Dynamic Scheduling Policies in Production-Inventory Systems with Returns or Two Classes of Demands

Jieling Han

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Apurva Jain, Chair

Hamed Mamani

Ming Fan

Program Authorized to Offer Degree:
Michael G. Foster School of Business

University of Washington

Abstract

Dynamic Scheduling Policies in Production-Inventory Systems with Returns or Two Classes of Demands

Jieling Han

Chair of the Supervisory Committee:
Associate Professor Apurva Jain
Information Systems & Operations Management

A major concern in operations and supply chain management research has been to determine optimal policies to minimize costs in production and inventory systems. This dissertation consists of two applications of production-inventory systems motivated by the real world observations.

The first model is motivated by a high-tech company data center server inventory control process. The firm has internalized the production process and faces server returns at the same time. There are demands for both new and used products. Returned servers can be used to serve the demand for use products directly, or be remanufactured into new condition by the same production capacity. Unmet demands are lost. The objective is to determine the optimal scheduling policy that minimizes the discounted total cost. We model the system with a production-inventory model with returns and show the optimal production inventory policy is of base stock type. Efficient heuristics are proposed and the base model is extended to scenarios that better described the real world situation. These scenarios include unmet demand are backordered, bulk demands and returns depending on the demands.

The second model analyzes a model of a continuous-time production-inventory system with a shared server serving two demands. Unsatisfied demands are backo-

rdered and standard holding and penalty costs apply at each inventory location. The objective is to determine the optimal scheduling policy that minimizes the discounted total cost. The distinguishing feature of our model is that the two demand classes differ in their variability characteristics. The motivation for analyzing such a system is driven by our observations in many real-life contexts where such a difference is reported to exist. Very briefly, practical examples of why the variability of two demand processes, served by the same capacity, may differ include (1) inherent differences in the predictability of the demand, (2) some customers' spiky purchasing pattern influenced by low prices during promotions, (3) mixing emergency and scheduled demands, and (4) the nature of two different markets. Our results allow us to shed some light on implications for managers in such circumstances.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

This dissertation is the summary of my PhD work in the area of Operations Management and also the result of help from many people.

My deepest thanks go to my advisor Dr. Apurva Jain who provided me with continuous support and guidance. He introduced me to the area of optimal control, and provided invalueable suggestions on conducting research. He is very approachable and consistently ready to help. Without him, this dissertation would not have been possible.

I would like to thank Dr. Yong-Pin Zhou for his vision and guidance. He is keen on every aspects of operations management and enthusiastic in both research to teaching. Talking with Yong-Pin always enlightens me with new ideas and understandings. My thanks also go to Dr. Yong Tan, Dr. Ted Klastorin, Dr. Hamed Mamani and Dr. Kamran Moinzadeh, who inspire me with new ideas in classes and discussions.

I am fortunate to have the opportunities to TA for Dr. Erich Studer-Ellis, Dr. Bruce Faaland, Dr. Shaosong Ou, Dr. Ming Fan, and Mr. Izic Choi. Teaching in business school is demanding and challenging, and I learnt a lot from these role models.

I am very grateful to my friends and family. My sincere thanks go to my husband Peng Dai, who provided valuable insights in my research and support to help me sustained. I would also like to thank my friend Wei-Ming Lee, who is always available to help, with patience. His persistence in doing research keeps me encouraged. Many thanks to my friends in the department of Information Systems and Operations Management. My life in University of Washington won't be colorful without you.

DEDICATION

To Hansee, for a strong spirit

Chapter 1

INTRODUCTION

1.1 Background

A major concern in operations and supply chain management research has been to determine optimal policies to minimize costs in production and inventory systems. Beginning with early single-location inventory models, research on centralized supply chains introduced models that integrate the flow of goods between multiple inventory locations (or echelons), such as a supplier's warehouse and retail stores. Relative to the research on multi-echelon inventory models, there has been much less work on models that integrate manufacturing or production systems with inventory models. Recent years, however, have seen a growing interest in determining optimal policies in production-inventory models. This dissertation formulates and analyzes two production-inventory models that are motivated by the observations in retail supply chains and data center operations.

Current theoretical interest in production-inventory models reflects the fact that such an approach allows for combining different parts of supply chain functions in the same framework and thus opens the possibility of looking for the best operating policy for the whole supply chain. While practical implementation requires researchers to look for decentralized solutions, the effectiveness of these solutions must be measured against the centralized solutions that, for example, can analyze the effect of production policy at one end of the supply chain on the inventory costs at the other end of supply chains. Production-inventory models are well-suited to address that challenge. These models are also well-equipped to consider another current supply chain issue, that of product differentiation. Supply chains are increasingly facing customers that demand individualized products and services. Limited capacity and inventory must be shared between different types of products and customers. Multiclass production-inventory models can analyze such sharing through determination of optimal scheduling policies to manage the capacity.

The two production-inventory models presented in this dissertation are motivated by business issues that exhibit both characteristics above, the need for an integrated analysis and the existence of multiple classes of demands.

1.1.1 Production-Inventory System with Returns

Value added recovery from used products or materials is gaining increasing importance for the firms as a profitable and sustainable strategy ([18, 70]). Material reuse and remanufacturing not only help firms reduce cost and protect environment, they also help firms build brand image for social responsibilities towards environment. The three simple reasons for why companies reuse products are summarized by [27] as profit, people and planet.

Products are returned for many reasons. Customers may return new/used products under generous service satisfaction warranty. Products (such as electronics) with value parts may be collected and returned for profit. In some regions, government enforce companies to recollect products with hazard material. Regardless of the reasons, the return flows complicates the inventory control [34, 17]. How to integrate the returned used products into inventory and production planning has been the concentration for recent research.

This model is motivated by the observations of a production inventory planning manager at a California based data center. There are demands for new servers and used servers which must both be satisfied by a shared capacity that either customizes new servers or upgrades used servers. Used servers may be returned due to variety of reasons such as underutilization or out-of-date. The focus is on optimizing production / inventory control operations in order to reduce total cost.

1.1.2 Production-Inventory System with Different Variability

Variability in supply chains is studied extensively in literature. The classic paper from [52] summarize four reasons for the Bullwhip effect: demand signal process, rationing game, order batching and price variation.

This model is related to the price variation. It is motivated by the observations in retail supply chains where a manufacturer uses shared production capacity to satisfy demands from multiple retailed. The retailers have different price promotion policies which bring variability in the demand process. Our research is in line with [41]. They collect a sample of data from the manufacture of private-label cookies who satisfy demands from several retailers. One retailer practices a Hi-Lo price promotion strategy which leads to variability in his ordering process. The process can be approximated with hyperexponential renewal process, and is supported by evidence from [41]. From the modeling perspective, this model falls within the queuing literature where variabilities come from the demand process, see [43].

In both these settings, managers would like to gain insights into the scheduling policy for the shared capacity that will allow them to minimize the total cost. Our models contribute to the literature by directly addressing these specific business settings and developing insights for operating managers.

1.2 Continuous-Time Production-Inventory Systems

The models in this dissertation fall in the general class of continuous-time models that integrate production capacity with inventory locations, usually in the context of multiple classes of demands. Production or manufacturing models in literature usually feature a queue-like behavior and inventory models usually feature a sales-replenishment cycle. To combine the two, addition of production models to inventory models recognizes that inventory replenishment lead time is a function of congestion delay and processing time in the production model. In early literature, this lead time was often considered as fixed and exogenous to the model. Detailed modeling of the production system allows for the lead time to be considered endogenous to the model and makes it possible to analyze the effect of shared capacity between different demand classes. Following the common practice in queuing and inventory literature, these models usually have continuous-time and discrete demands.

As the literature reviews presented in chapter 2 and chapter 4 describe, earlier models proposed static policies that made scheduling decisions without considering the current state of the system. Given the complexity of the problem, much of the work aimed at developing heuristics. In continuous-time models, under certain assumptions, it was shown that the form of the optimal policy can be determined. This dissertation advances this line of research by showing how to determine optimal policy in models that go beyond the standard modeling assumptions of this literature, Poisson demands and no returns.

1.2.1 Features of Models related to Optimal Control of Production-Inventory Systems with Shared Resources

To find the optimal policy in a production-inventory system with a shared resource is a complex challenge. As [75] stated, even with Poisson demand process and exponential service time, the derivation of the optimal policy requires the solution to a multidimensional Markov decision process which is extremely difficult to compute. Research emphasizing on describing the optimal policy usually falls within a restricted frame work:

- All the classes of demands occur according to memoryless processes, such as Poisson processes, Erlang process and etc.

- Servers are shared by performing multiclass of tasks.

- The server can only perform one task at a time.

- Service time follows memoryless distribution, such as exponential distribution.
- Holding and backorder (or lost sales) costs are convex with the number of net inventory.
- The goal is to find the optimal control policy to minimize the discounted total cost, or average cost.

Pioneering work by Ha [36] was the first to show the structure of the optimal policy in the context of two Poisson demand classes with equal production times. He transformed the continuous-time system into discrete time system, and modeled the system with a Markov decision process. Then, he characterized the structure of the optimal policy and used dynamic programming to determine optimal policy parameters.

The difficulty of the analysis lies within the combinatorial nature of choosing an optimal operation among a set of available operations at each decision epoch, anytime an arrival or service occurs. As is well-known, dynamic programming suffers from the curse of dimensionality which makes the system impractical to solve when the state space is large. To address this difficulty, heuristics are often proposed. Heuristics often use static rules that ignore the current state and are based only on fixed parameters like arrival rates and cost.

Our models follow the general assumptions as presented above. But we add features that are new to the literature. Specifically, one of our models uses a non-Poisson process to model arrivals and in the other model, we model a return process. While both of these additions are motivated by real business problems, the modeling complications they present allow us to advance the literature on continuous time production-inventory models.

1.3 Summary of Models and Contributions

In this section, we provide an overview of models and contributions.

1.3.1 Production-Inventory System with Returns

We develop and analyze a production-inventory model where the manufacturer faces demand for both new and used products. Demand for the new products occur according to a Poisson process and is satisfied out of the inventory of newly produced finished new products. Demand for used products follows a separate Poisson process and can be satisfied from the used product inventory. Customer may return products after some use; these are stocked in a return product inventory. Returned products can be used to serve the demand for use products directly, or be remanufactured into new products. The production process, modeled as a single server queue, can produce new finished products as well as refurbish return/used products into new products. Standard holding costs and penalty costs apply. We focus on developing the optimal production inventory policy for the firm in this setting.

While our model follows in the steps of existing production-inventory literature and extends it, our interest in considering this question is not purely theoretical. We are motivated by our observations in a California based data center of a large web-services provider. This data center has a very large number of servers to handle traffic for the firm's services. A traffic manager is responsible for maintaining a certain number of these servers in order to provide a high level of response to customers. Given the fast pace of technological change, these traffic managers often want to upgrade their servers by returning them and replacing them with either new or refurbished servers. Given the unpredictability of traffic volumes, these traffic managers often either return their existing servers or demand new/refurbished servers. These returns and demands are sent to the server inventory manager who has limited production capacity that can be either used to customize off-the-shelf servers received from outside supplier or to refurbish servers returned by the traffic managers. The server inventory manager is continuously making decisions about how to use the production capacity and about the inventory levels of new and refurbished servers.

Our model possesses the set of assumptions that describe the real world problem properly, and are technically challenging. These assumptions include: 1) Demands for both new and used products; 2) Unmet products are lost; 3) Product are returned as used ones, which can satisfy used demands directly or can be remanufactured into new products; and 4) Shared server can perform both producing and remanufacturing, which leads to the number of potential decisions being 3. It is also the largest number of decisions by a shared server that has been successfully analyzed. We show the structure of the optimal policy is of state dependent base stock type. Priority based heuristics are proposed to decide the production resource: 1) priority on using raw material; 2) priority on using used product; 3) two-level threshold for choosing between raw material and used product. In chapter 3, several related systems are studied.

1.3.2 Effects of Variability in a Production-Inventory System

We develop and analyze a model of a continuous-time production-inventory system with a shared server serving two demands. Unsatisfied demands are backordered and standard holding and penalty costs apply at each inventory location. The objective is to determine the optimal scheduling policy that minimizes the long term total cost. The distinguishing feature of our model is that the two demand classes differ in their variability characteristics. Because Poisson distributions are completely characterized by a single parameter, modeling both demand classes with Poisson processes does not allow us to have arrivals with, for example, equal means but different variances. To capture the differences in demand variability, we model the inter-arrival times for one demand class H as following hyperexponential distribution, and the other demand class M following exponential distribution. This allows us to keep the model tractable and understand the impact of variability differences on the structure of the optimal policy. We show the form of the optimal policy and prove properties of optimal policy parameters. Then we consider the cases where partial or full information about the demand arrival process is known.

The motivation for analyzing such a system is driven by our observations in many real-life contexts where such a difference is reported to exist. Very briefly, practical examples of why the variability of two demand processes, served by the same capacity, may differ include (1) inherent differences in the predictability of the demand as in [20], (2) some customers' spiky purchasing pattern influenced by low prices during promotions as in [41], (3) mixing emergency and scheduled demands as in [59], and (4) the nature of two different markets as in [25]. Our results allow us to shed some light on implications for managers in such circumstances.

The existing literature does not offer any insights about the structure of the optimal policy in case information of arrival process is known in the form of phase information. We develop observations and results about the properties of switching and stopping curves under different phases of class H arrival process. We start with the observation that, given other state variables remain the same, if it is optimal to produce class M when the class H arrival process is in on phase, it is also optimal to produce class M when the class H arrival process is in off phase. Building on this observation we are able to show, analytically and numerically, that the switching curves under *on* and *off* phases may overlap but will not cross. These results allow us to develop an understanding of the impact information about demand arrival process can have on the structure of the optimal policy. Numerical results allow us to show the impact on shapes of switching and stopping curves as the differences in the coefficient of variations of two demand arrival process changes.

Our next step is to consider the case where the phase information is only partially known. We model the system as a partially-observable Markov decision process. Given the prior belief, about the probability of being in phase *on* and the transition between the two phases, the belief at next decision epoch is updated according Bayes rule. With numerical analysis, we numerically compute the optimal policies and their corresponding costs of three models: (1) with full phase information, (2) with partial information of phase being in on, and (3) without phase information, but assume Poisson arrival process for class H. Among other observations, we notice that the discounted total costs of model (1) is no greater than for model (2), which is no greater than the cost for model (3). This result shows the value of knowing phase information.

The remainder of the dissertation is organized as follows. Chapter 2 studies a production-inventory model with returns and lost sales. Chapter 3 relax the assumptions in chapter 2 and analyzes several related models. Chapter 4 studies the optimal control in a production-inventory system with two classes of demands, and analyzes the effects of variability.

Chapter 2

DYNAMIC CONTROL FOR A PRODUCTION-INVENTORY SYSTEM WITH SHARED CAPACITY AND RETURNS

2.1 Introduction

We develop and analyze a production-inventory system where the manufacturer faces demand for both new and used products. Demand for the new products occur according to a Poisson process and is satisfied out of the inventory of newly produced finished new products. Demand for used products follows a separate Poisson process and can be satisfied from the used product inventory. Customer may return products after some use; these are stocked in a return product inventory. Returned products can be used to serve the demand for use products directly, or be remanufactured into new products. The production process, modeled as a single server queue, can produce new finished products as well as refurbish return/used products into new products. Standard holding costs and penalty costs apply. We focus on developing the optimal production inventory policy for the firm in this setting.

In presence of two different products that can be manufactured using the same limited capacity but that differ in their cost implications, production and inventory managers must determine a policy to decide which product should be produced next and what are the right levels of inventory for each product. The optimal decisions depend on the levels of inventory of each product at any given time. The production-inventory literature (reviewed below) in recent years has focused on developing optimal operating policy in these settings. Given the complexity of analyzing these models with stochastic demand and production times, well-performing heuristic policies are also attractive. In our model, we have an additional complication. Product return process that models return of units from consumers is also stochastic. Production and inventory decisions must also take into account the inventory level of returned products and the cost of carrying the return product. This not only increases the size of required state space in the model, we show that this also considerably increases the complexity in making optimal decision. We derive the form of the optimal policy, show how to compute optimal parameters and develop effective heuristics.

While our model follows in the steps of existing production-inventory literature and extends it, our interest in considering this question is not purely theoretical. We are motivated by our observations in a California-based data center of a large Web-services provider. This data center has a very large number of servers to handle traffic for the firm's services. A traffic manager is responsible for maintaining a certain number of these servers in order to provide a high level of response to customers. Given the fast pace of technological change, these traffic managers often want to upgrade their servers by returning them and replacing them with either new or refurbished servers. Given the unpredictability of traffic volumes, these traffic managers often either return their existing servers or demand new/refurbished servers. These returns and demands are sent to the server inventory manager who has limited production capacity that can be either used to customize off-the-shelf servers received from outside supplier or to refurbish servers returned by the traffic managers. The server inventory manager is continuously making decisions about how to use the production capacity and about the inventory levels of new and refurbished servers.

Data center operations are a fast growing area in high-tech businesses serving large numbers of end-users. Facebook spent $606 million on servers, storage, network gear and data center in 2011 and another $500 million in 2012 [57]. Google spent $951 million on its data center operations in just the fourth quarter of 2011, with infrastructure capital expenses of $3.4 billion for all of 2011 [57]. Some of the important aspects for a successful IT business include: maintain high site availability, site reliability, fast query responses and so on. Amazon cloud service EC2 guarantees 99.95% monthly availability, or partial credit back to the end users [1]. According to [86], 40% of users abandon a website that takes more than 3 seconds to load. Slow response to an e-commerce site may lead to huge loss of sales.

To meet and maintain these high standards of service, data centers use large numbers of servers to handle the traffic and to store user information. Sever capacity utilization rates are usually kept low. According to *Best Practices Guide for Energy-Efficient Data Center Design* [50], most servers run below 20% utilization most of the time while draw full power during the process. This leads to high costs and large carbon footprints. In this setting, server inventory management is rapidly emerging as an important area. The *2011 Data Center market Insight Report* concludes data center capacity planning and inventory control is one of the top concerns for data center operations [47].

Our observations of this industry suggest two current strategies data centers are using to better manage server inventory. The first strategy is that data centers are internalizing the process of customizing the servers in place of relying on using standardized purchases from vendors. Google [32] customizes the power path in the servers and eliminates the unnecessary hardware components that are default options in the vendor gears. In 2011, Facebook is reported to shift from buying vendor gear and lease data centers to build its own servers, racks and custom data centers, [56]. Internal resources for such customization allow data centers to gain cost efficiency and can also be used for upgrading existing servers. The second strategy is to use advanced tracking technology to continuously monitor sever characteristics such as heat output, power consumption and utilization. These real-time system measures enable managers to make frequent and faster decisions about retiring, adding, or upgrading servers. Combination of such strategies has allowed Google to upgrade servers by remanufacturing and re-purposing outdated servers and thus to avoid buying over 90,000 new, replacement machines since 2007 [33]. Given the increasing prominence of data center operations and specifically, the importance of server inventory management, our observations in the previous paragraph lead us to the question of effective operating policies that will allow the server inventory manager, who faces demands both for new and used servers from traffic managers, to best use the ability to remanufacture and upgrade servers. We first build a basic model that is aimed at capturing the theoretical trade-offs inherent in the question faced by the inventory manager. We present and analyze this model in the context of production-inventory literature and determine the optimal policy. We then add several extensions to this model that are designed to go beyond the theoretical trade-offs and capture real features that we observed during our visits to the California-based data center mentioned above.

In our basic model, demands for the new products (new servers) and used products (remanufactured servers) occur according to independent Poisson processes. Returned serves arrivals also follow Poisson process. The manager can choose to produce a new server, upgrade a returned server, or idle the production capacity. Finished new or used items are kept in different inventory locations. Different holding costs are charged at each inventory location. If there is not enough inventory to satisfy a demand, the demand is lost and a penalty cost is charged. We develop a continuous-time Markov dynamic program for this model. We prove that the optimal production inventory policy is of the state-dependent base stock form where the state is captured by the level of two types, new and used, of finished goods inventory and the level of returned goods inventory. We show how to compute the policy parameters. As the determination of optimal policy parameters is computationally intensive, we propose a heuristic method. We show that the heuristic achieves close to optimal results.

