes. $h = 2$, $c_1 = c_2 = 4$, $r_1 = 40$, $r_2 = 0$, $\lambda = 100$, $\phi^{(1)} = \phi^{(2)} = 0.75$, and $p_{0,1} = p_{0,2} = 0.5$. *The MIP solver could not solve the corresponding instance within two hours. †We did not solve the corresponding instance for SDNCP; FNCP-Ch, which is a special class of SDNCP, yields the globally optimal cost, and so does SDNCP.

rates of experimentation and product launch.

Conversely, if the NPD process speeds up at later stages, it is more desirable to aggressively promote projects; the percentage gaps are lower for NP under (monotonically) increasing rates of experimentation and product launch. But it is important to note that NP yields the globally optimal cost even when $\mu_1 = 0.5$, $\mu_2 = 1.5$, and $\mu_3 = 1$: New product ideas should always be tested at the initial stage as the later stages are faster, and also upon completion of the initial stage, projects should always be tested at the second stage as its throughput rate, which is constrained by the initial stage, is lower than the product launch rate.

Table 4.2 lists computation times for the numerical instances introduced in Table 4.1. We

Table 4.2 Numerical results for various rates of experimentation and product launch.

			Computation times (in seconds)					
μ_1	μ_2	μ_3	Opt. Soln.	SDNCP	FNCP-C	FNCP-Ch	FNCP-S	NP
0.5	0.5	0.5	0.24	**7200.00**	495.93	10.61	2.16	0.18
0.5	0.5	1.0	0.21	*	246.42	3.78	1.21	0.18
0.5	0.5	1.5	0.47	**7200.00**	286.50	5.79	1.10	0.15
0.5	1.0	0.5	0.22	98.22	451.77	6.81	2.88	0.15
0.5	1.0	1.0	0.25	*	216.31	6.71	2.35	0.17
0.5	1.0	1.5	0.46	*	199.80	3.16	2.37	0.19
0.5	1.5	0.5	0.27	68.86	681.22	14.44	2.65	0.18
0.5	1.5	1.0	0.29	*	433.61	7.72	1.42	0.16
0.5	1.5	1.5	0.43	*	260.83	2.54	1.14	0.22
1.0	0.5	0.5	0.16	**7200.00**	379.08	8.17	2.09	0.17
1.0	0.5	1.0	0.30	692.25	293.05	6.14	1.30	0.22
1.0	0.5	1.5	0.38	*	164.30	3.51	1.24	0.20
1.0	1.0	0.5	0.26	**7200.00**	717.47	12.18	2.26	0.19
1.0	1.0	1.0	0.29	**7200.00**	267.96	6.13	1.15	0.18
1.0	1.0	1.5	0.32	*	245.49	5.30	1.08	0.24
1.0	1.5	0.5	0.21	53.22	536.58	39.84	1.45	0.17
1.0	1.5	1.0	0.24	**7200.00**	273.39	7.34	2.50	0.21
1.0	1.5	1.5	0.32	*	363.88	4.72	2.02	0.19
1.5	0.5	0.5	0.18	**7200.00**	424.90	7.45	2.01	0.25
1.5	0.5	1.0	0.24	**7200.00**	277.02	4.33	1.52	0.25
1.5	0.5	1.5	0.25	*	201.47	3.81	1.74	0.24
1.5	1.0	0.5	0.20	4235.38	743.27	10.29	2.62	0.21
1.5	1.0	1.0	0.24	**7200.00**	282.34	7.12	1.37	0.24
1.5	1.0	1.5	0.29	**7200.00**	240.30	4.28	1.33	0.23
1.5	1.5	0.5	0.19	4426.84	742.87	9.24	2.70	0.21
1.5	1.5	1.0	0.25	**7200.00**	473.53	13.04	2.19	0.27
1.5	1.5	1.5	0.33	**7200.00**	251.32	6.10	1.73	0.24
Average			0.27	5331.93	375.95	8.17	1.84	0.20

Notes. $h = 2$, $c_1 = c_2 = 4$, $r_1 = 40$, $r_2 = 0$, $\lambda = 100$, $\phi^{(1)} = \phi^{(2)} = 0.75$, and $p_{0,1} = p_{0,2} = 0.5$. Single star (*) indicates that we did not solve the corresponding instance for SDNCP; FNCP-Ch yields the globally optimal cost, and so does SDNCP. Computation times equal to 7200 seconds indicate termination of the algorithm.

observe that computation times of FNCP-C are two orders of magnitude greater than those of FNCP-Ch and FNCP-S, which have the same order of magnitude on average. Recall from Table 4.1 performances of these three heuristics are comparable with respect to objective value. Thus we drop FNCP-C and FNCP-S from our experimental set; we opt to keep FNCP-Ch as it enables us to rigorously investigate the importance of allowing *state-dependent* promote-up-to levels, without computation time concerns. We also omit NP from our experimental set, as it yields unsatisfactory results due to our selection of experimentation and product launch rates in the remainder of this section. See Appendix C.4 for the omitted numerical results.

We next vary the holding cost rate and experimentation costs, all else being equal; see

Table 4.3 Numerical results for various holding and experimentation costs.

				Absolute gap		Percentage gap		Computation times	
h	c_1	c_2	Optimal Cost	SDNCP	FNCP-Ch	SDNCP	FNCP-Ch	SDNCP	FNCP-Ch
1	2	2	-8.485	*	0.074	*	0.873	**7200.00**	13.41
1	2	4	-7.288	0.000	0.108	0.000	1.475	534.29	9.84
1	2	6	-6.240	0.000	0.233	0.000	3.735	1630.05	5.92
1	4	2	-6.311	*	0.228	*	3.606	**7200.00**	6.80
1	4	4	-5.429	0.000	0.282	0.000	5.198	463.55	9.50
1	4	6	-4.581	0.000	0.232	0.000	5.058	96.83	8.17
1	6	2	-4.687	0.000	0.233	0.000	4.969	81.31	10.62
1	6	4	-3.790	*	0.200	*	5.272	**7200.00**	12.91
1	6	6	-2.956	0.000	0.164	0.000	5.538	38.05	8.47
2	2	2	-6.292	*	0.263	*	4.180	**7200.00**	12.51
2	2	4	-5.299	0.008	0.169	0.160	3.191	7082.82	5.49
2	2	6	-4.473	0.000	0.215	0.000	4.798	60.28	3.68
2	4	2	-4.498	0.000	0.224	0.000	4.989	100.21	10.23
2	4	4	-3.753	0.000	0.353	0.000	9.417	4270.56	8.84
2	4	6	-3.051	0.000	0.417	0.000	13.667	56.16	8.58
2	6	2	-3.111	0.000	0.450	0.000	14.465	48.85	13.82
2	6	4	-2.362	*	0.410	*	17.355	**7200.00**	9.40
2	6	6	-1.671	0.000	0.344	0.000	20.596	218.55	6.32
3	2	2	-4.777	*	0.271	*	5.667	**7200.00**	7.32
3	2	4	-3.945	0.059	0.311	1.483	7.874	95.16	4.57
3	2	6	-3.196	0.000	0.434	0.000	13.579	43.82	5.72
3	4	2	-3.199	0.000	0.437	0.000	13.655	3483.90	12.03
3	4	4	-2.507	0.000	0.508	0.000	20.269	104.37	3.01
3	4	6	-1.992	0.000	0.626	0.000	31.429	81.05	2.30
3	6	2	-1.999	0.000	0.633	0.000	31.652	44.03	5.84
3	6	4	-1.493	0.000	0.751	0.000	50.275	32.47	3.62
3	6	6	-0.992	0.000	0.873	0.000	88.034	48.43	3.76
Average				0.003	0.350	0.078	14.475	2289.44	7.88

Notes. $r_1 = 40$, $r_2 = 0$, $\mu_1 = \mu_2 = 1.5$, $\mu_3 = 0.5$, $\lambda = 100$, $\phi^{(1)} = \phi^{(2)} = 0.75$, and $p_{0,1} = p_{0,2} = 0.5$. Single star (*) indicates that the MIP solver could not solve the corresponding example within two hours. Computation times equal to 7200 seconds indicate termination of the algorithm.

Table 4.3. We observe that SDNCP minimizes the average costs in *most* of the instances we could solve without exceeding the time limit. We also observe that the optimality gap of FNCP-Ch increases as the holding cost rate and/or experimentation costs increase: It is almost always beneficial to terminate projects with little promise from the NPD process under high holding and/or experimentation costs. But negative optimal costs prove that it is still desirable to develop new products (from high-value categories); the benefit of introducing a new product largely relies on cost savings as a result of significantly reduced time-to-market (or congestion), which calls for a broader monitoring of projects across stages, in decision-making. For FNCP-Ch, however, promotion of projects at one stage depends only on projects at the

Table 4.4 Numerical results for various informativeness probabilities.

			Absolute gap		Percentage gap		Computation times	
$\phi^{(1)}$	$\phi^{(2)}$	Optimal Cost	SDNCP	FNCP-Ch	SDNCP	FNCP-Ch	SDNCP	FNCP-Ch
0.95	0.95	-5.630	0.007	0.321	0.123	5.703	5191.45	5.51
-	0.85	-5.570	0.000	0.352	0.000	6.320	57.93	5.94
-	0.75	-5.570	0.000	0.367	0.000	6.582	5724.65	8.75
-	0.65	-5.570	0.000	0.365	0.000	6.553	159.24	4.80
-	0.55	-5.570	0.000	0.361	0.000	6.481	51.74	5.54
0.85	0.95	-4.606	0.000	0.231	0.000	5.020	205.42	10.22
-	0.85	-4.239	*	0.174	*	4.105	**7200.00**	6.97
-	0.75	-4.213	0.000	0.219	0.000	5.201	77.71	5.11
-	0.65	-4.212	*	0.218	*	5.178	**7200.00**	4.48
-	0.55	-4.212	0.000	0.218	0.000	5.185	83.12	4.88
0.75	0.95	-3.643	0.000	0.200	0.000	5.495	1648.26	6.73
-	0.85	-3.063	0.000	0.098	0.000	3.206	86.18	13.79
-	0.75	-2.936	*	0.127	*	4.339	**7200.00**	11.93
-	0.65	-2.936	*	0.149	*	5.062	**7200.00**	6.83
-	0.55	-2.935	0.000	0.147	0.000	4.998	58.35	9.47
0.65	0.95	-2.960	0.012	0.115	0.416	3.892	130.70	9.53
-	0.85	-2.275	*	0.121	*	5.296	**7200.00**	10.89
-	0.75	-1.961	*	0.188	*	9.601	**7200.00**	11.76
-	0.65	-1.918	*	0.267	*	13.919	**7200.00**	12.24
-	0.55	-1.893	0.002	0.256	0.127	13.495	5483.98	9.94
0.55	0.95	-2.893	*	0.114	*	3.951	**7200.00**	7.20
-	0.85	-2.120	*	0.144	*	6.768	**7200.00**	9.31
-	0.75	-1.619	*	0.241	*	14.901	**7200.00**	13.65
-	0.65	-1.500	*	0.464	*	30.940	**7200.00**	11.03
-	0.55	-1.500	*	0.526	*	35.093	**7200.00**	12.93
Average			0.002	0.239	0.051	8.691	4214.35	8.78

Notes. $h = 2$, $c_1 = c_2 = 4$, $r_1 = 40$, $r_2 = 0$, $\mu_1 = \mu_2 = 1$, $\mu_3 = 0.5$, $\lambda = 100$, and $p_{0,1} = p_{0,2} = 0.5$. Single star (*) indicates that the MIP solver could not solve the corresponding example within two hours. Computation times equal to 7200 seconds indicate termination of the algorithm.

next stage. Hence, higher holding and/or experimentation costs lead to larger optimality gaps.

For FNCP-Ch, it is also important to note that an increment in experimentation cost of stage 1 leads to larger optimality gaps than in experimentation cost of stage 2: If the experimentation cost is higher at stage 1, an investment on a new product idea has a much lower rate of return than an investment on a project that shows promise upon completion of the initial stage. Thus, further caution must be taken while promoting new product ideas. But FNCP-Ch fails to do so as it does not incorporate projects at the last stage into decision-making for new product ideas, and thus a larger percentage gap results.

Lastly, we vary informativeness probabilities, all else being equal; see Table 4.4. Once again, SDNCP minimizes the average costs in *most* of the instances we could solve. For

FNCP-Ch, the largest percentage gaps occur when experiments are highly uninformative (i.e., when $\phi^{(1)} \leq 0.65$ and $\phi^{(2)} \leq 0.75$): As experiments become more informative, the system tends to terminate more projects from low-value categories. But then low-value categories have little impact on congestion; it is less valuable to keep track of the numbers of projects in *each* category. Since FNCP-Ch uses less information regarding the numbers of projects across categories, it performs closer to the global optimal policy when experiments are highly informative.

Our numerical results have led us to conclude that SDNCP performs significantly better than FNCP-Ch (i) when the NPD process slows down at downstream stages, (ii) when holding and/or experimentation costs are higher, or (iii) when experiments are highly uninformative. As for computation times, FNCP-Ch has distinct advantage over SDNCP.

4.7 Conclusions

We have studied the problem of resource allocation and project selection for NPD under Markovian assumptions. Each NPD project undergoes a different experiment at each stage of the NPD process. Signals created through experimentation enable project managers to resolve uncertainty around the true nature of the project. Projects are therefore categorized based on their signals; the state space consists of the numbers of projects in each category. NPD projects at different stages demand the use of different specialized resources, requiring us to make concurrent resource allocation decisions across stages at a time. A control policy specifies when and at what rate to utilize the resources at each stage, and on which projects.

We show the optimality of SDNCP in two special cases of the problem: (a) when there are multiple uninformative experiments, or (b) when there is a single informative experiment. An SDNCP policy implies that, at each stage, it is optimal to advance a project with the highest expected reward to the next stage if and only if the number of projects in each successor category is less than a state-dependent threshold. Furthermore, threshold values decrease in a non-strict sense as a later stage becomes more congested or as an earlier stage becomes less congested; a stage becomes more congested with an increase in the number of projects at this stage or with an increase in the expected reward of any project at this stage. To our knowledge, this study is the first to reveal the impact of congestion on optimal resource allocation decisions under stage-dependent resource constraints.

We also conduct numerical experiments to evaluate the use of SDNCP as a heuristic for the general problem. We find that SDNCP yields the globally optimal cost in over half of the 80 instances of the general NPD problem. Further, we numerically compare SDNCP to FNCP-Ch, which is a simpler version of SDNCP whose thresholds are constant across system states. Our numerical experiments demonstrate that SDNCP substantially outperforms FNCP-Ch, (i) when upstream stages are faster than downstream stages, or (ii) when expected margins of projects are lower, or (iii) when experiments are less informative.

An important avenue for future research is to explore the optimal policy for the general NPD problem. But further optimality results might require alternate metrics for congestion, which will capture the following properties: The incentive to promote projects increases if a project at any later stage is promoted to the next stage, returning a signal that will keep its expected reward below a certain threshold. Likewise, the incentive to promote projects increases if a project at any earlier stage is promoted to the next stage, returning a signal that will keep its expected reward above a certain threshold. Another direction for future research is to extend our optimality results to allow for general experimentation times. (It appears straightforward that SDNCP remains optimal under Erlang experimentation times when there are multiple uninformative experiments.) Also, extending our Bayesian framework to different conjugate priors is an interesting direction for future studies. On a wider level, research is also needed to incorporate competition between new and previously developed products into decision-making for resource allocation and project selection. But such an extension will likely require the use of game theoretical models.

Chapter 5

Conclusions

It is generally accepted that ATO and NPD problems are hard to analyze. This dissertation significantly broadens current knowledge of optimal policies for these complex, dynamic problems by employing the theory of MDP.

For the ATO problem, much of the complexity is attributable to the fact that inventory replenishment and allocation decisions should be handled simultaneously. The problem becomes even more difficult when replenishment and/or allocation decisions involve non-unitary changes in several component inventory levels, as is the case in our M-systems. We modeled the problem as an MDP; the state space consists of component inventory levels. We then developed novel functional characterizations restricted to certain lattices of the state space. The optimal cost function satisfies these functional characterizations if the state space is partitioned into disjoint lattices based on component requirements of products. With this we have managed to prove the optimality of a lattice-dependent base-stock and lattice-dependent rationing (LBLR) policy. Computational results verify the practicality of LBLR as a heuristic for the general ATO problem.

As for the NPD problem, much of the complexity comes from the evolution of projects throughout the NPD process, which is subject to stage-dependent resource constraints. We again modeled the problem as an MDP: Since beliefs about the true nature of a project undergo Bayesian updating after each experiment, we categorize projects based on posterior beliefs and keep in the state space the breakdown of projects into categories. Due to stage-dependent resource constraints, concurrent resource allocation decisions should be made across stages at a time. Therefore we define a separate set of actions for resource allocation at each stage. As a result, however, both the state and action spaces are unmanageably huge. We have been able to prove the optimality of a state-dependent non-congestive promotion (SDNCP) policy in two simpler versions of the problem, reporting the strong numerical performance of SDNCP for the general NPD problem.

Future work needs to be carried out to establish optimal policies for general ATO and NPD problems. But we are confident that the findings in this dissertation will serve as a basis for future studies on more general settings. More broadly, our treatments of the ATO and NPD problems highlight that MDPs might be very useful in handling complex problems that include dynamics and uncertainty.

Appendix A

Supplement to Optimal Structural Results for ATO M-Systems

This chapter includes supplementary material for Chapter 2: New Functional Characterizations and Optimal Structural Results for Assemble-to-Order M-Systems.

A.1 Proofs of the Results in Section 2.4.1

Lemma 2.4.1 (Restated)**.** $Super(\mathbf{r}, \mathbf{p}) \subseteq nSuper(\mathbf{r}, \mathbf{p})$, $\forall \mathbf{r}, \mathbf{p} \in \mathbb{N}_0^n$.

Proof. $f \in Super(\mathbf{r}, \mathbf{p})$ implies the following inequalities:

$$\begin{aligned} f(\mathbf{x} + p_1 e_1 + \mathbf{r}) - f(\mathbf{x} + p_1 e_1) &\geq f(\mathbf{x} + \mathbf{r}) - f(\mathbf{x}), \\ f(\mathbf{x} + p_1 e_1 + p_2 e_2 + \mathbf{r}) - f(\mathbf{x} + p_1 e_1 + p_2 e_2) &\geq f(\mathbf{x} + p_1 e_1 + \mathbf{r}) - f(\mathbf{x} + p_1 e_1), \\ &\vdots \\ f(\mathbf{x} + \sum_{j \leq n} p_j e_j + \mathbf{r}) - f(\mathbf{x} + \sum_{j \leq n} p_j e_j) &\geq f(\mathbf{x} + \sum_{j < n} p_j e_j + \mathbf{r}) - f(\mathbf{x} + \sum_{j < n} p_j e_j) \end{aligned}$$

Summation of the inequalities above implies $f(\mathbf{x} + \mathbf{p} + \mathbf{r}) - f(\mathbf{x} + \mathbf{p}) \geq f(\mathbf{x} + \mathbf{r}) - f(\mathbf{x})$, and therefore $f \in nSuper(\mathbf{r}, \mathbf{p})$. □

A.2 Proofs of the Results in Section 2.4.2

Lemma 2.4.2 (Restated)**.** (a) $T^{(j)} : Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q}) \to Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q})$, $\forall j$,

(b) $T_i : Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b}) \to Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b})$, $\forall i$, *and*

(c) $h \in Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q}) \cap Sub(\mathbf{b}) \cap Super(\mathbf{a}, \mathbf{b})$.

Proof. Recall that $T^{(j)}v(\mathbf{x}) = \min\{v(\mathbf{x} + q_j e_j), v(\mathbf{x})\}$, $T_i v(\mathbf{x}) = \min\{v(\mathbf{x}) + c_i, v(\mathbf{x} - b_i e_i)\}$ if $x_i \geq b_i$, and $T_i v(\mathbf{x}) = v(\mathbf{x}) + c_i$ otherwise, for $i \leq n$; and $T_{n+1}v(\mathbf{x}) = \min\{v(\mathbf{x}) + c_{n+1}, v(\mathbf{x} - \mathbf{a})\}$ if $x_j \geq a_j$ for all j, and $T_{n+1}v(\mathbf{x}) = v(\mathbf{x}) + c_{n+1}$ otherwise.

(a) Assume that $v \in Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q})$. We will show $T^{(j)}v \in Sub(\mathbf{q}) \cap Super(\mathbf{a}, \mathbf{q})$.

- First we show $T^{(j)}v \in Sub(\mathbf{q})$, i.e., $T^{(j)}v(\mathbf{x} + q_i e_i) - T^{(j)}v(\mathbf{x}) \geq T^{(j)}v(\mathbf{x} + q_i e_i + q_k e_k) - T^{(j)}v(\mathbf{x} + q_k e_k)$, $\forall k \neq i$. Pick arbitrary $k \in \{1, 2, .., n\}$. There are four different scenarios we need to consider depending on the optimal actions at $T^{(j)}v(\mathbf{x}+q_i e_i)$ and $T^{(j)}v(\mathbf{x}+q_k e_k)$ (if this inequality holds under suboptimal actions of $T^{(j)}v(\mathbf{x})$ and/or $T^{(j)}v(\mathbf{x}+q_i e_i+q_k e_k)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators). These four scenarios are as follows:

  (1) Suppose that $T^{(j)}v(\mathbf{x}+q_i e_i) = v(\mathbf{x}+q_i e_i) < v(\mathbf{x}+q_j e_j+q_i e_i)$ and $T^{(j)}v(\mathbf{x}+q_k e_k) = v(\mathbf{x}+q_k e_k) < v(\mathbf{x}+q_j e_j+q_k e_k)$. As we assume $v \in Sub(\mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x} + q_i e_i) - T^{(j)}v(\mathbf{x}) &\geq v(\mathbf{x} + q_i e_i) - v(\mathbf{x}) \\ &\geq v(\mathbf{x} + q_i e_i + q_k e_k) - v(\mathbf{x} + q_k e_k) \\ &\geq T^{(j)}v(\mathbf{x} + q_i e_i + q_k e_k) - T^{(j)}v(\mathbf{x} + q_k e_k) \end{aligned}$$

  (2) Suppose that $T^{(j)}v(\mathbf{x}+q_i e_i) = v(\mathbf{x}+q_j e_j+q_i e_i) < v(\mathbf{x}+q_i e_i)$ and $T^{(j)}v(\mathbf{x}+q_k e_k) = v(\mathbf{x}+q_k e_k) < v(\mathbf{x}+q_j e_j+q_k e_k)$. As we assume $v \in Sub(\mathbf{q})$, the following inequalities

hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_ie_i) - T^{(j)}v(\mathbf{x}) &\geq v(\mathbf{x}+q_je_j+q_ie_i) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+q_je_j) - v(\mathbf{x}+q_je_j+q_ke_k) \\ &\quad +v(\mathbf{x}+q_je_j+q_ie_i+q_ke_k) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i+q_ke_k) - v(\mathbf{x}+q_ke_k) \\ &\geq T^{(j)}v(\mathbf{x}+q_ie_i+q_ke_k) - T^{(j)}v(\mathbf{x}+q_ke_k) \end{aligned}$$

(3) Suppose that $T^{(j)}v(\mathbf{x}+q_ie_i) = v(\mathbf{x}+q_ie_i) < v(\mathbf{x}+q_je_j+q_ie_i)$ and $T^{(j)}v(\mathbf{x}+q_ke_k) = v(\mathbf{x}+q_je_j+q_ke_k) < v(\mathbf{x}+q_ke_k)$. If $j = i$, then it is easy to verify that

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_je_j) - T^{(j)}v(\mathbf{x}) &\geq v(\mathbf{x}+q_je_j) - v(\mathbf{x}+q_je_j) \\ &= v(\mathbf{x}+q_je_j+q_ke_k) - v(\mathbf{x}+q_je_j+q_ke_k) \\ &\geq T^{(j)}v(\mathbf{x}+q_je_j+q_ke_k) - T^{(j)}v(\mathbf{x}+q_ke_k) \end{aligned}$$

If $j \neq i$, as we assume $v \in Sub(\mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_ie_i) - T^{(j)}v(\mathbf{x}) &\geq v(\mathbf{x}+q_ie_i) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i) - v(\mathbf{x}+q_je_j) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i+q_ke_k) - v(\mathbf{x}+q_je_j+q_ke_k) \\ &\geq T^{(j)}v(\mathbf{x}+q_ie_i+q_ke_k) - T^{(j)}v(\mathbf{x}+q_ke_k) \end{aligned}$$

(4) Suppose that $T^{(j)}v(\mathbf{x}+q_ie_i) = v(\mathbf{x}+q_je_j+q_ie_i) < v(\mathbf{x}+q_ie_i)$ and $T^{(j)}v(\mathbf{x}+q_ke_k) = v(\mathbf{x}+q_je_j+q_ke_k) < v(\mathbf{x}+q_ke_k)$. As we assume $v \in Sub(\mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_ie_i) - T^{(j)}v(\mathbf{x}) &\geq v(\mathbf{x}+q_je_j+q_ie_i) - v(\mathbf{x}+q_je_j) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i+q_ke_k) - v(\mathbf{x}+q_je_j+q_ke_k) \\ &\geq T^{(j)}v(\mathbf{x}+q_ie_i+q_ke_k) - T^{(j)}v(\mathbf{x}+q_ke_k) \end{aligned}$$

Hence our inequality holds in each of the possible scenarios. Therefore, $T^{(j)}v \in Sub(\mathbf{q})$.

- Next we show $T^{(j)}v \in Super(\mathbf{a}, \mathbf{q})$, i.e., $T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) - T^{(j)}v(\mathbf{x} + \mathbf{a}) \geq T^{(j)}v(\mathbf{x} + q_i e_i) - T^{(j)}v(\mathbf{x})$, $\forall i$. Again, there are four different scenarios depending on the optimal actions at $T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a})$ and $T^{(j)}v(\mathbf{x})$:

  (1) Suppose that $T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) = v(\mathbf{x} + q_i e_i + \mathbf{a}) < v(\mathbf{x} + q_j e_j + q_i e_i + \mathbf{a})$ and $T^{(j)}v(\mathbf{x}) = v(\mathbf{x}) < v(\mathbf{x} + q_j e_j)$. As we assume $v \in Super(\mathbf{a}, \mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) - T^{(j)}v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + q_i e_i + \mathbf{a}) - v(\mathbf{x} + \mathbf{a}) \\ &\geq v(\mathbf{x} + q_i e_i) - v(\mathbf{x}) \\ &\geq T^{(j)}v(\mathbf{x} + q_i e_i) - T^{(j)}v(\mathbf{x}) \end{aligned}$$

  (2) Suppose that $T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) = v(\mathbf{x} + q_j e_j + q_i e_i + \mathbf{a}) < v(\mathbf{x} + q_i e_i + \mathbf{a})$ and $T^{(j)}v(\mathbf{x}) = v(\mathbf{x}) < v(\mathbf{x} + q_j e_j)$. As we assume $v \in Super(\mathbf{a}, \mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) - T^{(j)}v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + q_j e_j + q_i e_i + \mathbf{a}) - v(\mathbf{x} + \mathbf{a}) \\ &\geq v(\mathbf{x} + q_j e_j + q_i e_i) + v(\mathbf{x} + q_j e_j + \mathbf{a}) \\ &\quad -v(\mathbf{x} + q_j e_j) - v(\mathbf{x} + \mathbf{a}) \\ &\geq v(\mathbf{x} + q_j e_j + q_i e_i) - v(\mathbf{x}) \\ &\geq T^{(j)}v(\mathbf{x} + q_i e_i) - T^{(j)}v(\mathbf{x}) \end{aligned}$$

  (3) Suppose that $T^{(j)}v(\mathbf{x} + q_i e_i + \mathbf{a}) = v(\mathbf{x} + q_i e_i + \mathbf{a}) < v(\mathbf{x} + q_j e_j + q_i e_i + \mathbf{a})$ and $T^{(j)}v(\mathbf{x}) = v(\mathbf{x} + q_j e_j) < v(\mathbf{x})$. If $j = i$, then it is easy to verify that

$$\begin{aligned} T^{(j)}v(\mathbf{x} + q_j e_j + \mathbf{a}) - T^{(j)}v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + q_j e_j + \mathbf{a}) - v(\mathbf{x} + q_j e_j + \mathbf{a}) \\ &= v(\mathbf{x} + q_j e_j) - v(\mathbf{x} + q_j e_j) \\ &\geq T^{(j)}v(\mathbf{x} + q_j e_j) - T^{(j)}v(\mathbf{x}) \end{aligned}$$

If $j \neq i$, as we assume $v \in \mathit{Super}(\mathbf{a}, \mathbf{q})$ and $v \in \mathit{Sub}(\mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_ie_i+\mathbf{a}) - T^{(j)}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+q_ie_i+\mathbf{a}) - v(\mathbf{x}+\mathbf{a}) \\ &\geq v(\mathbf{x}+q_ie_i) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i) - v(\mathbf{x}+q_je_j) \\ &\geq T^{(j)}v(\mathbf{x}+q_ie_i) - T^{(j)}v(\mathbf{x}) \end{aligned}$$

(4) Suppose that $T^{(j)}v(\mathbf{x}+q_ie_i+\mathbf{a}) = v(\mathbf{x}+q_je_j+q_ie_i+\mathbf{a}) < v(\mathbf{x}+q_ie_i+\mathbf{a})$ and $T^{(j)}v(\mathbf{x}) = v(\mathbf{x}+q_je_j) < v(\mathbf{x})$. As we assume $v \in \mathit{Super}(\mathbf{a}, \mathbf{q})$, the following inequalities hold:

$$\begin{aligned} T^{(j)}v(\mathbf{x}+q_ie_i+\mathbf{a}) - T^{(j)}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+q_je_j+q_ie_i+\mathbf{a}) - v(\mathbf{x}+q_je_j+\mathbf{a}) \\ &\geq v(\mathbf{x}+q_je_j+q_ie_i) - v(\mathbf{x}+q_je_j) \\ &\geq T^{(j)}v(\mathbf{x}+q_ie_i) - T^{(j)}v(\mathbf{x}) \end{aligned}$$

Hence our inequality holds in all the possible scenarios. Therefore, $T^{(j)}v \in \mathit{Super}(\mathbf{a}, \mathbf{q})$.