The rest of the chapter is organized as following. Section 2.2 reviews the existing literature related to production-inventory systems with returns. Section 2.3 establishes the basic model and proves the optimal policy. Section 2.4 develops a heuristic method and compares its performance to optimal results. 2.5 concludes the paper. In chapter 3, we extend our basic model to incorporate additional features observed from practice.

2.2 Literature Review

2.2.1 Literature Review of Production-Inventory Systems with Returns

In this section, we review the existing literature related to continuous production-inventory systems with returns. We concentrate on the stochastic continuous-time models since they are most relevant to the system we study. For a complete literature review on deterministic models with returns, readers are referred to [72, 73, 39].

In the continuous-time models, decisions can be made any at point of time. The goal is to minimize (or maximize) the system cost (or profit), either for expected average cost (profit) rate or the expected total discounted cost (profit). Since our research concentrate on the structure of the optimal control policy, we will review literature that fall into this category.

Optimal Control in Production-Inventory Systems with Returns

Earlier literature in production-inventory system with returns study the system performance under given and mostly static control policies, several recent papers study the different perspectives in reverse supply chain setting and characterize the structure of the optimal control policy.

[30] extend the model of [38] by investigating a make-to-stock queue with product returns. There is capacity constraints, and unmet demand is backordered. Returns are as good as serviceable products, and can be used to serve demand directly. The author proves the optimality of a base stock policy to control production process and derives an exact formula for the optimal base stock level.

[89] extend the model of [30], and analyzes the effects of interdependence between the return and demand on the optimal control policy. But was only able to characterize the structure of the optimal control policy when returns happen instantly.

[26] study the effects of advanced return information by assuming returns are announced in advance and be returned with some probability after some exponentially distributed time. The structure of the optimal policy is characterized. Then they compare the systems with and without the advanced return information and examine the value of information.

[81] analyze an n-stage sequential production-inventory system with demand only for finished goods. There are independent product returns happening at each stages, and returns can be used for production directly.

In each case, the authors show the structure of the optimal control policy is state dependent base stock type. When there are n classes of products, the base stock for class $i, i \in [1, n]$ depends on the joint inventory for the rest $n-1$ classes of products. Also, there are dynamic priority rule, sometimes called switching rules, in producing different products. In most cases, the switching rules can divide the state space into regions and the boundary curves exhibit monotone properties.

To make the models tractable for characterizing the optimal structure, there are typical assumptions in the cost parameter, return process, and number of classes of demands. Although individual assumption appears frequently in existing literature, it is the combination of a set of assumptions that make the model unique and more advanced. Common assumptions for existing literature for production-inventory systems with returns include: (1) Products are returned as serviceable products, so no further re-manufacturing process is needed. This assumption usually helps simplify the analysis since production control for the returned products are not considered. (2) There is (are) only demand(s) for new products. This assumption makes the returned products as production resources, and simplify analysis by neglecting the tradeoff between satisfying demands between new and used products. (3) Unmet demands are lost. The advantage of lost sales vs. backorder scenario is that the state space of the system, usually measured by the product net inventory, is smaller than the backorder scenario. While lost sales scenario only deals with non-negative net inventory, the backorder scenario includes both positive and negative net inventory. So it generates extra difficulty in analyzing the optimal control policy over the whole state space. However, several literature analyzed both lost sales and backorder scenarios and showed both optimal control policy exhibit similar structure. (4) Return is independent from demand. Without this assumption, the system needs to keep track of the life time of products on the market. To the best of our knowledge, no existing paper has shown the structure of the optimal control policy with returns depending on demand.

2.2.2 Literature Review of Other Related Models

Besides the literature in production-inventory system with returns, our research is related to production-inventory system with inventory rationing for multiple classes of demands. Most of literature consider multiple classes of demands that differ in the cost parameters, for example, unit lost sales/backorder cost or demand arrival rate. [90] first consider dynamic scheduling in a make-to-stock system in the context of two identical products with same Poisson demand rates and equal exponential production times. The production decision in their model is governed by a simple base-stock policy for each product. [91] extend this model to multiple identical products and considered general production times. [36] is one of the earliest to show the structure of the optimal policy in the context of two Poisson demand classes with equal production times. [7] analyze the inventory rationing with multiple production facility producing multiple products. [6] analyze inventory rationing in an assemble-to-order system. [31] consider the inventory rationing in a production-inventory system with advance demand information. [8] analyze multiple customer classes with partial backordering. In each case, the authors show there are a rationing levels for different demand classes, below which the corresponding demand class cannot be satisfied. Unmet demand are backordered or lost, depending on the model setting. The ranking of the class parameters usually shed light on heuristics.

Also, our research is related to the admission control literature in queuing systems. In section 3.3, we extend the model to the case where multiple demand classes for new product exist and how admission control affect the structure of the optimal policies. For reviews of related literature, readers are referred to [71, 22, 48]. Recent examples of the application of admission control in production-inventory system include [8, 40].

2.3 Model

2.3.1 Problem Description

We study a production-inventory system where the manufacturer faces demand for both new and used products. Demands for the new products occurs according to a Poisson process with rate λ_x. They are satisfied if there is inventory, or lost with unit lost sales cost c_x. The lost sales cost includes the loss of brand image or good will for the manufacturer. In section 3.1, we show the model can be extended to the backorder case. New products once sold may be returned, which occurs according to an independent Poisson process with rate δ. Demands for the used products occur according to a Poisson process with rate λ_y. They are also satisfied if there is inventory, or lost with unit lost sales cost c_y. We assume $c_y < c_x$ since new products are usually the major product, which is pricier and customer pay more attention to it. The holding costs are h_x and h_y per item per unit time, with $h_y < h_x$ since holding cost is usually proportional the price.

The manufacturer has the flexibility of producing new product from raw material or remanufacturing used product. We assume both the production time follow the same exponential distribution with the rate μ. The same production time is reasonable when both processes consist of similar operations. As for the datacenter case, there is no significant time difference in building a new server from raw material or from a used server. We further assume both operations are preemptive and there is no set-up time in switching between operations. Because of the memoryless property

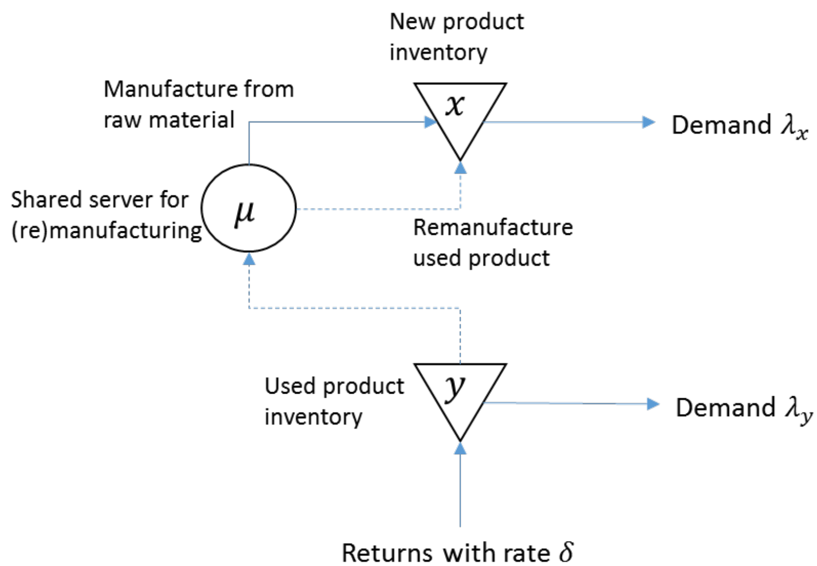

Figure 2.1: Model setting

of the exponential distribution, to continue a interrupted job is equivalent to start from the beginning. The assumption of no set-up time is also reasonable when the set-up time is negligible compared to the production times. The assumptions on the Poisson arrival of demands and exponential production times are similar to previous literature such as [36, 82, 8, 26]. The memoryless property enable us to model the system with Markov decision process, and show the structure of the optimal policy with dynamic programming. For the stability of the system, we assume $\delta < \lambda_x + \lambda_y$ and $\lambda_x < \mu$. The first condition means that return rates must be smaller than the total demand rate, otherwise the inventory for used product or new product will keep increasing. Practically, demand is satisfied and then products can be returned. Return rate cannot exceed its generating source, or the total demand rate. The second condition means that demand for new product must be smaller than the production rate. Figure 2.1 shows the configuration of the system.

The manager of the production-inventory system seeks the optimal control policy to minimize the discounted total cost over the infinite time horizon. At any time, he can decide on the production operations: 1) produce one unit of new product from raw material, 2) produce one unit of new product from used product if there is inventory, and 3) not produce. Let $X(t)$ be the inventory of new product at time t, and $Y(t)$ be the inventory for used product at time t, the system state can be described by $(X(t), Y(t))$, where $X(t), Y(t) \in \mathbb{N}$, $\mathbb{N}$ is the set of natural numbers. The manager only needs to make production decision when the state changes including demand occurrence, production completion or return occurrence.

In specifying details of our model, we have aimed at creating a balance between our motivating context, a data center, and the need for maintaining tractability. We are employing a continuous-time, discrete-state model. Availability of automated tracking technologies and algorithmic decision making that is popular in a high-tech industry like data centers makes is plausible to assume that changes in state will be recognized in real time and decisions can be made in continuous time. The assumption of continuous state recognizes that data centers use a very large number of servers and a fluid approximation may be reasonable. The assumption of continuous state, however, is primarily for the reasons of tractability. We choose to model the arrivals of new and used server demands as Poisson processes. The generation of these demands is the result of a complex process; traffic managers continuously observe changes in the data demand from Web-users and use a heuristic decision process to determine whether they should request a new server or a used server in case the demand from Web-users shows an upward trend. There are cost implications for traffic managers to hold theses servers. Therefore, in case the demand shows a sustained downward trend, traffic managers return servers with low usage. Thus the generation of new and used demands as well as returns is the result of a complex process that combines real observations with traffic managers' heuristic decision-making. In our basic model, rather than speculating about the exact nature of this process, we simply model the results of this process as Poisson processes for new and used demands and returns. In a later section, we extend our model to capture additional details of generation of demands and returns. In addition, private communications with the inventory managers of the data center suggest that, in their internal analysis, they too use Poisson processes to model these arrivals.

Our assumptions about cost structures follow standard approach taken in such models. They also reasonable reflect reality. We assume $c_y < c_x$ since new servers are more expensive than used servers. The holding costs are h_x and h_y per item per unit time, with $h_y < h_x$ since holding cost is usually proportional to the price. Our assumption of equal service rates for customizing new servers and for upgrading old ones is a reasonable approximation of what we observed at this data center. The assumption of exponential service times and preemptive-service times, however, are made for tractability and are standard in these types of production-inventory models. Finally, the basic model assumes that unmet demands are lost. Later, an extension considers the case of backlogged demands.

Let α be the time discount factor, and $N_i(t)$ be the units of lost sales over time $[0, t]$ for product i, $i \in \{x, y\}$. The goal is to find the optimal control policy π to minimizes the expected total discounted cost $v^\pi(x, y)$ with starting state $(x, y).\pi(t) \in \{0, 1, 2\}$, where

$$\pi(t) = \begin{cases} 0 & \text{Stay idle} \\ 1 & \text{Produce new product from raw material} \\ 2 & \text{Produce new product from used product} \end{cases}$$

The expected discounted cost over an infinite planning horizon with starting state (x, y) and policy π is defined as $v^\pi(x, y)$, where

$$v^\pi(x, y) = E^\pi_{(x,y)} \left[\int_0^\infty e^{-\alpha t} h(X(t), Y(t)) dt + \sum_{i=x,y} \int_0^T e^{-\alpha t} c_i dN_i(t) \right] \tag{2.1}$$

The optimal cost function $v^*(x, y)$ satisfies $v^*(x, y) = \min_\pi v^\pi(x, y)$.

Summary of Parameters:

- λ_x : Demand rate for the new product
- λ_y: Demand rate for the old product
- δ : Return rate for the old product
- μ : (Re)manufacture rate for both used and new product
- h_x: Unit holding cost for new product (per time period)
- h_y: Unit holding cost for used product (per time period)
- c_x: Unit lost sales cost for new product
- c_y : Unit lost sales cost for used product
- α : Discount rate

2.3.2 Uniformization and Bellman's Equation

The decision process can be solved with a continuous Markov decision process. Following [67, 53], we uniformize the process with $\beta = \lambda_x + \lambda_y + \mu + \delta$. The system can be transformed into a discrete Markov decision process with all transitions happen at the rate β.

Uniformization

The idea behind uniformization is to transform the continuous-time problem into a discrete-time problem. In the original settings, the times between events are exponentially distributed with different transition rates. For example, a demand for new product occurs on average every $1/\lambda_x$ unit of time; a return occurs on average every $1/\delta$ unit of time; a production completes on average after $1/\mu$ unit of time. In a complex system with multiple kinds of events, such as multiclass of demands, return, production and etc., it is difficult to analyze the system directly.

Following [67, section 11.5], the system can be discretized with a uniformized transition rate β. Consider the system where an event happens on average every $1/\beta$ unit of time, where $\beta = \lambda_x + \lambda_y + \mu + \delta$, and with probability λ_x/β the event is a demand for new product, with probability λ_y/β the event is a demand for used product, with probability δ/β the event is a return, and with probability μ/β the event is a production completion. This system is mathematically identical to the original system, but the transition rate between any event is exponentially distributed with the same rate β. Notice that in some cases, uniformization may introduce pseudo transitions into the current state itself. See an example in section 4.3.

To evaluate the total discounted cost with discount factor α for the infinite time horizon, [67, section 11.5, p.563] shows the mathematics deduction of using $1/\alpha+\beta$ as the coefficient of normalization, as seen in equation 2.3.

Optimality of the Bellman's Equation

The optimal cost function v^* satisfies the following optimality equation. It is also called Bellman's equation [5].

$$v^*(x, y) = Tv^*(x, y), \forall x, y \geq 0 \tag{2.2}$$

Operator T is a contraction mapping defined as

$$Tv(x, y) = \frac{1}{\alpha + \beta}[h(x, y) + \lambda_x T_1 v(x, y) + \lambda_y T_2 v(x, y) + \delta T_3 v(x, y) + \mu T_4 v(x, y)] \tag{2.3}$$

with $h(x, y) = h_x x + h_y y$ and operators T_i, $i = 1, 2, 3, 4$ defined as follows,

$$T_1 v(x,y) = \begin{cases} v(x,y) + c_x & \text{if } x = 0 \\ v(x-1,y) & \text{if } x > 0 \end{cases}$$
$$T_2 v(x,y) = \begin{cases} v(x,y) + c_y & \text{if } y = 0 \\ v(x,y-1) & \text{if } y > 0 \end{cases}$$
$$T_3 v(x,y) = v(x,y+1)$$
$$T_4 v(x,y) = \begin{cases} \min[v(x,y), v(x+1,y)] & \text{if } y = 0 \\ \min[v(x,y), v(x+1,y), v(x+1,y-1)] & \text{if } y > 0 \end{cases}$$

Operators T_1 and T_2 respectively indicates new and old demand must be satisfied if there is inventory, otherwise the demand is lost and corresponding lost sales cost is incurred. Operator T_3 corresponds to the used product return. Once return occurs, it must be accepted leading the inventory increase from (x, y) to $(x, y+1)$. Operator T_4 is associated with the optimal production decisions. The choice is within stay idle which leaves the state unchanged, produce new product from raw material which increase inventory from (x, y) to $(x + 1, y)$, or produce new product from returned product if available $(y > 0)$ which increase inventory from (x, y) to $(x + 1, y - 1)$. In summary, the optimal control policy chooses the feasible action that leads to the states with the lowest cost.

Since $Tv(x, y)$ is a contraction mapping [67, theorem 6.2.3, p.150], Banach fixed point theorem ensures that $\lim_{n\to\infty} v_n(x, y)$, with $v_{n+1}(x, y) = Tv_n(x, y)$, will converge to the fixed-point of the operator T. It is the optimal value function $v^*(x, y)$, and also the unique solution of $v^*(x, y) = Tv^*(x, y)$.

2.3.3 Optimal Policy under Discounted Total Cost

To analyze the structure of the optimal policy, we define these difference operators:

- First difference operators
  - $D_x v(x, y) = v(x + 1, y) - v(x, y)$

- $D_y v(x,y) = v(x,y+1) - v(x,y)$
- $D_{x-y} v(x,y) = v(x+1,y-1) - v(x,y)$

• Second difference operators

- $D_{x,x} v(x,y) = D_x D_x v(x,y) = D_x v(x+1,y) - D_x v(x,y)$
- $D_{y,y} v(x,y) = D_y D_y v(x,y) = D_y v(x,y+1) - D_y v(x,y)$
- $D_{x,y} v(x,y) = D_x D_y v(x,y) = D_y v(x+1,y) - D_y v(x,y)$
- $D_{x,x-y} v(x,y) = D_x D_{x-y} v(x,y) = D_{x-y} v(x+1,y) - D_{x-y} v(x,y)$
- $D_{y,x-y} v(x,y) = D_y D_{x-y} v(x,y) = D_{x-y} v(x,y+1) - D_{x-y} v(x,y)$

Notice that in the discrete case, the order of applying the operators doesn't affect the result, e.g: $D_{x,y}v = D_{y,x}v$, $D_{x,x-y}v = D_{x-y,x}v$, and etc.

The analogy of the difference operators in the continuous domain are partial derivatives. For a function $v(x,y)$ defined in the discrete and continuous domain, $D_x v \leftrightarrow \partial v/\partial x$, $D_{x,y} v \leftrightarrow \partial^2 v/\partial x \partial y$ and etc. Properties of partial derivatives in a continuous function determines the property for the function. For example, if for every x $\partial^2 v/\partial x^2 \geq 0$, v is convex in x. In the same manner, properties of the difference operators applied on the discrete function also capture the properties of the discrete function, if for every x $D_{x,x} v \geq 0$, v is convex in x.

Definition 2.1. Let U be a set of real-valued functions defined on $\mathbb{N}^2$. If $v \in U$, then for all $x, y \geq 0$:

A.1 $D_x v(x,y) \geq -c_x$, $\forall x, y \geq 0$

A.2 $D_{x,x} v(x,y) \geq D_{x,x-y} v(x,y)$, $\forall x \geq 0, y > 0$

A.3 $D_{x,x-y} v(x,y) \geq 0$, $\forall x \geq 0, y > 0$

A.4 $D_{x,y}v(x, y-1) \geq 0$, $\forall y > 0$

A.5 $D_{y,y}v(x, y) \geq 0$, $\forall x, y \geq 0$

A.6 $D_{y,x-y}v(x, y) \leq 0$, $\forall x \geq 0, y > 0$

A.7 $D_y v(x, y) \geq -c_y$, $\forall x, y \geq 0$

Lemma 2.1. *If $v \in U$, then $Tv \in U$. Moreover, the optimal value function $v^* \in U$.*

To prove this lemma, we first propose a set of function satisfying the above properties A.1 to A.7. Then by showing operator T preserves all the properties and T is a contraction mapping, we can conclude the optimal cost function v^* satisfies all the properties.

The difficulties of proof come from the number of properties, and the number of decision choices in the $\min[\cdots]$ part in $T_4 v$. For each operator $T_i v, i = 1 \ldots 4$, we need to show it preserves all the properties listed in definition 2.1. Since properties are based on difference operators and the state space is bounded below by 0, special care needs to be taken when hitting the boundary. Also, the combinatorial nature in comparing multiple the $\min[\cdots]$ creates great difficulty. To the best of our knowledge, among existing literature that involve difference operator based properties related to $\min[\cdots]$, the largest number of comparison values inside $\min[\cdots]$ is 3. Our model achieves this dimensionality.