(b) Assume that $v \in \mathit{Sub}(\mathbf{b}) \cap \mathit{Super}(\mathbf{a}, \mathbf{b})$. We will show $T_iv \in \mathit{Sub}(\mathbf{b}) \cap \mathit{Super}(\mathbf{a}, \mathbf{b})$.

Case I: Suppose that $i \leq n$.

- First we show $T_iv \in \mathit{Sub}(\mathbf{b})$, i.e., $T_iv(\mathbf{x}+b_je_j) - T_iv(\mathbf{x}) \geq T_iv(\mathbf{x}+b_je_j+b_ke_k) - T_iv(\mathbf{x}+b_ke_k)$, $\forall k \neq j$. Pick arbitrary $k \in \{1, 2, .., n\}$. There are four different scenarios we need to consider depending on the optimal actions at $T_iv(\mathbf{x}+b_je_j)$ and $T_iv(\mathbf{x}+b_ke_k)$ (if this inequality holds under suboptimal actions of $T_iv(\mathbf{x})$ and/or $T_iv(\mathbf{x}+b_je_j+b_ke_k)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators). These four scenarios are as follows:

  (1) Suppose that $T_iv(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j)+c_i$ and $T_iv(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k)+c_i$.

As we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned}
T_iv(\mathbf{x}+b_je_j) - T_iv(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j) + c_i - v(\mathbf{x}) - c_i \\
&\geq v(\mathbf{x}+b_je_j+b_ke_k) + c_i - v(\mathbf{x}+b_ke_k) - c_i \\
&\geq T_iv(\mathbf{x}+b_je_j+b_ke_k) - T_iv(\mathbf{x}+b_ke_k)
\end{aligned}$$

(2) Suppose that $x_i \geq b_i$, $T_iv(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j)+c_i$ and $T_iv(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k-b_ie_i)$. As we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned}
T_iv(\mathbf{x}+b_je_j) - T_iv(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j) + c_i - v(\mathbf{x}-b_ie_i) \\
&\geq v(\mathbf{x}) - v(\mathbf{x}+b_ke_k) \\
&\quad + v(\mathbf{x}+b_je_j+b_ke_k) + c_i - v(\mathbf{x}-b_ie_i) \\
&\geq v(\mathbf{x}+b_je_j+b_ke_k) + c_i - v(\mathbf{x}+b_ke_k-b_ie_i) \\
&\geq T_iv(\mathbf{x}+b_je_j+b_ke_k) - T_iv(\mathbf{x}+b_ke_k)
\end{aligned}$$

(3) Suppose that $x_i \geq b_i$ if $i \neq j$, $T_iv(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j-b_ie_i)$ and $T_iv(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k)+c_i$. If $i=j$, then it is easy to verify that

$$\begin{aligned}
T_iv(\mathbf{x}+b_ie_i) - T_iv(\mathbf{x}) &\geq v(\mathbf{x}) - v(\mathbf{x}) - c_i \\
&= v(\mathbf{x}+b_ke_k) - v(\mathbf{x}+b_ke_k) - c_i \\
&\geq T_iv(\mathbf{x}+b_ie_i+b_ke_k) - T_iv(\mathbf{x}+b_ke_k)
\end{aligned}$$

If $i \neq j$, as we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned}
T_iv(\mathbf{x}+b_je_j) - T_iv(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j-b_ie_i) - v(\mathbf{x}-b_ie_i) \\
&\geq v(\mathbf{x}+b_je_j) - v(\mathbf{x}) \\
&\geq v(\mathbf{x}+b_je_j+b_ke_k) + c_i - v(\mathbf{x}+b_ke_k) - c_i \\
&\geq T_iv(\mathbf{x}+b_je_j+b_ke_k) - T_iv(\mathbf{x}+b_ke_k)
\end{aligned}$$

(4) Suppose that $x_i \geq b_i$, $T_iv(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j-b_ie_i)$ and $T_iv(\mathbf{x}+b_ke_k) =$

$v(\mathbf{x} + b_k e_k - b_i e_i)$. As we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x}) &\geq v(\mathbf{x} + b_j e_j - b_i e_i) - v(\mathbf{x} - b_i e_i) \\ &\geq v(\mathbf{x} + b_j e_j + b_k e_k - b_i e_i) - v(\mathbf{x} + b_k e_k - b_i e_i) \\ &\geq T_i v(\mathbf{x} + b_j e_j + b_k e_k) - T_i v(\mathbf{x} + b_k e_k) \end{aligned}$$

Hence our inequality holds in all the possible scenarios. Therefore, $T_i v \in Sub(\mathbf{b})$.

- Next we show $T_i v \in Super(\mathbf{a}, \mathbf{b})$, i.e., $T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) \geq T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x})$, $\forall j$. Again, there are four different scenarios depending on the optimal actions at $T_i v(\mathbf{x} + b_j e_j + \mathbf{a})$ and $T_i v(\mathbf{x})$:

  (1) Suppose that $x_i \geq b_i$, $T_i v(\mathbf{x}+b_j e_j+\mathbf{a}) = v(\mathbf{x}+b_j e_j+\mathbf{a}-b_i e_i)$ and $T_i v(\mathbf{x}) = v(\mathbf{x}-b_i e_i)$. As we assume $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + b_j e_j + \mathbf{a} - b_i e_i) - v(\mathbf{x} + \mathbf{a} - b_i e_i) \\ &\geq v(\mathbf{x} + b_j e_j - b_i e_i) - v(\mathbf{x} - b_i e_i) \\ &\geq T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x}) \end{aligned}$$

  (2) Suppose that $x_i \geq b_i$, $T_i v(\mathbf{x}+b_j e_j+\mathbf{a}) = v(\mathbf{x}+b_j e_j+\mathbf{a})+c_i$ and $T_i v(\mathbf{x}) = v(\mathbf{x}-b_i e_i)$. As we assume $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + b_j e_j + \mathbf{a}) + c_i - v(\mathbf{x} + \mathbf{a} - b_i e_i) \\ &\geq v(\mathbf{x} + b_j e_j) + v(\mathbf{x} + \mathbf{a}) \\ &\quad -v(\mathbf{x}) + c_i - v(\mathbf{x} + \mathbf{a} - b_i e_i) \\ &\geq v(\mathbf{x} + b_j e_j) + c_i - v(\mathbf{x} - b_i e_i) \\ &\geq T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x}) \end{aligned}$$

  (3) Suppose that $x_i + a_i \geq b_i$ if $i \neq j$, $T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) = v(\mathbf{x} + b_j e_j + \mathbf{a} - b_i e_i)$ and

$T_i v(\mathbf{x}) = v(\mathbf{x}) + c_i$. If $i = j$, it is easy to verify that

$$\begin{aligned} T_i v(\mathbf{x} + b_i e_i + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + \mathbf{a}) - v(\mathbf{x} + \mathbf{a}) - c_i \\ &= v(\mathbf{x}) - v(\mathbf{x}) - c_i \\ &\geq T_i v(\mathbf{x} + b_i e_i) - T_i v(\mathbf{x}) \end{aligned}$$

If $i \neq j$, as we assume $v \in Sub(\mathbf{b})$ and $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + b_j e_j + \mathbf{a} - b_i e_i) - v(\mathbf{x} + \mathbf{a} - b_i e_i) \\ &\geq v(\mathbf{x} + b_j e_j + \mathbf{a}) - v(\mathbf{x} + \mathbf{a}) \\ &\geq v(\mathbf{x} + b_j e_j) + c_i - v(\mathbf{x}) - c_i \\ &\geq T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x}) \end{aligned}$$

(4) Suppose that $T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) = v(\mathbf{x} + b_j e_j + \mathbf{a}) + c_i$ and $T_i v(\mathbf{x}) = v(\mathbf{x}) + c_i$. As we assume $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_i v(\mathbf{x} + b_j e_j + \mathbf{a}) - T_i v(\mathbf{x} + \mathbf{a}) &\geq v(\mathbf{x} + b_j e_j + \mathbf{a}) + c_i - v(\mathbf{x} + \mathbf{a}) - c_i \\ &\geq v(\mathbf{x} + b_j e_j) + c_i - v(\mathbf{x}) - c_i \\ &\geq T_i v(\mathbf{x} + b_j e_j) - T_i v(\mathbf{x}) \end{aligned}$$

Hence our inequality holds in all the possible scenarios. Therefore, $T_i v \in Super(\mathbf{a}, \mathbf{b})$.

Case II: Suppose that $i = n + 1$.

- First we show $T_{n+1} v \in Sub(\mathbf{b})$, i.e., $T_{n+1} v(\mathbf{x} + b_j e_j) - T_{n+1} v(\mathbf{x}) \geq T_{n+1} v(\mathbf{x} + b_j e_j + b_k e_k) - T_{n+1} v(\mathbf{x} + b_k e_k), \forall k \neq j$. Pick arbitrary $k, j \in \{1, 2, .., n\}$. There are four possible scenarios depending on the optimal actions at $T_{n+1} v(\mathbf{x} + b_j e_j)$ and $T_{n+1} v(\mathbf{x} + b_k e_k)$:

  (1) Suppose that $T_{n+1} v(\mathbf{x} + b_j e_j) = v(\mathbf{x} + b_j e_j) + c_{n+1}$ and $T_{n+1} v(\mathbf{x} + b_k e_k) = v(\mathbf{x} +$

$b_k e_k) + c_{n+1}$. As we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j) + c_{n+1} - v(\mathbf{x}) - c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j+b_ke_k) + c_{n+1} - v(\mathbf{x}+b_ke_k) - c_{n+1} \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j+b_ke_k) - T_{n+1}v(\mathbf{x}+b_ke_k) \end{aligned}$$

(2) Suppose that $\mathbf{x}+b_ke_k \geq \mathbf{a}$, $T_{n+1}v(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j)+c_{n+1}$ and $T_{n+1}v(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k-\mathbf{a})$. As we assume $v \in Sub(\mathbf{b})$ and $v \in Super(\mathbf{a},\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j) + c_{n+1} - v(\mathbf{x}) - c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j+b_ke_k) - v(\mathbf{x}+b_ke_k) \\ &\geq v(\mathbf{x}+b_je_j+b_ke_k-\mathbf{a}) - v(\mathbf{x}+b_ke_k-\mathbf{a}) \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j+b_ke_k) - T_{n+1}v(\mathbf{x}+b_ke_k) \end{aligned}$$

(3) Suppose that $\mathbf{x}+b_je_j \geq \mathbf{a}$, $T_{n+1}v(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j-\mathbf{a})$ and $T_{n+1}v(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k)+c_{n+1}$. As we assume $v \in Super(\mathbf{a},\mathbf{b})$ and $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j-\mathbf{a}) - v(\mathbf{x}) - c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j) - v(\mathbf{x}+b_je_j+b_ke_k) \\ &\quad +v(\mathbf{x}+b_je_j+b_ke_k-\mathbf{a}) - v(\mathbf{x}) - c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j+b_ke_k-\mathbf{a}) - v(\mathbf{x}+b_ke_k) - c_{n+1} \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j+b_ke_k) - T_{n+1}v(\mathbf{x}+b_ke_k) \end{aligned}$$

(4) Suppose that $\mathbf{x}+b_je_j \geq \mathbf{a}$, $\mathbf{x}+b_ke_k \geq \mathbf{a}$, $T_{n+1}v(\mathbf{x}+b_je_j) = v(\mathbf{x}+b_je_j-\mathbf{a})$ and $T_{n+1}v(\mathbf{x}+b_ke_k) = v(\mathbf{x}+b_ke_k-\mathbf{a})$. Notice that, for $j \neq k$, $\mathbf{x}+b_je_j \geq \mathbf{a}$ and $\mathbf{x}+b_ke_k \geq \mathbf{a}$ imply, respectively, $x_t \geq a_t$ for all $t \neq j$ and $x_t \geq a_t$ for all $t \neq k$, and

therefore $\mathbf{x} \geq \mathbf{a}$. As we assume $v \in Sub(\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) &\geq v(\mathbf{x}+b_je_j-\mathbf{a}) - v(\mathbf{x}-\mathbf{a}) \\ &\geq v(\mathbf{x}+b_je_j+b_ke_k-\mathbf{a}) - v(\mathbf{x}+b_ke_k-\mathbf{a}) \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j+b_ke_k) - T_{n+1}v(\mathbf{x}+b_ke_k) \end{aligned}$$

Hence our inequality holds in all the possible scenarios. Therefore, $T_{n+1}v \in Sub(\mathbf{b})$.

- Next we show $T_{n+1}v \in Super(\mathbf{a}, \mathbf{b})$, i.e., $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) - T_{n+1}v(\mathbf{x}+\mathbf{a}) \geq T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x})$, $\forall j$. Again, there are four different scenarios depending on the optimal actions at $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a})$ and $T_{n+1}v(\mathbf{x})$:

  (1) Suppose that $\mathbf{x} \geq \mathbf{a}$, $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) = v(\mathbf{x}+b_je_j)$ and $T_{n+1}v(\mathbf{x}) = v(\mathbf{x}-\mathbf{a})$. As we assume $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) - T_{n+1}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+b_je_j) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+b_je_j-\mathbf{a}) - v(\mathbf{x}-\mathbf{a}) \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) \end{aligned}$$

  (2) Suppose that $\mathbf{x} \geq \mathbf{a}$, $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) = v(\mathbf{x}+b_je_j+\mathbf{a}) + c_{n+1}$ and $T_{n+1}v(\mathbf{x}) = v(\mathbf{x}-\mathbf{a})$. As we assume $v \in Super(\mathbf{a}, \mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) - T_{n+1}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+b_je_j+\mathbf{a}) + c_{n+1} - v(\mathbf{x}+\mathbf{a}) - c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j) - v(\mathbf{x}) \\ &\geq v(\mathbf{x}+b_je_j-\mathbf{a}) - v(\mathbf{x}-\mathbf{a}) \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j) - T_{n+1}v(\mathbf{x}) \end{aligned}$$

  (3) Suppose that $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) = v(\mathbf{x}+b_je_j)$ and $T_{n+1}v(\mathbf{x}) = v(\mathbf{x}) + c_{n+1}$. Then

it is easy to verify that

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a})-T_{n+1}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+b_je_j)-v(\mathbf{x}) \\ &= v(\mathbf{x}+b_je_j)+c_{n+1}-v(\mathbf{x})-c_{n+1} \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j)-T_{n+1}v(\mathbf{x}) \end{aligned}$$

(4) Suppose that $T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a}) = v(\mathbf{x}+b_je_j+\mathbf{a})+c_{n+1}$ and $T_{n+1}v(\mathbf{x}) = v(\mathbf{x})+c_{n+1}$. As we assume $v \in Super(\mathbf{a},\mathbf{b})$, the following inequalities hold:

$$\begin{aligned} T_{n+1}v(\mathbf{x}+b_je_j+\mathbf{a})-T_{n+1}v(\mathbf{x}+\mathbf{a}) &\geq v(\mathbf{x}+b_je_j+\mathbf{a})+c_{n+1}-v(\mathbf{x}+\mathbf{a})-c_{n+1} \\ &\geq v(\mathbf{x}+b_je_j)+c_{n+1}-v(\mathbf{x})-c_{n+1} \\ &\geq T_{n+1}v(\mathbf{x}+b_je_j)-T_{n+1}v(\mathbf{x}) \end{aligned}$$

Hence our inequality holds in all the possible scenarios. Therefore, $T_{n+1}v \in Super(\mathbf{a},\mathbf{b})$.

(c) We below show $h \in Sub(\mathbf{p}) \cap Super(\mathbf{r},\mathbf{p})$, for any $\mathbf{r}$ and $\mathbf{p}$.

- First we prove $h \in Sub(\mathbf{p})$ (i.e., $h(\mathbf{x}+p_je_j)-h(\mathbf{x}) \geq h(\mathbf{x}+p_je_j+p_ke_k)-h(\mathbf{x}+p_ke_k)$, $\forall k \neq j$): $h(\mathbf{x}+p_je_j)-h(\mathbf{x}) = \sum_{i\neq j} h_i(x_i)+h_j(x_j+p_j)-\sum_{i\neq j} h_i(x_i)-h_j(x_j) = h_j(x_j+p_j)-h_j(x_j) = \sum_{i\notin\{j,k\}} h_i(x_i)+h_j(x_j+p_j)+h_k(x_k+p_k)-\sum_{i\notin\{j,k\}} h_i(x_i)-h_j(x_j)-h_k(x_k+p_k) = h(\mathbf{x}+p_je_j+p_ke_k)-h(\mathbf{x}+p_ke_k)$, $\forall k \neq j$.
- Second we prove $h \in Super(\mathbf{r},\mathbf{p})$ (i.e., $h(\mathbf{x}+p_je_j+\mathbf{r})-h(\mathbf{x}+\mathbf{r}) \geq h(\mathbf{x}+p_je_j)-h(\mathbf{x})$, $\forall j$): $h(\mathbf{x}+p_je_j+\mathbf{r})-h(\mathbf{x}+\mathbf{r}) = \sum_{i\neq j} h_i(x_i+r_i)+h_j(x_j+p_j+r_j)-\sum_{i\neq j} h_i(x_i+r_i)-h_j(x_j+r_j) = h_j(x_j+p_j+r_j)-h_j(x_j+r_j) \geq h_j(x_j+p_j)-h_j(x_j) = \sum_{i\neq j} h_i(x_i)+h_j(x_j+p_j)-\sum_{i\neq j} h_i(x_i)-h_j(x_j) = h(\mathbf{x}+p_je_j)-h(\mathbf{x})$, $\forall j$. The inequality above follows from the assumption that h_j is a convex function, $\forall j$.

Since $h \in Sub(\mathbf{p}) \cap Super(\mathbf{r},\mathbf{p})$, for any $\mathbf{r}$ and $\mathbf{p}$, we have $h \in Sub(\mathbf{q}) \cap Super(\mathbf{a},\mathbf{q}) \cap Sub(\mathbf{b}) \cap Super(\mathbf{a},\mathbf{b})$. □

Lemma 2.4.3 (Restated)**.** *Under Assumption 1, if $v \in V^*$, then $Tv \in V^*$, where $Tv = h(\mathbf{x})+\sum_j \mu_j T^{(j)}v(\mathbf{x})+\sum_i \lambda_i T_i v(\mathbf{x})$. Furthermore, the optimal cost function v^* is an element of V^*.*

Proof. Define V^* as the set of functions satisfying the properties of $Sub(\mathbf{b})$, $Super(\mathbf{a},\mathbf{b})$, and $nSuper(\mathbf{a},\mathbf{b})$. Also, define the operator T on the set of real-valued functions v: $Tv(\mathbf{x}) = h(\mathbf{x}) + \sum_j \mu_j T^{(j)} v(\mathbf{x}) + \sum_i \lambda_i T_i v(\mathbf{x})$. First we show $T : V^* \to V^*$. By Lemma 2.4.1, $Super(\mathbf{r},\mathbf{p}) \subseteq nSuper(\mathbf{r},\mathbf{p})$, and therefore $Sub(\mathbf{p}) \cap Super(\mathbf{r},\mathbf{p}) \subseteq nSuper(\mathbf{r},\mathbf{p})$. This, combined with Lemma 2.4.2, yields $T^{(j)} : Sub(\mathbf{q}) \cap Super(\mathbf{a},\mathbf{q}) \cap nSuper(\mathbf{a},\mathbf{q}) \to Sub(\mathbf{q}) \cap Super(\mathbf{a},\mathbf{q}) \cap nSuper(\mathbf{a},\mathbf{q})$, and $T_i : Sub(\mathbf{b}) \cap Super(\mathbf{a},\mathbf{b}) \cap nSuper(\mathbf{a},\mathbf{b}) \to Sub(\mathbf{b}) \cap Super(\mathbf{a},\mathbf{b}) \cap nSuper(\mathbf{a},\mathbf{b})$. By Assumption 1, $\mathbf{q} = \mathbf{b}$; and therefore $T^{(j)}, T_i : Sub(\mathbf{b}) \cap Super(\mathbf{a},\mathbf{b}) \cap nSuper(\mathbf{a},\mathbf{b}) \to Sub(\mathbf{b}) \cap Super(\mathbf{a},\mathbf{b}) \cap nSuper(\mathbf{a},\mathbf{b})$. That is, $T^{(j)} : V^* \to V^*$ and $T_i : V^* \to V^*$. By Lemmas 2.4.1 and 2.4.2, we also know $h \in V^*$. Now let $v \in V^*$. Since $T^{(j)} v \in V^*$, $T_i v \in V^*$, and $h \in V^*$, and our second-order properties are preserved by linear transformations, $Tv \in V^*$. Hence, $T : V^* \to V^*$. Following Propositions 3.1.5 and 3.1.6 in Bertsekas (2007), we verify that $\lim_{k\to\infty}(T^k v_0)(\mathbf{x}) = v^*(\mathbf{x})$ where v_0 is the zero function, v^* is the optimal cost function, and T^k refers to k compositions of operator T. Since $v_0 \in V^*$ and $T : V^* \to V^*$, we have $T^k v_0 \in V^*$, and therefore $v^* \in V^*$. □

A.3 Proofs of the Results in Section 2.4.3

Theorem 2.4.1 (Restated)**.** *Under Assumption 2.4.1, there exists an optimal stationary policy that can be specified as follows.*

(1) *The optimal inventory replenishment policy for each component j is a lattice-dependent base-stock policy with lattice-dependent base-stock levels $S_j^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p},\mathbf{a})$, $\forall \mathbf{p}$: It is optimal to produce a batch of component j if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p},\mathbf{a})$ is less than $S_j^*(\mathbf{p})$.*

(2) *The optimal inventory allocation policy for each product $i \le n$ is a lattice-dependent rationing policy with lattice-dependent rationing levels $R_i^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p},\mathbf{a})$, $\forall \mathbf{p}$: It is optimal to fulfill a demand for product $i \le n$ if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p},\mathbf{a})$ is greater than or equal to $R_i^*(\mathbf{p})$.*

(3) *The optimal inventory allocation policy for product $n+1$ is a lattice-dependent rationing policy with lattice-dependent rationing levels $R_{n+1}^*(\mathbf{p}) \in \mathbb{L}(\mathbf{p},\mathbf{b})$, $\forall \mathbf{p}$: It is optimal to fulfill a demand for product $n+1$ if and only if $\mathbf{x} \in \mathbb{L}(\mathbf{p},\mathbf{b})$ is greater than or equal to*

$R^*_{n+1}(\mathbf{p})$.

The optimal policy has the following additional properties:

i. As the system moves to a difference lattice with an increment of b_k in the inventory level of component k, both the optimal base-stock level of component $j \neq k$ and the optimal rationing level for (individual) product $i \notin \{k, n+1\}$ increase in a non-strict sense, $\forall k$.

ii. As the system moves to a difference lattice with an increment of b_k in the inventory level of component k, the optimal rationing level for (master) product $n+1$ decreases in a non-strict sense, $\forall k$.

iii. It is optimal to fulfill a demand of (master) product $n+1$ if $x_j \geq a_j + b_j \left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$.

Proof. By Lemma 2.4.3, we know $v^* \in V^*$. Define, for $v^* \in V^*$,

$$S^*_j(\mathbf{p}) = \min\{\mathbf{p} + z\mathbf{a} : v^*(\mathbf{p} + z\mathbf{a} + q_j e_j) - v^*(\mathbf{p} + z\mathbf{a}) > 0,\ z \in \mathbb{N}_0\},\ \forall j,$$

$$R^*_i(\mathbf{p}) = \min\{\mathbf{p} + z\mathbf{a} : v^*(\mathbf{p} + z\mathbf{a}) - v^*(\mathbf{p} + z\mathbf{a} - b_i e_i) > -c_i,\ z \in \mathbb{N}_0, p_i + z a_i \geq b_i\}, \forall i \leq n,$$

$$R^*_{n+1}(\mathbf{p}) = \min\{\mathbf{p} + z\mathbf{b} : v^*(\mathbf{p} + z\mathbf{b}) - v^*(\mathbf{p} + z\mathbf{b} - \mathbf{a}) > -c_{n+1},\ z \in \mathbb{N}_0,\ \mathbf{p} + z\mathbf{b} \geq \mathbf{a}\}.$$

(1) Since $v^* \in Super(\mathbf{a}, \mathbf{b})$ and $\mathbf{q} = \mathbf{b}$, $v^*(\mathbf{p} + z\mathbf{a} + q_j e_j) - v^*(\mathbf{p} + z\mathbf{a})$ is increasing in z. As z increases, since the holding cost rate h is strictly increasing, this difference will eventually cross 0. Therefore, the lattice-dependent base-stock policy is optimal.

(2) Since $v^* \in Super(\mathbf{a}, \mathbf{b})$, $v^*(\mathbf{p} + z\mathbf{a}) - v^*(\mathbf{p} + z\mathbf{a} - b_i e_i)$ is increasing in z. We know that, as z increases, this difference will eventually cross 0. Therefore, as z increases, this difference should also cross $-c_i$. Hence, the lattice-dependent rationing policy is optimal.

(3) Since $v^* \in nSuper(\mathbf{a}, \mathbf{b})$, $v^*(\mathbf{p} + z\mathbf{b}) - v^*(\mathbf{p} + z\mathbf{b} - \mathbf{a})$ is increasing in z. As z increases, since the holding cost rate h is strictly increasing, this difference will eventually cross $-c_{n+1}$. Therefore, the lattice-dependent rationing policy is optimal.

Next we will prove properties (i)-(iii):

i. To prove property (i), first, we show that the optimal base-stock levels for each component j obey $S_j^*(\mathbf{p}+b_k e_k) \geq S_j^*(\mathbf{p})+b_k e_k$, $\forall k \neq j$. Let $S_j^*(\mathbf{p}) = \mathbf{p}+z_1\mathbf{a}$ and $S_j^*(\mathbf{p}+b_k e_k) = \mathbf{p}+b_k e_k+z_2\mathbf{a}$. Then, it is not optimal to produce a batch of component j at $\mathbf{x}=\mathbf{p}+z_1\mathbf{a}$ and $\mathbf{x}=\mathbf{p}+b_k e_k+z_2\mathbf{a}$. Since $v^* \in Sub(\mathbf{b})$, it is not optimal to produce a batch of component j at $\mathbf{x}=\mathbf{p}+z_2\mathbf{a}$, implying $z_2 \geq z_1$. Therefore, we must have $S_j^*(\mathbf{p}+b_k e_k) \geq S_j^*(\mathbf{p})+b_k e_k$.

   Second, we show that the optimal rationing levels for each product $i \leq n$ obey $R_i^*(\mathbf{p}+b_k e_k) \geq R_i^*(\mathbf{p})+b_k e_k$, $\forall k \neq i$. Let $R_i^*(\mathbf{p}) = \mathbf{p}+z_1\mathbf{a}$ and $R_i^*(\mathbf{p}+b_k e_k) = \mathbf{p}+b_k e_k+z_2\mathbf{a}$. Then, it is optimal to fulfill a demand for product i at $\mathbf{x}=\mathbf{p}+z_1\mathbf{a}$ and $\mathbf{x}=\mathbf{p}+b_k e_k+z_2\mathbf{a}$. Since $v^* \in Sub(\mathbf{b})$, it is also optimal to fulfill a demand for product i at $\mathbf{x}=\mathbf{p}+z_2\mathbf{a}$, implying $z_2 \geq z_1$. Therefore, we must have $R_i^*(\mathbf{p}+b_k e_k) \geq R_i^*(\mathbf{p})+b_k e_k$.

ii. To prove (ii), we will show that the optimal rationing levels for product $n+1$ obey $R_{n+1}^*(\mathbf{p}+b_k e_k) \leq R_{n+1}^*(\mathbf{p})+b_k e_k$, $\forall k$. Let $R_{n+1}^*(\mathbf{p}) = \mathbf{p}+z_1\mathbf{b}$ and $R_{n+1}^*(\mathbf{p}+b_k e_k) = \mathbf{p}+b_k e_k+z_2\mathbf{b}$. Then, it is optimal to fulfill a demand for product $n+1$ at $\mathbf{x}=\mathbf{p}+z_1\mathbf{b}$ and $\mathbf{x}=\mathbf{p}+b_k e_k+z_2\mathbf{b}$. Since $v^* \in Super(\mathbf{a},\mathbf{b})$, it is also optimal to fulfill a demand for product $n+1$ at $\mathbf{x}=\mathbf{p}+z_1\mathbf{b}+b_k e_k$, implying $z_1 \geq z_2$. Therefore, we must have $R_{n+1}^*(\mathbf{p})+b_k e_k \geq R_{n+1}^*(\mathbf{p}+b_k e_k)$.

iii. Lastly, we will prove that it is optimal to fulfill a demand of product $n+1$ if $x_j \geq a_j + b_j\left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$. Define $\widetilde{V}$ as the set of real-valued functions f defined on $\mathbb{N}_0^n$ such that $f(\mathbf{x}) - f(\mathbf{x}-\mathbf{a}) + c_{n+1} \geq 0$, for $x_j \geq a_j + b_j\left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$. Recall $Tv(\mathbf{x}) = h(\mathbf{x}) + \sum_j \mu_j T^{(j)}v(\mathbf{x}) + \sum_i \lambda_i T_i v(\mathbf{x})$. We show below $T : \widetilde{V} \to \widetilde{V}$.

   Assume that $v \in \widetilde{V}$. We want to prove $Tv \in \widetilde{V}$. Since h is an increasing convex function and $\sum_j \mu_j + \sum_i \lambda_i \le 1$, the following inequality holds:

$$\begin{aligned} Tv(\mathbf{x}) - Tv(\mathbf{x}-\mathbf{a}) + c_{n+1} &= h(\mathbf{x}) - h(\mathbf{x}-\mathbf{a}) + \sum_j \mu_j (T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a})) \\ &\quad + \sum_i \lambda_i (T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a})) + c_{n+1} \\ &\ge \sum_j \mu_j (T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a}) + c_{n+1}) \\ &\quad + \sum_i \lambda_i (T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a}) + c_{n+1}) \end{aligned}$$

To prove $Tv \in \widetilde{V}$, it suffices to show $T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge 0$, $\forall j$, and $T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge 0$, $\forall i$, where $x_k \ge a_k + b_k \left\lfloor \frac{x_k}{b_k} \right\rfloor$, $\forall k$. We prove these inequalities as follows:

- First we show $T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge 0$. There are two possible scenarios depending on the optimal action at $T^{(j)}v(\mathbf{x})$:
  (1) Suppose that $T^{(j)}v(\mathbf{x}) = v(\mathbf{x}+q_j e_j) < v(\mathbf{x})$: $T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge v(\mathbf{x}+q_j e_j) - v(\mathbf{x}+q_j e_j - \mathbf{a}) + c_{n+1} \ge 0$. The second inequality follows from the fact that $v \in \widetilde{V}$ and $x_j + q_j \ge a_j + b_j \left\lfloor \frac{x_j}{b_j} \right\rfloor + q_j = a_j + b_j \left\lfloor \frac{x_j+q_j}{b_j} \right\rfloor$. (By Assumption 1, $q_j = b_j$.)

  (2) Suppose that $T^{(j)}v(\mathbf{x}) = v(\mathbf{x}) \le v(\mathbf{x}+q_j e_j)$: $T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge v(\mathbf{x}) - v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge 0$. The second inequality follows from the assumption of $v \in \widetilde{V}$.

- Second we show $T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge 0$, for $i \le n$. There are two possible scenarios depending on the optimal action at $T_i v(\mathbf{x})$:
  (1) Suppose that $T_i v(\mathbf{x}) = v(\mathbf{x}) + c_i$: $T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge v(\mathbf{x}) + c_i - v(\mathbf{x}-\mathbf{a}) - c_i + c_{n+1} \ge 0$. The second inequality follows from the assumption of $v \in \widetilde{V}$.

  (2) Suppose that $x_i \ge b_i$ and $T_i v(\mathbf{x}) = v(\mathbf{x} - b_i e_i)$: $T_i v(\mathbf{x}) - T_i v(\mathbf{x}-\mathbf{a}) + c_{n+1} \ge v(\mathbf{x} - b_i e_i) - v(\mathbf{x} - \mathbf{a} - b_i e_i) + c_{n+1} \ge 0$. The second inequality follows from the fact that $v \in \widetilde{V}$ and $x_i - b_i \ge a_i + b_i \left\lfloor \frac{x_i}{b_i} \right\rfloor - b_i = a_i + b_i \left\lfloor \frac{x_i - b_i}{b_i} \right\rfloor$. Here notice that, as we assume $x_i \ge a_i + b_i \left\lfloor \frac{x_i}{b_i} \right\rfloor$ and $x_i \ge b_i$, we should have $x_i \ge a_i + b_i$,

implying $\mathbf{x} \geq \mathbf{a} + b_i e_i$.

- Lastly we show $T_{n+1}v(\mathbf{x}) - T_{n+1}v(\mathbf{x}-\mathbf{a}) + c_{n+1} \geq 0$. There are two possible scenarios depending on the optimal action at $T_{n+1}v(\mathbf{x})$:
  (1) Suppose that $T_{n+1}v(\mathbf{x}) = v(\mathbf{x}) + c_{n+1} < v(\mathbf{x} - \mathbf{a})$: $T_{n+1}v(\mathbf{x}) - T_{n+1}v(\mathbf{x} - \mathbf{a}) + c_{n+1} \geq v(\mathbf{x}) + c_{n+1} - v(\mathbf{x} - \mathbf{a}) - c_{n+1} + c_{n+1} \geq 0$. The second inequality follows from the assumption of $v \in \widetilde{V}$.
  (2) Suppose that $T_{n+1}v(\mathbf{x}) = v(\mathbf{x} - \mathbf{a}) \leq v(\mathbf{x}) + c_{n+1}$: $T_{n+1}v(\mathbf{x}) - T_{n+1}v(\mathbf{x} - \mathbf{a}) + c_{n+1} \geq v(\mathbf{x} - \mathbf{a}) - v(\mathbf{x} - \mathbf{a}) - c_{n+1} + c_{n+1} = 0$.

Since $\sum_j \mu_j (T^{(j)}v(\mathbf{x}) - T^{(j)}v(\mathbf{x} - \mathbf{a}) + c_{n+1}) + \sum_i \lambda_i (T_i v(\mathbf{x}) - T_i v(\mathbf{x} - \mathbf{a}) + c_{n+1}) \geq 0$, we have $Tv(\mathbf{x}) - Tv(\mathbf{x} - \mathbf{a}_j) + c_j \geq 0$. Hence, $T : \widetilde{V} \to \widetilde{V}$. Following Propositions 3.1.5 and 3.1.6 in Bertsekas (2007), we verify that $\lim_{k\to\infty}(T^k v_0)(\mathbf{x}) = v^*(\mathbf{x})$ where v_0 is the zero function, v^* is the optimal cost function, and T^k refers to k compositions of operator T. Since $v_0 \in \widetilde{V}$ and $T : \widetilde{V} \to \widetilde{V}$, we have $T^k v_0 \in \widetilde{V}$, and therefore $v^* \in \widetilde{V}$. Since $v^*(\mathbf{x}) - v^*(\mathbf{x} - \mathbf{a}) + c_{n+1} \geq 0$, for $x_j \geq a_j + b_j \left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$, it is optimal to fulfill a demand of product $n+1$ if $x_j \geq a_j + b_j \left\lfloor \frac{x_j}{b_j} \right\rfloor$, $\forall j$.

□

A.4 Proofs of the Results in Section 2.4.4

Proposition 2.4.1 (Restated)**.** *Suppose that Assumption 1 holds and the Markov chain governing the system is irreducible. There exists a stationary policy that is optimal under the average cost criterion. The policy retains all the properties of the optimal policy under the discounted cost criterion, as introduced in Theorem 1. Also, the optimal average cost is finite and independent of the initial state; there exists a finite constant v^* such that $v^*(\mathbf{x}) = v^*$, $\forall \mathbf{x}$.*

Proof. We first prove the following conditions: (i) There exists a stationary policy π that induces an irreducible positive recurrent Markov chain with finite average cost v^π, and (ii) the number of states for which $h(\mathbf{x}) \leq v^\pi$ is finite. To prove condition (i), consider a policy where the production of each component is controlled by a base-stock policy with an independent and fixed critical level, and inventory allocation follows a first-come-first-served policy. Notice that we have a finite-state Markov chain under this policy. Hence, this policy yields a finite average cost. It is easy to prove condition (ii) as the inventory holding cost rate for each component is increasing convex in its inventory level. Thus, for any positive value γ, the number of states for which $h(\mathbf{x}) \leq \gamma$ is always finite. Under conditions (i) and (ii), there exists a constant v^* and a function $f(\mathbf{x})$ such that $f(\mathbf{x}) + v^* = \inf\{h(\mathbf{x}) + \sum_j \mu_j T^{(j)} f(\mathbf{x}) + \sum_i \lambda_i T_i f(\mathbf{x})\}$ (Weber and Stidham 1987). The stationary policy that minimizes the righthand side of the above equation for each state $\mathbf{x}$ is an optimal policy for the average cost criterion and yields a constant average cost v^*. Hence, properties of the optimal policy for the average cost are determined through the function $f(\mathbf{x})$. Recall that properties of the optimal policy for the discounted costs are determined through $v^*(\mathbf{x})$. Since the same event operators are applied to $f(\mathbf{x})$, the optimal policy for the average cost retains the same structure as in the discounted cost case. $\square$

Appendix B

Supplement to Performance Evaluation of Lattice-Dependent Policies for ATO Systems

This chapter includes supplementary material for Chapter 3: Performance Evaluation of Lattice-Dependent Base-Stock and Rationing Policies for ATO Systems.

B.1 Proof of Proposition 3.2.1

Proposition 3.2.1 (Restated). $Z^* \leq Z_{LBLR} \leq Z_{SBSR} \leq Z_{FBFR}$

Proof. It is immediate that the first inequality holds since $\mathcal{LP}$ is a relaxation of all other MIP problems. It is also easy to verify that the third inequality holds since FBFR is a subclass of SBSR; SBSR becomes FBFR if all base-stock and rationing levels are constant across system states. To prove the second inequality, we will show that SBSR is a subclass of LBLR.

Recall that the difference vector of lattices for component i is defined as $\mathbf{\Delta}^i = (\Delta_1^i, \Delta_2^i, .., \Delta_m^i)$ where $\Delta_k^i \in \mathbb{N}_0$, $\forall i, k$, and the difference vector of lattices for product j is defined as $\mathbf{\Delta}_j = (\Delta_{j1}, \Delta_{j2}, .., \Delta_{jm})$ where $\Delta_{ji} \in \mathbb{N}_0$, $\forall i, j$. Now we choose any specific $\mathbf{\Delta}^i$ such that $\Delta_i^i \geq \sum_{k \neq i} \Delta_k^i$, $\forall i$ (recall LBLR chooses the optimal $\mathbf{\Delta}^i$). We then consider the inventory replenishment decisions for component i under an LBLR policy. The only constraint on these decisions is that, if a batch of component i is not produced at inventory level $\mathbf{x}$, then it is not produced at inventory level $\mathbf{x} + \mathbf{\Delta}^i$. But this is also true under an SBSR policy: If a batch of component i is not produced at inventory level $\mathbf{x}$, then the base-stock level of component i is less than x_i at inventory level $\mathbf{x}$. We know from property (b) of SBSR that, if the inventory level of component $k \neq i$ increases by Δ^i_k, $\forall k$, then the base-stock level of component i increases by at most $\sum_{k \neq i} \Delta^i_k$ units. Consequently, the base-stock level of component i is less than $x_i + \sum_{k \neq i} \Delta^i_k$ at inventory level $\mathbf{x} + \boldsymbol{\Delta}^i$. As we assume $\Delta^i_i \geq \sum_{k \neq i} \Delta^i_k$, a batch of component i is not produced at inventory level $\mathbf{x} + \boldsymbol{\Delta}^i$.

We next consider the inventory allocation decisions for product j under an LBLR policy. The only constraint on these decisions is that, if a demand for product j is satisfied at inventory level $\mathbf{x}$, then it is satisfied at inventory level $\mathbf{x} + \boldsymbol{\Delta}_j$. Again this is also true under an SBSR policy: It is immediate from property (c) that if a demand for product j is satisfied at inventory level $\mathbf{x}$, then it is also satisfied at inventory level $\mathbf{y} \geq \mathbf{x}$. Hence, SBSR is a subclass of LBLR, and the second inequality holds. □

B.2 Additional Numerical Results for Nested Structure

Tables B.1 – B.4 exhibit our numerical results for Examples (b) and (c) in Section 3.3.1.

Table B.1 Numerical results for Example (b).

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_A	λ_B	Average cost	f_A	f_B	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	31.974	0.508	0.792	0.000	1.742	2.687	1.76	335.07	4.87
-	1.0	55.396	0.224	0.730	0.000	1.589	2.413	2.27	**1000**	11.02
-	1.5	89.495	0.013	0.589	0.000	0.098	1.891	2.58	18.67	3.04
-	2.0	133.028	0.000	0.453	0.000	0.075	1.532	2.69	6.54	2.77
-	2.5	180.194	0.000	0.364	0.000	0.046	1.004	2.43	8.19	3.41
1.0	0.5	39.059	0.457	0.782	0.000	1.526	2.898	1.89	84.08	5.11
-	1.0	64.497	0.172	0.719	0.000	1.504	3.295	2.40	**1000**	8.02
-	1.5	99.455	0.010	0.588	0.000	0.129	1.739	2.36	9.29	3.99
-	2.0	143.028	0.000	0.453	0.000	0.070	1.425	2.91	5.12	2.27
-	2.5	190.194	0.000	0.364	0.000	0.044	0.952	2.16	3.60	2.35
1.5	0.5	47.243	0.373	0.756	0.000	1.410	2.961	2.24	**1000**	8.56
-	1.0	74.046	0.137	0.711	0.000	1.225	2.928	2.64	**1000**	9.15
-	1.5	109.428	0.008	0.588	0.000	0.142	1.607	2.67	55.06	4.43
-	2.0	153.028	0.000	0.453	0.000	0.065	1.332	2.38	3.81	2.31
-	2.5	200.194	0.000	0.364	0.000	0.041	0.904	2.60	5.06	5.88
2.0	0.5	56.289	0.300	0.736	0.000	1.014	2.620	2.51	67.22	8.70
-	1.0	83.797	0.108	0.709	0.000	0.967	2.593	2.31	**1000**	10.90
-	1.5	119.407	0.012	0.585	0.000	0.147	1.489	2.43	251.57	5.42
-	2.0	163.028	0.000	0.453	0.000	0.061	1.250	2.29	7.27	2.59
-	2.5	210.194	0.000	0.364	0.000	0.039	0.861	3.11	2.21	4.15
2.5	0.5	65.748	0.235	0.765	0.000	0.794	2.313	2.15	426.50	7.08
-	1.0	93.644	0.088	0.715	0.000	0.805	2.277	2.42	**1000**	10.29
-	1.5	129.391	0.010	0.584	0.000	0.148	1.387	2.45	74.47	5.78
-	2.0	173.028	0.000	0.453	0.000	0.058	1.178	2.72	4.20	2.33
-	2.5	220.194	0.000	0.364	0.000	0.038	0.822	2.35	4.28	2.68

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

Table B.2 Numerical results for Example (b).

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_A/c_B	Average cost	f_A	f_B	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	48.715	0.096	0.789	0.000	1.271	1.490	2.65	**1000**	5.36
-	-	0.4	65.892	0.186	0.759	0.000	0.645	0.827	2.73	**1000**	5.47
-	-	0.6	81.278	0.261	0.712	0.000	0.268	0.436	2.81	58.92	6.23
-	-	0.8	95.296	0.357	0.628	0.000	0.086	0.329	2.56	90.89	7.56
-	-	1.0	106.140	0.589	0.400	0.000	0.005	0.710	2.88	119.72	7.20
-	3	0.2	57.606	0.134	0.765	0.000	1.142	2.680	2.83	**1000**	8.56
-	-	0.4	73.881	0.231	0.718	0.000	0.854	2.030	2.45	**1000**	6.05
-	-	0.6	88.301	0.326	0.645	0.000	0.495	1.598	2.53	**1000**	5.39
-	-	0.8	100.865	0.423	0.553	0.000	0.227	1.391	2.42	3.70	6.13
-	-	1.0	110.557	0.601	0.381	0.000	0.278	2.150	2.46	42.28	6.59
-	5	0.2	64.497	0.172	0.719	0.000	1.504	3.295	2.36	**1000**	8.19
-	-	0.4	79.868	0.270	0.676	0.000	1.277	2.736	2.14	**1000**	6.26
-	-	0.6	93.359	0.361	0.607	0.000	0.551	2.184	2.16	37.68	7.21
-	-	0.8	104.877	0.466	0.506	0.000	0.205	1.970	2.40	27.39	5.30
-	-	1.0	114.194	0.580	0.397	0.000	0.118	2.740	2.08	21.94	4.06
3	1	0.2	55.704	0.093	0.758	0.000	1.075	1.635	2.49	**1000**	4.68
-	-	0.4	72.863	0.203	0.715	0.000	0.531	0.895	2.49	**1000**	5.83
-	-	0.6	87.484	0.331	0.647	0.000	0.096	0.177	2.58	170.89	4.39
-	-	0.8	100.184	0.418	0.573	0.000	0.169	0.314	2.85	75.94	8.11
-	-	1.0	110.179	0.575	0.418	0.000	0.001	0.465	2.49	23.68	3.33
-	3	0.2	64.894	0.110	0.730	0.000	1.187	2.931	2.59	**1000**	10.22
-	-	0.4	81.334	0.238	0.681	0.000	0.813	1.960	2.62	**1000**	9.62
-	-	0.6	95.115	0.362	0.611	0.000	0.273	1.294	2.43	164.92	5.09
-	-	0.8	106.805	0.457	0.526	0.000	0.138	1.339	2.63	59.24	5.27
-	-	1.0	115.950	0.587	0.399	0.000	0.172	1.750	2.64	23.93	4.90
-	5	0.2	71.610	0.116	0.686	0.000	1.396	2.828	2.24	**1000**	8.09
-	-	0.4	87.455	0.246	0.656	0.000	0.991	2.745	2.47	**1000**	10.14
-	-	0.6	100.331	0.412	0.556	0.000	0.463	2.111	2.66	38.59	5.80
-	-	0.8	111.307	0.482	0.494	0.000	0.226	2.200	2.45	12.27	5.81
-	-	1.0	120.293	0.595	0.388	0.000	0.318	2.723	2.47	21.74	4.43
5	1	0.2	60.431	0.100	0.731	0.000	0.811	1.867	1.99	613.64	4.55
-	-	0.4	77.354	0.222	0.686	0.000	0.346	0.937	2.09	598.84	5.24
-	-	0.6	91.803	0.340	0.618	0.000	0.125	0.315	2.52	84.92	4.54
-	-	0.8	104.066	0.423	0.560	0.000	0.352	0.495	2.61	152.06	9.06
-	-	1.0	113.656	0.560	0.431	0.000	0.002	0.347	2.73	10.07	5.79
-	3	0.2	69.735	0.129	0.690	0.000	1.116	2.841	2.24	**1000**	6.62
-	-	0.4	86.234	0.217	0.665	0.000	0.546	1.780	2.27	558.06	8.69
-	-	0.6	99.999	0.385	0.568	0.000	0.121	1.114	2.73	9.48	4.54
-	-	0.8	111.637	0.476	0.494	0.000	0.164	1.326	2.58	21.10	7.06
-	-	1.0	120.581	0.585	0.399	0.000	0.057	1.493	2.51	9.64	5.78
-	5	0.2	76.270	0.102	0.651	0.000	1.290	2.920	2.11	**1000**	5.85
-	-	0.4	92.500	0.279	0.591	0.000	0.879	2.435	2.13	322.30	7.16
-	-	0.6	105.490	0.399	0.539	0.000	0.248	1.696	2.22	17.21	4.51
-	-	0.8	116.544	0.527	0.431	0.000	0.024	2.070	2.54	6.21	6.24
-	-	1.0	125.545	0.585	0.391	0.000	0.315	2.487	2.66	11.83	8.12

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

Table B.3 Numerical results for Example (c).

		Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
λ_A	λ_B	Average cost	f_A	f_B	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
0.5	0.5	60.414	0.595	0.417	0.000	0.000	1.170	1.57	1.56	2.00
-	1.0	99.799	0.357	0.356	0.000	0.882	2.526	1.98	11.51	10.96
-	1.5	144.392	0.269	0.273	0.000	0.268	1.941	2.04	47.06	7.26
-	2.0	191.741	0.242	0.213	0.000	0.071	1.468	2.09	111.46	10.80
-	2.5	240.147	0.019	0.193	0.000	0.010	0.928	2.08	**1000**	17.45
1.0	0.5	76.209	0.488	0.294	0.000	0.000	2.010	1.73	2.14	1.93
-	1.0	118.449	0.279	0.307	0.000	0.939	2.100	1.74	51.46	5.25
-	1.5	163.994	0.213	0.245	0.000	0.354	1.641	2.18	15.49	5.74
-	2.0	211.639	0.193	0.194	0.000	0.101	1.360	2.07	551.63	4.33
-	2.5	260.130	0.016	0.192	0.000	0.014	0.863	2.10	132.75	13.52
1.5	0.5	93.828	0.382	0.212	0.000	0.000	1.674	1.59	2.78	5.47
-	1.0	137.517	0.266	0.260	0.000	0.874	1.865	1.81	19.48	9.89
-	1.5	183.736	0.166	0.233	0.000	0.386	1.422	1.84	41.09	5.07
-	2.0	231.571	0.160	0.182	0.000	0.114	1.218	2.19	535.52	8.41
-	2.5	280.116	0.014	0.191	0.000	0.016	0.806	2.13	139.29	3.00
2.0	0.5	112.533	0.313	0.160	0.000	0.000	1.348	1.90	2.42	2.36
-	1.0	156.912	0.220	0.237	0.000	0.821	1.654	2.31	12.79	12.97
-	1.5	203.538	0.160	0.212	0.000	0.401	1.262	2.07	**1000**	9.20
-	2.0	251.522	0.138	0.173	0.000	0.120	1.101	1.99	175.88	7.16
-	2.5	300.105	0.013	0.190	0.000	0.018	0.757	2.00	91.46	13.36
2.5	0.5	131.697	0.222	0.266	0.000	0.060	1.140	2.04	9.07	4.41
-	1.0	176.514	0.187	0.221	0.000	0.758	1.462	2.09	40.45	7.75
-	1.5	223.350	0.138	0.203	0.000	0.420	1.152	2.15	24.47	9.38
-	2.0	271.485	0.120	0.166	0.000	0.121	1.003	2.34	435.99	6.69
-	2.5	320.096	0.012	0.189	0.000	0.019	0.712	1.98	301.15	8.41

Notes. $q_\phi = q_\gamma = 1$, $h_\phi = h_\gamma = 9$, $\mu_\phi = \mu_\gamma = 1$, $c_A = 40$, and $c_B = 100$. Computation times equal to 1000 seconds indicate termination of the algorithm.

Table B.4 Numerical results for Example (c).