The structure of the optimal control policy can then be inferred from the properties. For continuity of the dissertation, we leave the details of the proof in appendix A.1.

Intuitions behind the Properties

The intuitions behind the set of properties is the key to show the structure of the optimal policy. Here we explain the intuition behind each property when applied to

v^*.

- A.1 implies that when net hand inventory for new product is $x+1$, it is better to satisfy the demand by reducing one unit on hand inventory than to keep the inventory unchanged and incur a lost sales cost of c_x. It guarantees demand from new product is always satisfied.

- A.2 implies the marginal cost difference in producing new product from raw material and from returned product, or $D_x v - D_{x-y} v$, is non-decreasing in new product inventory x. So given y, there is a threshold value for x where $D_x v - D_{x-y} v$ changes sign. The threshold value of x serves as a switching point of producing from raw material vs. producing from returned product.

- A.3 implies the marginal cost of producing new product from returned product, or $D_{x-y} v$, is non-decreasing with new product inventory x. So given y, there is a threshold value for x where $D_{x-y} v$ changes sign. The threshold value of x serves as a switching point of producing from returned product vs. being idle.

- A.4 implies that marginal cost of satisfying a unit demand for used product, or $D_y v$, is non-decreasing with inventory of x. So given y, there is a threshold value for x where $D_y v$ changes sign.

- A.5 implies the marginal cost of having an extra unit of used product, or $D_y v$, is non-decreasing with inventory of y. It implies v is convex in y.

- A.6 implies the marginal cost of producing one unit of new product with one unit of used product, or $D_{x-y} v$, is non-increasing with inventory of y.

- A.7 is similar to A.1. It implies that when net hand inventory for used product is $y+1$, it is better to satisfy the demand by reducing one unit on hand inventory than to keep the inventory unchanged and incur a lost sales cost of c_y. It guarantees demand from used product is always satisfied.

Theorem 2.1. *Let* $B(y) = \min[x \geq 0 | D_x v(x,y) > 0, D_{x-y} v(x,y) > 0]$, $C(y) = \min[x \geq 0 | D_x v(x,y) - D_{x-y} v(x,y) > 0]$. *Given the inventory level* (x,y), *it is optimal to be idle if the inventory for the new product* x *is no less than* $B(y)$. *If not this case, it is optimal to produce new product from used ones if the inventory for the new product* x *is no less than* $C(y)$, *and to produce new produce from raw material otherwise.*

$B(y)$ *and* $C(y)$ *satisfy these properties:*

1. $B(y+1) + 1 \geq B(y)$

2. $C(y) \geq C(y+1)$

Proof. At $(x = B(y+1), y+1)$, by definition $D_x v(x, y+1) > 0$ and $D_{x-y} v(x, y+1) > 0$. By A.6 and A.3, $D_{x-y} v(x+1, y) \underset{\text{A.3}}{\geq} D_{x-y} v(x,y) \underset{\text{A.6}}{\geq} D_{x-y} v(x, y+1) > 0$. By A.3, $D_x v(x+1, y) \underset{\text{A.3}}{\geq} D_x v(x, y+1) > 0$. Since $x + 1 = B(y+1) + 1$, we have $B(y+1) + 1 \geq B(y)$.

At $(x = C(y), y)$, by definition $D_x v(x,y) - D_{x-y} v(x,y) > 0$. By A.4 $D_{x,y} v(x,y) \geq 0$, and A.6 $D_{y,x-y} v(x,y) \leq 0$, so $D_{x,y} v(x,y) - D_{y,x-y} v(x,y) \geq 0$. Thus $D_x v(x, y+1) - D_{x-y} v(x, y+1) > D_x v(x,y) - D_{x-y} v(x,y) > 0$. We have $C(y) \geq C(y+1)$. □

An Example of the Structure of the Optimal Policy

Figure 2.2 illustrates an example of the optimal control policy for a given set of parameters ($\mu = 1, \lambda_x = 0.85, \lambda_y = 0.85, \delta = 0.8, h_x = 2, h_y = 1, c_x = 11.4, c_y = 5.7, \alpha = 0.05$). The state space is divided by $B(y)$, the production on/off curve, and $C(y)$, the production switching curve, into several regions.

- Some notations:

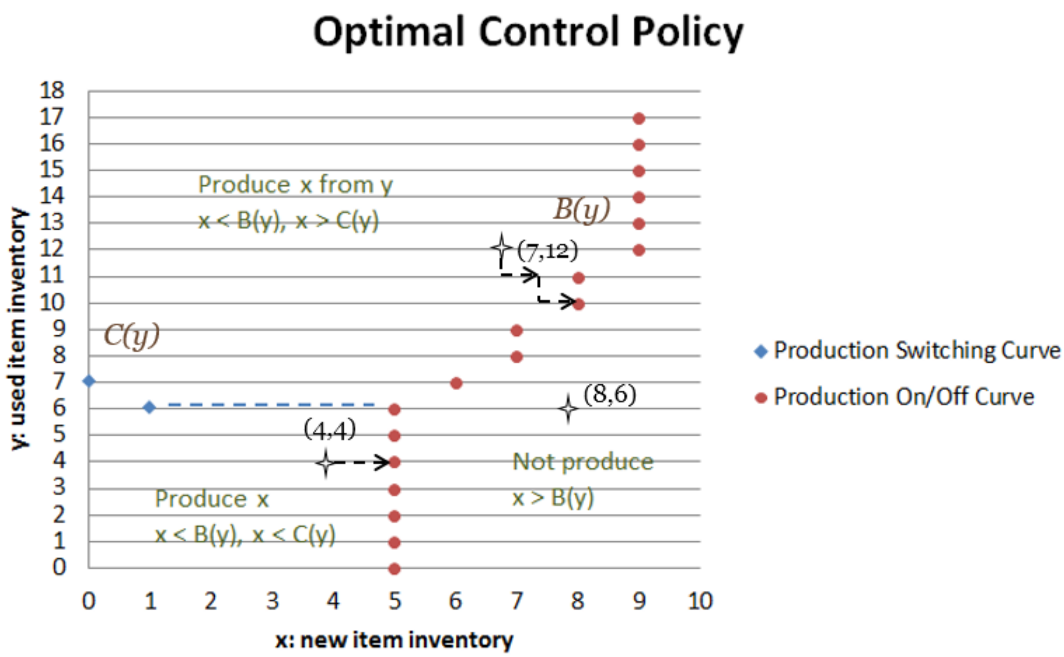

Figure 2.2: The optimal control policy with parameter settings: $(\mu = 1, \lambda_x = 0.85, \lambda_y = 0.85, \delta = 0.8, h_x = 2, h_y = 1, c_x = 11.4, c_y = 5.7, \alpha = 0.05)$

  - The blue diamond dots is for $C(y)$: the production switching curve between producing new product from raw material or from used product.
  - The red dots is for $B(y)$: the production on/off curve, when x is beyond $B(y)$ the production will stay idle. For example, $B(y = 4) = 5$
  - The graph is connected from the left, meaning for state (x, y), $x > 0$, that is on $B(y)$ or $C(y)$ the optimal action is the same as in state $(x - 1, y)$.
  - The black dotted directed lines indicate the potential sample paths after applying the corresponding actions.

- Sample paths:

  - At state $(x = 4, y = 4)$, $B(y = 4) = x < B(y)$ and $x < C(y)$, then the optimal policy is to produce a new product from raw material. The potential next state will be $(x = 4, y = 5)$ if the production completion happens first among all the potential events (demands, returns, and etc.).
  - At state $(x = 7, y = 12)$, $x < B(y)$ and $x > C(y)$, then the optimal policy is to produce new produce from used product. The potential next state will be $(x = 8, y = 11)$ if the production completion happens first among all the potential events.
  - At state $(x = 8, y = 6)$, $x > B(y)$, then the optimal policy is to be idle. The potential next state can be $(x = 7, y = 6)$: a demand for new product occurs first; $(x = 8, y = 5)$: a demand for used product occurs first; or $(x = 8, y = 7)$: a return occurs first.

2.3.4 Optimal Policy under Average Cost

The similar results can be extended to the average cost case. Given a policy π, the average cost function is

$$J^{\pi}(x,y) = \lim_{T\to\infty} \sup \frac{1}{T} E^{\pi}_{(x,y)} \left[\int_0^T e^{-\alpha t} h(X(t),Y(t))dt + \sum_{i=x,y} \int_0^T e^{-\alpha t} c_i dN_i(t) \right] \quad (2.4)$$

The optimal policy π^* satisfies $J^*(x,y) = \inf_{\pi} J^{\pi}(x,y)$ for all the states (x,y). As shown in [84], the optimal policy for the average cost is the limit of the discounted cost policy when the discount factor α goes to 0. Thus they share the same structure in the optimal policy. Moreover, the optimal average cost J^* is the limit of $\alpha v^*(x,y)$ when α goes to 0.

Theorem 2.2. *The optimal policy under the average cost criterion share the same structure as the optimal policy under the discounted total cost criterion. Furthermore, there exists a finite constant average cost J^* such that $J^*(x,y) = J^*$ for all states (x,y).*

The proof is similar to the proofs in [21, 8].

Proof. To show there exist an optimal policy under the average cost scenario, and for the average cost to by finite and independent of the initial state, we need to show the following two conditions hold [13, 84]: a) there exists a stationary policy π that induces an irreducible positive recurrent Markov chain with finite average cost J^{π}, and b) the number of states for which one stage cost $h(x,y) = h_x x + h_y y \leq J^{\pi}$ is finite.

To prove condition a), define a two-threshold control policy (s_x, s_y), with $s_x, s_y > 0$. The production decision is determined as: 1) produce new product from used product when $x < s_x$ and $y \geq s_y$; 2) produce new product from raw material when $x < s_x$ and $y < s_y$; 3) stay idle when $x \geq s_x$. It is straightforward to show that the optimal cost under the two-threshold (s_x, s_y) policy is finite, and the transition of states is bounded by $(s_x, 0)$. More details about the heuristics can be found in section 2.4.4, heuristic H_3, with a transition diagram.

As for condition b), since the one stage cost $h(x,y) = h_x x + h_y y$ is non-decreasing in x and y, the number of states for which $h(x,y) = h_x x + h_y y \leq \gamma$ is finite for any positive value γ.

By [84], there exists a function $f(x,y)$ and a constant average cost J^* satisfying the following inequality:

$$f(x,y) + J^* \geq h(x,y) + \lambda_x T_1 f(x,y) + \lambda_y T_2 f(x,y) + \delta T_3 f(x,y) + \mu T_4 f(x,y)$$

The optimal policy under the average cost scenario is the stationary policy that minimize the right hand side of the above inequality, and yields a constant average cost J^*. The properties of the optimal policy is decried by the properties of $f(x,y)$. Since the same operators T_1 to T_4 are applied to both $v(x,y)$ and $f(x,y)$, the structure of the optimal policy under the average cost is the same as that of the discounted total cost. □

2.4 Heuristics

In this numerical study, we concentrate on the average cost problems since they don't depend on the discount factor and the starting state. The purpose of this numerical study is to propose heuristics that are easy to implement, since the optimal policy is state dependent and thus not easy to compute.

2.4.1 Computational Procedure

The average cost is computed by solving the corresponding dynamic program with value iteration algorithm described in [67, Chapter 8]. The states space is truncated with a positive value of boundary where the average cost is no longer sensitive to the value of the boundary.

From section 2.3.4, we show the structure of the optimal policy under the average cost is similar to that under the discounted cost. Thus the optimal policy for the average cost is also of base-stock type. The heuristics are based on this intuition, where heuristics H_1 and H_2 are within one parameter (s_x) base stock policies with average cost $C(s_x)$, and heuristics H_3 is of two-parameter (s_x, s_y) base stock policy with average cost $C(s_x, s_y)$. To find the optimal value for the parameter(s) in each heuristics, we make the plausible assumption that the cost functions $C(s_x)$ (for H_1, H_2) and $C(s_x, s_y)$ (for H_3) are unimodal. The assumption is checked on several instances with exhaustive search. Based on the unimodularity assumption, we can find the optimal s_x and s_y with brute force search or maximal gradient search.

The performances of the heuristics are measured by the cost increase percentage of heuristics over the cost with optimal policy ΔJ^{H_i}, where $\Delta J^{H_i} = {}^{J^{H_i}-J^*}/_{J^*}$, $J^*$is the optimal average cost and J^{H_i} is the optimal average cost under heuristic H_i, $i = 1, 2, 3$. The lower ΔJ^{H_i} is, the better the heuristics H_i approximates the optimal policy.

2.4.2 Heuristics H_1: Used Product First

Under heuristics H_1, always produce new product from used product when new product is below the threshold value s_x. The production can be described as: 1) produce new product from used product if $x < s_x$, and $y > 0$; 2) produce new product from raw material if $x < s_x$, and $y = 0$; 3) stay idle when $x \geq s_x$. Since the production priority is completely assigned to using used product.

2.4.3 Heuristics H_2: Raw Material First

Heuristics H_2 always produce new product from the raw material when new product inventory is below the threshold value s_x. The production can be described as: 1) produce new product from raw material if $x < s_x$; 2) stay idle when $x \geq s_x$.

This Markov chain, with state space finite in one dimension x and infinity in the other y, is called *1D Markov chain* [64]. While classic approaches provide exact expressions for the state probabilities, they may be difficult to analyze and complicated to evaluate numerically. Algorithmic methods including matrix geometric methods [60] and matrix analytic methods [51] can calculate the computational probability

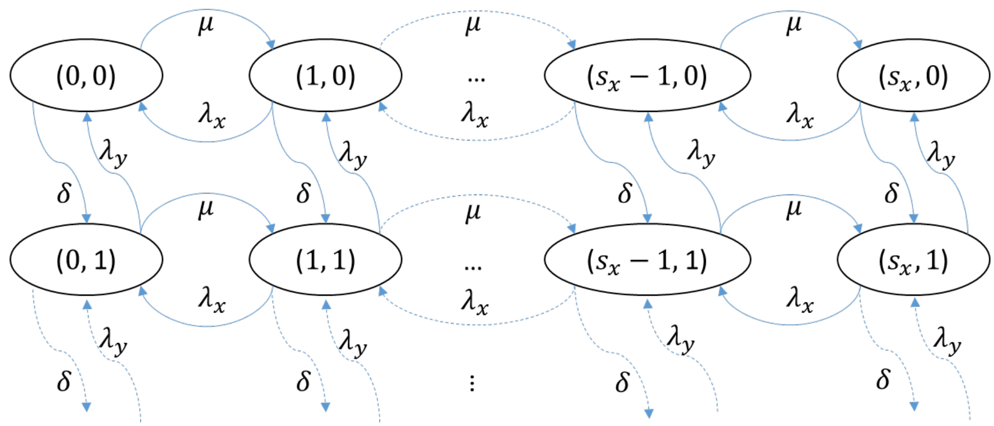

Figure 2.3: Transition diagram of the system under heuristics H_2

efficiently when the 1D Markov chain has certain structures, such as the one shown in figure 2.3.

2.4.4 Heuristics H_3: Two-Threshold Policy

Heuristic H_3 is a two-parameter base-stock policy (s_x, s_y), $s_x, s_y \geq 0$. The production process is turned on when the inventory for new product is less than threshold value s_x, otherwise the serve will stay idle. If needs to produce new product, produce new product from raw material when used product inventory is less than threshold value s_y; or produce new product from the used product. The production decision is determined as: 1) produce new product from used product when $x < s_x$ and $y \geq s_y$; 2) produce new product from raw material when $x < s_x$ and $y < s_y$; 3) stay idle when $x \geq s_x$. Notice that heuristics H_1 , used product first, is a special case of H_3 with $s_y = 1$; and heuristics H_2 , raw material first, is a special case of H_3 with $s_y = +\infty$. The system transition diagram is shown in figure 2.4.

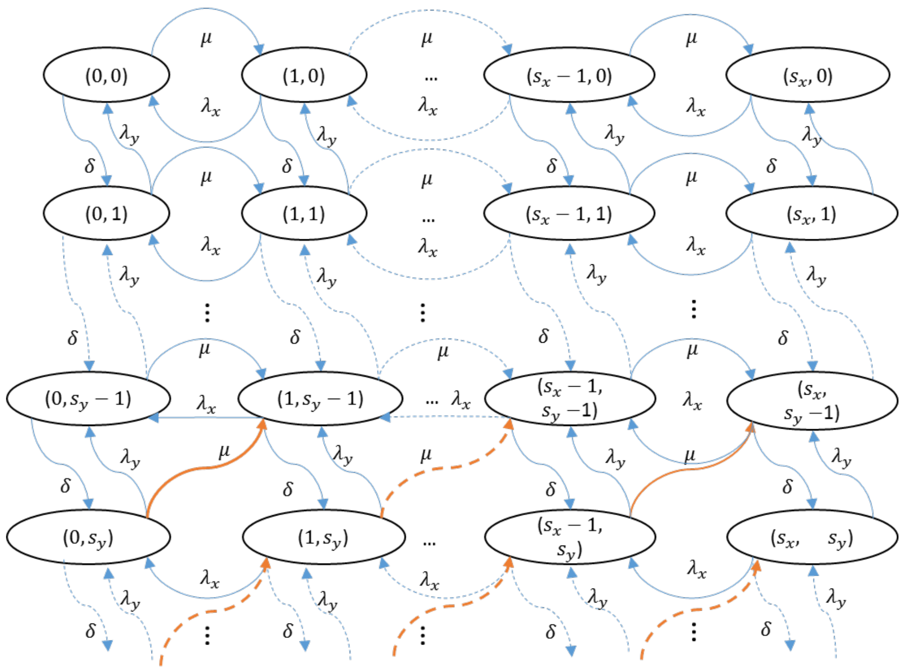

Figure 2.4: Transition diagram with heuristics $H_3 : (s_x, s_y)$

2.4.5 Experimental Design

The model evaluating average cost includes 8 parameters $(\lambda_x, \lambda_y, \delta, \mu, h_x, h_y, c_x, c_y)$. Without loss of generality, set $\mu = 1$ and $h_y = 1$ which is equivalent to choosing time and monetary unit, see [26, p395]. To evaluate the system, we consider parameter settings with 1) holding cost ratio $\frac{h_x}{h_y} \in \{1, 1.5, 2\}$; 2) newsvendor ratio $\frac{c_i}{c_i+h_i} \in \{0.8, 0.9, 0.99\}$, $i = x, y$; 3) system utilization $\frac{\lambda_x}{\mu} \in \{0.8, 0.9, 0.99\}$; 4) demand ratio $\frac{\lambda_y}{\lambda_x} \in \{0.25, 0.5, 0.75\}$; and 5) return ratio $\frac{\delta}{\lambda_y} \in \{0.75, 1, 1.25\}$. The ratios are typical measurements of the queuing system parameters, and we select three values for each representing low, median and high levels. Based on the potential parameter values, we further select three values for all the parameters, highlighted in figure 2.1. Specifically $h_x \in \{1, 1.5, 2\}, c_x \in \{4, 13.5, 198\}, c_y \in \{4, 9, 99\}, \lambda_x \in \{0.8, 0.9, 0.99\}, \lambda_y \in \{0.2, 0.45, 0.74\}, \delta \in \{0.15, 0.45, 0.93\}$. We did numerical experiments for the 729 combinations of the above values with $\mu = 1$ and $h_y = 1$.

Table 2.1: Experimental design

	value given						value calculated		
hx/hy	1	1.5	2		hx	(hy*value)	1	1.5	2
c/(c+h)	0.8	0.9	0.99		cx	(h*value/(1-value))	4	13.5	198
					cy		4	9	99
lambdax/mu	0.8	0.9	0.99		lambdax	(mu*value)	0.8	0.9	0.99
							0.2	0.4	0.6
							0.225	0.45	0.675
lambday/lambdax	0.25	0.5	0.75		lambday	(lambdax*value)	0.2475	0.495	0.7425
delta/lambday	0.75	1	1.25		delta	(lambday*value)	0.15	0.3	0.45
							0.16875	0.3375	0.50625
							0.185625	0.37125	0.556875
							0.2	0.4	0.6
							0.225	0.45	0.675
							0.2475	0.495	0.7425
							0.25	0.5	0.75
							0.28125	0.5625	0.84375
							0.309375	0.61875	0.928125

The program runs on a personal computer with Intel Core i5 2.5GHz processor and 8 GB memory. The computing time for each instance varies from around 10 seconds to 10 minutes. With brute force search on the optimal parameters (s_x, s_y), the computational time will be very large if the optimal values is large.

2.4.6 Results Discussion

There are 729 combinations of settings for each of the heuristics. Out of the 729 combinations of the parameter values, $newsratio(x){=}{}^{c_x}/_{c_x+h_x}$ ranges from 0.67 to 0.99, $newsratio(y)\ ={}^{c_y}/_{c_y+h_y} \in \{0.8, 0.9, 0.99\}$ and ${}^{\delta}/_{\lambda_y}$ ranges from 0.2 to 4.65.

For H_1, used product first, the average cost increase is 22.4%, with minimum approaching 0 and maximum at 146%. The distribution of the cost increase is positively skewed, as shown in figure 2.5.

For H_2, raw material first, the average cost increase is 127%, with minimum approaching 0, and maximum at 852%. The distribution of the cost increase is positively

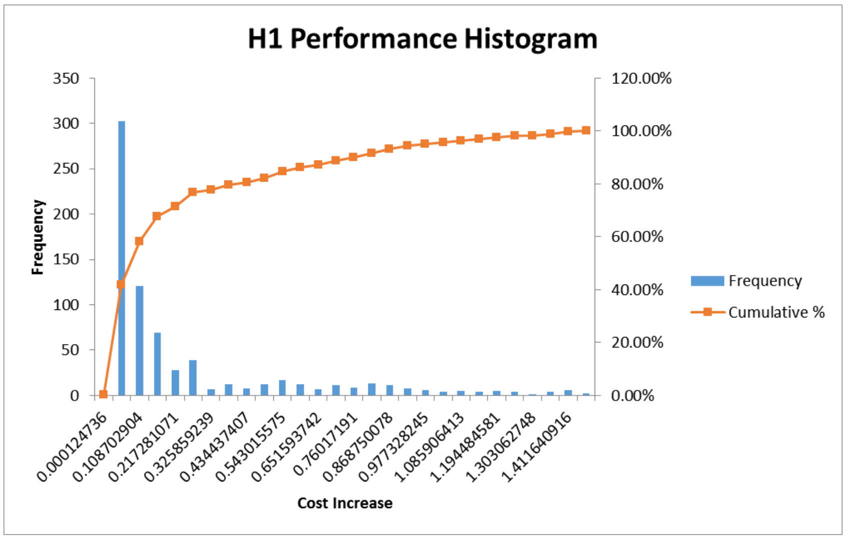

Figure 2.5: H_1: used product first performance histogram

skewed, as shown in figure 2.6. Near 300 instances have cost increase less than 30%, and the cost increase for the rest instances spread out relatively evenly from 30% to 442%.

We concentrate the analysis on the performance of the two-threshold heuristics H_3 since it is the general case of H_1 and H_2, and always yields the best policy among these three. The distribution of the cost increase for H_3 is shown in figure 2.7. The maximum value of cost increase percentage ΔJ, is 33.2% and the average is 4%.

When $\delta/\lambda_y < 1$, the maximum value of ΔJ is 7.2% with parameter settings as $(\lambda_x = 0.99, \lambda_y = 0.74, \delta = 0.45, \mu = 1, h_x = 1, h_y = 1, c_x = 13.5, c_y = 99)$; the average value is 0.2%; and it achieves near optimal cost for many cases with near 0 cost increase percentage. The $\delta/\lambda_y < 1$ constrain comes from the intuition of demand process for used product is satisfied from the on hand inventory, which is replenished solely by an independent return process with rate δ. With $\delta/\lambda_y < 1$, the on hand inventory for y will not go to infinity even without the remanufacturing process. Even with $\delta/\lambda_y \geq 1$,

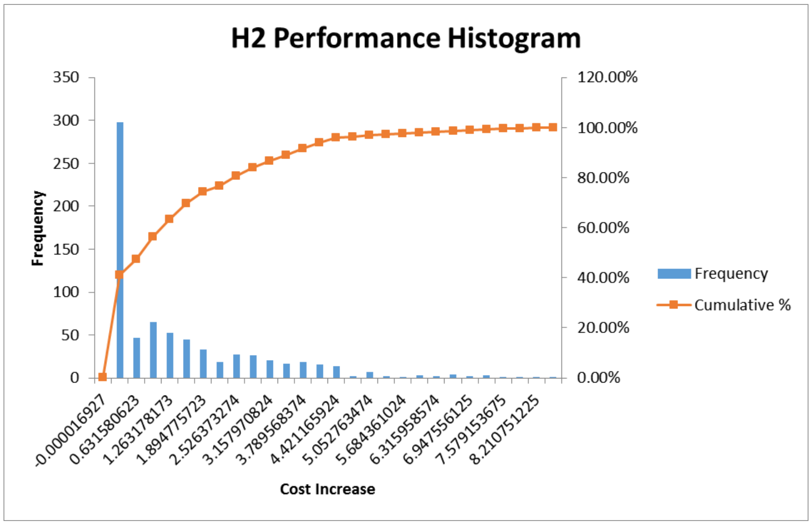

Figure 2.6: H_2: raw material first performance histogram

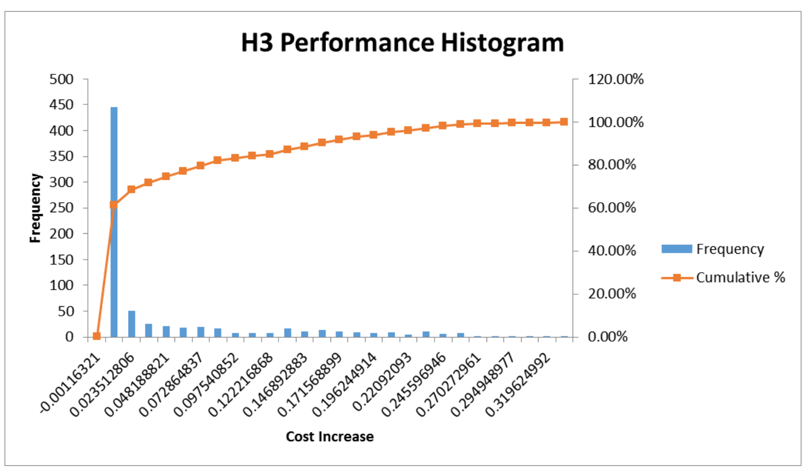

Figure 2.7: H_3: two-threshold policy performance histogram

the system may still be stable with the remanufacturing process. Thus we consider both cases when $\delta/\lambda_y < 1$ and $\delta/\lambda_y \geq 1$.

Even though heuristics H_3 outperforms H_1 and H_2, a nature question to ask is when will single parameter heuristics H_1 and H_2 perform well? We analyze the instances for H_3 with optimal $s_x = 1$, instances for H_3 with optimal $s_x =$bound(x) , and observe what parameter settings do they have.

When will H_1 (Used Product First) Perform Optimally?

184 combinations has optimal value at $s_y = 1$, in which case H_1will also perform optimally (with the optimal s_x and $s_y = 1$). This instances all occur with corresponding $\delta/\lambda_y > 1$ (but not the other way around), and newsratio for y never takes value of 0.99. The intuition is that when return rate is larger than demand for the used product, or δ/λ_y> 1; and if c_y is not too large compared to h_y, or *newsratioy* $\in \{0.8, 0.9\}$, there are enough replenishment for used product, and it is not too expensive to incur a lost sales for used product. Strict priority to always remanufacture used product can be optimal in these cases.

When will H_2(Raw Material First) Perform Optimally?

53 combinations has optimal value with s_y equals the boundary for y (since space needs to be truncated for the value iteration algorithm). In these cases, H_2 with s_y set to a very large number (1000, in this case) performs very well with $\triangle J^{H_2}$ at the level of 10^{-8} or less. The 53 instances all occur with corresponding $\delta/\lambda_y < 1$ (but not the other way around), and the newsratio for y being 0.99 (also not the other way around). The intuition is that when there is not enough replenishment from returns to satisfy the demand for used product, and the lost sales cost for used product is very high compared to the holding cost, always use raw material to produce new product can be optimal in these cases.

The Effects of Return Rate δ

Figure 2.8 shows the performance among all heuristics v.s. δ/λ_y. In each graph, x-axis shows the cost increase measure by $\triangle J^{H_i}$, and y-axis shows the value of δ/λ_y. In the left upper graph, three heuristics are labeled by different colors. We can see that on average the black one, or H_3, performs best with narrow spread of cost increase under all values of δ/λ_y. Red one, or H_1, performs better than the green one, or H_2.

Heuristics H_3 outperform H_1 and H_2 is as expected, since H_3 is the general case of the other two. The intuition behind H_1 outperform H_2 (in the experiment settings) is that H_1 will always utilize used product, instead of solely holding them on hand to satisfy demand from used product. When the holding cost of used product is relatively high (for example, $newsratio_y < 0.9$), in most cases it will be beneficial to use H_1 over H_2.

We analyze the effects of δ on the cost increase percentage (ΔJ^{H_3}) using heuristics (s_x, s_y) vs. using the optimal control policy. δ varies from 0.1 to 0.9. $\mu = 1, \lambda_y = 1$.

We consider two cases based on newsvendor ratio, or $newsratio = \frac{c_i}{c_i+h_i}, i \in \{x, y\}$. Figure 2.9, shows the result when newsratio is high, both 0.99 for x and y, with parameter settings ($\lambda_x = 0.8, \lambda_y = 0.4, \delta, \mu = 1, h_x = 2, h_y = 1, c_x = 500, c_y = 250$). Figure 2.10 shows the result when newsratio is low, both 0.7 for x and y, with parameter settings ($\lambda_x = 0.8, \lambda_y = 0.4, \delta, \mu = 1, h_x = 2, h_y = 1, c_x = 5, c_y = 2.5$). Our observations are:

- s_x is non-decreasing with δ. The intuition is when more return occurs, or δ increases, it is more and more beneficial to convert the used product to new product to keep the cost low, thus not decrease s_x. Otherwise, the holding cost of the used product will increase the average cost. The effect is more obvious when the holding cost for used product is relatively high to the lost sales, as shown the flat trend of s_x in figure 2.9 vs. the increasing trend of s_x in figure

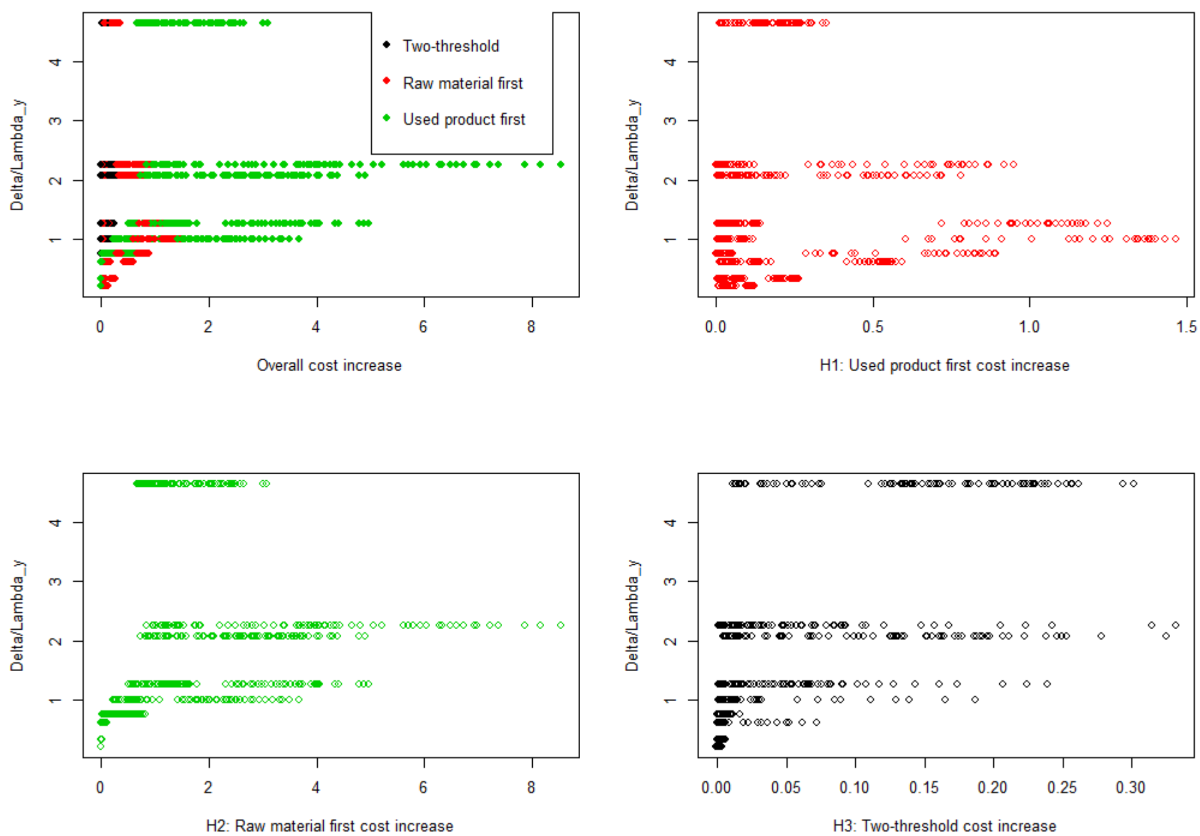

Figure 2.8: $\frac{\delta}{\lambda_y}$ v.s. cost increase

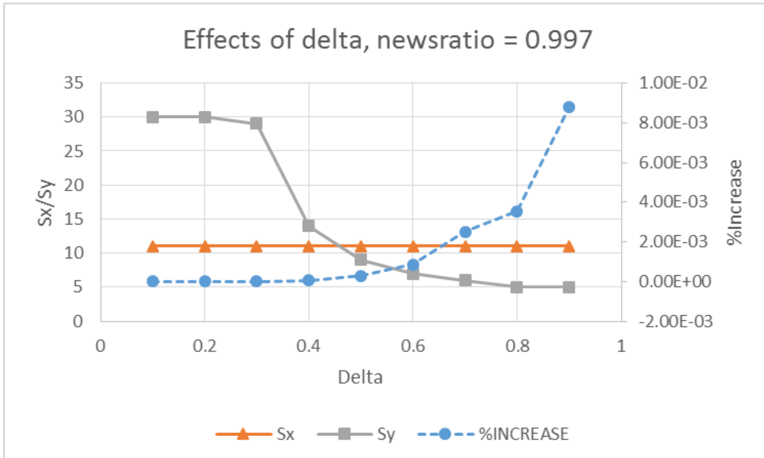

Figure 2.9: δ vs. ΔJ^{H_3}, with $\frac{c}{c+h} = 0.997$

2.10.

- s_y is non-increasing with δ. It is intuitive since when more returns occur, or δ increases, we can keep less s_y to hedge for the demand for used product. It will be replenished quickly.

- The cost increase percentage, ΔJ^{H_3}, is non-decreasing with δ. When δ is very low (at 0.1), there are not enough source for returned products, and on hand inventory stays as low level. Since most of the demand for used products will be lost, there won't be many situation to decide on the production control. Thus the optimal control policy doesn't bring much benefit over heuristics. As δ increases, the inventory for used products increases and optimal control policy outperforms heuristics by always choosing the optimal production decision.

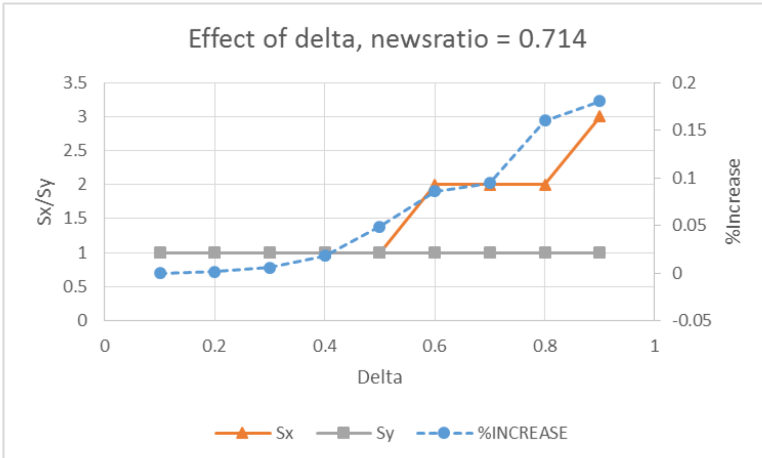

Figure 2.10: δ vs. ΔJ^{H_3} when $\frac{c}{c+h} = 0.714$

Effects of Holding Cost h_x

We analyze the effects of h_x on the cost increase percentage (ΔJ^{H_3}) . Two cases are considered based on level of newsratio. Figure 2.11, shows the result when newsratio is high. The parameter settings is $(\lambda_x = 0.8, \lambda_y = 0.4, \delta = 0.3, \mu = 1, h_x, h_y = 1, c_x = 500, c_y = 250)$, and h_x ranges from 1 to 10. The newsratio for both x and y are higher than 0.99 for x and y with all instances of h_x.

Figure 2.12 shows the result when newsratio varies. The parameter settings is $(\lambda_x = 0.8, \lambda_y = 0.4, \delta = 0.3, \mu = 1, h_x, h_y = 1, c_x = 5, c_y = 2.5)$, and h_x ranges from 0.1 to 5. The newsratio for x ranges from 0.98 to 0.5, decreasing with the value of h_x. Newsratio for y is 0.7 for all instances of h_x. Our observations are:

- s_x is non-increasing with h_x. This is intuitive since when h_x increases, it is more expensive to hold new product in inventory. The benefit of holding new product inventory for future demand over not holding enough incurring lost sales is decreasing with h_x. To keep the balance between holding cost vs. lost sales cost, s_x is non-increasing with h_x.

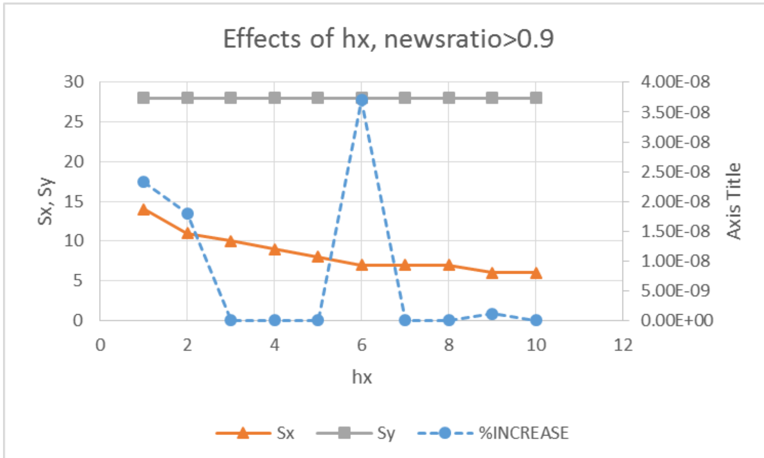

Figure 2.11: h_x vs. ΔJ^{H_3} when $newsratio_x > 0.98$

- s_y is not affected by h_x. Since cost parameters for used product, (h_y and c_y) don't change, conjecture: s_y depends on newsratio $\frac{c_y}{c_y+h_y}$ when it's large enough, e.g: >0.9

- The cost increase percentage ΔJ^{H_3}, is very low (at level of 10^{-8}) for all the values of h_x, even with the cost increase percentage jumps at $h_x = 6$. ΔJ^{H_3} is not monotone with h_x, and there is no systematic pattern to relate the trend of ΔJ^{H_3} vs. the value of h_x.