			Optimal solution			Percentage difference from optimal cost			Computation times (in seconds)		
h_ϕ	h_γ	c_A/c_B	Average cost	f_A	f_B	LBLR	SBSR	FBFR	LBLR	SBSR	FBFR
1	1	0.2	74.483	0.004	0.494	0.000	0.000	0.729	2.57	4.73	2.80
-	-	0.4	94.325	0.019	0.487	0.000	0.000	0.618	2.73	8.27	4.18
-	-	0.6	108.481	0.588	0.203	0.000	0.079	0.199	2.63	72.24	4.65
-	-	0.8	114.719	0.748	0.121	0.000	0.009	0.131	2.89	22.72	5.03
-	-	1.0	119.118	0.803	0.089	0.000	0.000	0.145	3.03	2.83	4.79
-	5	0.2	83.572	0.014	0.489	0.000	0.001	0.672	2.28	26.71	2.83
-	-	0.4	102.492	0.186	0.404	0.000	0.012	0.416	2.63	8.63	6.06
-	-	0.6	114.344	0.483	0.255	0.000	0.118	0.247	2.92	182.43	8.35
-	-	0.8	122.984	0.636	0.176	0.000	0.073	0.263	2.41	270.42	11.07
-	-	1.0	129.337	0.715	0.133	0.000	0.060	0.362	2.72	55.27	13.58
-	9	0.2	91.285	0.045	0.463	0.000	0.011	0.584	2.19	13.26	2.87
-	-	0.4	107.906	0.320	0.329	0.000	0.403	0.678	2.34	198.77	4.72
-	-	0.6	118.665	0.506	0.238	0.000	0.104	0.344	2.66	11.44	9.26
-	-	0.8	128.232	0.599	0.186	0.000	0.134	0.519	3.10	182.74	11.74
-	-	1.0	135.512	0.653	0.159	0.000	0.046	0.562	2.84	26.79	10.80
5	1	0.2	78.377	0.007	0.487	0.000	0.000	2.649	1.86	3.68	4.41
-	-	0.4	97.974	0.059	0.464	0.000	0.006	2.298	2.31	9.62	4.41
-	-	0.6	112.693	0.422	0.281	0.000	0.032	0.894	2.36	4.85	7.07
-	-	0.8	121.971	0.615	0.182	0.000	0.000	0.639	2.20	21.85	4.24
-	-	1.0	128.871	0.690	0.141	0.000	0.000	0.667	2.04	8.09	6.26
-	5	0.2	89.465	0.036	0.470	0.000	0.000	1.771	2.17	3.03	3.32
-	-	0.4	108.034	0.271	0.355	0.000	0.081	1.376	2.08	45.81	4.95
-	-	0.6	121.254	0.406	0.284	0.000	0.234	1.108	2.33	8.91	4.15
-	-	0.8	131.881	0.538	0.212	0.000	0.086	0.689	2.65	39.65	6.99
-	-	1.0	140.432	0.590	0.184	0.000	0.014	0.589	2.33	4.44	5.89
-	9	0.2	97.785	0.036	0.436	0.000	0.026	1.341	1.94	3.21	3.09
-	-	0.4	114.619	0.276	0.326	0.000	0.369	1.189	2.11	64.06	10.21
-	-	0.6	126.975	0.444	0.247	0.000	0.093	0.932	2.06	4.05	4.19
-	-	0.8	137.696	0.470	0.229	0.000	0.001	0.715	2.24	5.08	3.86
-	-	1.0	147.058	0.572	0.181	0.000	0.023	0.770	2.34	32.20	8.33
9	1	0.2	81.346	0.011	0.482	0.000	0.000	4.117	1.86	3.56	3.72
-	-	0.4	100.440	0.103	0.439	0.000	0.000	4.236	1.98	7.26	2.86
-	-	0.6	114.918	0.373	0.302	0.000	0.000	1.907	1.89	8.06	2.82
-	-	0.8	125.899	0.513	0.233	0.000	0.000	0.920	2.09	4.96	4.17
-	-	1.0	134.148	0.632	0.169	0.000	0.000	1.066	2.21	3.14	4.04
-	5	0.2	93.368	0.035	0.460	0.000	0.012	3.353	1.79	15.69	2.74
-	-	0.4	111.341	0.193	0.385	0.000	0.315	2.618	2.15	4.63	8.27
-	-	0.6	124.910	0.411	0.279	0.000	0.101	1.377	2.27	11.45	5.55
-	-	0.8	136.407	0.432	0.264	0.000	0.000	1.159	2.17	4.06	4.08
-	-	1.0	146.016	0.535	0.205	0.000	0.008	0.908	2.26	4.72	2.60
-	9	0.2	101.886	0.084	0.407	0.000	0.043	2.619	1.92	4.65	2.98
-	-	0.4	118.449	0.279	0.307	0.000	0.939	2.100	1.85	50.04	5.33
-	-	0.6	131.368	0.413	0.240	0.000	0.011	1.117	1.99	6.95	3.81
-	-	0.8	142.775	0.433	0.231	0.000	0.003	0.994	2.07	2.81	2.62
-	-	1.0	153.310	0.521	0.194	0.000	0.072	1.033	2.06	3.61	7.22

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

Appendix C

Supplement to Optimal Portfolio Strategies for New Product Development

This chapter includes supplementary material for Chapter 4: Optimal Portfolio Strategies for New Product Development.

C.1 Proofs of the Results in Section 4.3.

Lemma 4.3.1 (Restated)**.** *There is no loss of generality in assuming that $y_j \in \{0,1\}$, $\forall j$.*

Proof. We will first prove that, without loss of generality, the action space of the operator $T_{B,i}$ can be reduced to the set of binary variables. Pick arbitrary stage i. For a given state $\mathbf{x}$, let y_j^* denote the optimal rate of resources of stage i allocated to a project in category $j \in \mathcal{W}_i$. Also, define $Y^* = \sum_{j\in\mathcal{W}_i} y_j^*$ as the optimal rate of utilized resources at stage i. Therefore, $1-Y^*$ is the optimal rate of unused resources at stage i. Now suppose that $Y^* > 0$. Then the operator $T_{B,i}$ can be written as follows:

$$T_{B,i}v(\mathbf{x}) = \sum_{j\in\mathcal{W}_i}\left(c_{ij}(y_j^*) + y_j^*\sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j\to j'}\right) + (1-Y^*)\, v\left(\mathbf{x}\right),$$

We next introduce an auxiliary variable $\hat{y} \in [0,1]$. Let $\hat{y}_j^*$, $\forall j \in \mathcal{W}_i$, solve the following problem:

$$\min \sum_{j \in \mathcal{W}_i} c_{ij}(\hat{y}_j), \quad \text{subject to} \sum_{j \in \mathcal{W}_i} \hat{y}_j = Y^*.$$

Notice that, since $c_{ij}(\hat{y}_j)$ is concave in $\hat{y}_j$, there exists a category $\hat{k} \in \mathcal{W}_i$ such that $\hat{y}^*_{\hat{k}} = Y^*$ and $\hat{y}^*_j = 0$, $\forall j \neq \hat{k}$. We now introduce another auxiliary variable $\tilde{y}_j \in [0, 1]$. Let $\tilde{y}^*_j$, $\forall j \in \mathcal{W}_i$, solve the following problem:

$$\min \sum_{j \in \mathcal{W}_i} \tilde{y}_j \sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'}, \quad \text{subject to} \sum_{j \in \mathcal{W}_i} \tilde{y}_j = Y^*.$$

There exists a category $\tilde{k} \in \mathcal{W}_i$ s.t. $\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k} \to j'} \leq \sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'}$, $\forall j \in \mathcal{W}_i$. This implies that $\tilde{y}^*_{\tilde{k}} = Y^*$ and $\tilde{y}^*_j = 0$, $\forall j \neq \tilde{k}$, is an optimal solution.

We then derive the following lower bound $T^L_{B,i}$ for the operator $T_{B,i}$ (by assumption, $c_{i\hat{k}}(y) = c_{i\tilde{k}}(y)$, $\forall y \in [0, 1]$):

$$\begin{aligned}
T^L_{B,i} v(\mathbf{x}) &= \sum_{j \in \mathcal{W}_i} \left(c_{ij}(\hat{y}^*_j) + \tilde{y}^*_j \sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'} \right) + (1 - Y^*)\, v(\mathbf{x}) \\
&= c_{i\hat{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k} \to j'} + (1 - Y^*)\, v(\mathbf{x}) \\
&= c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k} \to j'} + (1 - Y^*)\, v(\mathbf{x}) \\
&\leq \sum_{j \in \mathcal{W}_i} \left(c_{ij}(y^*_j) + y^*_j \sum_{j'} v\left(\mathbf{x} - e_j + e_{j'}\right) f_{j \to j'} \right) + (1 - Y^*)\, v(\mathbf{x}) = T_{B,i} v(\mathbf{x})
\end{aligned}$$

Notice that $y^*_{\tilde{k}} = Y^*$ and $y^*_j = 0$, $\forall j \neq \tilde{k}$, is an optimal solution for the operator $T_{B,i}$. Therefore, there is no loss of generality in assuming that resources are utilized by only one project when $Y^* > 0$.

We next show that, without loss of generality, Y^* can be restricted to take values zero or one. Suppose that there exists $0 < Y^* < 1$ such that $c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k} \to j'} + (1 - Y^*)\, v(\mathbf{x}) < \min\{c_{i\tilde{k}}(1) + \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k} \to j'}, v(\mathbf{x})\}$. Then the following inequalities must hold for fixed ϵ:

$$T_{B,i}v(\mathbf{x}) = c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^*)\, v\left(\mathbf{x}\right)$$

$$\leq c_{i\tilde{k}}(Y^* + \epsilon) + (Y^* + \epsilon) \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^* - \epsilon)\, v\left(\mathbf{x}\right), \qquad \text{(a)}$$

$$T_{B,i}v(\mathbf{x}) = c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^*)\, v\left(\mathbf{x}\right)$$

$$\leq c_{i\tilde{k}}(Y^* - \epsilon) + (Y^* - \epsilon) \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^* + \epsilon)\, v\left(\mathbf{x}\right) \qquad \text{(b)}$$

If inequality (a) holds with equality, $Y^*+\epsilon$ is also an optimal rate of utilized resources. Likewise, if inequality (b) holds with equality, $Y^* - \epsilon$ is also an optimal rate of utilized resources. If $Y^* + \epsilon$ (or $Y^* - \epsilon$) is an optimal rate of utilized resources, we can replace Y^* with $Y^* + \epsilon$ (or $Y^* - \epsilon$) and repeat the same argument for the same value of ϵ iteratively. Then we need to consider the following scenarios:

(i) Inequality (a) holds with equality until we hit 1. Inequality (b) holds with equality until we hit 0.

(ii) Inequality (a) holds with equality until we hit 1. There exists $\kappa_b \in \{0, 1, ..\}$ such that inequality (b) holds with equality at $Y^* - (\kappa_b - 1)\epsilon$ but fails at $Y^* - \kappa_b\epsilon \geq 0$.

(iii) There exists $\kappa_a \in \{0, 1, ..\}$ such that inequality (a) holds with equality at $Y^* + (\kappa_a - 1)\epsilon$ but fails at $Y^* + \kappa_a\epsilon \leq 1$. Inequality (b) holds with equality until we hit 0.

(iv) There exists $\kappa_a \in \{0, 1, ..\}$ such that inequality (a) holds with equality at $Y^* + (\kappa_a - 1)\epsilon$ but fails at $Y^* + \kappa_a\epsilon \leq 1$. There exists $\kappa_b \in \{0, 1, ..\}$ such that inequality (b) holds with equality at $Y^* - (\kappa_b - 1)\epsilon$ but fails at $Y^* - \kappa_b\epsilon \geq 0$.

But scenarios (i)–(iii) are infeasible as we assumed $c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^*)\, v\left(\mathbf{x}\right) < \min\{c_{i\tilde{k}}(1) + \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'}, v\left(\mathbf{x}\right)\}$ where $0 < Y^* < 1$. Now consider scenario (iv): $\{Y^* - (\kappa_b - 1)\epsilon, Y^* - (\kappa_b - 2)\epsilon, .., Y^* + (\kappa_a - 1)\epsilon\}$ is a set of optimal rates of utilized resources, implying the following (strict) inequalities:

$$
\begin{aligned}
& c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) + \left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right)\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} \\
& + \left(1 - Y^* - \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) v\left(\mathbf{x}\right) \\
& < c_{i\tilde{k}}(Y^* + \kappa_a\epsilon) + (Y^* + \kappa_a\epsilon)\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^* - \kappa_a\epsilon)\, v\left(\mathbf{x}\right), \qquad \text{(a}'\text{)} \\
& c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) + \left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right)\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} \\
& + \left(1 - Y^* - \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) v\left(\mathbf{x}\right) \\
& < c_{i\tilde{k}}(Y^* - \kappa_b\epsilon) + (Y^* - \kappa_b\epsilon)\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^* + \kappa_b\epsilon)\, v\left(\mathbf{x}\right). \qquad \text{(b}'\text{)}
\end{aligned}
$$

Inequalities (a′) and (b′) imply the following inequalities, respectively:

$$
\begin{aligned}
& c_{i\tilde{k}}(Y^* + \kappa_a\epsilon) - c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) + \frac{(\kappa_a + \kappa_b)\epsilon}{2}\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} \\
& > \frac{(\kappa_a + \kappa_b)\epsilon}{2} v\left(\mathbf{x}\right) \\
& > c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) - c_{i\tilde{k}}\left(Y^* - \kappa_b\epsilon\right) + \frac{(\kappa_a + \kappa_b)\epsilon}{2}\sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'}.
\end{aligned}
$$

Thus:

$$
c_{i\tilde{k}}(Y^* + \kappa_a\epsilon) - c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) > c_{i\tilde{k}}\left(Y^* + \frac{(\kappa_a - \kappa_b)\epsilon}{2}\right) - c_{i\tilde{k}}\left(Y^* - \kappa_b\epsilon\right)
$$

But the above inequality contradicts the assumption that $c_{ij}(y_j)$ is concave in y_j. Therefore, we cannot have $0 < Y^* < 1$ such that $c_{i\tilde{k}}(Y^*) + Y^* \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'} + (1 - Y^*)\, v\left(\mathbf{x}\right) < \min\{c_{i\tilde{k}}(1) + \sum_{j'} v\left(\mathbf{x} - e_{\tilde{k}} + e_{j'}\right) f_{\tilde{k}\to j'}, v\left(\mathbf{x}\right)\}$. If $Y^* = 0$ is an optimal solution, then $y_j^* = 0$, $\forall j \in \mathcal{W}_i$; without loss of generality, the action space of the operator $T_{B,i}$ can be reduced to the set of binary variables. If $Y^* = 1$ is an optimal solution, there is no loss of generality in assuming that resources are utilized by only one project; again the action space can be reduced to the set of binary variables.

We will now prove that, without loss of generality, the action space of the operator T_C can be reduced to the set of binary variables. For a given state $\mathbf{x}$, let y_j^* denote the optimal rate of resources of stage $m+1$ allocated to a project in category $j \in \mathcal{W}_{m+1}$. Thus:

$$T_C v(\mathbf{x}) = \sum_{j \in \mathcal{W}_{m+1}} y_j^* \left(v\left(\mathbf{x} - e_j\right) - \rho_j \right) + \left(1 - \sum_{j \in \mathcal{W}_{m+1}} y_j^* \right) v\left(\mathbf{x}\right)$$

There exists a category $k \in \mathcal{W}_{m+1}$ such that $v\left(\mathbf{x} - e_k\right) - \rho_k \leq v\left(\mathbf{x} - e_j\right) - \rho_j$, $\forall j \in \mathcal{W}_{m+1}$. If $v\left(\mathbf{x} - e_k\right) - \rho_k \leq v\left(\mathbf{x}\right)$, then $y_k^* = 1$ and $y_j^* = 0$, $\forall j \neq k$, is an optimal solution. If $v\left(\mathbf{x} - e_k\right) - \rho_k > v\left(\mathbf{x}\right)$, then $y_j^* = 0$, $\forall j \in \mathcal{W}_{m+1}$, is an optimal solution. Therefore, without loss of generality, the action space of the operator T_C can be reduced to the set of binary variables. □

C.2 Proofs of the Results in Section 4.4.

Lemma 4.4.1 (Restated)**.** *Under Assumptions 4.4.1 and 4.4.2, if* $v \in \widehat{V}$*, then* $Tv \in \widehat{V}$*, where* $Tv(\mathbf{x}) = h(\mathbf{x}) + \mu_1 T_B v(\mathbf{x}) + \mu_2 T_C v(\mathbf{x})$*. Furthermore, the optimal cost function* v^* *is an element of* $\widehat{V}$*.*

Proof. Define $\widehat{V}$ as the set of real-valued functions on $\mathbb{N}_0^m$ that satisfy Properties 1–7. Also, define the operator T on the set of real-valued functions v as follows: $Tv(\mathbf{x}) = h(\mathbf{x}) + \mu_1 T_B v(\mathbf{x}) + \mu_2 T_C v(\mathbf{x})$. We below show that $T_B : \widehat{V} \to \widehat{V}$, $T_C : \widehat{V} \to \widehat{V}$, and $h \in \widehat{V}$. We will then prove that $T : \widehat{V} \to \widehat{V}$.

$T_B : \widehat{V} \to \widehat{V}$. Assuming v satisfies Properties 1–7, we want to show $T_B v$ satisfies Properties 1–7.

Property 1. We will prove $T_B v$ satisfies Property 1 (i.e., $T_B v(\mathbf{x} + e_w) \geq T_B v(\mathbf{x} + e_q)$). There are two different scenarios we need to consider depending on the optimal action at $T_B v(\mathbf{x} + e_w)$ (if this inequality holds under a suboptimal action of $T_B v(\mathbf{x} + e_q)$, it also holds under the optimal action of this operator, and thus we do not enforce the optimal action at this operator). These two scenarios are as follows:

(1) Suppose that $T_B v(\mathbf{x} + e_w) = v(\mathbf{x} + e_w)$. As we assume v satisfies Property 1, the following inequalities hold: $T_B v(\mathbf{x} + e_w) = v(\mathbf{x} + e_w) \geq v(\mathbf{x} + e_q) \geq T_B v(\mathbf{x} + e_q)$.

(2) Suppose that $T_B v(\mathbf{x} + e_w) = \sum_j v(\mathbf{x} + e_w + e_j) f_{0\to j} + c$. As we assume v satisfies Property 1, the following inequalities hold: $T_B v(\mathbf{x}+e_w) = \sum_j v(\mathbf{x} + e_w + e_j) f_{0\to j} + c \geq \sum_j v(\mathbf{x} + e_q + e_j) f_{0\to j} + c \geq T_B v(\mathbf{x} + e_q)$.

Therefore $T_B v$ satisfies Property 1.

Property 2. We will prove $T_B v$ satisfies Property 2 (i.e., $T_B v(\mathbf{x} + e_w) \geq T_B v(\mathbf{x}) - \rho_w$). There are two different scenarios we need to consider depending on the optimal action at $T_B v(\mathbf{x} + e_w)$:

(1) Suppose that $T_B v(\mathbf{x}+e_w) = v(\mathbf{x}+e_w)$. As we assume v satisfies Property 2, the following inequalities hold: $T_B v(\mathbf{x} + e_w) = v(\mathbf{x} + e_w) \geq v(\mathbf{x}) - \rho_w \geq T_B v(\mathbf{x}) - \rho_w$.

(2) Suppose that $T_B v(\mathbf{x} + e_w) = \sum_j v(\mathbf{x} + e_w + e_j) f_{0\to j} + c$. As we assume v satisfies Property 2, the following inequalities hold: $T_B v(\mathbf{x}+e_w) = \sum_j v(\mathbf{x} + e_w + e_j) f_{0\to j} + c \geq \sum_j (v(\mathbf{x} + e_j) - \rho_w) f_{0\to j} + c = \sum_j v(\mathbf{x} + e_j) f_{0\to j} + c - \rho_w \geq T_B v(\mathbf{x}) - \rho_w$.

Therefore $T_B v$ satisfies Property 2.

Property 3. We will prove $T_B v$ satisfies Property 3 (i.e., $T_B v(\mathbf{x} + e_q) - \rho_w \geq T_B v(\mathbf{x} + e_w) - \rho_q$). There are two different scenarios we need to consider depending on the optimal action at $T_B v(\mathbf{x} + e_q)$:

(1) Suppose that $T_B v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q)$. As we assume v satisfies Property 3, the following inequalities hold: $T_B v(\mathbf{x}+e_q) - \rho_w = v(\mathbf{x}+e_q) - \rho_w \geq v(\mathbf{x}+e_w) - \rho_q \geq T_B v(\mathbf{x}+e_w) - \rho_q$.

(2) Suppose that $T_B v(\mathbf{x} + e_q) = \sum_j v(\mathbf{x} + e_q + e_j) f_{0\to j} + c$. As we assume v satisfies Property 3, the following inequalities hold: $T_B v(\mathbf{x} + e_q) - \rho_w = \sum_j v(\mathbf{x} + e_q + e_j) f_{0\to j} + c - \rho_w \geq \sum_j v(\mathbf{x} + e_w + e_j) f_{0\to j} + c - \rho_q \geq T_B v(\mathbf{x} + e_w) - \rho_q$.

Therefore $T_B v$ satisfies Property 3.

Property 4. We will prove $T_B v$ satisfies Property 4 (i.e., $T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) \geq T_B v(\mathbf{x} + e_l) - T_B v(\mathbf{x} + e_d)$, $\forall d, l, q \in \{1, 2, .., n\}$ where $d \leq l \leq q$). We consider the following scenarios depending on the optimal actions at $T_B v(\mathbf{x} + e_q + e_l)$ and $T_B v(\mathbf{x} + e_d)$ (if this inequality holds under suboptimal actions of $T_B v(\mathbf{x} + e_q + e_d)$ and/or $T_B v(\mathbf{x} + e_l)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators):

(1) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = v(\mathbf{x} + e_q + e_l)$ and $T_B v(\mathbf{x} + e_d) = v(\mathbf{x} + e_d)$. As we assume v satisfies Property 4, the following inequalities hold:

$$\begin{aligned} T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) &\geq v(\mathbf{x} + e_q + e_l) - v(\mathbf{x} + e_q + e_d) \\ &\geq v(\mathbf{x} + e_l) - v(\mathbf{x} + e_d) \\ &\geq T_B v(\mathbf{x} + e_l) - T_B v(\mathbf{x} + e_d) \end{aligned}$$

(2) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = v(\mathbf{x} + e_q + e_l)$ and $T_B v(\mathbf{x} + e_d) = \sum_j v(\mathbf{x} + e_d + e_j) f_{0\to j}$ $+c$. As we assume v satisfies Properties 4 and 7, the following inequalities hold:

$$\begin{aligned} & T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) \\ & \geq v(\mathbf{x} + e_q + e_l) - v(\mathbf{x} + e_q + e_d) \\ & \geq v(\mathbf{x} + e_l) - v(\mathbf{x} + e_d) \\ & \geq \sum_j v(\mathbf{x} + e_l + e_j) f_{0\to j} + c - \sum_j v(\mathbf{x} + e_d + e_j) f_{0\to j} - c \\ & \geq T_B v(\mathbf{x} + e_l) - T_B v(\mathbf{x} + e_d) \end{aligned}$$

(3) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c$ and $T_B v(\mathbf{x} + e_d) = v(\mathbf{x} + e_d)$. As we assume v satisfies Properties 4 and 6, the following inequalities hold:

$$\begin{aligned} & T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) \\ & \geq \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_q + e_d) \\ & \geq \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_q + e_l) \\ & + v(\mathbf{x} + e_l) - v(\mathbf{x} + e_d) \\ & \geq \sum_j v(\mathbf{x} + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_d) \\ & \geq T_B v(\mathbf{x} + e_l) - T_B v(\mathbf{x} + e_d) \end{aligned}$$

(4) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c$ and $T_B v(\mathbf{x} + e_d) = \sum_j v(\mathbf{x} + e_d + e_j) f_{0\to j} + c$. As we assume v satisfies Property 4, the following inequalities

hold:

$$
\begin{aligned}
& T_B v(\mathbf{x}+e_q+e_l) - T_B v(\mathbf{x}+e_q+e_d) \\
& \geq \sum_j v(\mathbf{x}+e_q+e_l+e_j) f_{0\to j} + c - \sum_j v(\mathbf{x}+e_q+e_d+e_j) f_{0\to j} - c \\
& = \sum_j \left(v(\mathbf{x}+e_q+e_l+e_j) - v(\mathbf{x}+e_q+e_d+e_j)\right) f_{0\to j} \\
& \geq \sum_j \left(v(\mathbf{x}+e_l+e_j) - v(\mathbf{x}+e_d+e_j)\right) f_{0\to j} \\
& = \sum_j v(\mathbf{x}+e_l+e_j) f_{0\to j} + c - \sum_j v(\mathbf{x}+e_d+e_j) f_{0\to j} - c \\
& \geq T_B v(\mathbf{x}+e_l) - T_B v(\mathbf{x}+e_d)
\end{aligned}
$$

Therefore $T_B v$ satisfies Property 4.

Property 5. We will prove $T_B v$ satisfies Property 5 (i.e., $T_B v(\mathbf{x}+e_q+e_l) - T_B v(\mathbf{x}+e_q+e_d) \geq T_B v(\mathbf{x}+e_w+e_l) - T_B v(\mathbf{x}+e_w+e_d)$, $\forall d,l,q,w \in \{1,2,..,n\}$ where $d \leq l \leq q \leq w$). We consider the following scenarios depending on the optimal actions at $T_B v(\mathbf{x}+e_q+e_l)$ and $T_B v(\mathbf{x}+e_w+e_d)$:

(1) Suppose that $T_B v(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$ and $T_B v(\mathbf{x}+e_w+e_d) = v(\mathbf{x}+e_w+e_d)$. As we assume v satisfies Property 5, the following inequalities hold:

$$
\begin{aligned}
T_B v(\mathbf{x}+e_q+e_l) - T_B v(\mathbf{x}+e_q+e_d) & \geq v(\mathbf{x}+e_q+e_l) - v(\mathbf{x}+e_q+e_d) \\
& \geq v(\mathbf{x}+e_w+e_l) - v(\mathbf{x}+e_w+e_d) \\
& \geq T_B v(\mathbf{x}+e_w+e_l) - T_B v(\mathbf{x}+e_w+e_d)
\end{aligned}
$$

(2) Suppose that $T_B v(\mathbf{x}+e_w+e_d) = \sum_j v(\mathbf{x}+e_w+e_d+e_j) f_{0\to j} + c$ and $T_B v(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$. As we assume v satisfies Properties 5 and 7, the following inequalities

hold:

$$
\begin{aligned}
& T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) \\
& \geq v(\mathbf{x} + e_q + e_l) - v(\mathbf{x} + e_q + e_d) \\
& \geq v(\mathbf{x} + e_w + e_l) - v(\mathbf{x} + e_w + e_d) \\
& \geq \sum_j v(\mathbf{x} + e_w + e_l + e_j) f_{0\to j} - \sum_j v(\mathbf{x} + e_w + e_d + e_j) f_{0\to j} \\
& \geq T_B v(\mathbf{x} + e_w + e_l) - T_B v(\mathbf{x} + e_w + e_d)
\end{aligned}
$$

(3) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c$ and $T_B v(\mathbf{x} + e_w + e_d) = v(\mathbf{x} + e_w + e_d)$. As we assume v satisfies Properties 5 and 7, the following inequalities hold:

$$
\begin{aligned}
& T_B v(\mathbf{x} + e_q + e_l) - T_B v(\mathbf{x} + e_q + e_d) \\
& \geq \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_q + e_d) \\
& \geq \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_q + e_l) \\
& + v(\mathbf{x} + e_w + e_l) - v(\mathbf{x} + e_w + e_d) \\
& \geq \sum_j v(\mathbf{x} + e_w + e_l + e_j) f_{0\to j} + c - v(\mathbf{x} + e_w + e_d) \\
& \geq T_B v(\mathbf{x} + e_w + e_l) - T_B v(\mathbf{x} + e_w + e_d)
\end{aligned}
$$

(4) Suppose that $T_B v(\mathbf{x} + e_q + e_l) = \sum_j v(\mathbf{x} + e_q + e_l + e_j) f_{0\to j} + c$ and $T_B v(\mathbf{x} + e_w + e_d) = \sum_j v(\mathbf{x} + e_w + e_d + e_j) f_{0\to j} + c$. As we assume v satisfies Property 5, the following

inequalities hold:

$$\begin{aligned}
&T_Bv(\mathbf{x}+e_q+e_l) - T_Bv(\mathbf{x}+e_q+e_d)\\
&\geq \sum_j v(\mathbf{x}+e_q+e_l+e_j)f_{0\to j} + c - \sum_j v(\mathbf{x}+e_q+e_d+e_j)f_{0\to j} - c\\
&= \sum_j \left(v(\mathbf{x}+e_q+e_l+e_j) - v(\mathbf{x}+e_q+e_d+e_j)\right)f_{0\to j}\\
&\geq \sum_j \left(v(\mathbf{x}+e_w+e_l+e_j) - v(\mathbf{x}+e_w+e_d+e_j)\right)f_{0\to j}\\
&= \sum_j v(\mathbf{x}+e_w+e_l+e_j)f_{0\to j} + c - \sum_j v(\mathbf{x}+e_w+e_d+e_j)f_{0\to j} - c\\
&\geq T_Bv(\mathbf{x}+e_w+e_l) - T_Bv(\mathbf{x}+e_w+e_d)
\end{aligned}$$

Therefore T_Bv satisfies Property 5.