Effects of Lost Sales Cost c_x

We analyze the effects of c_x on the cost increase percentage (ΔJ^{H_3}). Parameter settings is ($\lambda_x = 0.8, \lambda_y = 0.4, \delta = 0.3, \mu = 1, h_x = 2, h_y = 1, c_x, c_y = 2.5$). c_x varies from 2.75 to 25, and newsratio varies from 0.58 to 0.93. It increases with the value of c_x. Figure 2.13 shows the result. Our observations are:

- s_x is non-decreasing with c_x. When c_x increases, it costs more to have a lost

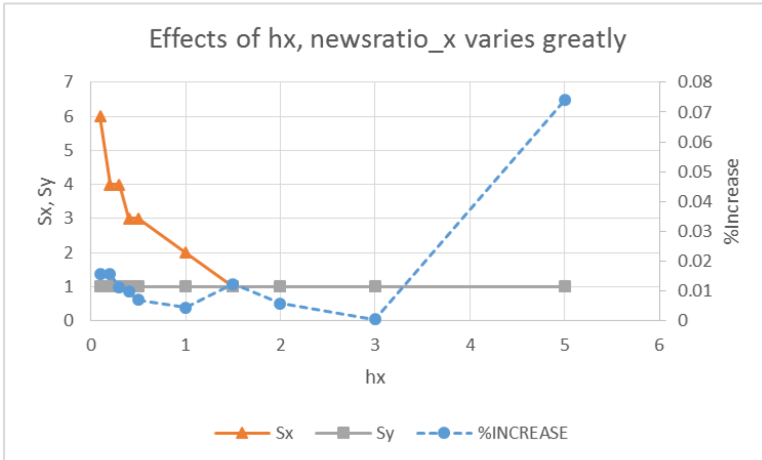

Figure 2.12: h_x vs. ΔJ^{H_3} when $newsratio_x$ varies from 0.98 to 0.5

sales for new product. s_x will not decrease to account for the increasing lost sales cost.

- s_y is non-decreasing with c_x. Consider the inventory changes for used product separately, both the replenishment and the demand process for the used product are independent from c_x. So the base stock s_y will not be affected. If consider inventory for used product as a production source for the new product, since increase of c_x will lead to higher s_x, then s_y will no decrease with c_x.

- The cost increase percentage ΔJ^{H_3} is not monotone with c_x.

Effect of Demand Rate λ_x

To analyze the effect of demand rate λ_x on the cost increase percentage ΔJ^{H_3}, we set the parameters to $(\lambda_x, \lambda_y = 0.4, \delta = 0.3, \mu = 1, h_x = 2, h_y = 1, c_x = 500, c_y = 250)$ and let λ_x to vary from 0.4 to 0.98. The newsratio for both x and y are 0.99. Figure 2.14 shows the result.

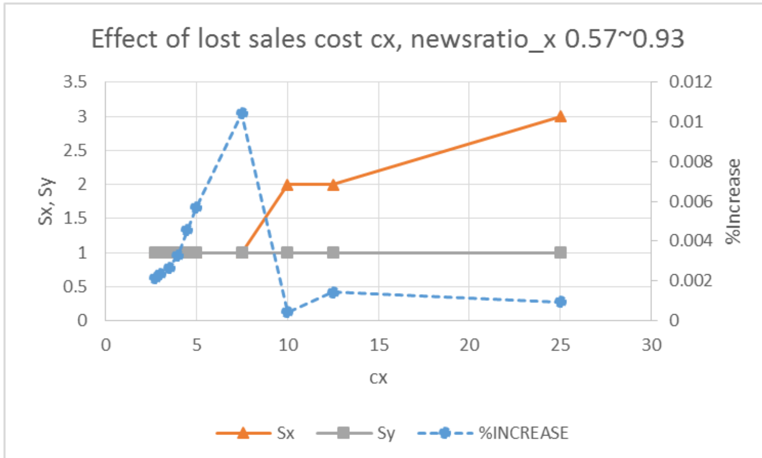

Figure 2.13: c_x vs. ΔJ^{H_3} when $newsratio_x$ varies 0.58 to 0.93

- s_x is non-decreasing with λ_x. This is intuitive since the higher the λ_x, the more demand for x will be. To satisfy the increasing demand, s_x will not decrease.

- s_y is non-decreasing with λ_x. Consider the inventory changes for used product separately, both the replenishment and the demand process for the used product are independent from c_x. So the base stock s_y will not be affected. If consider inventory for used product as a production source for the new product, since increase of c_x will lead to higher s_x, then s_y will no decrease with c_x.

- The cost increase percentage ΔJ^{H_3} is not monotone with c_x.

Summary of Effects of Different Parameters on H_3

Table 2.2 shows the summary of the results. For the parameters given, it is interesting to see that demand processes for new and used products are relatively independent. For example, when changing the parameters for new product, λ_x, h_x, c_x, the value of s_y will not be affected. When changing the parameters for used product, λ_y, h_y, c_y, the

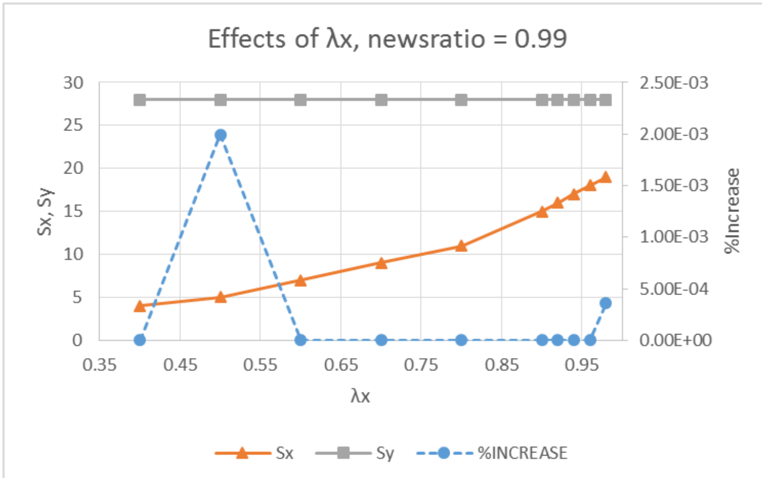

Figure 2.14: λ_x vs. ΔJ^{H_3} when *newsratio* for x and y are 0.99

value of s_x will not be affected. The potential reason can be the production process of the new product, no matter which option to choose, will not affect the lead time for the used product. So to balance the demand and replenish process for the used product, s_y will not be affected by the changed in parameters for new product.

2.5 Conclusion and Extensions

In this research, we consider a production-inventory system satisfying demand for both new and used products. Customer may return products after some use, and they are stocked in a return product inventory. Returned products served as an extra production resource for the new product, and it is the only resource to satisfy demand for used products. Unmet demands for both new and used products are lost. Standard holding costs and penalty costs apply.

Our major contribution is that we show the structure of the optimal policy of model under this set of assumptions: 1) Demands for both new and used products; 2) Unmet products are lost; 3) Product are returned as used ones, which can satisfy used

Table 2.2: Summary of Effects of Different Parameters on H_3

	s_x	s_y
δ	↑	↓
λ_x	↑	→
λ_y	→	↑
h_x	↓	→
c_x	↓	→
c_y	→	↑

demands directly or can be remanufactured into new products; and 4) Shared server can perform both producing and remanufacturing. The structure for the optimal production policy is of state dependent base-stock type where the base-stock level for the used product is non-increasing with the inventory of new product. We then propose three heuristics with one parameter or two-parameter. These parameters serve as fixed thresholds for new and used products below which the operation will stock the corresponding product. The two-parameter heuristics H_3 performs very well when return rate is no greater than used product demand rate. The cost increase percentage is no larger than 7% with an average increase of 0.2%. When return rate is greater than demand rate for used product, the cost increase percentage can be as high as 33% with an average of 4%.

There are many directions for future research. First, for the two-parameter heuristics, analytical solution will provide strong justification for its the good performance. Existing literature (such as [43]) that analyze the system performance provide different techniques (such as z-transformation) to solve the steady state probability, which can be explored to our model.

Second cost parameter based priority rules can be explored. [36] analyze a model with two classes of demands served by a shared serve. They propose heuristics with static priority rule based on backorder cost and production rate, and dynamic priority rule based on the previous static rule plus the current inventories for both products. These approaches shed light on how to fast choose the best options at any decision epoch without solving the dynamic program.

The model we analyze in this chapter is confined to a strict set of assumptions including unit demand that must be satisfied; unmet demands are backordered; there is one demand class for new product and one for used product. They enable us to characterize the structure of the optimal policy, but assume more than strict conditions for a real world system. In chapter 3, we relax several assumptions and analyze related systems that are more practical in real world setting. Although provide structural results are not available, numerical analysis suggest similar result such as state dependent type base stock policies to be optimal.

Chapter 3

RELATED SYSTEMS FOR PRODUCTION-INVENTORY SYSTEMS WITH RETURNS

We have focused on developing and analyzing a model with a set of assumptions that, we believe, captures the main features of our motivating context, server replacement in a data center. In order to maintain analytical tractability, however, the basic model assumed away some of the complexities that we observed during our visits to the data center. In this section, we report on these complexities and extend our model to address them. While we lose the ability to derive optimal policies for all of these extensions, the framework developed in the previous section allows us to numerically develop managerial insights. We relax the constraints from chapter 2 and analyze several related systems.

3.1 Backorder Situation

The first of these extensions acknowledges the fact that when there is not enough inventory to satisfy traffic managers' demand for new or used servers, these demands are not necessarily lost. The basic model had assumed that such unsatisfied demands, once lost, may come back in future but that connection was not explicitly captured in the model. Our actual observations suggest the traffic managers usually decide to wait till they get what they need. In the context of our model, this is backordering. We investigate if our results about the form of optimal policy in the basic model continue to hold with this extensions. We rewrite the dynamic formulations for the backorders scenario and show that the analysis of backorder model follows the same steps as shown in the basic model. This allows us to prove the form of the optimal policy for the backorder model.

If the unmet demands for both new and used products are backordered instead of lost, the optimal policy under the discounted total cost can be analyzed in a similar way. The state space is described by $(\tilde{X}(t), \tilde{Y}(t)) \in \mathbb{Z}^2$, $\mathbb{Z}$ is the set of the integers, and $\tilde{X}(t), \tilde{Y}(t)$, are the net inventories for new and used products respectively. Since returns serve as the only source for used products, it is required $\delta > \lambda_y$ for the system being stable. Otherwise, the backorder queue may grow to infinity and incur the corresponding backorder cost. Let b_x and b_y be the per unit time backorder cost for new and used products.

Summary of Parameters:

- λ_x : Demand rate for the new product
- λ_y: Demand rate for the old product
- δ : Return rate for the old product
- μ : (Re)manufacture rate for both used and new product
- h_x: Unit holding cost for new product (per time period)
- h_y: Unit holding cost for used product (per time period)
- b_x: Unit backorder cost for new product (per time period)
- b_y : Unit backorder cost for used product (per time period)
- α : Discount rate

Let $(\tilde{X}(0), \tilde{Y}(0)) = (x, y)$. The optimal cost function starting from time zero, with discount factor α satisfies

$v^*(x,y) = \tilde{T}v^*(x,y), \forall (x,y) \in \mathbb{Z}$

where $\tilde{T}$ is defined as

$\tilde{T}v(x,y) = \frac{1}{\alpha+\beta}[\tilde{h}(x,y) + \lambda_x \tilde{T}_1 v(x,y) + \lambda_y \tilde{T}_2 v(x,y) + \delta T_3 v(x,y) + \mu T_4 v(x,y)]$,

with $\tilde{h}(x,y) = h_x x^+ + h_y y^+ + b_x x^- + b_y y^-$. Operators T_3 and T_4 are defined in section 2.3.2, and $\tilde{T}_1$ and $\tilde{T}_2$ are defined as

$\tilde{T}_1 v(x,y) = v(x-1,y)$

$\tilde{T}_2 v(x,y) = v(x,y-1)$

Define a set of functions $\tilde{U_{BO}}$ that satisfies all the properties as U expect for $D_x v(x,y) \geq -c_x$ and $D_y v(x,y) \geq -c_y$.

Definition 3.1. Let $\tilde{U_{BO}}$ be a set of real-valued functions defined on $\mathbb{N}^2$. If $v \in \tilde{U_{BO}}$, then for all $x, y \geq 0$:

BO.1 $\quad D_{x,x}v(x,y) \geq D_{x,x-y}v(x,y), \forall x \geq 0, y > 0$

BO.2 $\quad D_{x,x-y}v(x,y) \geq 0, \forall x \geq 0, y > 0$

BO.3 $\quad D_{x,y}v(x,y-1) \geq 0, \forall y > 0$

BO.4 $\quad D_{y,y}v(x,y) \geq 0, \forall x, y \geq 0$

BO.5 $\quad D_{y,x-y}v(x,y) \leq 0, \forall x \geq 0, y > 0$

Similar to proof of lemma 2.1, it can be shown that $\tilde{T}$ preserves all the properties in $\tilde{U}$, and the optimal cost function v^*satisfies all the properties. Then the structure of the optimal policy is similar with backorders, as stated in theorem 3.1.

Theorem 3.1. *There exists a production on/off curve* $\tilde{B}(y) = \min[x \geq 0 | D_x v(x,y) > 0, D_{x-y}v(x,y) > 0]$ *such that it is optimal to be idle when inventory for new product* x *is no less than* $\tilde{B}(y)$*. There exists a production switching curve* $\tilde{C}(y) = \min[x \geq 0 | D_x v(x,y) - D_{x-y}v(x,y) > 0]$ *such that when producing for new product, it is optimal*

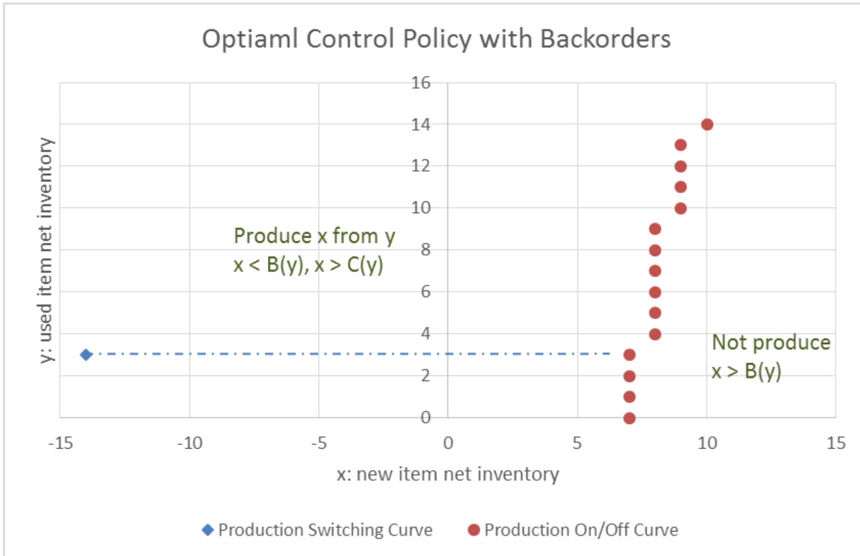

Figure 3.1: Optimal control policy with backorders

to produce from the used product when x is no less than $\tilde{C}(y)$, and to produce from raw material otherwise.

$\tilde{B}(y)$ and $\tilde{C}(y)$ satisfy these properties:

1. $\tilde{B}(y+1)+1 \geq \tilde{B}(y)$

2. $\tilde{C}(y) \geq \tilde{C}(y+1)$

Figure 3.1 shows an example of the optimal control policy with backorder with parameter settings ($h_x = 2, h_y = 1, b_x = 20, b_y = 10, \lambda_x = 0.85, \lambda_y = 0.2, \delta = 0.4, \mu = 1, \alpha = 0.05$). Now the blue switching curve expands to the region with negative values of x. The structure is very similar to the lost sales scenario.

3.2 Non-unitary Demand Situation

This extension reflects another observation we made about traffic managers' ordering behavior. Given that they can continuously monitor the server performance and user traffic, the traffic managers can order one unit of server at any time. This is what we assumed in our basic model. In reality, however, traffic managers' decision making process exhibits a batching behavior. They often order multiple units at the same time. This behavior appears to emanate from a natural tendency to focus on such ordering decisions occasionally rather than continuously. In the context of our model, we capture this behavior by allowing the demand arrivals to follow bulk Poisson processes. We show how to modify the basic model's dynamic formulation to incorporate bulk arrivals.

Assume the demand for new and used products follow Poisson process with non-unitary demand size d_i, $i = x, y$. The order size is bounded above by D_i, $i = x, y$. Let $\Pr\{d_i = d\} = p_i(d)$, $d = 1, ..., D_i$, and $i = x, y$. The demand can be partially fulfilled if the on hand inventory is less than the demand size, but must be fulfilled with all the on hand inventory. The non-satisfied proportion is lost inuring lost sales cost of c_i per unit product, $i = x, y$. We further assume returns happen in unit size. The optimal cost function under the discounted cost scenario satisfies

$v^*(x, y) = \tilde{T}v^*(x, y)$, $\forall(x, y) \in \mathbb{Z}$

where $\tilde{T}$ is defined as

$$\tilde{T}v(x,y) = \frac{1}{\alpha+\beta}[h(x,y) + \lambda_x \sum_{i=1}^{D_x} p_x(d_x)\tilde{T}_1 v(x,y) + \lambda_y \sum_{i=1}^{D_y} p_y(d_y)\tilde{T}_2 v(x,y) + \delta T_3 v(x,y) + \mu T_4 v(x,y)],$$

with $h(x, y) = h_x x + h_y y$. Operators T_3 and T_4 are defined in section 2.3.2, and $\tilde{T}_1$ and $\tilde{T}_2$ are defined as

$\tilde{T}_1 v(x, y) = v(x - q_x, y) + (d_x - q_x)c_x$

where $q_x = \min\{x, d_x\}$

$\tilde{T}_2 v(x, y) = v(x, y - q_y) + (d_y - q_y)c_y$

where $q_y = \min\{y, d_y\}$

Figure 3.2 shows an example of optimal control with bulk demands with demand for new product uniformly distributed from 1 to 10. Demand for used product is uniformly distributed from 1 to 5. Parameter settings are: $h_x = 2, h_y = 1, b_x =$

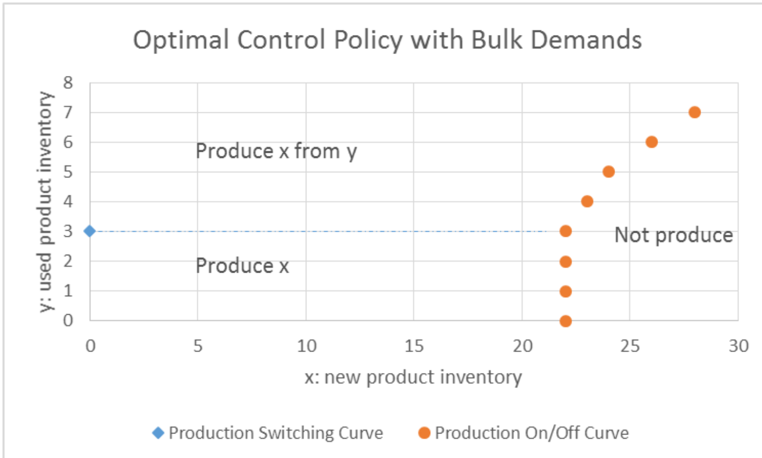

Figure 3.2: Optimal control with bulk demands for new and used products. Demand for new product: U[1, 10]. Demand for used product: U[1, 5]. Parameter settings: $h_x = 2, h_y = 1, b_x = 20, b_y = 10, \lambda_x = 0.85, \lambda_y = 0.3, \delta = 0.1, \mu = 1, \alpha = 0.05$

$20, b_y = 10, \lambda_x = 0.85, \lambda_y = 0.3, \delta = 0.1, \mu = 1, \alpha = 0.05.$

3.3 Multiple Demand Classes for New Product

In this section, we consider the model with multiple demand classes for new product. As for the data center case, initiatives and projects corresponding to different types for demand classes that differ from the cost structure and demand processes.

Assume there are n classes of customers demanding the new products. Class l has holding cost of h_l per unit per time period, and c_l per unit lost cost. The demand for the new product for class l incur in unit size and follows Poisson process with rate λ_l. $l \in \{1, 2, 3, \cdots, n\}$. The state can be described with $X(t) = (x_1(t), x_2(t), \cdots, x_N(t), y(t))$, where $x_l(t)$ is the on hand new product inventory for class l. The optimal cost function under the discounted cost scenario satisfies

$\tilde{v}^*(\mathbf{x}, y) = \tilde{T}v^*(\mathbf{x}, y), \forall (x, y) \in \mathbb{Z}$

where $\tilde{T}$ is defined as

$$\tilde{T}v(\mathbf{x}, y) = \frac{1}{\alpha+\beta}[h(\mathbf{x}, y) + \sum_{l=1}^{n}\lambda_l\tilde{T}_1 v_l(\mathbf{x}, y) + \lambda_y T_2 v(\mathbf{x}, y) + \delta T_3 v(\mathbf{x}, y) + \mu T_4 v(\mathbf{x}, y)],$$

with $h(\mathbf{x}, y) = \sum_{l=1}^{n} h_l x_l + h_y y$. Operators T_2, T_3 and T_4 are defined in section 2.3.2. $\tilde{T}_1$ is defined as $\tilde{T}_1 v_l(\mathbf{x}, y) = \min[v(\mathbf{x} - \mathbf{e_l}, y), v(\mathbf{x}, y) + c_l]$, where $\mathbf{e_l}$ is the l^{th} unit vector of dimension n.

$\tilde{T}_1$ corresponds to the optimal decision of satisfying demand from class l. It is possible to reject a demand from a class and incur a lost sales cost c_l, to save the inventory for the future demand from other classes.

3.4 Dependent Returns on Demands

For the final extension, we dig a little deeper into the how the traffic manager makes the ordering decisions. We have described earlier the main drivers of the traffic managers' decisions to demand a new or user server and to return one. These drivers are: changes in user-traffic, rapid growth in server technology leading to obsolescence in current server population, and the costs for traffic manager to keep a certain number of servers in the population. While we do not offer a detailed model for the traffic manager's decision making, we capture how these drivers influence the demand and return processes in our model. In this exercise, we discover and formalize the dependence of the return process on the number of servers in the population. As we had discussed earlier, traffic managers decide to return servers when the traffic volumes shift to lower levels and it is possible to meet required service levels with fewer servers. Thus fluctuations in traffic intensity drive the return process. Rather than capturing all details of traffic fluctuations and the heuristics behind traffic manager's decision to return a server, we focus on the main modeling issue in this setting, that the return of a server is dependent on the size of server population in a traffic manager's department.

We consider a model that considers the correlation between returns and demands, as shown in figure 3.3. A similar model is described in [89] that assumes all the

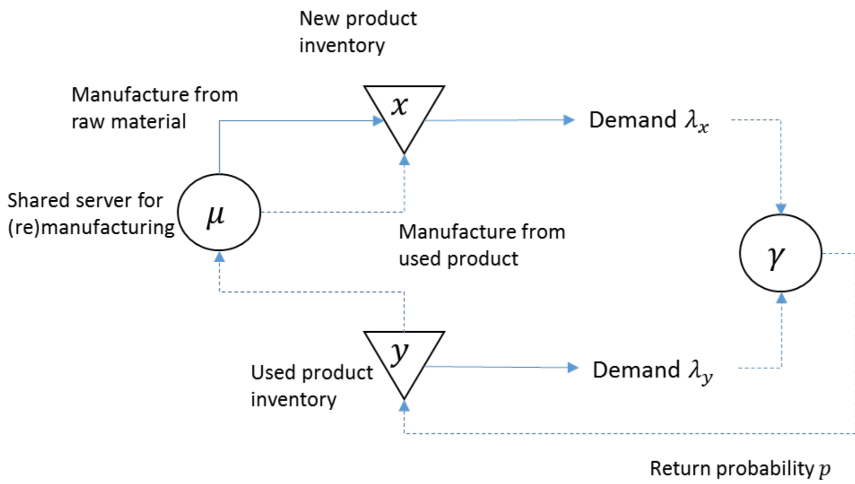

Figure 3.3: Dependent returns on demands

products are returned as new, and there is no demand for used products.

Comparing to the base model in section 2.3, this model assumes a satisfied demand will lead to a return with probability p after a stochastic time. The time before the return occurs is exponentially distributed with rate γ. This assumption is tested by [16] on real data. They conclude with a statistical test that the assumption is rejected for some products but not all.

To capture the dynamics of the returns, an extra variable is needed to record the number of satisfied demand that may be returned. The state of the system is described by $(X(t), Y(t), N(t))$ where $X(t)$ is the on-hand inventory of the new product, $Y(t)$ is the on-hand inventory for the used product, and $N(t)$ is the number of products (including new and used) on the market.

Following the approach in [89], we categorize the system by whether $N(t)$ is observable or not.

- When $N(t)$ is not observable, the production and inventory manager only knows the returns will be a proportion p of the total demand.

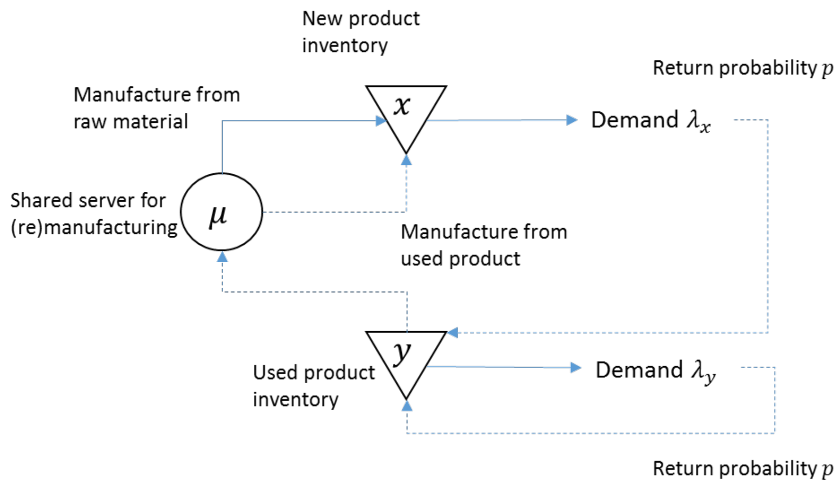

Figure 3.4: Dependent returns on demands with zero lead-time

- When $N(t)$ is observable, the production and inventory manager knows exactly how many products are out in the market that may be returned. For data center operations, advanced inventory monitoring technologies, such as RFID, enable the real time tracking of servers in use, which is $N(t)$.

3.4.1 Unobservable Market Population with Zero Return Lead-Time

We concentrate the analysis on the case where $N(t)$ is unobservable and return lead-time is zero, or $1/\gamma = 0$. This limit case simplifies the analysis by neglecting $N(t)$, and it also provides a good estimates of the case with short return period for the customers. Returned product returns will be stored in the used product inventory. Figure 3.4 shows the system flow.

Optimal Value Function Let $v^*(x, y)$ be the optimal cost function with starting state (x, y). The optimal cost function v^* satisfies the following Bellman's equation

$$v^*(x, y) = Tv^*(x, y), \forall x, y \geq 0$$

Operator T is a contraction mapping defined as

$T_B v(x,y) = \frac{1}{\alpha+\beta}[h(x,y) + \lambda_x T_1 v(x,y) + \lambda_y T_2 v(x,y) + \mu T_4 v(x,y)]$

with $h(x,y) = h_x x + h_y y$ and operators T_i, $i = 1,2,3,4$ defined as follows,

$$T_1 v(x,y) = \begin{cases} v(x,y) + c_x & \text{if } x = 0 \\ (1-p)v(x-1,y) + pv(x-1,y+1) & \text{if } x > 0 \end{cases}$$

$$T_2 v(x,y) = \begin{cases} v(x,y) + c_y & \text{if } y = 0 \\ (1-p)v(x,y-1) + pv(x,y) & \text{if } y > 0 \end{cases}$$

$$T_4 v(x,y) = \begin{cases} \min[v(x,y), v(x+1,y)] & \text{if } y = 0 \\ \min[v(x,y), v(x+1,y), v(x+1,y-1)] & \text{if } y > 0 \end{cases}$$

Structure of the Optimal Value Function

Definition. Let U_C be a set of real-valued functions in $\mathbb{N}^2$. If $v \in U_C$, then for all $x, y \geq 0$:

C.1 $pD_y v(x,y) \leq D_x v(x,y) - c_x$, $\forall x, y \geq 0$

C.2 $*D_{x,x} v(x,y) \geq 0$, $\forall x \geq 0, y > 0$

C.3 $*D_{y,y} v(x,y) \geq 0$, $\forall x, y \geq 0$

C.4 $D_y v(x,y) \geq -\dfrac{c_y}{1-p}$, $\forall x, y \geq 0$

Conjecture. *If a value function $v \in U_C$, then $T_C v \in U_C$. Moreover, the optimal value function $v^* \in U_C$.*

There does not exist intuitive simple characterization of the structure of the optimal value function. However, it is numerically shown that the optimal policy is state dependent base stock type on both x and y. Comparing to the base model with independent returns, this model does not satisfy properties similar to A.3 ($D_{x,x-y} v$) and

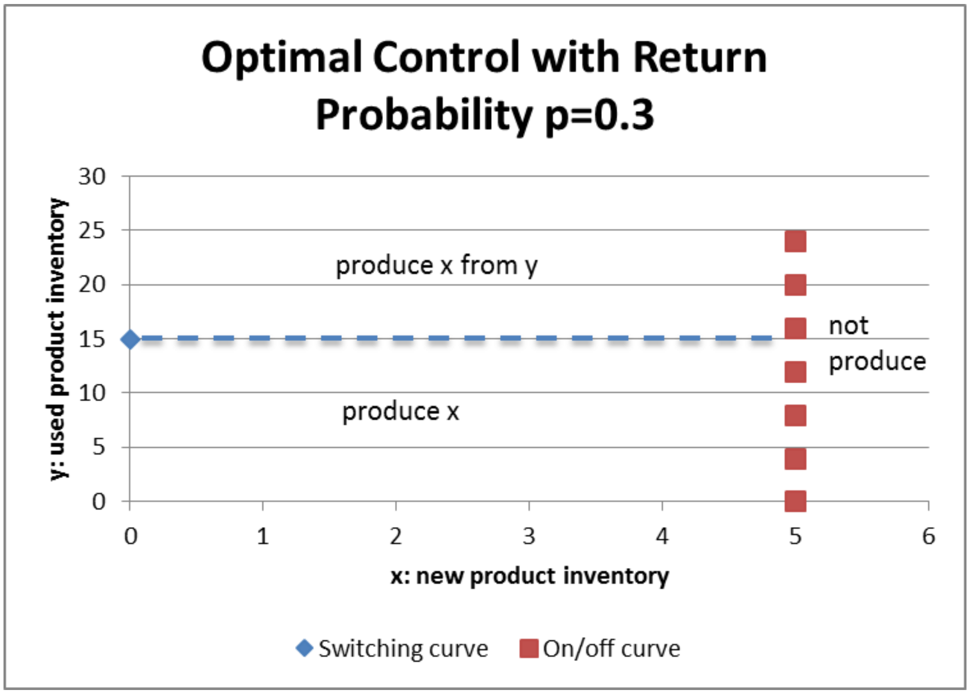

Figure 3.5: The optimal control policy with return probability $p = 0.3$ and zero return leadtime

A.4 ($D_{x,y}v$), without which the order of $D_xT_4v(x,y)$,$D_yT_4v(x,y)$, and $D_{x-y}T_4v(x,y)$ cannot be determined. Then the combinatorial nature of the difference operators applied on $T_4v(x,y)$ cannot go through the proof for all the properties. An example of the optimal control policy is shown in figure 3.5. The parameter settings are $(\mu = 1, \lambda_x = 0.85, \lambda_y = 0.6, p = 0.3, h_x = 2, h_y = 1, c_x = 200, c_y = 100, \alpha = 0.05)$

3.5 Conclusions

In this chapter, we have considered several extensions for the base model in chapter 2, a production-inventory system with returns. The extensions are motivated by the observations from real life data center operations. For example, the unmet demand for servers will usually be backordered since they are project based. Or the demand are usually requested in bulk to satisfy a reasonable customer size. Although we are not able to characterize the structure of the optimal policy for the extension models, numerically it is shown the optimal policies is of state dependent base stock type.

The assumptions of exponential production time, order interarrival time and return lead time reduce the complexity of the problems. The model can be extended by substituting the exponential distribution to phase type distribution which approximate other distributions more flexibly, and still retains the Markovian property.

Another potential direction for research is to combine the models into an integraded system, which is more practical. In a complex system which includes features such as backorder, lost sales, substitution, bulk order or dependent returns, it is not realistic to prove the structure of the optimal policy. Numerical example indicates the power and base stock policy, and efforts can be put into finding heuristics that are efficient and simple to implement.

Chapter 4

DYNAMIC CONTROL FOR A PRODUCTION-INVENTORY SYSTEM WITH DEMAND CLASSES WITH DIFFERENT VARIABILITY

4.1 Introduction

We develop and analyze a model of a continuous-time production-inventory system with a shared server serving two demands. Unsatisfied demands are backordered and standard holding and penalty costs apply at each inventory location. The objective is to determine the optimal scheduling policy that minimizes the long term total cost, and to develop insights into managing variability in such supply chains.

The distinguishing feature of our model is that the two demand classes differ in their variability characteristics. It is motivated by the observations of real-life contexts where a production-inventory system with shared server serves two classes of demands with different variability. Very briefly, practical examples similar to this situation include (1) inherent differences in the predictability of the demand as in [20], (2) some customers' spiky purchasing pattern influenced by low prices during promotions as in [41], (3) mixing emergency and scheduled demands as in [59], and (4) the nature of two different markets as in [25].

Since Poisson distributions are completely characterized by a single parameter, modeling both demand classes with Poisson processes does not allow us to have arrivals with, for example, equal means but different variances. To capture the differences in demand variability, we model the inter-arrival times for one demand class as following hyperexponential distribution. This allows us to keep the model tractable and understand the impact of variability differences on the structure of the optimal policy. We partially characterize the structure of the optimal policy and prove properties of optimal policy parameters. Then we consider the cases where partial or full information about the demand arrival process is known. Our results allow us to shed some light on implications for managers in such circumstances.

The rest of the chapter is organized as follows. Section 4.2 provides a review of related literature. Section 4.3 describes the assumptions and basic model setting. Section 4.4 analyze the model with full information and provide analytical results. Section 4.5 extend the model to only knowing partial information. Section 4.6 provides our conclusions and possible extensions.

4.2 Literature Review

There has been a growing interest in modeling of production-inventory systems and in using the tools of continuous-time Markov decision process (MDP) to characterize and compute the optimal policy for such systems. A large amount of literature utilize make-to-stock queues framework to analyze system performance under given policies. [90] first considered dynamic scheduling in a make-to-stock system in the context of two identical products with the same Poisson demand rates and equal exponential production times. They use a simple base-stock policy for each product, and results in a corresponding production order. They also compare the performance of FCFS rule with a longest queue rule that uses the current information about inventory levels and then produces the class that needs more units to achieve its base-stock level. [91] extends this model to multiple identical products and considered general production times. [80] considers a multiclass $M/M/1$ make-to-stock queue with non-identical products (demand rates, costs, and production times are different) and proposed and evaluated several heuristic scheduling policies. The same system was then considered by [65], who proposed a general approach to construct a policy and suggested a very effective heuristic scheduling scheme.

Literature related to optimal control in continuous-time production-inventory systems have concentrate on proving the structure of the optimal policy. [36] is one of the earliest to show the structure of the optimal policy in the context of two Poisson demand classes with equal production times. Unmet demands are backordered. He model the system with MDP and show the structure of the optimal policy is of state dependent base stock type. [35] analyze a similar model with multiple classes of demands with lost sales. They show the similar structure from [36] extend to this model. Rationing between the multiple classes is based on order of cost parameters. Since then the technique has been applied to models that include a variety of features, such as advance demand information, admission control, stock rationing and etc. [9] analyze the admission control in an assembly system with multiple stages and multiple demand classes. The demands occur according to Poisson arrival processes. Products are produced in variable batch sizes, one batch at a time, with exponentially distributed batch production times. They show the structure of the optimal production policy and inventory allocation policy. [31] consider a make-to-stock supplier serving multiple classes demands with a shared server. Demands occur according to Poisson process. Imperfect advance demand information is provided, but announced demand may be canceled. They model the system with MDP and show the structure of the optimal production and inventory allocation decisions. In each case, the authors show the optimal production/allocation policy for each product is of state-dependent base-stock type, with certain monotonicity properties. For a detailed review on control theory application to the production-inventory system, readers are referred to [63].

In the literature related to variability in supply chain, research have been conducted to target different sources of variability such as processing time, demand arrival process, and demand size variations. A few papers develop multiclass queuing inventory models where arrivals or service can be general and thus can accommodate differences in variability. [61] allows general arrival processes for different classes (MTO and MTS products) in her queuing-inventory model, and then focuses on evaluating the performance measures. [65] also allow for general arrival and service processes for different classes in their MTS queuing model, and develop a dynamic scheduling policy at the manufacturer's queue. These papers have general models also develop heuristics and approximations for performance measures. As for demand size variations, [21] study a production-inventory system with multiple classes of demands. Demand sizes are non-unitary and follow a discrete distribution. With the total production time following k_0-Erlang distribution, and each demand class following Erlang distribution, the system is formulated with a MDP and structure of the optimal policy is characterized.

The paper that comes closest to our work in focusing on differences in variability characteristics of two demand classes in the context of production-inventory system is [43]. That paper considers the scheduling problem in a make-to-stock queue with two demand classes, one class experience Poisson arrivals and other experiences hyperexponential renewal arrivals. Exact analysis is provided for the case when the demand class with higher variability is given non-preemptive priority. They also proposed a heuristics based on existing dynamic scheduling rule. We analyze a similar system but focus on determining the optimal policy and understanding the impact of knowing partial information about the structure of demand arrivals.

4.3 Model

Consider a production facility that produces and stocks two products, denoted as class H and class M. The stochastic demand from any product is satisfied from the on-hand inventory. Unmet demands are backordered. There are linear costs for holding inventory and delaying orders. At any time, the production manager can make one of the production decisions: not to produce, produce class H, produce class M. The objective is to find the production policy that minimizes the discounted expected total cost over infinite time horizon.

The production time for both classes is exponentially distributed with rate μ. The changeover time for production is negligible, and the production process is preemptive. The demand process for class M follows an independent Poisson process with rate λ_m. The demand process for class H follows a hypoexponential renewal arrival process, i.e., the distribution of the inter-arrival time is hypoexponentially distributed. Hyperexponential distribution is a mixture of exponential distributions weighted by a set of probabilities. The density function of a hyperexponential distributed random variable H is

$$f_H(h) = \sum_{j=1}^{n} f_{Y_j}(h) k_j$$

where Y_i is exponentially distributed with rate r_j, and $k_j \in [0, 1]$ is the probability of H taking on the exponential distribution with rate r_j. The hyperexponential distribution is a common choice in queuing theory to give a high variability [85]. Its coefficient of variation (standard deviation/mean) c_v is at least than 1. Compared to exponential distribution has a coefficient of variation of 1, the parameter setting in hyperexponential distribution can be adjusted to reflect a wide range of variations. Besides, the mixture of exponential distribution nature provides a nice structure that can be analyzed as a time invariant system. Existing literature [43] utilizes this structure to propose efficient heuristics.