Property 6. We will prove T_Bv satisfies Property 6, i.e.,

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_Bv(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_Bv(\mathbf{x}+e_q+e_j)f_{0\to j} - T_Bv(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} T_Bv(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_Bv(\mathbf{x}+e_j)f_{0\to j} - T_Bv(\mathbf{x}),\ \forall l,q\in\{1,..,n\} \text{ where } l\leq q.
\end{aligned}$$

We consider the following scenarios depending on the optimal actions at $T_Bv(\mathbf{x}+e_q+e_l)$, $T_Bv(\mathbf{x}+e_q+e_j)$ for $j>l$, and $T_Bv(\mathbf{x})$ (if this inequality holds under suboptimal actions of $T_Bv(\mathbf{x}+e_q)$, $T_Bv(\mathbf{x}+e_l)$, and/or $T_Bv(\mathbf{x}+e_j)$ for $j>l$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators):

(1) Suppose that $T_Bv(\mathbf{x}) = v(\mathbf{x})$. As we assume v satisfies Property 6, $T_Bv(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$, $T_Bv(\mathbf{x}+e_q+e_j) = v(\mathbf{x}+e_q+e_j)$, $\forall j>l$, and the inequalities below hold:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_Bv(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_Bv(\mathbf{x}+e_q+e_j)f_{0\to j} - T_Bv(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} v(\mathbf{x}+e_q+e_j)f_{0\to j} - v(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} v(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} v(\mathbf{x}+e_j)f_{0\to j} - v(\mathbf{x})\\
&\geq \sum_{1\leq j\leq l} T_Bv(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_Bv(\mathbf{x}+e_j)f_{0\to j} - T_Bv(\mathbf{x})
\end{aligned}$$

(2) Suppose that $T_B v(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$, $T_B v(\mathbf{x}+e_q+e_j) = v(\mathbf{x}+e_q+e_j)$ for $j^* > j > l$, $T_B v(\mathbf{x}+e_q+e_j) = \sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c$ for $j \geq j^*$, and $T_B v(\mathbf{x}) = \sum_{j'} v(\mathbf{x}+e_{j'}) f_{0\to j'} + c$. As we assume v satisfies Properties 4 and 6, the following inequalities hold:

$$
\begin{aligned}
&\sum_{1\leq j\leq l} T_B v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{n\geq j>l} T_B v(\mathbf{x}+e_q+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{l<j<j^*} v(\mathbf{x}+e_q+e_j) f_{0\to j}\\
&+ \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c\right) f_{0\to j} - \sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_j) f_{0\to j}\\
&- \sum_{l<j<j^*} v(\mathbf{x}+e_q+e_j) f_{0\to j} - \sum_{j^*\leq j} v(\mathbf{x}+e_q+e_j) f_{0\to j} - c\\
&= \sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c\right) f_{0\to j}\\
&- \sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_j) f_{0\to j} - \sum_{j^*\leq j} v(\mathbf{x}+e_q+e_j) f_{0\to j} - c\\
&= \sum_{1\leq j\leq l} \left(v(\mathbf{x}+e_q+e_l) - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j}\\
&+ \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j}\\
&\geq \sum_{1\leq j\leq l} \left(v(\mathbf{x}+e_l) - v(\mathbf{x}+e_j)\right) f_{0\to j}\\
&+ \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j}\\
&\geq \sum_{1\leq j\leq l} \left(v(\mathbf{x}+e_l) - v(\mathbf{x}+e_j)\right) f_{0\to j}\\
&+ \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_j+e_{j'}) f_{0\to j'} - v(\mathbf{x}+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j}\\
&= \sum_{1\leq j\leq l} v(\mathbf{x}+e_l) f_{0\to j} + \sum_{j^*\leq j}\left(\sum_{j'} v(\mathbf{x}+e_j+e_{j'}) f_{0\to j'} + c\right) f_{0\to j}\\
&- \sum_{1\leq j\leq l} v(\mathbf{x}+e_j) f_{0\to j} - \sum_{j^*\leq j} v(\mathbf{x}+e_j) f_{0\to j} - c
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{1 \le j \le l} v(\mathbf{x}+e_l) f_{0\to j} + \sum_{l<j<j^*} v(\mathbf{x}+e_j) f_{0\to j} \\
&+ \sum_{j^* \le j} \left(\sum_{j'} v(\mathbf{x}+e_j+e_{j'}) f_{0\to j'} + c \right) f_{0\to j} - \sum_{1 \le j \le l} v(\mathbf{x}+e_j) f_{0\to j} \\
&- \sum_{l<j<j^*} v(\mathbf{x}+e_j) f_{0\to j} - \sum_{j^* \le j} v(\mathbf{x}+e_j) f_{0\to j} - c \\
&\ge \sum_{1 \le j \le l} T_B v(\mathbf{x}+e_l) f_{0\to j} + \sum_{n \ge j > l} T_B v(\mathbf{x}+e_j) f_{0\to j} - T_B v(\mathbf{x})
\end{aligned}
$$

(3) Suppose that $T_B v(\mathbf{x}+e_q+e_l) = \sum_{j'} v(\mathbf{x}+e_q+e_l+e_{j'}) f_{0\to j'} + c$. As we assume v satisfies Property 7, $T_B v(\mathbf{x}+e_q+e_j) = \sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c$ for $j > l$. As we assume v satisfies Property 6, $T_B v(\mathbf{x}) = \sum_{j'} v(\mathbf{x}+e_{j'}) f_{0\to j'} + c$. As we assume v satisfies Property 6, the following inequalities hold:

$$
\begin{aligned}
&\sum_{1 \le j \le l} T_B v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{n \ge j > l} T_B v(\mathbf{x}+e_q+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_q) \\
&\ge \sum_{1 \le j \le l} \left(\sum_{j'} v(\mathbf{x}+e_q+e_l+e_{j'}) f_{0\to j'} + c \right) f_{0\to j} \\
&+ \sum_{n \ge j > l} \left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c \right) f_{0\to j} \\
&- \sum_{j'} v(\mathbf{x}+e_q+e_{j'}) f_{0\to j'} - c \\
&= \sum_{j'} \Bigg[\sum_{1 \le j \le l} v(\mathbf{x}+e_q+e_l+e_{j'}) f_{0\to j} + \sum_{n \ge j > l} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j} \\
&\qquad - v(\mathbf{x}+e_q+e_{j'}) \Bigg] f_{0\to j'} \\
&\ge \sum_{j'} \left[\sum_{1 \le j \le l} v(\mathbf{x}+e_l+e_{j'}) f_{0\to j} + \sum_{n \ge j > l} v(\mathbf{x}+e_j+e_{j'}) f_{0\to j} - v(\mathbf{x}+e_{j'}) \right] f_{0\to j'} \\
&= \sum_{1 \le j \le l} \left(\sum_{j'} v(\mathbf{x}+e_l+e_{j'}) f_{0\to j'} + c \right) f_{0\to j} \\
&+ \sum_{n \ge j > l} \left(\sum_{j'} v(\mathbf{x}+e_j+e_{j'}) f_{0\to j'} + c \right) f_{0\to j} \\
&- \sum_{j'} v(\mathbf{x}+e_{j'}) f_{0\to j'} - c \\
&\ge \sum_{1 \le j \le l} T_B v(\mathbf{x}+e_l) f_{0\to j} + \sum_{n \ge j > l} T_B v(\mathbf{x}+e_j) f_{0\to j} - T_B v(\mathbf{x})
\end{aligned}
$$

Therefore $T_B v$ satisfies Property 6.

Property 7. We will prove $T_B v$ satisfies Property 7, i.e.,

$$\begin{aligned}
&\sum_{1\le j\le l} T_B v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{n\ge j>l} T_B v(\mathbf{x}+e_q+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_q)\\
&\ge \sum_{1\le j\le l} T_B v(\mathbf{x}+e_w+e_l) f_{0\to j} + \sum_{n\ge j>l} T_B v(\mathbf{x}+e_w+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_w),\\
&\forall l,q,w \in \{1,..,n\} \text{ where } l \le q < w.
\end{aligned}$$

We consider the following scenarios depending on the optimal actions at $T_B v(\mathbf{x}+e_q+e_l)$, $T_B v(\mathbf{x}+e_q+e_j)$ for $j>l$, and $T_B v(\mathbf{x}+e_w)$:

(1) Suppose that $T_B v(\mathbf{x}+e_w) = v(\mathbf{x}+e_w)$. As we assume v satisfies Properties 6 and 7, $T_B v(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$ and $T_B v(\mathbf{x}+e_q+e_j) = v(\mathbf{x}+e_q+e_j)$ for $j>l$. As we assume v satisfies Property 7, the following inequalities hold:

$$\begin{aligned}
&\sum_{1\le j\le l} T_B v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{n\ge j>l} T_B v(\mathbf{x}+e_q+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_q)\\
&\ge \sum_{1\le j\le l} v(\mathbf{x}+e_q+e_l) f_{0\to j} + \sum_{n\ge j>l} v(\mathbf{x}+e_q+e_j) f_{0\to j} - v(\mathbf{x}+e_q)\\
&\ge \sum_{1\le j\le l} v(\mathbf{x}+e_w+e_l) f_{0\to j} + \sum_{n\ge j>l} v(\mathbf{x}+e_w+e_j) f_{0\to j} - v(\mathbf{x}+e_w)\\
&\ge \sum_{1\le j\le l} T_B v(\mathbf{x}+e_w+e_l) f_{0\to j} + \sum_{n\ge j>l} T_B v(\mathbf{x}+e_w+e_j) f_{0\to j} - T_B v(\mathbf{x}+e_w)
\end{aligned}$$

(2) Suppose that $T_B v(\mathbf{x}+e_q+e_l) = v(\mathbf{x}+e_q+e_l)$, $T_B v(\mathbf{x}+e_q+e_j) = v(\mathbf{x}+e_q+e_j)$ for $j^* > j > l$, $T_B v(\mathbf{x}+e_q+e_j) = \sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'}) f_{0\to j'} + c$ for $j \ge j^*$, and $T_B v(\mathbf{x}+e_w) = \sum_{j'} v(\mathbf{x}+e_w+e_{j'}) f_{0\to j'} + c$. As we assume v satisfies Properties 5 and

7, the following inequalities hold:

$$
\begin{aligned}
&\sum_{1\le j\le l} T_B v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\ge j>l} T_B v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_B v(\mathbf{x}+e_q) \\
&\ge \sum_{1\le j\le l} v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{l<j<j^*} v(\mathbf{x}+e_q+e_j)f_{0\to j} \\
&+ \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j} - \sum_{1\le j\le l} v(\mathbf{x}+e_q+e_j)f_{0\to j} \\
&- \sum_{l<j<j^*} v(\mathbf{x}+e_q+e_j)f_{0\to j} - \sum_{j^*\le j} v(\mathbf{x}+e_q+e_j)f_{0\to j} - c \\
&= \sum_{1\le j\le l} v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j} \\
&- \sum_{1\le j\le l} v(\mathbf{x}+e_q+e_j)f_{0\to j} - \sum_{j^*\le j} v(\mathbf{x}+e_q+e_j)f_{0\to j} - c \\
&= \sum_{1\le j\le l} \left(v(\mathbf{x}+e_q+e_l) - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j} \\
&+ \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j} \\
&\ge \sum_{1\le j\le l} \left(v(\mathbf{x}+e_w+e_l) - v(\mathbf{x}+e_w+e_j)\right) f_{0\to j} \\
&+ \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} - v(\mathbf{x}+e_q+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j} \\
&\ge \sum_{1\le j\le l} \left(v(\mathbf{x}+e_w+e_l) - v(\mathbf{x}+e_w+e_j)\right) f_{0\to j} \\
&+ \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_w+e_j+e_{j'})f_{0\to j'} - v(\mathbf{x}+e_w+e_j)\right) f_{0\to j} - c\sum_{j<j^*} f_{0\to j} \\
&= \sum_{1\le j\le l} v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_w+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j} \\
&- \sum_{1\le j\le l} v(\mathbf{x}+e_w+e_j)f_{0\to j} - \sum_{j^*\le j} v(\mathbf{x}+e_w+e_j)f_{0\to j} - c \\
&= \sum_{1\le j\le l} v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{l<j<j^*} v(\mathbf{x}+e_w+e_j)f_{0\to j} \\
&+ \sum_{j^*\le j}\left(\sum_{j'} v(\mathbf{x}+e_w+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j} - \sum_{1\le j\le l} v(\mathbf{x}+e_w+e_j)f_{0\to j} \\
&- \sum_{l<j<j^*} v(\mathbf{x}+e_w+e_j)f_{0\to j} - \sum_{j^*\le j} v(\mathbf{x}+e_w+e_j)f_{0\to j} - c
\end{aligned}
$$

$$\geq \sum_{1\leq j\leq l} T_B v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_B v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_B v(\mathbf{x}+e_w)$$

(3) Suppose that $T_B v(\mathbf{x}+e_q+e_l) = \sum_{j'} v(\mathbf{x}+e_q+e_l+e_{j'})f_{0\to j'} + c$. As we assume v satisfies Property 7, $T_B v(\mathbf{x}+e_q+e_j) = \sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} + c$ for $j > l$. As we assume v satisfies Properties 6 and 7, $T_B v(\mathbf{x}+e_w) = \sum_{j'} v(\mathbf{x}+e_w+e_{j'})f_{0\to j'} + c$. As we assume v satisfies Property 7, the following inequalities hold:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_B v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_B v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_B v(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l}\left(\sum_{j'} v(\mathbf{x}+e_q+e_l+e_{j'})f_{0\to j'} + c\right) f_{0\to j}\\
&+ \sum_{n\geq j>l}\left(\sum_{j'} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j}\\
&- \sum_{j'} v(\mathbf{x}+e_q+e_{j'})f_{0\to j'} - c\\
&= \sum_{j'}\Bigg[\sum_{1\leq j\leq l} v(\mathbf{x}+e_q+e_l+e_{j'})f_{0\to j} + \sum_{n\geq j>l} v(\mathbf{x}+e_q+e_j+e_{j'})f_{0\to j}\\
&\qquad -v(\mathbf{x}+e_q+e_{j'})\Bigg] f_{0\to j'}\\
&\geq \sum_{j'}\Bigg[\sum_{1\leq j\leq l} v(\mathbf{x}+e_w+e_l+e_{j'})f_{0\to j} + \sum_{n\geq j>l} v(\mathbf{x}+e_w+e_j+e_{j'})f_{0\to j}\\
&\qquad -v(\mathbf{x}+e_w+e_{j'})\Bigg] f_{0\to j'}\\
&= \sum_{1\leq j\leq l}\left(\sum_{j'} v(\mathbf{x}+e_w+e_l+e_{j'})f_{0\to j'} + c\right) f_{0\to j}\\
&+ \sum_{n\geq j>l}\left(\sum_{j'} v(\mathbf{x}+e_w+e_j+e_{j'})f_{0\to j'} + c\right) f_{0\to j}\\
&- \sum_{j'} v(\mathbf{x}+e_w+e_{j'})f_{0\to j'} - c\\
&\geq \sum_{1\leq j\leq l} T_B v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_B v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_B v(\mathbf{x}+e_w)
\end{aligned}$$

Therefore $T_B v$ satisfies Property 7. Hence we showed that $T_B v$ satisfies Properties 1–7; $T_B : \widehat{V} \to \widehat{V}$.

$T_C : \widehat{V} \to \widehat{V}$. Assuming v satisfies Properties 1–7, we want to show $T_C v$ satisfies Properties 1–7.

Property 1. We will prove $T_C v$ satisfies Property 1 (i.e., $T_C v(\mathbf{x} + e_w) \geq T_C v(\mathbf{x} + e_q)$). As we assume v satisfies Properties 2 and 3, it is always optimal to launch a new product by choosing a project with highest expected reward: $T_C v(\mathbf{x} + e_w) = v(\mathbf{x} + e_w - e_b) - \rho_b$ where b is the smallest j such that $x_j + I_{j=w} \geq 1$ ($I_{j=w} = 1$ if $j = w$, and $I_{j=w} = 0$ otherwise). Suppose that $b = w$. Then it is easy to verify that $T_C v(\mathbf{x} + e_w) = v(\mathbf{x}) - \rho_w \geq v(\mathbf{x}) - \rho_q \geq T_C v(\mathbf{x} + e_q)$. Now suppose that $b \neq w$. As we assume v satisfies Property 1, $T_C v(\mathbf{x}+e_w) = v(\mathbf{x}+e_w - e_b) - \rho_b \geq v(\mathbf{x}+e_q - e_b) - \rho_b \geq T_C v(\mathbf{x}+e_q)$. Therefore $T_C v$ satisfies Property 1.

Property 2. We will prove $T_C v$ satisfies Property 2 (i.e., $T_C v(\mathbf{x} + e_w) \geq T_C v(\mathbf{x}) - \rho_w$). As we assume v satisfies Properties 2 and 3, $T_C v(\mathbf{x} + e_w) = v(\mathbf{x} + e_w - e_b) - \rho_b$ where b is the smallest j such that $x_j + I_{j=w} \geq 1$. Suppose that $b = w$. Then it is easy to verify that $T_C v(\mathbf{x} + e_w) = v(\mathbf{x}) - \rho_w \geq T_C v(\mathbf{x}) - \rho_w$. Now suppose that $b \neq w$. As we assume v satisfies Property 2, $T_C v(\mathbf{x}+e_w) = v(\mathbf{x} + e_w - e_b) - \rho_b \geq v(\mathbf{x} - e_b) - \rho_b - \rho_w \geq T_C v(\mathbf{x}) - \rho_w$. Therefore $T_C v$ satisfies Property 2.

Property 3. We will prove $T_C v$ satisfies Property 3 (i.e., $T_C v(\mathbf{x} + e_q) - \rho_w \geq T_C v(\mathbf{x} + e_w) - \rho_q$). Again, as we assume v satisfies Properties 2 and 3, $T_C v(\mathbf{x}+e_q) = v(\mathbf{x} + e_q - e_b) - \rho_b$ where b is the smallest j such that $x_j + I_{j=q} \geq 1$. Suppose that $b = q$. Then it is easy to verify that $T_C v(\mathbf{x} + e_q) - \rho_w = v(\mathbf{x}) - \rho_q - \rho_w \geq T_C v(\mathbf{x} + e_w) - \rho_q$. Now suppose that $b \neq q$. As we assume v satisfies Property 3, $T_C v(\mathbf{x} + e_q) - \rho_w = v(\mathbf{x} + e_q - e_b) - \rho_b - \rho_w \geq v(\mathbf{x} + e_w - e_b) - \rho_b - \rho_q \geq T_C v(\mathbf{x} + e_w) - \rho_q$. Therefore $T_C v$ satisfies Property 3.

Property 4. We will prove $T_C v$ satisfies Property 4 (i.e., $T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) \geq T_C v(\mathbf{x} + e_l) - T_C v(\mathbf{x} + e_d)$, $\forall d, l, q \in \{1, 2, .., n\}$ where $d \leq l \leq q$). Recall that as we assume v satisfies Properties 2 and 3, it is always optimal to launch a new product by choosing a project with highest expected reward. We consider the following scenarios:

(1) Suppose that $x_j = 0$, $\forall j \leq l$. Then it is easy to verify the following equalities:

$$\begin{aligned} T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) &= v(\mathbf{x} + e_q) - \rho_l - v(\mathbf{x} + e_q) + \rho_d \\ &= v(\mathbf{x}) - \rho_l - v(\mathbf{x}) + \rho_d \\ &= T_C v(\mathbf{x} + e_l) - T_C v(\mathbf{x} + e_d) \end{aligned}$$

(2) Suppose that $x_j = 0$, $\forall j \leq d$, and $x_j \geq 1$, $\exists j \in \{d+1, d+2, .., l\}$. Define b as the smallest j such that $x_j \geq 1$. As we assume v satisfies Property 4, the following inequality holds:

$$\begin{aligned} T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) &= v(\mathbf{x} + e_q + e_l - e_b) - \rho_b - v(\mathbf{x} + e_q) + \rho_d \\ &\geq v(\mathbf{x} + e_l - e_b) - \rho_b - v(\mathbf{x}) + \rho_d \\ &= T_C v(\mathbf{x} + e_l) - T_C v(\mathbf{x} + e_d) \end{aligned}$$

(3) Suppose that $x_j \geq 1$, $\exists j \leq d$. Again define b as the smallest j such that $x_j \geq 1$. As we assume v satisfies Property 4, the following inequality holds:

$$\begin{aligned} & T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) \\ & = v(\mathbf{x} + e_q + e_l - e_b) - \rho_b - v(\mathbf{x} + e_q + e_d - e_b) + \rho_b \\ & \geq v(\mathbf{x} + e_l - e_b) - \rho_b - v(\mathbf{x} + e_d - e_b) + \rho_b \\ & = T_C v(\mathbf{x} + e_l) - T_C v(\mathbf{x} + e_d) \end{aligned}$$

Therefore $T_C v$ satisfies Property 4.

Property 5. We will prove $T_C v$ satisfies Property 5 (i.e., $T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) \geq T_C v(\mathbf{x} + e_w + e_l) - T_C v(\mathbf{x} + e_w + e_d)$, $\forall d, l, q, w \in \{1, 2, .., n\}$ where $d \leq l \leq q \leq w$). Recall that as we assume v satisfies Properties 2 and 3, it is always optimal to launch a new product by choosing a project with highest expected reward. We consider the following scenarios:

(1) Suppose that $x_j = 0$, $\forall j \leq l$. Then it is easy to verify the following equalities:

$$\begin{aligned} T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) &= v(\mathbf{x} + e_q) - \rho_l - v(\mathbf{x} + e_q) + \rho_d \\ &= v(\mathbf{x} + e_w) - \rho_l - v(\mathbf{x} + e_w) + \rho_d \\ &= T_C v(\mathbf{x} + e_w + e_l) - T_C v(\mathbf{x} + e_w + e_d) \end{aligned}$$

(2) Suppose that $x_j = 0$, $\forall j \leq d$, and $x_j \geq 1$, $\exists j \in \{d+1, d+2, .., l\}$. Define b as the smallest j such that $x_j \geq 1$. As we assume v satisfies Property 5, the following inequality holds:

$$\begin{aligned} T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) &= v(\mathbf{x} + e_q + e_l - e_b) - \rho_b - v(\mathbf{x} + e_q) + \rho_d \\ &\geq v(\mathbf{x} + e_w + e_l - e_b) - \rho_b - v(\mathbf{x} + e_w) + \rho_d \\ &= T_C v(\mathbf{x} + e_w + e_l) - T_C v(\mathbf{x} + e_w + e_d) \end{aligned}$$

(3) Suppose that $x_j \geq 1$, $\exists j \leq d$. Again define b as the smallest j such that $x_j \geq 1$. As we assume v satisfies Property 5, the following inequality holds:

$$\begin{aligned} & T_C v(\mathbf{x} + e_q + e_l) - T_C v(\mathbf{x} + e_q + e_d) \\ & = v(\mathbf{x} + e_q + e_l - e_b) - \rho_b - v(\mathbf{x} + e_q + e_d - e_b) + \rho_b \\ & \geq v(\mathbf{x} + e_w + e_l - e_b) - \rho_b - v(\mathbf{x} + e_w + e_d - e_b) + \rho_b \\ & = T_C v(\mathbf{x} + e_w + e_l) - T_C v(\mathbf{x} + e_w + e_d) \end{aligned}$$

Therefore $T_C v$ satisfies Property 5.

Property 6. We will prove $T_C v$ satisfies Property 6, i.e.,

$$\begin{aligned} & \sum_{1 \leq j \leq l} T_C v(\mathbf{x} + e_q + e_l) f_{0 \to j} + \sum_{n \geq j > l} T_C v(\mathbf{x} + e_q + e_j) f_{0 \to j} - T_C v(\mathbf{x} + e_q) \\ & \geq \sum_{1 \leq j \leq l} T_C v(\mathbf{x} + e_l) f_{0 \to j} + \sum_{n \geq j > l} T_C v(\mathbf{x} + e_j) f_{0 \to j} - T_C v(\mathbf{x}), \ \forall l, q \in \{1, .., n\} \text{ where } l \leq q. \end{aligned}$$

Recall that as we assume v satisfies Properties 2 and 3, it is always optimal to launch a new product by choosing a project with highest expected reward. Taking $q = n$, we consider the following scenarios:

(1) Suppose that $x_j = 0$, $\forall j \leq q$. As we assume v satisfies Property 2, the following inequality holds:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\
&= \sum_{1\leq j\leq l} (v(\mathbf{x}+e_q)-\rho_l)f_{0\to j} + \sum_{n\geq j>l} (v(\mathbf{x}+e_q)-\rho_j)f_{0\to j} - v(\mathbf{x}) + \rho_q\\
&= v(\mathbf{x}+e_q) - \sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{n\geq j>l} \rho_j f_{0\to j} - v(\mathbf{x}) + \rho_q\\
&\geq -\sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{n\geq j>l} \rho_j f_{0\to j}\\
&= \sum_{1\leq j\leq l} (v(\mathbf{x})-\rho_l)f_{0\to j} + \sum_{n\geq j>l} (v(\mathbf{x})-\rho_j)f_{0\to j} - v(\mathbf{x})\\
&= \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_j)f_{0\to j} - T_C v(\mathbf{x})
\end{aligned}$$

(2) Suppose that $x_j = 0$, $\forall j \leq l$, and $x_j \geq 1$, $\exists j \in \{l+1, l+2, .., q\}$. Define b as the smallest j such that $x_j \geq 1$. As we assume v satisfies Property 6, the following inequality holds:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\
&= \sum_{1\leq j\leq l} (v(\mathbf{x}+e_q)-\rho_l)f_{0\to j} + \sum_{b\geq j>l} (v(\mathbf{x}+e_q)-\rho_j)f_{0\to j}\\
&+ \sum_{n\geq j>b} (v(\mathbf{x}+e_q+e_j-e_b)-\rho_b)f_{0\to j} - v(\mathbf{x}+e_q-e_b) + \rho_b\\
&= \sum_{1\leq j\leq b} v(\mathbf{x}+e_q)f_{0\to j} + \sum_{n\geq j>b} v(\mathbf{x}+e_q+e_j-e_b)f_{0\to j} - v(\mathbf{x}+e_q-e_b)\\
&- \sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{b\geq j>l} \rho_j f_{0\to j} + \sum_{1\leq j\leq b} \rho_b f_{0\to j}\\
&\geq \sum_{1\leq j\leq b} v(\mathbf{x})f_{0\to j} + \sum_{n\geq j>b} v(\mathbf{x}+e_j-e_b)f_{0\to j} - v(\mathbf{x}-e_b)\\
&- \sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{b\geq j>l} \rho_j f_{0\to j} + \sum_{1\leq j\leq b} \rho_b f_{0\to j}
\end{aligned}$$

$$= \sum_{1\le j\le l} (v(\mathbf{x}) - \rho_l) f_{0\to j} + \sum_{b\ge j>l} (v(\mathbf{x}) - \rho_j) f_{0\to j}$$
$$+ \sum_{n\ge j>b} (v(\mathbf{x} + e_j - e_b) - \rho_b) f_{0\to j} - v(\mathbf{x} - e_b) + \rho_b$$
$$= \sum_{1\le j\le l} T_C v(\mathbf{x} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_j) f_{0\to j} - T_C v(\mathbf{x})$$

(3) Suppose that $x_j \ge 1$, $\exists j \le l$. Again define b as the smallest j such that $x_j \ge 1$. As we assume v satisfies Property 6, the following inequality holds:

$$\sum_{1\le j\le l} T_C v(\mathbf{x} + e_q + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_q + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_q)$$
$$= \sum_{1\le j\le l} (v(\mathbf{x} + e_q + e_l - e_b) - \rho_b) f_{0\to j} + \sum_{n\ge j>l} (v(\mathbf{x} + e_q + e_j - e_b) - \rho_b) f_{0\to j}$$
$$-v(\mathbf{x} + e_q - e_b) + \rho_b$$
$$\ge \sum_{1\le j\le l} (v(\mathbf{x} + e_l - e_b) - \rho_b) f_{0\to j} + \sum_{n\ge j>l} (v(\mathbf{x} + e_j - e_b) - \rho_b) f_{0\to j}$$
$$-v(\mathbf{x} - e_b) + \rho_b$$
$$= \sum_{1\le j\le l} T_C v(\mathbf{x} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_j) f_{0\to j} - T_C v(\mathbf{x})$$

Therefore $T_C v$ satisfies Property 6 when $q = n$. As $T_C v$ satisfies Property 7, it also satisfies Property 6 for $l \le q \le n$:

$$\sum_{1\le j\le l} T_C v(\mathbf{x} + e_q + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_q + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_q)$$
$$\ge \sum_{1\le j\le l} T_C v(\mathbf{x} + e_{q+1} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_{q+1} + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_{q+1}),$$
$$\sum_{1\le j\le l} T_C v(\mathbf{x} + e_{q+1} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_{q+1} + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_{q+1})$$
$$\ge \sum_{1\le j\le l} T_C v(\mathbf{x} + e_{q+2} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_{q+2} + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_{q+2}),$$
$$\vdots$$
$$\sum_{1\le j\le l} T_C v(\mathbf{x} + e_n + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_n + e_j) f_{0\to j} - T_C v(\mathbf{x} + e_n)$$
$$\ge \sum_{1\le j\le l} T_C v(\mathbf{x} + e_l) f_{0\to j} + \sum_{n\ge j>l} T_C v(\mathbf{x} + e_j) f_{0\to j} - T_C v(\mathbf{x}).$$

Summation of the above inequalities implies

$$\begin{aligned}&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\ &\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_j)f_{0\to j} - T_C v(\mathbf{x}) \text{ where } l\leq q\leq n.\end{aligned}$$

Therefore $T_C v$ satisfies Property 6.