When $n = 2$, there are two exponential distributions involved, and the hyperexponential distribution is referred to as H_2. Let $\bar{\lambda}_h$ be the intensity of the hyperexponential renewal process, parameters for H_2 can be determined as

$$k_1 = 0.5\left(1 + \sqrt{\frac{c_v^2 - 1}{c_v^2 + 1}}\right)$$

$$k_2 = 1 - k_1$$

$$r_1 = 2k_1\bar{\lambda}_h$$

$$r_2 = 2k_2\bar{\lambda}_h$$

The arrival process of H_2 can be modeled as an *Interrupted Poisson Process* (IPP) with two phases *on* and *off*. In phase *on*, the demand arrives following a Poisson process with rate λ_h; in phase *off*, no demand arrives. The two phases transit according to an underlying Markov process. The duration of phase being *on* and *off* are exponentially distributed with rate η_1 and η_0. The parameters of the IPP arrival process with respect to the hyperexponential distribution are [23]:

$$\lambda_h = k_1 r_1$$
$$\eta_1 = \frac{k_1 k_2 (r_1 - r_2)^2}{\lambda_h} \qquad \text{phase } on$$
$$\eta_0 = \frac{r_1 r_2}{\lambda_h} \qquad \text{phase } off$$

We continue our analysis based on modeling arrival process for class H as an IPP. It enables us to model the system with a continuous time Markov decision process and simplify the analysis. The same system has been studied by [43]. It provides an exact analysis when class H has non-preemptive priority, and them emphasizes on analyzing heuristics. We emphasize on characterizing the structure of the optimal policy. Our model is similar to [36], which is one of the earliest papers that prove the structure of the optimal policy in multiclass make-to-stock queuing systems. In [36], there are two demand classes with both demand arrivals following Poisson processes. Our model assumes two demand arrival processes, with one following Poisson process and the other following hyperexponential process. The hyperexponential process or the equivalent IPP brings great difficulty in analyzing the optimal policy. To our knowledge, we are the first paper to analyze the optimal policy of a system with such level of complexity.

In section 4.4, we consider the case where the phase information is fully observable by the decision maker, and show the partial structure of the optimal policy. Then in section 4.5, we consider the case where the decision maker only knows the probability of the phase information. By comparing these two systems, the value of knowing phase information is evaluated.

4.4 Knowing Phase Information

Considering the case where the phase information of the IPP is known, e.g, the production and inventory manager observes whether the current demand from class H is in phase *on* or *off*. The state of the system at time t can be described as

$X(t) = (X_h(t), X_m(t), I(t))$ where $X_j(t) \in \mathbb{Z}$ is the net inventory of class j at time t, $j \in \{h, m\}$. $I(t) = 1$ denotes demand H being in phase *on* and $I(y) = 0$ denotes demand H being in phase *off*. The inventory cost rate function at time t can be denoted as $c(X(t)) = \sum_{j=h,m} h_j X_j(t)^+ + b_j X_j(t)^-$. We use $x = (x_h, x_m, i)$ to denote the initial inventory levels $(X_h(0), X_m(0), i)$, and $\alpha \in (0, 1)$ for the discount rate. The goal is to find the optimal control policy u to minimize the expected total discounted costs $v^u(x)$ with starting state $x = (x_h, x_m, i)$.

$$v^u(x) = E_x^u[\int_0^\infty e^{-\alpha t} c(X(t))dt] \tag{4.1}$$

Control policy $u(t) \in \{0, 1, 2\}$ where

$$u(t) = \begin{cases} 0 & \text{Stay idle} \\ 1 & \text{Produce product for class } H \\ 2 & \text{Produce product for class } M \end{cases}$$

Let $v^*(x_h, x_m, i)$ be the optimal expected discounted inventory cost over an infinite time horizon with initial inventory (x_h, x_m, i). The optimal cost function satisfy $v^*(x_h, x_m, i) = \min_u v^u(x_h, x_m, i)$, $i \in \{0, 1\}$.

Summary of Parameters

- λ_h: Demand rate for class H when phase is *on*
- λ_m: Demand rate for class M
- μ: Production rate for both H and M
- $\frac{1}{\eta_0}$: Average time of being in phase *off* in the IPP process
- $\frac{1}{\eta_1}$: Average time of being in phase *on* in the IPP process
- h_h: Unit holding cost for class H per time period

- h_m: Unit holding cost for class M per time period
- b_h: Unit backorder cost for class H per time period
- b_m : Unit backorder cost for class M per time period
- α : Discount rate

Uniformization and Transition Probabilities

The decision process can be modeled as a continuous Markov decision process, since all the transition time between any two events is exponentially distributed. However, the different transition rates between events complicate the problem. To solve that, we uniformize the inventory process by defining the uniformization rate $\gamma = \mu + \lambda_h + \lambda_m + \eta_0 + \eta_1$ [53, 67]. The system is transformed into a discrete Markov decision process with all transitions happen at the rate γ, and actual event happens with a corresponding probability as illustrated below:

- Transition probability: start from state $(x_h, x_m, 1)$
  - If action is *notProduce*, then transit to
    - $(x_h, x_m, 0)$ with probability $\frac{\eta_1}{\gamma}$; $(x_h - 1, x_m, 1)$ with probability $\frac{\lambda_h}{\gamma}$; $(x_h, x_m - 1, 1)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h, x_m, 1)$ with probability $\frac{\eta_0+\mu}{\gamma}$;
  - If action is *produceH*, then transit to
    - $(x_h, x_m, 0)$ with probability $\frac{\eta_1}{\gamma}$; $(x_h - 1, x_m, 1)$ with probability $\frac{\lambda_h}{\gamma}$; $(x_h, x_m - 1, 1)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h + 1, x_m, 1)$ with probability $\frac{\mu}{\gamma}$; $(x_h, x_m, 1)$ with probability $\frac{\eta_0}{\gamma}$;
  - If action is *produceM*, then transit to

  * $(x_h, x_m, 0)$ with probability $\frac{\eta_1}{\gamma}$; $(x_h - 1, x_m, 1)$ with probability $\frac{\lambda_h}{\gamma}$; $(x_h, x_m - 1, 1)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h, x_m + 1, 1)$ with probability $\frac{\mu}{\gamma}$; $(x_h, x_m, 1)$ with probability $\frac{\eta_0}{\gamma}$.

- Transition probability: start from state $(x_h, x_m, 0)$
  - If action is *notProduce*, then transit to
    * $(x_h, x_m, 1)$ with probability $\frac{\eta_0}{\gamma}$; $(x_h, x_m - 1, 0)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h, x_m, 0)$ with probability $\frac{\eta_1+\lambda_h+\mu}{\gamma}$;
  - if action is *produceH*, then transit to
    * $(x_h, x_m, 1)$ with probability $\frac{\eta_0}{\gamma}$; $(x_h, x_m - 1, 0)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h + 1, x_m, 0)$ with probability $\frac{\mu}{\gamma}$; $(x_h, x_m, 0)$ with $\frac{\eta_1+\lambda_h}{\gamma}$;
  - if action is *produceM*, then transit to
    * $(x_h, x_m, 1)$ with probability $\frac{\eta_0}{\gamma}$; $(x_h, x_m - 1, 0)$ with probability $\frac{\lambda_m}{\gamma}$; $(x_h, x_m + 1, 0)$ with probability $\frac{\mu}{\gamma}$; $(x_h, x_m, 0)$ with probability $\frac{\eta_1+\lambda_h}{\gamma}$.

Notice that the uniformization creates pseudo transitions into the state itself. The intuition is that time span is sliced into pieces with average length of $1/\gamma$, starting from$(x_h, x_m, 0)$ has different set of next potential states than starting from $(x_h, x_m, 1)$. For example, there is transition from $(x_h, x_m, 1)$ to $(x_h - 1, x_m, 1)$ but not from $(x_h, x_m, 0)$ to $(x_h - 1, x_m, 0)$. This difference creates the pseudo transitions into the current state. As compared to model in the reverse supply chain (section 2.3.2 equation 2.3), no pseudo transition is created. More about uniformization can be found in section 2.3.2.

The optimal cost functions $v^*(x_h, x_m, i), i = (0, 1)$, $x_h, x_m \in \mathbb{Z}$ satisfy the following optimality equations:

$$\begin{aligned} v^*(x_h, x_m, 0) \quad = \quad & \frac{1}{\alpha+\gamma}[c(x_h, x_m) + \eta_0 v^*(x_h, x_m, 1) + \lambda_m v^*(x_h, x_m - 1, 0) \\ & +(\lambda_h + \eta_1 + \mu)v^*(x_h, x_m, 0) + \mu T_0 v^*(x_h, x_m, 0)] \end{aligned} \tag{4.2}$$

$$v^*(x_h, x_m, 1) = \frac{1}{\alpha+\gamma}[c(x_h, x_m) + \eta_1 v^*(x_h, x_m, 0) + \lambda_m v^*(x_h, x_m - 1, 1) + \lambda_h v^*(x_h - 1, x_m, 1) + (\eta_0 + \mu) v^*(x_h, x_m, 1) + \mu T_1 v^*(x_h, x_m, 1)] \quad (4.3)$$

where the operators T_i, $i = 0, 1$ are defined as:

$T_i v(x_h, x_m, i) = min\{0, D_h v(x_h, x_m, i), D_m v(x_h, x_m, i)\}$

Operator T_i corresponding to the decision of whether or not to produce, and which product to produce in phase i.

Structure of the Optimal Policy under Discounted Total Cost

To characterize the structure of the optimal policy, define these finite difference operators:

- First difference operators
  - $D_h v(x_h, x_m, i) = v(x_h + 1, x_m, i) - v(x_h, x_m, i)$
  - $D_m v(x_h, x_m, i) = v(x_h, x_m + 1, i) - v(x_h, x_m, i) i = 0$
- Second difference operators:
  - $D_{h,h} v(x_h, x_m, i) = D_h D_h v(x_h, x_m, i) = D_h v(x_h + 1, x_m, i) - D_h v(x_h, x_m, i)$
  - $D_{m,m} v(x_h, x_m, i) = D_m D_m v(x_h, x_m, i) = D_m v(x_h, x_m + 1, i) - D_m v(x_h, x_m, i)$
  - $D_{h,m} v(x_h, x_m, i) = D_h D_m v(x_h, x_m, i) = D_h v(x_h, x_m + 1, i) - D_h v(x_h, x_m, i)$

The order of applying the operators doesn't affect the result, e.g: $D_{h,m} v = D_{m,h} v$ and etc.

Properties of Switching Curves

In this section, we examine the structure of the optimal policy.

Definition 4.1. Let V be the set of functions defined on $\mathbb{Z}^2 \times \{0,1\}$, where $\mathbb{Z}$ is the set of integers. If $v \in V$, then for all $(x_h, x_m, i) \in \mathbb{Z}^2 \times \{0,1\}$:

D.1 *Supermodularity:* $(D_{h,m}v(x_h, x_m, i) \geq 0)$

Intuition: $D_h v(x_h, x_m, i)$ is increasing in x_m, or equivalently, $D_m v(x_h, x_m, i)$ is increasing in x_h. The property is equivalent to $v(x_h+1, x_m+1, i) + v(x_h, x_m, i) \geq v(x_h+1, x_m, i) + v(x_h, x_m+1, i)$.

D.2 *Diagonal dominance on H:* $D_{h,h}v(x_h, x_m, i) \geq D_{h,m}v(x_h, x_m, i)$

Intuition: $D_m v(x_h, x_m, i) - D_h v(x_h, x_m, i)$ is decreasing in x_h. The property is equivalent to $v(x_h+2, x_m, i) + v(x_h, x_m+1, i) \geq v(x_h+1, x_m, i) + v(x_h+1, x_m+1, i)$.

D.3 *Diagonal dominance on M:*$D_{m,m}v(x_h, x_m, i) \geq D_{h,m}v(x_h, x_m, i)$

Intuition: $D_m v(x_h, x_m, i) - D_h v(x_h, x_m, i)$ is increasing in x_m. The property is equivalent to $v(x_h, x_m+2, i) + v(x_h+1, x_m, i) \geq v(x_h, x_m+1, i) + v(x_h+1, x_m+1, i)$.

D.4 $(D_m - D_h)v(x_h, x_m, 0) \leq (D_m - D_h)v(x_h, x_m, 1)$.

Intuition: when it's *off*, the difference in producing one extra unit of M relative to H is not higher than when it's *on*. It is equivalent to $D_m v(x_h, x_m, 0) - D_h v(x_h, x_m, 0) \leq D_m v(x_h, x_m, 1) - D_h v(x_h, x_m, 1)$, meaning $v(x_h, x_m+1, 1) + v(x_h+1, x_m, 0) \geq v(x_h, x_m+1, 0) + v(x_h, x_m, 1)$.

D.5 $D_h v(x_h, x_m, 1) \leq D_h v(x_h, x_m, 0)$

Intuition: the marginal cost of producing one extra unit of H is less when phase is *on* than when it is *off*.

Notice that for a function $v \in V$, supermodularity D.1 and diagonal dominance D.2 implies v is convex in x_h and x_m.

Theorem 4.1. *If $v \in V$, then $T_i v \in V$. Furthermore, the optimal cost function $v^*(x_h, x_m, i) \in V$. $i \in \{0, 1\}$.*

Proof. Details of the proof is in appendix B.1. □

Theorem 4.2. *Define $B(x_h, i) = min\{x_m | D_h v^*(x_h, x_m, i) > 0, D_m v^*(x_h, x_m, i) > 0\}$ and $C(x_h, i) = min\{x_m | D_m v^*(x_h, x_m, i) - D_h v^*(x_h, x_m, i) > 0\}$. Given the inventory and phase information (x_h, x_m, i), it is optimal to be idle if the inventory for class M is no less than $B(x_h, i)$. If not this case, it is optimal to produce class H when inventory for class M is no less than $C(x_h, i)$, and to produce class M otherwise.*

$B(x_h, i)$ and $C(x_h, i)$ satisfy these properties:

1. *$B(x_h, i)$ is non-increasing in x_h, for $i = 0, 1$.*

2. *$C(x_h, i)$ is non-decreasing in x_h, for $i = 0, 1$.*

3. *$C(x_h, 1) \leq C(x_h, 0)$.*

Proof. Since the optimal value function $v^* \in V$, so v^* satisfies the properties with $B(x_h, i)$ and $C(x_h, i)$.

Part 1: At $(x_h, x_m = B(x_h, i), i)$, $i = 0, 1$, by definition $D_h v(x_h, x_m = B(x_h, i), i) > 0$ and $D_m v(x_h, x_m = B(x_h, i), i) > 0$. By property $D_{h,h} v(x_h, x_m, i) \geq 0$, $D_h v(x_h + 1, x_m = B(x_h, i), i) \geq D_h v(x_h, x_m = B(x_h, i), i) > 0$. In the same manner, by property $D_{h,m} v(x_h, x_m, i) \geq 0$, $D_m v(x_h + 1, x_m = B(x_h, i), i) \geq D_m v(x_h, x_m = B(x_h, i), i) > 0$. So $B(x_h, i) \geq B(x_h + 1, i)$ by definition of $B(x_h + 1, i)$.

Part 2: At $(x_h + 1, x_m = C(x_h + 1, i), i)$, $i = 0, 1$, by definition $D_m v(x_h + 1, x_m = C(x_h + 1, i), i) - D_h v(x_h + 1, x_m = C(x_h + 1, i), i) > 0$. By property $D_{h,h} v(x_h, x_m, i) \geq D_{h,m} v(x_h, x_m, i)$, we get $D_m v(x_h, x_m = C(x_h + 1, i), i) - D_h v(x_h, x_m = C(x_h + 1, i), i) \geq$

$D_m v(x_h + 1, x_m = C(x_h + 1, i), i) - D_h v(x_h + 1, x_m = C(x_h + 1, i), i) > 0$. So $C(x_h, i) \leq C(x_h + 1, i)$ by definition of $C(x_h, i)$.

Part 3: By definition of $C(x_h, i)$, $(D_m - D_h)v(x_h, C(x_h, 1) - 1, 1) < 0$. By property $(D_m - D_h)v(x_h, x_m, 0) < (D_m - D_h)v(x_h, x_m, 1)$ we have $(D_m - D_h)v(x_h, C(x_h, 1) - 1, 0) \leq (D_m - D_h)v(x_h, C(x_h, 1) - 1_m, 1) < 0$. Since $C(x_h, 0) - 1$ is the largest x_m such that $(D_m - D_h)v(x_h, C(x_h, 0) - 1, 0) < 0$, it must be $C(x_h, 1) \leq C(x_h, 0)$. □

Proposition 4.1. *Given (x_h, x_m), if at phase* on *it is optimal to produce* M *then it is also optimal to produce* M *at phase off.*

Proof. At state $(x_h, x_m, i = 1)$, if it is optimal to produce M means $x_m < B(x_h, i = 1)$ and $x_m < C(x_h, i = 1)$. With property $C(x_h, 1) \geq C(x_h, 0)$, we know $x_m < C(x_h, i = 0)$. □

4.4.1 Numerical Analysis for Knowing Phase Information Model

First, we provide a numerical examples to illustrate the structure of the optimal policy when phase information is known. The optimal policy is calculated with value iteration, details in appendix C.1. The state space is truncated to a finite space with $x_h, x_m \in [\underline{n}, \bar{n}]$. We gradually change $\underline{n}$ and $\bar{n}$ until the optimal cost function is not sensitive to the changes in the boundaries. In this case, $[\underline{n} = -30, \bar{n} = 30]$.

Approximation of the Value Functions

Since the state space is truncated to a finite space, the value functions at or beyond the boundary states need to be approximated. We follow the approach in [36, section 4.4], which analyze a similar model with two Poisson demand processes. [36] shows that "when the inventory holding and backorder costs are linear, the value function under all the policies in the experiments are asymptotically linear". Specifically, when the state variable $x_h, x_m \in [\underline{n}, \bar{n}]$, for $i = 0, 1$

$$v(\underline{n}, x_m, i) = v(\underline{n} + 1, x_m, i) + \frac{b_h}{\alpha}$$

$$v(x_h, \underline{n}, i) = v(x_h, \underline{n}+1, i) + \frac{b_m}{\alpha}$$
$$v(\bar{n}, x_m, i) = v(\bar{n}-1, x_m, i) + \frac{h_h}{\alpha}$$
$$v(x_h, \bar{n}, i) = v(x_h, \bar{n}-1, i) + \frac{h_m}{\alpha}$$

An Example of the Optimal Policy

Consider a production-inventory system producing two classes of products with the following pa maters: ($\mu = 1, \lambda_h = 2, \lambda_m = 0.8, \eta_0 = 0.77, \eta_1 = 0.23, h_h = 3, h_m = 3, b_h = 80, b_m = 100, \alpha = 0.05$). Figure 4.1 shows the optimal policy for this problem.

The state space is divided into several regions. The rectangular points are for the switching curves $C(x_h, i)$, and the crosses are for the production on/off curves $B(x_h, i)$. Blue points and crosses are for demand phase of H being *on*, and red points and crosses for demand phase of H being *off*. B is non-increasing and C is non-decreasing, as proved in theorem 4.2.

Notice that when $x_h < 0$, the optimal policy for both phase *on* and *off* is to produce class M, the switching curves $C(x_h, i)$ never cross x_h axis. This is true for this example since $b_h = 80$ is less than $b_m = 100$. [36] proves the static priority rule when $x_h < 0$, $b_h < b_m$ and production rate for both classes are the same. The intuition is when x_h is backordered, the system behave as a make-to-order queue. Producing one unit of H will save an immediate cost of b_h at the rate of μ. Producing one unit of M will save b_m at the rate of μ as well. Since $b_h < b_m$, always M will save more, which leads to the static priority in the region of $x_h < 0$. The result corresponds to the $c\mu$ rule in a make-to-order system as seen in [3].