Property 7. We will prove $T_C v$ satisfies Property 7, i.e.,

$$\begin{aligned}&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\ &\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_w),\\ &\forall l,q,w\in\{1,2,..,n\} \text{ where } l\leq q<w.\end{aligned}$$

Recall that as we assume v satisfies Properties 2 and 3, it is always optimal to launch a new product by choosing a project with highest expected reward. Taking $q=w-1$, we consider the following scenarios:

(1) Suppose that $x_j=0$, $\forall j\leq q$. As we assume v satisfies Property 3, the following inequality holds:

$$\begin{aligned}&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\ &= \sum_{1\leq j\leq l} (v(\mathbf{x}+e_q)-\rho_l)f_{0\to j} + \sum_{q\geq j>l} (v(\mathbf{x}+e_q)-\rho_j)f_{0\to j}\\ &+ \sum_{n\geq j\geq w} (v(\mathbf{x}+e_j)-\rho_q)f_{0\to j} - v(\mathbf{x})+\rho_q\\ &= \sum_{1\leq j\leq l} (v(\mathbf{x}+e_q)+\rho_q)f_{0\to j} + \sum_{q\geq j>l} (v(\mathbf{x}+e_q)+\rho_q)f_{0\to j} + \sum_{n\geq j\geq w} v(\mathbf{x}+e_j)f_{0\to j}\\ &-v(\mathbf{x}) - \sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{q\geq j>l} \rho_j f_{0\to j}\\ &\geq \sum_{1\leq j\leq l} (v(\mathbf{x}+e_w)+\rho_w)f_{0\to j} + \sum_{q\geq j>l} (v(\mathbf{x}+e_w)+\rho_w)f_{0\to j} + \sum_{n\geq j\geq w} v(\mathbf{x}+e_j)f_{0\to j}\\ &-v(\mathbf{x}) - \sum_{1\leq j\leq l} \rho_l f_{0\to j} - \sum_{q\geq j>l} \rho_j f_{0\to j}\end{aligned}$$

$$= \sum_{1 \le j \le l} (v(\mathbf{x} + e_w) - \rho_l) f_{0 \to j} + \sum_{q \ge j > l} (v(\mathbf{x} + e_w) - \rho_j) f_{0 \to j}$$
$$+ \sum_{n \ge j \ge w} (v(\mathbf{x} + e_j) - \rho_w) f_{0 \to j} - v(\mathbf{x}) + \rho_w$$
$$= \sum_{1 \le j \le l} T_C v(\mathbf{x} + e_w + e_l) f_{0 \to j} + \sum_{n \ge j > l} T_C v(\mathbf{x} + e_w + e_j) f_{0 \to j} - T_C v(\mathbf{x} + e_w)$$

(2) Suppose that $x_j = 0$, $\forall j \le l$, and $x_j \ge 1$, $\exists j \in \{l+1, l+2, .., q\}$. Define b as the smallest j such that $x_j \ge 1$. As we assume v satisfies Property 7, the following inequality holds:

$$\sum_{1 \le j \le l} T_C v(\mathbf{x} + e_q + e_l) f_{0 \to j} + \sum_{n \ge j > l} T_C v(\mathbf{x} + e_q + e_j) f_{0 \to j} - T_C v(\mathbf{x} + e_q)$$
$$= \sum_{1 \le j \le l} (v(\mathbf{x} + e_q) - \rho_l) f_{0 \to j} + \sum_{b \ge j > l} (v(\mathbf{x} + e_q) - \rho_j) f_{0 \to j}$$
$$+ \sum_{n \ge j > b} (v(\mathbf{x} + e_q + e_j - e_b) - \rho_b) f_{0 \to j} - v(\mathbf{x} + e_q - e_b) + \rho_b$$
$$= \sum_{1 \le j \le b} v(\mathbf{x} + e_q) f_{0 \to j} + \sum_{n \ge j > b} v(\mathbf{x} + e_q + e_j - e_b) f_{0 \to j} - v(\mathbf{x} + e_q - e_b)$$
$$- \sum_{1 \le j \le l} \rho_l f_{0 \to j} - \sum_{b \ge j > l} \rho_j f_{0 \to j} + \sum_{1 \le j \le b} \rho_b f_{0 \to j}$$
$$\ge \sum_{1 \le j \le b} v(\mathbf{x} + e_w) f_{0 \to j} + \sum_{n \ge j > b} v(\mathbf{x} + e_w + e_j - e_b) f_{0 \to j} - v(\mathbf{x} + e_w - e_b)$$
$$- \sum_{1 \le j \le l} \rho_l f_{0 \to j} - \sum_{b \ge j > l} \rho_j f_{0 \to j} + \sum_{1 \le j \le b} \rho_b f_{0 \to j}$$
$$= \sum_{1 \le j \le l} (v(\mathbf{x} + e_w) - \rho_l) f_{0 \to j} + \sum_{b \ge j > l} (v(\mathbf{x} + e_w) - \rho_j) f_{0 \to j}$$
$$+ \sum_{n \ge j > b} (v(\mathbf{x} + e_w + e_j - e_b) - \rho_b) f_{0 \to j} - v(\mathbf{x} + e_w - e_b) + \rho_b$$
$$= \sum_{1 \le j \le l} T_C v(\mathbf{x} + e_w + e_l) f_{0 \to j} + \sum_{n \ge j > l} T_C v(\mathbf{x} + e_w + e_j) f_{0 \to j} - T_C v(\mathbf{x} + e_w)$$

(3) Suppose that $x_j \ge 1$, $\exists j \le l$. Again define b as the smallest j such that $x_j \ge 1$. As we

assume v satisfies Property 7, the following inequality holds:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\
&= \sum_{1\leq j\leq l} (v(\mathbf{x}+e_q+e_l-e_b)-\rho_b)f_{0\to j} + \sum_{n\geq j>l} (v(\mathbf{x}+e_q+e_j-e_b)-\rho_b)f_{0\to j}\\
&-v(\mathbf{x}+e_q-e_b)+\rho_b\\
&\geq \sum_{1\leq j\leq l} (v(\mathbf{x}+e_w+e_l-e_b)-\rho_b)f_{0\to j} + \sum_{n\geq j>l} (v(\mathbf{x}+e_w+e_j-e_b)-\rho_b)f_{0\to j}\\
&-v(\mathbf{x}+e_w-e_b)+\rho_b\\
&= \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_w)
\end{aligned}$$

Therefore $T_C v$ satisfies Property 7 when $q = w-1$. This implies that $T_C v$ satisfies Property 7 for $l \leq q \leq w-1$:

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_{q+1}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_{q+1}+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_{q+1}),\\
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_{q+1}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_{q+1}+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_{q+1})\\
&\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_{q+2}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_{q+2}+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_{q+2}),\\
&\qquad\qquad\vdots\\
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_{w-1}+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_{w-1}+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_{w-1})\\
&\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_w).
\end{aligned}$$

Summation of the above inequalities implies

$$\begin{aligned}
&\sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_q+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_q)\\
&\geq \sum_{1\leq j\leq l} T_C v(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} T_C v(\mathbf{x}+e_w+e_j)f_{0\to j} - T_C v(\mathbf{x}+e_w),
\end{aligned}$$

where $l \leq q \leq w-1$. Therefore $T_C v$ satisfies Property 7. Hence we showed that $T_C v$ satisfies

Properties 1–7; $T_C : \widehat{V} \to \widehat{V}$.

$\underline{\boldsymbol{h} \in \widehat{\boldsymbol{V}}.}$ We below show h satisfies Properties 1–7.

Property 1. h satisfies Property 1: $h(\mathbf{x}+e_w) = h'(\sum x_j + 1) = h(\mathbf{x}+e_q)$.

Property 2. h satisfies Property 2: As h is increasing in the number of projects, $h(\mathbf{x}+e_w) = h'(\sum x_j + 1) \geq h'(\sum x_j) - \rho_w = h(\mathbf{x}) - \rho_w$.

Property 3. h satisfies Property 3: $h(\mathbf{x}+e_q)-\rho_w = h'(\sum x_j+1)-\rho_w \geq h'(\sum x_j+1)-\rho_q = h(\mathbf{x}+e_w) - \rho_q$.

Property 4. h satisfies Property 4: $h(\mathbf{x}+e_q+e_l) - h(\mathbf{x}+e_q+e_d) = h'(\sum x_j + 2) - h'(\sum x_j + 2) = h'(\sum x_j + 1) - h'(\sum x_j + 1) = h(\mathbf{x}+e_l) - h(\mathbf{x}+e_d)$.

Property 5. h satisfies Property 5: $h(\mathbf{x}+e_q+e_l) - h(\mathbf{x}+e_q+e_d) = h'(\sum x_j + 2) - h'(\sum x_j + 2) = h(\mathbf{x}+e_w+e_l) - h(\mathbf{x}+e_w+e_d)$.

Property 6. h satisfies Property 6: As h is convex in the number of projects,

$$\begin{aligned}
&\sum_{1\leq j\leq l} h(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} h(\mathbf{x}+e_q+e_j)f_{0\to j} - h(\mathbf{x}+e_q)\\
&= \sum_{1\leq j\leq l} h'(\sum x_{j'}+2)f_{0\to j} + \sum_{n\geq j>l} h'(\sum x_{j'}+2)f_{0\to j} - h'(\sum x_{j'}+1)\\
&= h'(\sum x_{j'}+2) - h'(\sum x_{j'}+1) \geq h'(\sum x_{j'}+1) - h'(\sum x_{j'})\\
&= \sum_{1\leq j\leq l} h'(\sum x_{j'}+1)f_{0\to j} + \sum_{n\geq j>l} h'(\sum x_{j'}+1)f_{0\to j} - h'(\sum x_{j'})\\
&= \sum_{1\leq j\leq l} h(\mathbf{x}+e_l)f_{0\to j} + \sum_{n\geq j>l} h(\mathbf{x}+e_j)f_{0\to j} - h(\mathbf{x}).
\end{aligned}$$

Property 7. h satisfies Property 7:

$$\begin{aligned}
&\sum_{1\leq j\leq l} h(\mathbf{x}+e_q+e_l)f_{0\to j} + \sum_{n\geq j>l} h(\mathbf{x}+e_q+e_j)f_{0\to j} - h(\mathbf{x}+e_q)\\
&= \sum_{1\leq j\leq l} h'(\sum x_{j'}+2)f_{0\to j} + \sum_{n\geq j>l} h'(\sum x_{j'}+2)f_{0\to j} - h'(\sum x_{j'}+1)\\
&= \sum_{1\leq j\leq l} h(\mathbf{x}+e_w+e_l)f_{0\to j} + \sum_{n\geq j>l} h(\mathbf{x}+e_w+e_j)f_{0\to j} - h(\mathbf{x}+e_w).
\end{aligned}$$

Hence h satisfies Properties 1–7; $h \in \widehat{V}$.

$\underline{\boldsymbol{T} : \widehat{\boldsymbol{V}} \to \widehat{\boldsymbol{V}}.}$ Assume v satisfies Properties 1–7 (i.e., $v \in \widehat{V}$). We proved that $T_B v$, $T_C v$, and h satisfy Properties 1–7. It is immediate that Tv satisfies Properties 1, 4, 5, 6, and 7, as these properties are preserved by linear transformations. Next we will prove Tv satisfies Property 2 (i.e., $Tv(\mathbf{x}+e_w) \geq Tv(\mathbf{x}) - \rho_w$). Since (i) h is increasing in the number of projects in the system, (ii) $T_B v$ and $T_C v$ satisfy Property 2, and (iii) $\mu_1 + \mu_2 \leq 1$, Property 2 holds: $Tv(\mathbf{x} + e_w) = h(\mathbf{x} + e_w) + \mu_1 T_B v(\mathbf{x} + e_w) + \mu_2 T_C v(\mathbf{x} + e_w) \geq h(\mathbf{x}) + \mu_1 (T_B v(\mathbf{x}) - \rho_w) + \mu_2 (T_C v(\mathbf{x}) - \rho_w) \geq Tv(\mathbf{x}) - \rho_w$. Lastly we will prove Tv satisfies Property 3 (i.e., $Tv(\mathbf{x} + e_q) - \rho_w \geq Tv(\mathbf{x} + e_w) - \rho_q$). Since (i) $T_B v$ and $T_C v$ satisfy Property 3, (ii) $\mu_1 + \mu_2 \leq 1$, and (iii) $\rho_w \leq \rho_q$, Property 3 holds: $Tv(\mathbf{x} + e_q) = h(\mathbf{x} + e_q) + \mu_1 T_B v(\mathbf{x} + e_q) + \mu_2 T_C v(\mathbf{x} + e_q) \geq h(\mathbf{x}+e_w) + \mu_1 (T_B v(\mathbf{x} + e_w) + \rho_w - \rho_q) + \mu_2 (T_C v(\mathbf{x} + e_w) + \rho_w - \rho_q) \geq Tv(\mathbf{x}+e_w) + \rho_w - \rho_q$. Therefore Tv satisfies Properties 2 and 3, as well. Hence $Tv \in \widehat{V}$; $T : \widehat{V} \to \widehat{V}$. Following Propositions 3.1.5 and 3.1.6 in Bertsekas (2007), we verify that $\lim_{k\to\infty}(T^k v_0)(\mathbf{x}) = v^*(\mathbf{x})$ where v_0 is the zero function, v^* is the optimal cost function, and T^k refers to k compositions of operator T. Since $v_0 \in \widehat{V}$ and $T : \widehat{V} \to \widehat{V}$, we have $T^k v_0 \in \widehat{V}$, and therefore $v^* \in \widehat{V}$. □

Theorem 4.4.1 (Restated)**.** *Under Assumptions 4.4.1 and 4.4.2, the optimal portfolio strategy is a state-dependent noncongestive-promotion policy with state-dependent promote-up-to levels* $S_j^*(\mathbf{x}_{-j})$*: It is optimal to allocate resources of the experimental stage to a new product idea if and only if* $x_j < S_j^*(\mathbf{x}_{-j})$, $\forall j$, *where* $\mathbf{x}_{-j} = (x_1, .., x_{j-1}, x_{j+1}, .., x_n)$ *is an* $n-1$ *dimensional vector of the numbers of projects in categories* $k \neq j$. *The optimal policy has the following additional properties:*

i. The optimal promote-up-to level $S_j^*(\mathbf{x}_{-j})$ *weakly decreases as the number of projects in category* $k \neq j$ *increases,* $\forall j$.

ii. The optimal promote-up-to level $S_j^*(\mathbf{x}_{-j})$ *weakly decreases as the expected reward of a project in category* $k \neq j$ *increases,* $\forall j$.

iii. It is always optimal to launch a new product if there are projects available for the product launch stage.

iv. It is always optimal to allocate resources of the product launch stage to a project with highest expected reward.

v. It is never optimal to interrupt any experiment.

Proof. By Lemma 4.4.1, we know $v^* \in \widehat{V}$. Define, for $v^* \in \widehat{V}$,

$$S_j^*(\mathbf{x}_{-j}) \;=\; \min\left\{ z_j : \sum_{1\le k\le n} v^*(\mathbf{z}+e_k)f_{0\to k} - v^*(\mathbf{z}) > -c,\ z_{j'} = x_{j'}, \forall j' \neq j,\ z_j \in \mathbb{N}_0 \right\},\ \forall j.$$

Since v^* satisfies Property 6, $\sum_{1\le k\le n} v^*(\mathbf{z}+e_k)f_{0\to k} - v^*(\mathbf{z})$ is increasing in z_j. As z_j increases, since the holding cost rate h is strictly increasing, this difference will eventually cross 0. Thus a state-dependent noncongestive promotion policy is optimal. Next we will prove properties (i)-(v):

i. Pick arbitrary j and k such that $j \neq k$. We will show that the optimal promote-up-to level for category j obeys $S_j^*(\mathbf{z}_{-j}) \le S_j^*(\mathbf{x}_{-j})$, where $z_{j'} = x_{j'}$, $\forall j' \notin \{j,k\}$, and $z_k = x_k + 1$. Suppose that $S_j^*(\mathbf{z}_{-j}) > S_j^*(\mathbf{x}_{-j})$. By definition, it is optimal to initiate an experiment at $\mathbf{z}$ if $z_j < S_j^*(\mathbf{z}_{-j})$, and it is not optimal to do so at $\mathbf{x}$ if $x_j = S_j^*(\mathbf{x}_{-j}) < S_j^*(\mathbf{z}_{-j})$. But we have a contradiction when $z_j = x_j = S_j^*(\mathbf{x}_{-j})$; if it is optimal to initiate an experiment at $\mathbf{z}$, it should also be optimal to do so at state $\mathbf{x}$ (due to Property 6). Thus we must have $S_j^*(\mathbf{z}_{-j}) \le S_j^*(\mathbf{x}_{-j})$.

ii. Pick arbitrary j, k, and k' such that $j \notin \{k,k'\}$ and $k' < k$. We will show that the optimal promote-up-to level for category j obeys $S_j^*(\mathbf{z}_{-j}) \le S_j^*(\mathbf{x}_{-j})$, where $z_{j'} = x_{j'}$, $\forall j' \notin \{j,k,k'\}$, $z_k + 1 = x_k$ and $z_{k'} = x_{k'} + 1$. Suppose that $S_j^*(\mathbf{z}_{-j}) > S_j^*(\mathbf{x}_{-j})$. By definition, it is optimal to initiate an experiment at $\mathbf{z}$ if $z_j < S_j^*(\mathbf{z}_{-j})$, and it is not optimal to do so at $\mathbf{x}$ if $x_j = S_j^*(\mathbf{x}_{-j}) < S_j^*(\mathbf{z}_{-j})$. But we have a contradiction when $z_j = x_j = S_j^*(\mathbf{x}_{-j})$; if it is optimal to initiate an experiment at $\mathbf{z}$, it should also be optimal to do so at $\mathbf{x}$ (due to Property 7). Thus we must have $S_j^*(\mathbf{z}_{-j}) \le S_j^*(\mathbf{x}_{-j})$.

iii. Suppose that $\exists k$ such that $x_k > 0$. Due to Property 2, it is always optimal to launch a new product:

$$\begin{aligned} T_C v(\mathbf{x}) &= \min\left\{ v(\mathbf{x}), \min_{1\le j\le n \ s.t.\ \mathbf{x}\ge e_j} v(\mathbf{x}-e_j) - \rho_j \right\} \\ &= \min_{1\le j\le n \ s.t.\ \mathbf{x}\ge e_j} v(\mathbf{x}-e_j) - \rho_j. \end{aligned}$$

iv. Suppose that $\exists k$ such that $x_k > 0$. Point (iii) implies that

$$T_C v(\mathbf{x}) = \min_{1 \le j \le n \ s.t.\ \mathbf{x} \ge e_j} v(\mathbf{x} - e_j) - \rho_j.$$

Due to Property 3, it is always optimal to choose a project with highest expected reward: $T_C v(\mathbf{x}) = v(\mathbf{x} - e_{j^*}) - \rho_{j^*}$ where j^* is the smallest j such that $x_j > 0$.

v. Lastly, we will prove it is never optimal to interrupt any experiment. Assume that an experiment is optimally initiated at a given state $\mathbf{x}$. We will then consider the following two cases:

- Suppose that $\exists j$ such that $x_j > 0$. Point (iii) implies that it is optimal to launch a new product at state $\mathbf{x}$. Suppose that a new product is placed on the market before the experiment is complete. Thus the system moves to a state $\mathbf{z}$ where $z_{j^*} = x_{j^*} - 1$ and $z_j = x_j$, $\forall j \neq j^*$ (j^* is the category with highest expected reward among available categories). Due to Property 6, it is optimal to initiate an experiment at state $\mathbf{z}$; the experiment initiated at $\mathbf{x}$ can be resumed at $\mathbf{z}$. Now suppose that a new product is placed on the market after the experiment is complete. But then the experiment is not interrupted.
- Suppose that $x_j = 0$, $\forall j$. The system can move to a new state only after the experiment is complete. Therefore the experiment is not interrupted.

□

C.3 Proofs of the Results in Section 4.5.

We will use the following auxiliary lemma to prove Lemma 4.5.1:

Lemma C.3.1. *A real-valued function on $\mathbb{N}_0^n$ satisfying Property 10 also satisfies Properties 11–13.*

Proof. Assuming a real-valued function g on $\mathbb{N}^n$ satisfies Property 10, we want to show g satisfies Properties 11, 12, and 13. First, we will show g satisfies Property 11. Pick arbitrary

l, q, and w such that $l, q, w \in \{1, 2, .., m+1\}$ and $l \le q < w \le m+1$. As we assume g satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
g(\mathbf{x}+e_l+e_q) - g(\mathbf{x}+e_{l-1}+e_q) &\ge g(\mathbf{x}+e_l+e_{q+1}) - g(\mathbf{x}+e_{l-1}+e_{q+1}) \\
g(\mathbf{x}+e_l+e_{q+1}) - g(\mathbf{x}+e_{l-1}+e_{q+1}) &\ge g(\mathbf{x}+e_l+e_{q+2}) - g(\mathbf{x}+e_{l-1}+e_{q+2}) \\
&\vdots \\
g(\mathbf{x}+e_l+e_{w-2}) - g(\mathbf{x}+e_{l-1}+e_{w-2}) &\ge g(\mathbf{x}+e_l+e_{w-1}) - g(\mathbf{x}+e_{l-1}+e_{w-1}) \\
g(\mathbf{x}+e_l+e_{w-1}) - g(\mathbf{x}+e_{l-1}+e_{w-1}) &\ge g(\mathbf{x}+e_l+e_w) - g(\mathbf{x}+e_{l-1}+e_w)
\end{aligned}$$

Summation of the above inequalities implies $g(\mathbf{x}+e_l+e_q) - g(\mathbf{x}+e_{l-1}+e_q) \ge g(\mathbf{x}+e_l+e_w) - g(\mathbf{x}+e_{l-1}+e_w)$. Thus g satisfies Property 11.

Second, we will show g satisfies Property 12. Pick arbitrary l, q, and w such that $l, q, w \in \{0, 1, .., m+1\}$ and $0 \le l < q \le w-1$. As we assume g satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
g(\mathbf{x}+e_w+e_l) - g(\mathbf{x}+e_{w-1}+e_l) &\ge g(\mathbf{x}+e_w+e_{l+1}) - g(\mathbf{x}+e_{w-1}+e_{l+1}) \\
g(\mathbf{x}+e_w+e_{l+1}) - g(\mathbf{x}+e_{w-1}+e_{l+1}) &\ge g(\mathbf{x}+e_w+e_{l+2}) - g(\mathbf{x}+e_{w-1}+e_{l+2}) \\
&\vdots \\
g(\mathbf{x}+e_w+e_{q-2}) - g(\mathbf{x}+e_{w-1}+e_{q-2}) &\ge g(\mathbf{x}+e_w+e_{q-1}) - g(\mathbf{x}+e_{w-1}+e_{q-1}) \\
g(\mathbf{x}+e_w+e_{q-1}) - g(\mathbf{x}+e_{w-1}+e_{q-1}) &\ge g(\mathbf{x}+e_w+e_q) - g(\mathbf{x}+e_{w-1}+e_q)
\end{aligned}$$

Summation of the above inequalities implies $g(\mathbf{x}+e_w+e_l) - g(\mathbf{x}+e_{w-1}+e_l) \ge g(\mathbf{x}+e_w+e_q) - g(\mathbf{x}+e_{w-1}+e_q)$. Thus g satisfies Property 12.

Lastly, we will show g satisfies Property 13. Since g satisfies Property 11, the following inequality holds: $g(\mathbf{x}+e_l+e_l) - g(\mathbf{x}+e_{l-1}+e_l) \ge g(\mathbf{x}+e_l) - g(\mathbf{x}+e_{l-1})$ for $l \in \{1, .., m\}$. Since g satisfies Property 12, the following inequality holds: $g(\mathbf{x}+e_w) - g(\mathbf{x}+e_{w-1}) \ge g(\mathbf{x}+e_w+e_{w-1}) - g(\mathbf{x}+e_{w-1}+e_{w-1})$ for $w \in \{2, .., m\}$. When $l = w \ne 1$, summation of these two inequalities implies $g(\mathbf{x}+e_l+e_l) - g(\mathbf{x}+e_{l-1}+e_l) \ge g(\mathbf{x}+e_l+e_{l-1}) - g(\mathbf{x}+e_{l-1}+e_{l-1})$ for $l \in \{2, .., m\}$. Now suppose that $l = 1$. Since g satisfies Property 11, the following inequality holds: $g(\mathbf{x}+2e_1) - g(\mathbf{x}+e_1) \geq g(\mathbf{x}+e_1) - g(\mathbf{x})$. Thus g satisfies Property 13. □

Lemma 4.5.1 (Restated). *Under Assumption 4.5.1, if $v \in \widetilde{V}$, then $Tv \in \widetilde{V}$, where $Tv(\mathbf{x}) = h(\mathbf{x}) + \lambda T_A v(\mathbf{x}) + \sum_{1\leq i\leq m} \mu_i T_{B,i} v(\mathbf{x}) + \mu_{m+1} T_C v(\mathbf{x})$. Furthermore, the optimal cost function v^* is an element of $\widetilde{V}$.*

Proof. Define $\widetilde{V}$ as the set of real-valued functions on $\mathbb{N}_0^n$ that satisfy Properties 8–13. Also, define the operator T on the set of real-valued functions v as follows: $Tv(\mathbf{x}) = h(\mathbf{x}) + \lambda T_A v(\mathbf{x}) + \sum_{1\leq i\leq m} \mu_i T_{B,i} v(\mathbf{x}) + \mu_{m+1} T_C v(\mathbf{x})$. We below show that $T_A : \widetilde{V} \to \widetilde{V}$, $T_{B,i} : \widetilde{V} \to \widetilde{V}$, $\forall i$, $T_C : \widetilde{V} \to \widetilde{V}$, and $h \in \widetilde{V}$. We will then prove that $T : \widetilde{V} \to \widetilde{V}$.

$T_A : \widetilde{V} \to \widetilde{V}$. Assuming v satisfies Properties 8–13, we below show $T_A v$ satisfies Properties 8–13.

Property 8. We will prove $T_A v$ satisfies Property 8 (i.e., $T_A v(\mathbf{x}+e_q) \geq T_A v(\mathbf{x}+e_w)$, $\forall q, w \in \{1, 2, .., m\}$ where $q < w$). Pick arbitrary q and w such that $q, w \in \{1, 2, .., m\}$ and $q < w$. There are two different scenarios we need to consider depending on the optimal action at $T_A v(\mathbf{x}+e_q)$ (if this inequality holds under a suboptimal action of $T_A v(\mathbf{x}+e_w)$, it also holds under the optimal action of this operator, and thus we do not enforce the optimal action at this operator):

(1) Suppose that $T_A v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q)$. As we assume v satisfies Property 8, the following inequalities hold: $T_A v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q) \geq v(\mathbf{x}+e_w) \geq T_A v(\mathbf{x}+e_w)$.

(2) Suppose that $\mathbf{x}+e_q \geq e_l$ and $T_A v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q-e_l)$ where $l \geq 1$. Also, suppose that $l \neq q$. Hence we should have $x_l > 0$. As we assume v satisfies Property 8, the following inequalities hold: $T_A v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q-e_l) \geq v(\mathbf{x}+e_w-e_l) \geq T_A v(\mathbf{x}+e_w)$. Now suppose that $l = q$. Then it is easy to verify that $T_A v(\mathbf{x}+e_q) = v(\mathbf{x}) \geq T_A v(\mathbf{x}+e_w)$.

Therefore $T_A v$ satisfies Property 8.

Property 9. We will prove $T_A v$ satisfies Property 9 (i.e., $T_A v(\mathbf{x}+e_m) \geq T_A v(\mathbf{x}) - \rho$). There are two different scenarios we need to consider depending on the optimal action at $T_A v(\mathbf{x}+e_m)$:

(1) Suppose that $T_A v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m)$. As we assume v satisfies Property 9, the following inequalities hold: $T_A v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m) \geq v(\mathbf{x}) - \rho \geq T_A v(\mathbf{x}) - \rho$.

(2) Suppose that $\mathbf{x}+e_m \geq e_l$ and $T_A v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m-e_l)$ where $l \geq 1$. Also, suppose that $l \neq m$. Hence we should have $x_l > 0$. As we assume v satisfies Property 9, the following inequalities hold: $T_A v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m-e_l) \geq v(\mathbf{x}-e_l)-\rho \geq T_A v(\mathbf{x})-\rho$. Now suppose that $l = m$. Then it is easy to verify that $T_A v(\mathbf{x}+e_m) = v(\mathbf{x}) \geq v(\mathbf{x}) - \rho \geq T_A v(\mathbf{x}) - \rho$.

Therefore $T_A v$ satisfies Property 9.

Property 10. We will prove $T_A v$ satisfies Property 10 (i.e., $T_A v(\mathbf{x}+e_q+e_{w-1}) - T_A v(\mathbf{x}+e_{q-1}+e_{w-1}) \geq T_A v(\mathbf{x}+e_q+e_w) - T_A v(\mathbf{x}+e_{q-1}+e_w)$, $\forall q, w \in \{1, 2, .., m+1\}$ where $q \neq w$). Pick arbitrary q and w such that $q, w \in \{1, 2, .., m+1\}$ and $q \neq w$. There are four different scenarios we need to consider depending on the optimal actions at $T_A v(\mathbf{x}+e_q+e_{w-1})$ and $T_A v(\mathbf{x}+e_{q-1}+e_w)$ (if this inequality holds under suboptimal actions of $T_A v(\mathbf{x}+e_{q-1}+e_{w-1})$ and/or $T_A v(\mathbf{x}+e_q+e_w)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators):

(1) Suppose that $T_A v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1})$ and $T_A v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w)$. As we assume v satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
&T_A v(\mathbf{x}+e_q+e_{w-1}) - T_A v(\mathbf{x}+e_{q-1}+e_{w-1}) \\
&\geq v(\mathbf{x}+e_q+e_{w-1}) - v(\mathbf{x}+e_{q-1}+e_{w-1}) \\
&\geq v(\mathbf{x}+e_q+e_w) - v(\mathbf{x}+e_{q-1}+e_w) \\
&\geq T_A v(\mathbf{x}+e_q+e_w) - T_A v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}$$

(2) Suppose that $\mathbf{x}+e_{q-1}+e_w \geq e_l$, $T_A v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1})$, and $T_A v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w-e_l)$ where $l \geq 1$. Also, suppose that $l < w$. Since $\mathbf{x}+e_{q-1}+e_w \geq e_l$, we should have $\mathbf{x}+e_{q-1} \geq e_l$. As we assume v satisfies Properties

10 and 12, the following inequalities hold:

$$T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1})$$
$$\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_l)$$
$$\geq v(\mathbf{x} + e_{q-1} + e_{w-1}) + v(\mathbf{x} + e_q + e_w)$$
$$-v(\mathbf{x} + e_{q-1} + e_w) - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_l)$$
$$\geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w - e_l)$$
$$\geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)$$

Now suppose that $l \geq w$. As we assume v satisfies Property 8, if a project is to be terminated, it is optimal to choose this from the earliest possible stage. Hence we should have $l = w$. Again as we assume v satisfies Property 8, the following inequalities hold:

$$T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1})$$
$$\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1})$$
$$\geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1})$$
$$\geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)$$

(3) Suppose that $\mathbf{x} + e_q + e_{w-1} \geq e_l$, $T_A v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1} - e_l)$ where $l \geq 1$, and $T_A v(\mathbf{x} + e_{q-1} + e_w) = v(\mathbf{x} + e_{q-1} + e_w)$. Also, suppose that $l < q$. Since $\mathbf{x} + e_q + e_{w-1} \geq e_l$, we should have $\mathbf{x} + e_{w-1} \geq e_l$. As we assume v satisfies Properties 10 and 12, the following inequalities hold:

$$T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1})$$
$$\geq v(\mathbf{x} + e_q + e_{w-1} - e_l) - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_l)$$
$$\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1} + e_{w-1})$$
$$\geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w)$$
$$\geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)$$

Now suppose that $l \geq q$. As we assume v satisfies Property 8, we should have $l = q$. Again as we assume v satisfies Property 8, the following inequalities hold:

$$
\begin{aligned}
& T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_{w-1}) - v(\mathbf{x} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w) \\
& \geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

(4) Suppose that $\mathbf{x} + e_q + e_{w-1} \geq e_l$, $\mathbf{x} + e_{q-1} + e_w \geq e_d$, $T_A v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1} - e_l)$ where $l \geq 1$, and $T_A v(\mathbf{x} + e_{q-1} + e_w) = v(\mathbf{x} + e_{q-1} + e_w - e_d)$ where $d \geq 1$. First, suppose that $q < w$ and $l \neq q$. As we assume v satisfies Property 8, we should have $l < q$ and $x_l > 0$. Also we should have $l = d$. As we assume v satisfies Property 10, the following inequalities hold:

$$
\begin{aligned}
& T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_q + e_{w-1} - e_l) - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_l) \\
& \geq v(\mathbf{x} + e_q + e_w - e_l) - v(\mathbf{x} + e_{q-1} + e_w - e_l) \\
& \geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Second, suppose that $q < w$ and $l = q$. As we assume v satisfies Property 8, we should have $x_i = 0$, $\forall i < q$. Therefore, $d = q - 1$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
& T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_{w-1}) - v(\mathbf{x} + e_{w-1}) \\
& = v(\mathbf{x} + e_w) - v(\mathbf{x} + e_w) \\
& \geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Third, suppose that $q > w$ and $d \neq w$. As we assume v satisfies Property 8, we should have $d < w$ and $x_d > 0$. Also we should have $d = l$. As we assume v satisfies Property

10, the following inequalities hold:

$$
\begin{aligned}
&T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1} - e_d) - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_d) \\
&\geq v(\mathbf{x} + e_q + e_w - e_d) - v(\mathbf{x} + e_{q-1} + e_w - e_d) \\
&\geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Lastly, suppose that $q > w$ and $d = w$. As we assume v satisfies Property 8, we should have $x_i = 0$, $\forall i < w$. Therefore, $l = w - 1$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
&T_A v(\mathbf{x} + e_q + e_{w-1}) - T_A v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q) - v(\mathbf{x} + e_{q-1}) \\
&\geq T_A v(\mathbf{x} + e_q + e_w) - T_A v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Thus $T_A v$ satisfies Property 10. By Lemma C.3.1, $T_A v$ also satisfies Properties 11–13. Hence we showed that $T_A v$ satisfies Properties 8–13; $T_A : \widetilde{V} \to \widetilde{V}$.

$\boldsymbol{T_{B,i} : \widetilde{V} \to \widetilde{V}}$**.** Assuming v satisfies Properties 8–13, we below show $T_{B,i}v$ satisfies Properties 8–13, $\forall i$.

Property 8. We will prove $T_{B,i}v$ satisfies Property 8 (i.e., $T_{B,i}v(\mathbf{x} + e_q) \geq T_{B,i}v(\mathbf{x} + e_w)$, $\forall q, w \in \{1, 2, .., m\}$ where $q < w$). Pick arbitrary q and w such that $q, w \in \{1, 2, .., m\}$ and $q < w$. There are two different scenarios we need to consider depending on the optimal action at $T_{B,i}v(\mathbf{x} + e_q)$ (if this inequality holds under a suboptimal action of $T_{B,i}v(\mathbf{x} + e_w)$, it also holds under the optimal action of this operator, and thus we do not enforce the optimal action at this operator):

(1) Suppose that $T_{B,i}v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q)$. As we assume v satisfies Property 8, the following inequalities hold: $T_{B,i}v(\mathbf{x} + e_q) = v(\mathbf{x} + e_q) \geq v(\mathbf{x} + e_w) \geq T_{B,i}v(\mathbf{x} + e_w)$.

(2) Suppose that $\mathbf{x} + e_q \geq e_{i-1}$ and $T_{B,i}v(\mathbf{x} + e_q) = v(\mathbf{x} + e_q - e_{i-1} + e_i) + c_i$. Also, suppose that $i \neq q + 1$. Since $\mathbf{x} + e_q \geq e_{i-1}$, we should have $\mathbf{x} \geq e_{i-1}$. As we assume v satisfies

Property 8, the following inequalities hold: $T_{B,i}v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q-e_{i-1}+e_i)+c_i \geq v(\mathbf{x}+e_w-e_{i-1}+e_i)+c_i \geq T_{B,i}v(\mathbf{x}+e_w)$. Now suppose that $i = q+1$. As we assume v satisfies Property 8 and $q < w$, it is easy to verify that $T_{B,i}v(\mathbf{x}+e_q) = v(\mathbf{x}+e_{q+1})+c_{q+1} \geq v(\mathbf{x}+e_w) \geq T_{B,i}v(\mathbf{x}+e_w)$.

Therefore $T_{B,i}v$ satisfies Property 8.

Property 9. We will prove $T_{B,i}v$ satisfies Property 9 (i.e., $T_{B,i}v(\mathbf{x}+e_m) \geq T_{B,i}v(\mathbf{x})-\rho$). There are two different scenarios we need to consider depending on the optimal action at $T_{B,i}v(\mathbf{x}+e_m)$:

(1) Suppose that $T_{B,i}v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m)$. As we assume v satisfies Property 9, the following inequalities hold: $T_{B,i}v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m) \geq v(\mathbf{x})-\rho \geq T_{B,i}v(\mathbf{x})-\rho$.

(2) Suppose that $\mathbf{x}+e_m \geq e_{i-1}$ and $T_{B,i}v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m-e_{i-1}+e_i)+c_i$. Since $i \leq m$, we should have $\mathbf{x} \geq e_{i-1}$. As we assume v satisfies Property 9, the following inequalities hold: $T_{B,i}v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m-e_{i-1}+e_i)+c_i \geq v(\mathbf{x}-e_{i-1}+e_i)+c_i-\rho \geq T_{B,i}v(\mathbf{x})-\rho$.

Therefore $T_{B,i}v$ satisfies Property 9.

Property 10. We will prove $T_{B,i}v$ satisfies Property 10 (i.e., $T_{B,i}v(\mathbf{x}+e_q+e_{w-1})-T_{B,i}v(\mathbf{x}+e_{q-1}+e_{w-1}) \geq T_{B,i}v(\mathbf{x}+e_q+e_w)-T_{B,i}v(\mathbf{x}+e_{q-1}+e_w)$, $\forall q,w \in \{1,2,..,m+1\}$ where $q \neq w$). Pick arbitrary q and w such that $q,w \in \{1,2,..,m+1\}$ and $q \neq w$. There are four different scenarios we need to consider depending on the optimal actions at $T_{B,i}v(\mathbf{x}+e_q+e_{w-1})$ and $T_{B,i}v(\mathbf{x}+e_{q-1}+e_w)$ (if this inequality holds under suboptimal actions of $T_{B,i}v(\mathbf{x}+e_{q-1}+e_{w-1})$ and/or $T_{B,i}v(\mathbf{x}+e_q+e_w)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators):

(1) Suppose that $T_{B,i}v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1})$ and $T_{B,i}v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w)$. As we assume v satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
&T_{B,i}v(\mathbf{x}+e_q+e_{w-1})-T_{B,i}v(\mathbf{x}+e_{q-1}+e_{w-1})\\
&\geq v(\mathbf{x}+e_q+e_{w-1})-v(\mathbf{x}+e_{q-1}+e_{w-1})\\
&\geq v(\mathbf{x}+e_q+e_w)-v(\mathbf{x}+e_{q-1}+e_w)\\
&\geq T_{B,i}v(\mathbf{x}+e_q+e_w)-T_{B,i}v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}$$

(2) Suppose that $\mathbf{x} + e_{q-1} + e_w \geq e_{i-1}$, $T_{B,i}v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1})$, and $T_{B,i}v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w-e_{i-1}+e_i)+c_i$. Also, suppose that $i \neq q$. Since $\mathbf{x} + e_{q-1} + e_w \geq e_{i-1}$, we should have $\mathbf{x} + e_w \geq e_{i-1}$. As we assume v satisfies Property 10, the following inequalities hold:

$$
\begin{aligned}
&T_{B,i}v(\mathbf{x} + e_q + e_{w-1}) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w) \\
&\geq v(\mathbf{x} + e_q + e_w - e_{i-1} + e_i) + c_i - v(\mathbf{x} + e_{q-1} + e_w - e_{i-1} + e_i) - c_i \\
&\geq T_{B,i}v(\mathbf{x} + e_q + e_w) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Now suppose that $i = q$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
&T_{B,i}v(\mathbf{x} + e_q + e_{w-1}) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_q + e_{w-1}) - c_q \\
&= v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_q + e_w) - c_q \\
&\geq T_{B,i}v(\mathbf{x} + e_q + e_w) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

(3) Suppose that $\mathbf{x}+e_q+e_{w-1} \geq e_{i-1}$, $T_{B,i}v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1}-e_{i-1}+e_i)+c_i$, and $T_{B,i}v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w)$. Also, suppose that $i \neq w$. Since $\mathbf{x}+e_q+e_{w-1} \geq e_{i-1}$, we should have $\mathbf{x} + e_q \geq e_{i-1}$. As we assume v satisfies Property 10, the following inequalities hold:

$$
\begin{aligned}
&T_{B,i}v(\mathbf{x} + e_q + e_{w-1}) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1} - e_{i-1} + e_i) + c_i - v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_w - e_{i-1} + e_i) + c_i - v(\mathbf{x} + e_q + e_w) \\
&+v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_w - e_{i-1} + e_i) + c_i - v(\mathbf{x} + e_{q-1} + e_w) \\
&\geq T_{B,i}v(\mathbf{x} + e_q + e_w) - T_{B,i}v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Now suppose that $i = w$. Then it is easy to verify the following inequalities:

$$\begin{aligned}
&T_{B,i}v(\mathbf{x}+e_q+e_{w-1}) - T_{B,i}v(\mathbf{x}+e_{q-1}+e_{w-1})\\
&\geq v(\mathbf{x}+e_q+e_w)+c_w - v(\mathbf{x}+e_{q-1}+e_w) - c_w\\
&= v(\mathbf{x}+e_q+e_w) - v(\mathbf{x}+e_{q-1}+e_w)\\
&\geq T_{B,i}v(\mathbf{x}+e_q+e_w) - T_{B,i}v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}$$

(4) Suppose that $\mathbf{x}+e_q+e_{w-1} \geq e_{i-1}$, $\mathbf{x}+e_{q-1}+e_w \geq e_{i-1}$, $T_{B,i}v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1}-e_{i-1}+e_i)+c_i$, and $T_{B,i}v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w-e_{i-1}+e_i)+c_i$. Since $\mathbf{x}+e_q+e_{w-1} \geq e_{i-1}$, $\mathbf{x}+e_{q-1}+e_w \geq e_{i-1}$, and $q \neq w$, we should have $\mathbf{x} \geq e_{i-1}$. As we assume v satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
&T_{B,i}v(\mathbf{x}+e_q+e_{w-1}) - T_{B,i}v(\mathbf{x}+e_{q-1}+e_{w-1})\\
&\geq v(\mathbf{x}+e_q+e_{w-1}-e_{i-1}+e_i)+c_i - v(\mathbf{x}+e_{q-1}+e_{w-1}-e_{i-1}+e_i) - c_i\\
&\geq v(\mathbf{x}+e_q+e_w-e_{i-1}+e_i)+c_i - v(\mathbf{x}+e_{q-1}+e_w-e_{i-1}+e_i) - c_i\\
&\geq T_{B,i}v(\mathbf{x}+e_q+e_w) - T_{B,i}v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}$$

Thus $T_{B,i}v$ satisfies Property 10. By Lemma C.3.1, $T_{B,i}v$ also satisfies Properties 11–13. Hence we showed that $T_{B,i}v$ satisfies Properties 8–13; $T_{B,i} : \widetilde{V} \to \widetilde{V}$.

$T_C : \widetilde{V} \to \widetilde{V}$. Assuming v satisfies Properties 8–13, we below show $T_C v$ satisfies Properties 8–13.

Property 8. We will prove $T_C v$ satisfies Property 8 (i.e., $T_C v(\mathbf{x}+e_q) \geq T_C v(\mathbf{x}+e_w)$, $\forall q, w \in \{1, 2, .., m\}$ where $q < w$). Pick arbitrary q and w such that $q, w \in \{1, 2, .., m\}$ and $q < w$. There are two different scenarios we need to consider depending on the optimal action at $T_C v(\mathbf{x}+e_q)$ (if this inequality holds under a suboptimal action of $T_C v(\mathbf{x}+e_w)$, it also holds under the optimal action of this operator, and thus we do not enforce the optimal action at this operator):

(1) Suppose that $T_C v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q)$. As we assume v satisfies Property 8, the following inequalities hold: $T_C v(\mathbf{x}+e_q) = v(\mathbf{x}+e_q) \geq v(\mathbf{x}+e_w) \geq T_C v(\mathbf{x}+e_w)$.

(2) Suppose that $\mathbf{x} + e_q \geq e_m$ and $T_C v(\mathbf{x} + e_q) = v(\mathbf{x} + e_q - e_m) - \rho$. Since $q < w \leq m$, we should have $q < m$ and $x_m > 0$. Therefore, as we assume v satisfies Property 8, the following inequalities hold: $T_C v(\mathbf{x} + e_q) = v(\mathbf{x} + e_q - e_m) - \rho \geq v(\mathbf{x} + e_w - e_m) - \rho \geq T_C v(\mathbf{x} + e_w)$.

Therefore $T_C v$ satisfies Property 8.

Property 9. We will prove $T_C v$ satisfies Property 9 (i.e., $T_C v(\mathbf{x} + e_m) \geq T_C v(\mathbf{x}) - \rho$). There are two different scenarios we need to consider depending on the optimal action at $T_C v(\mathbf{x} + e_m)$:

(1) Suppose that $T_C v(\mathbf{x}+e_m) = v(\mathbf{x}+e_m)$. As we assume v satisfies Property 9, the following inequalities hold: $T_C v(\mathbf{x} + e_m) = v(\mathbf{x} + e_m) \geq v(\mathbf{x}) - \rho \geq T_C v(\mathbf{x}) - \rho$.

(2) Suppose that $T_C v(\mathbf{x} + e_m) = v(\mathbf{x}) - \rho$. Then, it is easy to verify that $T_C v(\mathbf{x} + e_m) = v(\mathbf{x}) - \rho \geq T_C v(\mathbf{x}) - \rho$.

Therefore $T_C v$ satisfies Property 9.

Property 10. We will prove $T_C v$ satisfies Property 10 (i.e., $T_C v(\mathbf{x}+e_q+e_{w-1}) - T_C v(\mathbf{x}+e_{q-1}+e_{w-1}) \geq T_C v(\mathbf{x}+e_q+e_w) - T_C v(\mathbf{x}+e_{q-1}+e_w)$, $\forall q, w \in \{1, 2, .., m+1\}$ where $q \neq w$). Pick arbitrary q and w such that $q, w \in \{1, 2, .., m+1\}$ and $q \neq w$. There are four different scenarios we need to consider depending on the optimal actions at $T_C v(\mathbf{x} + e_q + e_{w-1})$ and $T_C v(\mathbf{x}+e_{q-1}+e_w)$ (if this inequality holds under suboptimal actions of $T_C v(\mathbf{x}+e_{q-1}+e_{w-1})$ and/or $T_C v(\mathbf{x} + e_q + e_w)$, it also holds under optimal actions of these operators, and thus we do not enforce the optimal actions at these operators):

(1) Suppose that $T_C v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1})$ and $T_C v(\mathbf{x} + e_{q-1} + e_w) = v(\mathbf{x} + e_{q-1} + e_w)$. As we assume v satisfies Property 10, the following inequalities hold:

$$\begin{aligned}
&T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w) \\
&\geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}$$

(2) Suppose that $\mathbf{x}+e_{q-1}+e_w \geq e_m$, $T_C v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1})$, and $T_C v(\mathbf{x}+e_{q-1}+e_w) = v(\mathbf{x}+e_{q-1}+e_w-e_m)-\rho$. As we assume v satisfies Property 9, it is always optimal to launch a new product if it is feasible. But $T_C v(\mathbf{x}+e_q+e_{w-1}) = v(\mathbf{x}+e_q+e_{w-1})$; we should have $x_m = 0$, $q \neq m$, and $w \neq m+1$. Since $\mathbf{x}+e_{q-1}+e_w \geq e_m$, we should also have $q = m+1$ and/or $w = m$. First, suppose that $q = m+1$ and $w \neq m$. Then it is easy to verify the following inequalities (recall that e_{m+1} is a zero vector of dimension m):

$$
\begin{aligned}
& T_C v(\mathbf{x}+e_q+e_{w-1}) - T_C v(\mathbf{x}+e_{q-1}+e_{w-1}) \\
& \geq v(\mathbf{x}+e_q+e_{w-1}) - v(\mathbf{x}+e_{w-1}) + \rho \\
& = v(\mathbf{x}+e_{w-1}) - v(\mathbf{x}+e_{w-1}) + \rho \\
& = v(\mathbf{x}+e_w) - v(\mathbf{x}+e_w) + \rho \\
& = v(\mathbf{x}+e_q+e_w) - v(\mathbf{x}+e_w) + \rho \\
& \geq T_C v(\mathbf{x}+e_q+e_w) - T_C v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}
$$

Second, suppose that $q \neq m+1$ and $w = m$. Thus $q < m$. As we assume v satisfies Property 11, the following inequalities hold:

$$
\begin{aligned}
& T_C v(\mathbf{x}+e_q+e_{w-1}) - T_C v(\mathbf{x}+e_{q-1}+e_{w-1}) \\
& \geq v(\mathbf{x}+e_q+e_{w-1}) - v(\mathbf{x}+e_{q-1}+e_{w-1}) \\
& \geq v(\mathbf{x}+e_q) - \rho - v(\mathbf{x}+e_{q-1}) + \rho \\
& \geq T_C v(\mathbf{x}+e_q+e_w) - T_C v(\mathbf{x}+e_{q-1}+e_w)
\end{aligned}
$$

Lastly, suppose that $q = m+1$ and $w = m$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
&T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q + e_{w-1}) - v(\mathbf{x} + e_{w-1}) + \rho \\
&= v(\mathbf{x} + e_{w-1}) - v(\mathbf{x} + e_{w-1}) + \rho \\
&= v(\mathbf{x} + e_w) - v(\mathbf{x} + e_w) + \rho \\
&= v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_w) + \rho \\
&\geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

(3) Suppose that $\mathbf{x} + e_q + e_{w-1} \geq e_m$, $T_C v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1} - e_m) - \rho$, and $T_C v(\mathbf{x} + e_{q-1} + e_w) = v(\mathbf{x} + e_{q-1} + e_w)$. As we assume v satisfies Property 9, we should have $x_m = 0$, $q \neq m + 1$, and $w \neq m$. Since $\mathbf{x} + e_q + e_{w-1} \geq e_m$, we should also have $q = m$ and/or $w = m + 1$. First, suppose that $q = m$ and $w \neq m + 1$. Thus $w < m$. As we assume v satisfies Property 11, the following inequalities hold:

$$
\begin{aligned}
&T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_{w-1}) - \rho - v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_w) - \rho - v(\mathbf{x} + e_{q-1} + e_w) \\
&\geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Second, suppose that $q \neq m$ and $w = m + 1$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
&T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
&\geq v(\mathbf{x} + e_q) - \rho - v(\mathbf{x} + e_{q-1}) + \rho \\
&= v(\mathbf{x} + e_q) - v(\mathbf{x} + e_{q-1}) \\
&= v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w) \\
&\geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Lastly, suppose that $q = m$ and $w = m + 1$. Then it is easy to verify the following inequalities:

$$
\begin{aligned}
& T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_q) - \rho - v(\mathbf{x} + e_{q-1}) + \rho \\
& = v(\mathbf{x} + e_q + e_w) - v(\mathbf{x} + e_{q-1} + e_w) \\
& \geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

(4) Suppose that $\mathbf{x} + e_q + e_{w-1} \geq e_m$, $\mathbf{x} + e_{q-1} + e_w \geq e_m$, $T_C v(\mathbf{x} + e_q + e_{w-1}) = v(\mathbf{x} + e_q + e_{w-1} - e_m) - \rho$, and $T_C v(\mathbf{x} + e_{q-1} + e_w) = v(\mathbf{x} + e_{q-1} + e_w - e_m) - \rho$. Since $\mathbf{x} + e_q + e_{w-1} \geq e_m$, $\mathbf{x} + e_{q-1} + e_w \geq e_m$, and $q \neq w$, we should have $\mathbf{x} \geq e_m$. As we assume v satisfies Property 10, the following inequalities hold:

$$
\begin{aligned}
& T_C v(\mathbf{x} + e_q + e_{w-1}) - T_C v(\mathbf{x} + e_{q-1} + e_{w-1}) \\
& \geq v(\mathbf{x} + e_q + e_{w-1} - e_m) - \rho - v(\mathbf{x} + e_{q-1} + e_{w-1} - e_m) + \rho \\
& \geq v(\mathbf{x} + e_q + e_w - e_m) - \rho - v(\mathbf{x} + e_{q-1} + e_w - e_m) - \rho \\
& \geq T_C v(\mathbf{x} + e_q + e_w) - T_C v(\mathbf{x} + e_{q-1} + e_w)
\end{aligned}
$$

Thus $T_C v$ satisfies Property 10. By Lemma C.3.1, $T_C v$ also satisfies Properties 11–13. Hence, we showed that $T_C v$ satisfies Properties 8–13; $T_C : \widetilde{V} \to \widetilde{V}$.

$\boldsymbol{h \in \widetilde{V}}$**.** We below show h satisfies Properties 8–13.

Property 8. h satisfies Property 8: $h(\mathbf{x} + e_q) = h'(\sum_i x_i + 1) = h(\mathbf{x} + e_w)$.

Property 9. h satisfies Property 9: As h is increasing in the number of projects in the system, $h(\mathbf{x} + e_m) = h'(\sum_i x_i + 1) \geq h'(\sum_i x_i) - \rho = h(\mathbf{x}) - \rho$.

Property 10. We next prove h satisfies Property 10. Pick arbitrary q and w such that $q \neq w$.

(1) Suppose that $m \geq q \geq 2$ and $m \geq w \geq 2$. Then, it is easy to verify $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i + 2) - h'(\sum_i x_i + 2) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$.

(2) Suppose that $m \geq q \geq 2$ and $w \in \{1, m+1\}$. If $w = 1$, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} +$

$e_{w-1}) = h'(\sum_i x_i + 1) - h'(\sum_i x_i + 1) = h'(\sum_i x_i + 2) - h'(\sum_i x_i + 2) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$. If $w = m + 1$, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i + 2) - h'(\sum_i x_i + 2) = h'(\sum_i x_i + 1) - h'(\sum_i x_i + 1) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$.

(3) Suppose that $q \in \{1, m + 1\}$ and $m \geq w \geq 2$. If $q = 1$, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i + 2) - h'(\sum_i x_i + 1) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$. If $q = m + 1$, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i + 1) - h'(\sum_i x_i + 2) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$.

(4) Suppose that $q, w \in \{1, m + 1\}$. If $q = 1$ and $w = m + 1$, as h is convex in the number of projects in the system, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i + 2) - h'(\sum_i x_i + 1) \geq h'(\sum_i x_i + 1) - h'(\sum_i x_i) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$. If $q = m + 1$ and $w = 1$, as h is convex in the number of projects in the system, $h(\mathbf{x} + e_q + e_{w-1}) - h(\mathbf{x} + e_{q-1} + e_{w-1}) = h'(\sum_i x_i) - h'(\sum_i x_i + 1) \geq h'(\sum_i x_i + 1) - h'(\sum_i x_i + 2) = h(\mathbf{x} + e_q + e_w) - h(\mathbf{x} + e_{q-1} + e_w)$.

Therefore h satisfies Property 10. By Lemma C.3.1, h also satisfies Properties 11–13. Thus $h \in \widetilde{V}$.

<u>$\boldsymbol{T : \widetilde{V} \to \widetilde{V}}$.</u> Assume v satisfies Properties 8–13 (i.e., $v \in \widetilde{V}$). We proved that $T_A v$, $T_{B,i} v$, $T_C v$, and h satisfy Properties 8–13. It is immediate that Tv satisfies Properties 8, 10, 11, 12, and 13, as these properties are preserved by linear transformations. Next we will prove Tv satisfies Property 9 (i.e., $Tv(\mathbf{x} + e_m) \geq Tv(\mathbf{x}) - \rho$). Since (i) h is increasing in the number of projects in the system, (ii) $T_A v$, $T_{B,i} v$, and $T_C v$ satisfy Property 9, and (iii) $\lambda + \sum_i \mu_i + \mu_{m+1} \leq 1$, Property 9 holds: $Tv(\mathbf{x} + e_m) = h(\mathbf{x} + e_m) + \lambda T_A v(\mathbf{x} + e_m) + \sum_{1 \leq i \leq m} \mu_i T_{B,i} v(\mathbf{x} + e_m) + \mu_{m+1} T_C v(\mathbf{x} + e_m) \geq h(\mathbf{x}) + \lambda\left(T_A v(\mathbf{x}) - \rho\right) + \sum_{1 \leq i \leq m} \mu_i \left(T_{B,i} v(\mathbf{x}) - \rho\right) + \mu_{m+1}\left(T_C v(\mathbf{x}) - \rho\right) \geq Tv(\mathbf{x}) - \rho$. Hence $Tv \in \widetilde{V}$; $T : \widetilde{V} \to \widetilde{V}$. Following Propositions 3.1.5 and 3.1.6 in Bertsekas (2007), we verify that $\lim_{k \to \infty} (T^k v_0)(\mathbf{x}) = v^*(\mathbf{x})$ where v_0 is the zero function, v^* is the optimal cost function, and T^k refers to k compositions of operator T. Since $v_0 \in \widetilde{V}$ and $T : \widetilde{V} \to \widetilde{V}$, we have $T^k v_0 \in \widetilde{V}$, and therefore $v^* \in \widetilde{V}$. □

Theorem 4.5.1 (Restated)**.** *Under Assumption 4.5.1, the optimal portfolio strategy at each stage i is a state-dependent noncongestive-promotion policy with state-dependent promote-up-to levels $S_i^*(\mathbf{x}_{-i})$: It is optimal to promote a project to stage i if and only if $x_i < S_i^*(\mathbf{x}_{-i})$,*

where $\mathbf{x}_{-i} = (x_1, .., x_{i-1}, x_{i+1}, .., x_m)$ *is an* $m-1$ *dimensional vector of the numbers of projects at stages* $k \neq i$. *The optimal policy has the following additional properties:*

i. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ *weakly increases as the number of projects at stage* $j > i$ *decreases.*

ii. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ *weakly increases as the number of projects at stage* $j < i$ *increases.*

iii. The optimal promote-up-to level $S_i^*(\mathbf{x}_{-i})$ *weakly increases as projects at stage* $j \neq i-1$ *move along the process.*

iv. It is always optimal to launch a new product if there are projects available for the product launch stage.

v. It is never optimal to interrupt any experiment.

vi. It is never optimal to terminate any project.

Proof. By Lemma 4.5.1, we know $v^* \in \widetilde{V}$. Define, for $v^* \in \widetilde{V}$,

$$S_1^*(\mathbf{x}_{-1}) = \min \{z_1 : v^*(\mathbf{z}+e_1) - v^*(\mathbf{z}) > -c_1,\ z_k = x_k, \forall k > 1,\ z_1 \in \mathbb{N}_0\},\ \text{and}$$

$$S_i^*(\mathbf{x}_{-i}) = \min\{x_{i-1}, \min \{z_i : v^*(\mathbf{z} - e_{i-1} + e_i) - v^*(\mathbf{z}) > -c_i,\ z_k = x_k, \forall k \neq i,\ z_i \in \mathbb{N}_0\}\},$$
$$\forall i \in \{2, .., m\}.$$

Since v^* satisfies Property 11, $v^*(\mathbf{z} - e_{i-1} + e_i) - v^*(\mathbf{z})$ is increasing in z_i (take $l = q = i$ and $w = m+1$), $\forall i \in \{1, 2, .., m\}$. Consider $S_1^*(\mathbf{x}_{-1})$. As z_1 increases, since the holding cost rate h is strictly increasing in the number of projects in the system, $v^*(\mathbf{z}+e_1) - v^*(\mathbf{z})$ will eventually cross $-c_1$. Thus there exists a finite $S_1^*(\mathbf{x}_{-1})$. Now consider $S_i^*(\mathbf{x}_{-i})$ for $i \in \{2, .., m\}$. Notice that, since the holding cost rate h is strictly increasing in the number of projects in the system, x_{i-1} should be finite. Thus there exists a finite $S_i^*(\mathbf{x}_{-i})$. Hence the optimal control policy is a noncongestive promotion policy with state-dependent promote-up-to levels $S_i^*(\mathbf{x}_{-i})$, $\forall i$. Next we will prove properties (i)-(vi):

i. Pick arbitrary i and j such that $i < j$. We will show that the optimal promote-up-to level at stage i obeys $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$, where $z_{i'} = x_{i'}$, $\forall i' \notin \{i, j\}$, and $z_j + 1 = x_j$.

Suppose that $S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. By definition, it is optimal to promote a project to stage i at $\mathbf{x}$ if $x_i < S_i^*(\mathbf{x}_{-i})$, and it is not optimal to do so at $\mathbf{z}$ if $z_i = S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. But we have a contradiction when $x_i = z_i = S_i^*(\mathbf{z}_{-i})$; if it is optimal to promote a project to stage i at $\mathbf{x}$, it should also be optimal to do so at $\mathbf{z}$ (due to Property 11 when $l = i$, $q = j$, and $w = m+1$). Thus we must have $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$.

ii. Pick arbitrary i and j such that $i > j$. We will show that the optimal promote-up-to level at stage i obeys $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$, where $z_{i'} = x_{i'}$, $\forall i' \notin \{i, j\}$, and $z_j = x_j + 1$. Suppose that $S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. By definition, it is optimal to promote a project to stage i at $\mathbf{x}$ if $x_i < S_i^*(\mathbf{x}_{-i})$, and it is not optimal to do so at $\mathbf{z}$ if $z_i = S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. But we have a contradiction when $x_i = z_i = S_i^*(\mathbf{z}_{-i})$; if it is optimal to promote a project to stage i at $\mathbf{x}$, it should also be optimal to do so at $\mathbf{z}$ (due to Property 12 when $l = 0$, $q = j$, and $w = i$). Thus we must have $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$.

iii. Pick arbitrary i and j such that $i \neq j+1$. We will show that the optimal promote-up-to level at stage i obeys $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$, where $z_{i'} = x_{i'}$, $\forall i' \notin \{i, j, j+1\}$, $z_j + 1 = x_j$ and $z_{j+1} = x_{j+1} + 1$. Suppose that $S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. By definition, it is optimal to promote a project to stage i at $\mathbf{x}$ if $x_i < S_i^*(\mathbf{x}_{-i})$, and it is not optimal to do so at $\mathbf{z}$ if $z_i = S_i^*(\mathbf{z}_{-i}) < S_i^*(\mathbf{x}_{-i})$. But we have a contradiction when $x_i = z_i = S_i^*(\mathbf{z}_{-i})$; if it is optimal to promote a project to stage i at $\mathbf{x}$, it should also be optimal to do so at $\mathbf{z}$ (due to Property 10 when $q = i$ and $w = j+1$). Thus we must have $S_i^*(\mathbf{z}_{-i}) \geq S_i^*(\mathbf{x}_{-i})$.

iv. Suppose that $x_m > 0$. Since v^* satisfies Property 9, it is always optimal to launch a new product: $T_C v^*(\mathbf{x}) = \min\{v^*(\mathbf{x}), v^*(\mathbf{x} - e_m) - \rho\} = v^*(\mathbf{x} - e_m) - \rho$.

v. We will prove it is never optimal to interrupt any experiment. Assume that a project is optimally promoted to stage i at a given state $\mathbf{x}$, or equivalently, an experiment at stage i is optimally initiated at $\mathbf{x}$ (i.e., $x_i < S_i^*(\mathbf{x}_{-i})$). We will then consider the following cases:

- Suppose that $i \geq 2$. Also, suppose that the system moves to a state $\mathbf{z}$ such that $z_1 = x_1 + 1$ and $z_j = x_j$, $\forall j > 1$. Point (ii) implies that it is optimal to promote a project to stage i at $\mathbf{z}$: The experiment at stage i, which has been initiated at $\mathbf{x}$, can be resumed at $\mathbf{z}$.

- Pick arbitrary $j \geq 1$ such that $j+1 \neq i$ and $j+1 \leq m$. Suppose that the system moves to a state $\mathbf{z}$ such that $z_j + 1 = x_j$, $z_{j+1} = x_{j+1} + 1$, and $z_{j'} = x_{j'}$, $\forall j' \notin \{j, j+1\}$. Point (iii) implies that it is optimal to promote a project to stage i at $\mathbf{z}$: Again, the experiment at stage i, which has been initiated at $\mathbf{x}$, can be resumed at $\mathbf{z}$.

- Suppose that the system moves to a state $\mathbf{z}$ such that $z_m + 1 = x_m$ and $z_j = x_j$, $\forall j < m$. Also, suppose that $i < m$. Point (i) implies that it is optimal to promote a project to stage i at state $\mathbf{z}$: Once again, the experiment at stage i, which has been initiated at $\mathbf{x}$, can be resumed at $\mathbf{z}$. Next suppose that $i = m$: It is optimal to promote a project to stage m at state $\mathbf{z}$, since $z_m < x_m < S^*_m(\mathbf{x}_{-m}) = S^*_m(\mathbf{z}_{-m})$. Once again, the experiment at stage i, which has been initiated at $\mathbf{x}$, can be resumed at $\mathbf{z}$.

Therefore the experiment at stage i is never interrupted.

vi. As v^* satisfies Property 8, it is easy to verify that $v^*(\mathbf{x} + e_q + e_w) - v^*(\mathbf{x} + e_w) \geq v^*(\mathbf{x} + e_q + e_w) - v^*(\mathbf{x} + e_q)$ for $q < w$. Therefore, if a project is to be terminated, it is optimal to select this from the earliest possible stage. Pick arbitrary state $\mathbf{x}$. Let i^* denote the earliest stage with at least one available project. Thus $x_i = 0$, $\forall i < i^*$. Suppose that it is not optimal to terminate a project from category i^* at state $\mathbf{x}$ (i.e., $v^*(\mathbf{x}) \leq v^*(\mathbf{x} - e_{i^*})$). Then we consider the following scenarios:

- Suppose that the system moves to a state $\mathbf{z}$ such that $z_1 = x_1 + 1$ and $z_j = x_j$, $\forall j > 1$. This implies that $v^*(\mathbf{x} + e_1) + c_1 \leq v^*(\mathbf{x})$. Since $c_1 \geq 0$, we should have $v^*(\mathbf{x} + e_1) \leq v^*(\mathbf{x})$. Thus it is not optimal to terminate a project at stage 1. As v^* satisfies Property 8, we should also have $0 \geq v^*(\mathbf{x} + e_1) - v^*(\mathbf{x}) \geq v^*(\mathbf{x} + e_1) - v^*(\mathbf{x} + e_1 - e_{i^*})$. Thus, it is not optimal to terminate any project at stage i^*.

- Suppose that the system moves to a state $\mathbf{z}$ such that $z_{i^*} + 1 = x_{i^*}$, $z_{i^*+1} = x_{i^*+1} + 1$, and $z_j = x_j$, $\forall j \notin \{i^*, i^*+1\}$. This implies that $v^*(\mathbf{x}) \geq v^*(\mathbf{z}) + c_{i^*+1}$. Also, suppose that $x_{i^*} \geq 2$. Since $v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{x})$ and v^* satisfies Property 12, we should have $0 \geq v^*(\mathbf{x}) - v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{z}) - v^*(\mathbf{z} - e_{i^*})$. Thus it is not optimal to terminate any project at stage i^*. Now suppose that $x_{i^*} = 1$. Notice that $z_i = 0$, $\forall i \leq i^*$. If a project is to be terminated at $\mathbf{z}$, it is optimal to select this from stage $i^* + 1$. But it is not optimal to terminate such a project: Since $v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{x})$ and $v^*(\mathbf{x}) \geq v^*(\mathbf{z}) + c_{i^*+1}$, we should have $v^*(\mathbf{z} - e_{i^*+1}) = v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{z})$.

- Pick arbitrary j such that $j > i^*$ and $j < m$. Suppose that the system moves to a state $\mathbf{z}$ such that $z_j + 1 = x_j$, $z_{j+1} = x_{j+1} + 1$, and $z_{j'} = x_{j'}$, $\forall j' \notin \{j, j+1\}$. Since $v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{x})$ and v^* satisfies Property 12, we should have $0 \geq v^*(\mathbf{x}) - v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{z}) - v^*(\mathbf{z} - e_{i^*})$. Thus it is not optimal to terminate any project at stage i^*.

- Suppose that the system moves to a state $\mathbf{z}$ such that $z_m + 1 = x_m$ and $z_j = x_j$, $\forall j < m$. Also, suppose that either $i^* < m$ or $i^* = m$ and $x_m \geq 2$. Since $v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{x})$ and v^* satisfies Property 12, we should have $0 \geq v^*(\mathbf{x}) - v^*(\mathbf{x} - e_{i^*}) \geq v^*(\mathbf{z}) - v^*(\mathbf{z} - e_{i^*})$. Thus it is not optimal to terminate any project at stage i^*. Now, suppose that $i^* = m$ and $x_m = 1$. But then $z_i = 0$, $\forall i$; there is no project in the system.

Therefore it is never optimal to terminate any project during the NPD process. □

C.4 Additional Numerical Results for FNCP-C, FNCP-S, and NP

Tables C.1 and C.2 exhibit additional numerical results for FNCP-C, FNCP-S, and NP.

Table C.1 Numerical results for various holding and experimentation costs.

			% difference from optimal cost			Computation times (in seconds)		
h	c_1	c_2	FNCP-C	FNCP-S	NP	FNCP-C	FNCP-S	NP
1	2	2	0.876	0.877	1.041	467.01	3.49	0.23
1	2	4	1.610	1.637	2.316	572.58	3.96	0.19
1	2	6	4.165	4.202	5.801	1754.26	2.92	0.17
1	4	2	3.600	4.143	5.866	446.86	2.34	0.28
1	4	4	5.134	5.202	12.701	608.42	2.98	0.21
1	4	6	5.333	5.344	17.141	743.41	2.88	0.18
1	6	2	4.888	7.017	17.960	820.93	3.43	0.27
1	6	4	5.169	5.161	23.841	855.63	2.49	0.29
1	6	6	5.846	4.016	34.411	697.96	2.02	0.30
2	2	2	4.174	4.633	13.184	500.77	3.59	0.21
2	2	4	3.429	4.284	18.775	686.85	2.91	0.15
2	2	6	5.328	6.207	27.641	675.33	2.57	0.13
2	4	2	4.958	6.345	32.294	460.01	2.68	0.22
2	4	4	9.721	8.698	51.362	749.09	2.52	0.21
2	4	6	13.965	12.486	79.718	817.25	2.10	0.15
2	6	2	14.397	13.709	78.627	672.89	3.21	0.22
2	6	4	17.338	17.405	123.464	198.22	2.99	0.22
2	6	6	20.758	20.781	201.705	383.51	1.62	0.22
3	2	2	5.767	5.874	25.632	541.60	1.76	0.13
3	2	4	8.254	8.252	33.446	406.77	1.76	0.13
3	2	6	14.371	14.208	46.302	645.48	1.53	0.13
3	4	2	13.796	13.746	82.622	466.25	2.88	0.15
3	4	4	20.520	20.688	114.935	504.34	1.59	0.17
3	4	6	31.680	31.680	164.425	402.99	2.15	0.12
3	6	2	31.532	31.532	201.586	65.11	1.52	0.24
3	6	4	50.322	50.322	317.756	212.77	2.35	0.16
3	6	6	88.438	88.438	531.542	410.44	1.60	0.15
Average			14.643	14.700	83.929	583.95	2.51	0.19

Notes. $r_1 = 40$, $r_2 = 0$, $\mu_1 = \mu_2 = 1.5$, $\mu_3 = 0.5$, $\lambda = 100$, $\phi^{(1)} = \phi^{(2)} = 0.75$, $p_{0,1} = p_{0,2} = 0.5$.

Table C.2 Numerical results for various values of informativeness probabilities.

		% difference from optimal cost			Computation times (in seconds)		
$\phi^{(1)}$	$\phi^{(2)}$	FNCP-C	FNCP-S	NP	FNCP-C	FNCP-S	NP
0.95	0.95	6.233	5.876	10.490	511.84	2.85	0.14
-	0.85	6.945	6.203	13.586	751.28	2.12	0.17
-	0.75	7.264	6.214	13.595	746.72	2.35	0.14
-	0.65	7.233	6.219	13.599	783.31	2.40	0.21
-	0.55	7.146	6.226	13.607	920.46	2.73	0.13
0.85	0.95	5.393	5.122	10.593	642.52	1.77	0.21
-	0.85	4.447	4.425	14.772	552.05	2.44	0.15
-	0.75	5.660	5.504	22.397	490.95	2.69	0.16
-	0.65	5.654	5.518	23.604	535.78	2.92	0.16
-	0.55	5.685	5.529	23.612	510.77	2.88	0.14
0.75	0.95	5.815	5.512	13.176	570.38	2.61	0.19
-	0.85	3.312	3.477	20.353	385.34	2.31	0.14
-	0.75	4.633	4.233	33.686	709.66	2.13	0.14
-	0.65	5.472	4.272	43.773	900.67	2.14	0.13
-	0.55	5.458	4.286	43.799	662.20	1.75	0.14
0.65	0.95	4.031	5.594	22.755	433.17	3.89	0.17
-	0.85	5.548	8.865	39.539	400.55	2.91	0.20
-	0.75	10.602	13.619	63.344	595.26	2.92	0.18
-	0.65	15.972	14.190	84.591	831.34	3.64	0.21
-	0.55	15.480	13.331	87.340	612.26	2.99	0.15
0.55	0.95	4.076	3.809	44.499	257.75	3.32	0.25
-	0.85	7.075	5.830	75.781	264.97	1.89	0.29
-	0.75	15.745	12.454	118.978	278.54	2.04	0.24
-	0.65	44.550	31.473	152.407	263.22	2.37	0.21
-	0.55	50.000	33.333	156.773	49.10	1.84	0.14
Average		10.377	8.845	46.426	546.40	2.56	0.18

Notes. $h = 2$, $c_1 = c_2 = 4$, $r_1 = 40$, $r_2 = 0$, $\mu_1 = \mu_2 = 1$, $\mu_3 = 0.5$, $\lambda = 100$, $p_{0,1} = p_{0,2} = 0.5$.

Bibliography

Adler, P. S., A. Mandelbaum, V. Nguyen, E. Schwerer. 1995. From project to process management: An empirically-based framework for analyzing product development time. *Management Sci.* **41** 458–484.

Agrawal, N., M. A. Cohen. 2001. Optimal material control in an assembly system with component commonality. *Naval Res. Logist.* **48** 409–429.

Artmann, C. 2009. The value of information updating in new product development. *Lecture Notes in Economics and Mathematical Systems, Vol: 620*. Springer Berlin Heidelberg.

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

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

Bertsekas, D. P. 2007. *Dynamic Programming and Optimal Control, Vol. 2*. Athena Scientific, Nashua, NH.

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

Brucker, P., A. Drexl, R. Möhring, K. Neumann, E. Pesch. 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur. J. Oper. Res.* **112** 3–41.

Chen, F., Y. S. Zheng. 1994. Lower bounds for multi-echelon stochastic inventory systems. *Management Sci.* **40** 1426–1443.

Clark, A. J., H. Scarf. 1960. Optimal policies for a multiechelon inventory problem. *Management Sci.* **6** 475–490.

Cooper, R. G. 2008. Perspective: The stage-gate idea-to-launch process – update, what's new, and NexGen systems. *J. Prod. Innov. Management.* **25** 213–232.

DiMasi, J. A., R. W. Hansen, H. G. Grabowski. 2003. The price of innovation: New estimates of drug development costs. *Journal of Health Economics.* **22** 151–185.

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

Ehrhardt, R. 1984. (s, S) policies for a dynamic inventory model with stochastic lead times. *Oper. Res.* **32** 121–132.

ElHafsi, M., H. Camus, E. Craye. 2008. Optimal control of a nested-multiple-product assemble-to-order system. *Internat. J. Production Res.* **46** 5367–5392.

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

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

Gayon, J. P., S. Benjaafar, F. de Vericourt. 2009. Using imperfect advance demand information in production-inventory systems with multiple customer classes. *Manufacturing Service Oper. Management* **11** 128–143.

Gerchak, Y., M. Henig. 1989. Component commonality in assemble-to-order systems: Models and properties. *Naval Res. Logist.* **36** 61–68.

Gittins, J. C., D. M. Jones. 1972. A dynamic allocation index for the sequential design of experiments. *Progress in Statistics: European Meeting of Statisticians.*

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

Ha, A. 1997. Optimal dynamic scheduling policy for a make-to-stock production system. *Oper. Res.* **45** 42-53.

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

Hariharan, R., P. Zipkin. 1995. Customer-order information, leadtimes, and inventories. *Management Sci.* **41** 1599–1607.

Hillier, M. S. 2000. Component commonality in multiple-period, assemble-to-order systems. *IIE Trans.* **32** 755–766.

Janakiraman, G., J. A. Muckstadt. 2004. Inventory control in directed networks: A note on linear costs. *Oper. Res.* **52** 491–495.

Kaplan, R. S. 1970. A dynamic inventory model with stochastic lead times. *Management Sci.* **16** 491–507.

Kavadias, S., C. H. Loch. 2004. *Project Selection Under Uncertainty: Dynamically Allocating Resources to Maximize Value.* Kluwer Academic Publishers, Boston, MA.

Kavadias, S., R. Chao. 2008. Resource allocation and new product development portfolio management. C. H. Loch, S. Kavadias, eds. *Handbook of New Product Development Management.* Elsevier, Oxford, UK.

Kleywegt, A. J., J. D. Papastavrou. 2001. The dynamic and stochastic knapsack problem with random sized items. *Oper. Res.* **49** 26-41.

Koole, G. 2006. Monotonicity in Markov reward and decision chains: Theory and applications. *Foundations and Trends in Stochastic Systems* **1** 1–76.

Lippman, S. 1975. Applying a new device in the optimization of exponential queueing systems. *Oper. Res.* **23** 687–710.

Lorie, J. H., L. J. Savage. 1955. Three problems in rationing capital. *The Journal of Business* **28** 229–239.

Lu, L., J. S. Song, H. Zhang. 2010. Optimal policies for an assemble-to-order N-system. *Working paper.*

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

Neumann, K., C. Schwindt, J. Zimmermann. 2002. *Project Scheduling with Time Windows and Scarce Resources*. Springer-Verlag, Berlin.

Papastavrou, J. D., S. Rajagopalan, A. J. Kleywegt. 1996. The dynamic and stochastic knapsack problem with deadlines. *Management Sci.* **42** 1706–1718.

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

Roberts, K., M. L. Weitzman. 1981. Funding criteria for research, development, and exploration projects. *Econometrica* **49** 1261–1288.

Rosling, K. 1989. Optimal inventory policies for assembly systems under random demands. *Oper. Res.* **37** 565–579.

Schmidt, C. P., S. Nahmias. 1985. Optimal policy for a two-stage assembly system under random demand. *Oper. Res.* **33** 1130–1145.

Sobel, M. J., R. Q. Zhang. 2001. Inventory policies for systems with stochastic and deterministic demand. *Oper. Res.* **49** 157–162.

Sommer, S. C., C. H. Loch, M. T. Pich. 2007. Project risk management in new product development. C. H. Loch, S. Kavadias, eds. *Handbook of New Product Development Management*. Elsevier, Oxford, UK.

Song, J. S. 2000. A note on assemble-to-order systems with batch ordering. *Management Sci.* **46** 739–743.

Song, J. S., C. A. Yano, P. Lerssrisuriya. 2000. Contract assembly: Dealing with combined supply lead time and demand quantity uncertainty. *Manufacturing Service Oper. Management* **2** 287–296.

Song, J. S., P. Zipkin. 1993. Inventory control in a fluctuating demand environment. *Oper. Res.* **41** 351–370.

Song, J. S., P. Zipkin. 1996. Inventory control with information about supply conditions. *Management Sci.* **42** 1409–1419.

Song, J. S., P. Zipkin. 2003. Supply chain operations: Assemble-to-order systems. T. De Kok, S. Graves, eds. *Handbooks in Operations Research and Management Science, Vol. 30: Supply Chain Management.* North-Holland.

Thomke, S. 2008. Learning by experimentation: Prototyping and testing. C. H. Loch, S. Kavadias, eds. *Handbook of New Product Development Management.* Elsevier, Oxford, UK.

Topkis, D. M. 1998. *Supermodularity and Complementarity.* Princeton University Press, Princeton, NJ.

Topkis, D. M. 1968. Optimal ordering and rationing policies in a nonstationary dynamic inventory model with n demand classes. *Management Sci.* **15** 160–176.

Ulrich, K., S. D. Eppinger. 2004. *Product Design and Development.* 3rd ed., McGraw-Hill/Irwin, New York, NY.

Van Mieghem, J. A., N. Rudi. 2001. Newsvendor networks: Inventory management and capacity investment with discretionary activities. *Manufacturing Service Oper. Management* **4** 313–335.

Veinott Jr, A. F. 1965. Optimal policy for a multi-product, dynamic, nonstationary inventory problem. *Management Sci.* **12** 206–222.

de Véricourt, F., F. Karaesmen, Y. Dallery. 2002. Optimal stock allocation for a capacitated supply system. *Management Sci.* **48** 1486–1501.

Weber, R. R., S. Stidham. 1987. Optimal control of service rates in networks of queues. *Adv. Appl. Probab.* **19** 202–218.

Wein, L. 1992. Dynamic scheduling of a multiclass make-to-stock queue. *Oper. Res.* **40** 724-735.

Zipkin, P. 1986. Stochastic leadtimes in continuous-time inventory models. *Naval Res. Logist. Quart.* **33** 763–774.