The graph is connect from top. For the states with $x_m = B(x_h, i)$ or $C(x_h, i)$, the optimal policy is the same as states with $x_m = B(x_h, i) + 1$ and $C(x_h, i) + 1$. In this example, $B(x_h, 0)$ and $C(x_h, 0)$ intersect at $(x_h, x_m) = (7, 3)$ and $B(x_h, 1)$ and $C(x_h, 1)$ intersect at $(x_h, x_m) = (9, 3)$. Consider the stock levels $(s_h, s_m) = (\max\{7, 9\}, \max\{3, 3\}) = (9, 3)$, when the system starts from region $X(s_h, s_m) =$

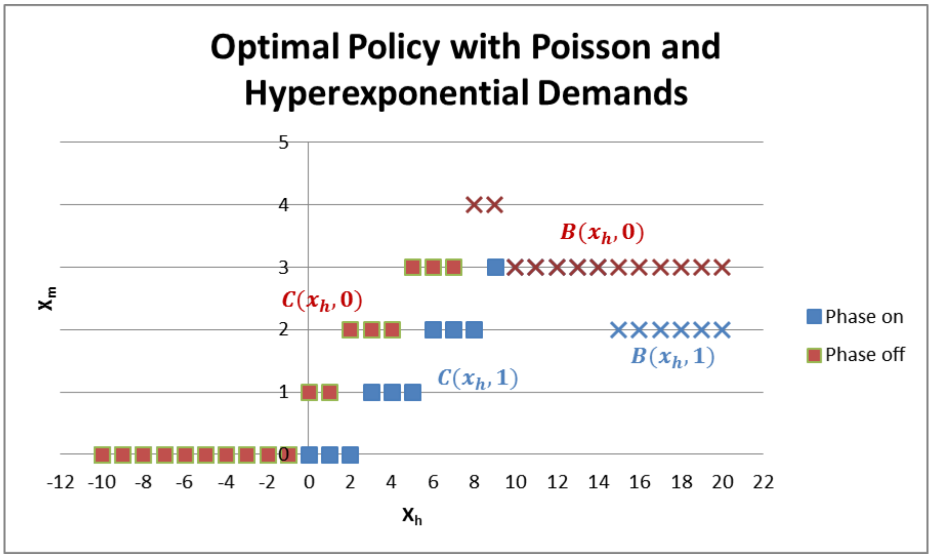

Figure 4.1: Optimal policy for production-inventory system with Poisson and hyper-exponential demands

$\{(x_h, x_m) : x_h \le s_h, x_m \le s_m\}$ it will never leave. So the optimal policy when states starts from this region can be described as a base stock policy (s_h, s_m) for *on/off*, together with $C(x_h, 0)$ and $C(x_h, 1)$ for switching production between class H and M.

Effects of c_v on the Optimal Policy

We analyze the effects of $c_v$on the optimal policy on a set of problems with different combinations of $(\eta_1, \eta_2, \lambda_h)$. Since $^1/_{\eta_1}$ is the average time being in phase *on* and $^1/_{\eta_2}$ the average time being in phase *off* , and these two phases transit according to an underlying Markov process. Then the probability of phase being in either value is

$$\Pr(on) = \frac{\frac{1}{\eta_1}}{\frac{1}{\eta_1} + \frac{1}{\eta_0}} = \frac{\eta_0}{\eta_1 + \eta_0}$$

$$\Pr(off) = 1 - \Pr(on) = \frac{\eta_1}{\eta_1 + \eta_0}$$

Increasing η_1 will decrease the percentage of time where there is positive demand

Table 4.1: IPP parameters with $\lambda_m = 0.45$

	Two Poisson	Poisson + Hyperexponential		
	Problem I	Problem II	Problem III	Problem IV
η_1	0.5	0.1	0.77	0.9
η_0	0.5	0.9	0.23	0.1
λ_h	0.45	0.5	1.96	2.0
c_v	1	1.05	2.0	3.0

from class H. Increase λ_h will increase the intensity of demand for class H when it is on. If demand from class H seldom happens, but occurs with great intensity, overall hyperexponential renewal process will be variable. This is the intuition for the test cases Problem I to IV. Problem I has two demand classes both following Poisson process with identical parameter. Problem II to IV all have two demand classes with one following Poisson process and the other following hyperexponential renewal process. In table 4.1, we change the value combination of $(\eta_1, \eta_2, \lambda_h)$ to vary c_v. The other parameters are $(\mu = 1, \lambda_h, \lambda_m = 0.45, \eta_0, \eta_1, h_h = 3, h_m = 3, b_h = 80, b_m = 100, \alpha = 0.05)$.

Figure 4.2 shows the comparison between Problem I and II. For Problem I, there is one production switching curve (in blue) and production stopping curve (in red). It is consistent with the result from [36] which analyzes a similar model. For Problem II, there are two set of curves corresponding to hyperexponential demand phase being *on* in blue and *off* being red. The transition between the phases make the two separate sets of control curves.

Figure 4.3 shows the comparison between Problem III and IV. With the increase of c_v, the gap between $C(x_h, 0)$ and $C(x_h, 1)$ is increasing. With a smaller c_v (Problem III), the duration for phase *on* is longer but with a lower intensity. With $(x_h = 5, 6, 7,$

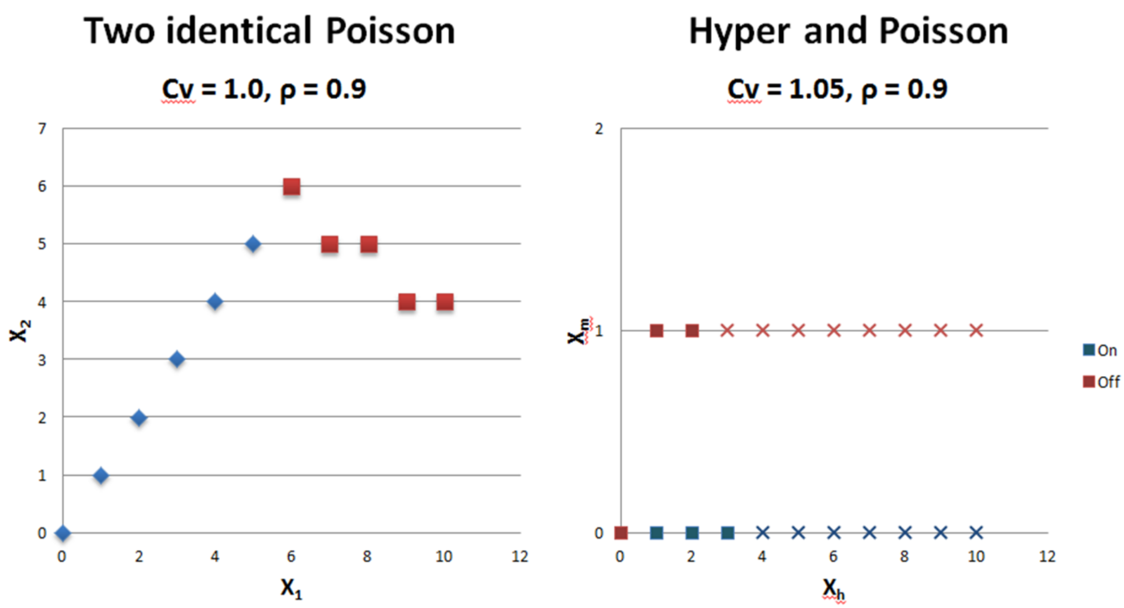

Figure 4.2: Problem I vs. Problem II

x_m =3) the optimal action is to produce class H.

4.5 Knowing Phase Probability

In many situations, demand information is not shared between supplier and retailer. The manager for the production-inventory system (or the supplier) observes stochastic demand from the retailer and use his prior knowledge to guess for the demand pattern from the retailer. What kind of control policy should the manager use to achieve a low total cost? In this section, we study this problem which is practical in real world situations.

Suppose the manager cannot observe phase information (*on/off*), but he has a belief of demand from class H being in phase *on* or *off*. Let $P_1(t)$ be the probability of phase being in phase *on* at time t, and $1 - P_1(t)$ be the probability of phase being

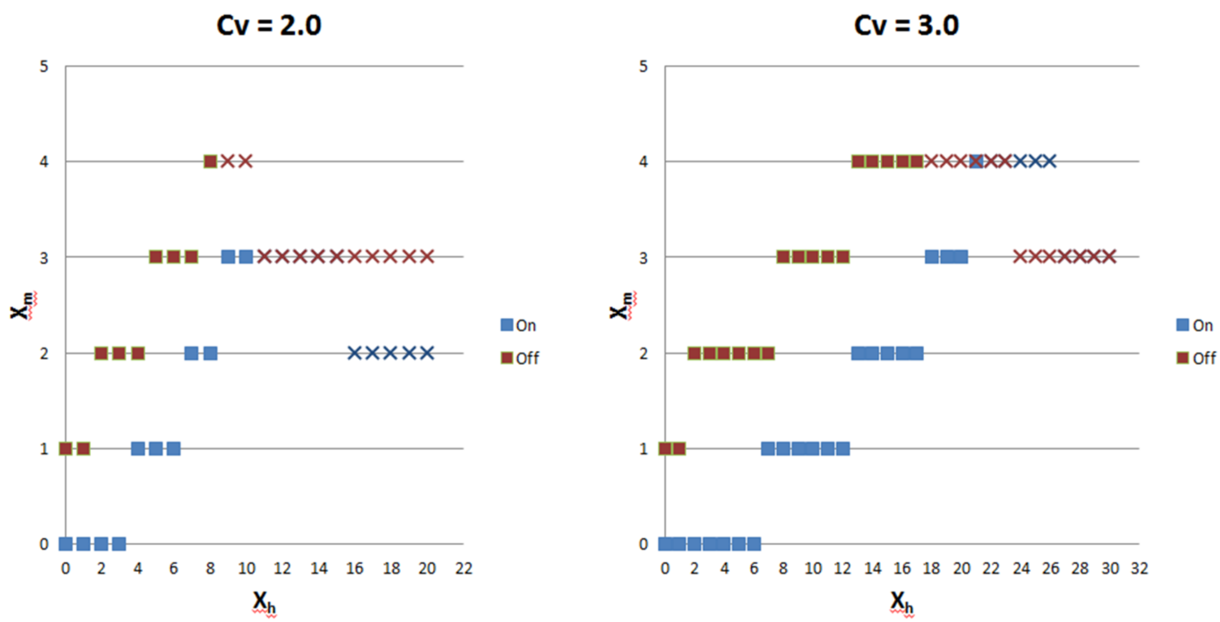

Figure 4.3: Problem III vs. Problem IV

in *off* at time t. Consider a system that is identical to the one in section 4.3 except for phase information is described by the probability p_1. There are demands for class H and M, with arrival processes follow Poisson process and hyperexponential renewal process. At any point of time, the manager can decide to produce for H or M or stay idle. Production time for both classes follow exponential distribution. Production is preemptive. Standard linear holding and backorder costs apply. The notations are the same as in section 4.3.

The state of the system can be described as $(X_h(t), X_m(t), P_1(t))$, with $X_h(t)$, $X_m(t)$ being the net inventory for class H and M at time t and $P_1(t)$ as defined previously. The manager makes decisions when the state changes, for example, production completion and demand satisfied.

To analyze the system, we transform the continuous-time model to discrete time model by uniformization with rate $\gamma = \mu + \lambda_h + \lambda_m + \eta_0 + \eta_1$. Explanation of uniformization can be found in section 2.3.2. With uniformization, the system becomes time invariant and can be described by the state (x_h, x_m, p_1). There are pseudo transitions from the current state to itself, and the probability p_1 will also update according to different production decisions.

Why $p_1 = \Pr(on)$ *May Change?*

Consider the case when phase information is known. Starting from $(x_h, x_m, ?)$, suppose the action is "not produce": if the current actual phase is *on*, then with probability $1 - \frac{\eta_1}{\gamma}$ the next state will be on; if the original phase is *off*, then with probability $\frac{\eta_0}{\gamma}$ the next state will be off. Notice that $1 - \frac{\eta_1}{\gamma} \neq \frac{\eta_0}{\gamma}$, meaning even with the same action, the current phase may change the probability of being *on*. Without the phase information, the current phase is characterize by probability p_1, and it will be updated for the next stage.

Let $p_1'(x, y; \text{action}) = \Pr((x, y)|(x_h, x_m, p_1; \text{action}))$ be the new probability of being in *on* after any transition, we calculate $p_1'(x, y; \text{action})$ for different actions.

Notations and Transition Probabilities:

- State: $(x_h, x_m, p_1), x_h, x_m \in \mathbb{Z}, p_1 \in [0, 1]$. It implies that $p_1 = 0$ means phase is *off*, and $p_1 = 1$ means phase is *on*.

- Transition probability: start from state (x_h, x_m, p_1)

- If action is *notProduce/idle*:

  – In the next states, the potential inventory levels for H and M include (x_h, x_m) - no demand; $(x_h - 1, x_m)$ - demand for H occurs; and $(x_h, x_m - 1)$ - demand for M occurs.

  – Consider possible transition to (x_h, x_m, p_1'). Let $\alpha_1 = \Pr((x_h, x_m), (x_h, x_m, p_1; \text{idle}))$, being the probability of transit to a state with inventory levels at (x_h, x_m)

if start from state (x_h, x_m, p_1) and taking action *idle.* Then,

$$\begin{aligned}\alpha_1 &= \Pr((x_h, x_m), (x_h, x_m, p_1; \text{idle})) \\ &= \sum_{i=0,1} p_i \Pr(x_h, x_m | x_h, x_m, i;\ \text{idle}) \\ &= \sum_{i=0,1} p_i \sum_{j=0,1} \Pr(x_h, x_m, j | x_h, x_m, i;\ \text{idle})\end{aligned}$$

Then the belief of phase being in *on* can be updated in the following way.

$$\begin{aligned}p_1^{'}(x_h, x_m; \text{idle}) &= \Pr((x_h, x_m, 1) | (x_h, x_m, p_1; \text{idle})) \\ &= \frac{\Pr((x_h, x_m, 1), (x_h, x_m, p_1; \text{idle}))}{\Pr((x_h, x_m), (x_h, x_m, p_1; \text{idle}))} \\ &= \frac{\sum_{i=0,1} p_i \Pr(x_h, x_m, 1 | x_h, x_m, i;\ \text{idle})}{\alpha_1}\end{aligned}$$

We know that

$$\begin{aligned}\Pr(on) &= p_1 \\ \Pr((x_h, x_m, 0) | (x_h, x_m, 1;\text{idle})) &= \frac{\eta_1}{\gamma} \\ \Pr((x_h, x_m, 1) | (x_h, x_m, 1; \text{idle})) &= \frac{\eta_0 + \mu}{\gamma}\end{aligned}$$

Also,

$$\begin{aligned}\Pr(off) &= 1 - p_1 \\ \Pr((x_h, x_m, 0) | (x_h, x_m, 0; \text{idle})) &= \frac{\eta_1 + \mu + \lambda_h}{\gamma} \\ \Pr((x_h, x_m, 1) | (x_h, x_m, 0; \text{idle})) &= \frac{\eta_0}{\gamma}\end{aligned}$$

Define

$$\alpha_1 = P((x_h, x_m, p_1'), (x_h, x_m, p_1; \text{idle}))$$

then

$$\begin{aligned}\alpha_1 &= p_1 \frac{\eta_0 + \eta_1 + \mu}{\gamma} + (1 - p_1) \frac{\lambda_h + \eta_1 + \mu + \eta_0}{\gamma} \\ &= \frac{\eta_0 + \eta_1 + \mu + (1 - p_1)\lambda_h}{\gamma}\end{aligned}$$

So

$$p_1' = \frac{p_1\frac{\eta_0+\mu}{\gamma} + (1-p_1)\frac{\eta_0}{\gamma}}{\alpha_1} = \frac{\eta_0 + p_1\mu}{\eta_0 + \eta_1 + \mu + (1-p_1)\lambda_h}$$

- For $p_1'(x_h - 1, x_m; \text{idle})$: $\Pr((x_h - 1, x_m, 1)|(x_h, x_m, 1; \text{idle})) = \frac{\lambda_h}{\gamma}$. Define $\alpha_2 = \Pr((x_h - 1, x_m); (x_h, x_m, p_1; \text{idle}))$, then $\alpha_2 = p_1\frac{\lambda_h}{\gamma}$ and $p_1'(x_h - 1, x_m; \text{idle}) = 1$
- For $p_1'(x_h, x_m - 1; \text{idle})$: $\Pr((x_h, x_m - 1, 1)|(x_m, x_h, 1; \text{idle})) = \frac{\lambda_m}{\gamma}$, and $\Pr((x_h, x_m-1, 0)|(x_m, x_h, 0; \text{idle})) = \frac{\lambda_m}{\gamma}$. Define $\alpha_3 = \Pr((x_h, x_m-1); (x_h, x_m, p_1; \text{idle}))$, then $\alpha_3 = \frac{\lambda_m}{\gamma}$ and $p_1'(x_h, x_m - 1; \text{idle}) = p_1$.

- If action is *prodceH*, in a similar manner, the transition probabilities can be calculated as:
  - $\alpha_1 = \Pr((x_h, x_m); (x_h, x_m, p_1; \text{produceH})) = \frac{\eta_0+\eta_1+(1-p_1)\lambda_h}{\gamma}$, $p_1^{'}(x_h, x_m; \text{produceH}) = \frac{\eta_0}{\eta_0+\eta_1+(1-p_1)\lambda_h}$;
  - $\alpha_2 = \Pr((x_h-1, x_m); (x_h, x_m, p_1; \text{produceH})) = p_1\frac{\lambda_h}{\gamma}$, $p_1^{'}(x_h-1, x_m; \text{produceH}) = 1$;
  - $\alpha_3 = \Pr P((x_h, x_m-1); (x_h, x_m, p_1; \text{produceH})) = \frac{\lambda_m}{\gamma}$, $p_1^{'}(x_h, x_m-1; \text{produceH}) = p_1$;
  - $\alpha_4 = \Pr((x_h+1, x_m); (x_h, x_m, p_1; \text{produceH})) = \frac{\mu}{\gamma}$, $p_1^{'}(x_h+1, x_m; \text{produceH}) = p_1$.

- If action is *produceM*, in a similar manner, the transition probabilities can be calculated as:
  - $\alpha_1 = \Pr((x_h, x_m); (x_h, x_m, p_1; \text{produceM})) = \frac{\eta_0+\eta_1+(1-p_1)\lambda_h}{\gamma}$, $p_1^{'}(x_h, x_m; \text{produceM}) = \frac{\eta_0}{\eta_0+\eta_1+(1-p_1)\lambda_h}$;
  - $\alpha_2 = \Pr((x_h-1, x_m); (x_h, x_m, p_1; \text{produceM})) = p_1\frac{\lambda_h}{\gamma}$, $p_1^{'}(x_h-1, x_m; \text{produceM}) = 1$;

- $\alpha_3 = \Pr P((x_h, x_m{-}1); (x_h, x_m, p_1; \text{produceM})) = \frac{\lambda_m}{\gamma}$, $p_1^{'}(x_h, x_m{-}1; \text{produceM}) = p_1$;
- $\alpha_4 = \Pr((x_h, x_m{+}1); (x_h, x_m, p_1; \text{produceM})) = \frac{\mu}{\gamma}$, $p_1^{'}(x_h, x_m{+}1; \text{produceM}) = p_1$;

The optimal value function $v^*(x_h, x_m, p_1)$ can be shown to satisfy the following equation:

$$v^*(x_h, x_m, p_1) = c(x_h, x_m) + p_1\lambda_h v^*(x_h - 1, x_m, 1) + \lambda_m v^*(x_h, x_m - 1, p_1) + T_0 v^*(x_h, x_m, p_1) \tag{4.4}$$

where

$$T_0 v(x_h, x_m, p_1) = min \left\{ \begin{array}{l} (\eta_0 + \eta_1 + \mu + (1 - p_1)\lambda_h) v(x_h, x_m, \\ \qquad \frac{\eta_0 + p_1\mu}{(\eta_0 + \eta_1 + \mu + (1 - p_1)\lambda_h)}) \\ (\eta_0 + \eta_1 + (1 - p_1)\lambda_h) v(x_h, x_m, \frac{\eta_0}{\eta_0 + \eta_1 + (1 - p_1)\lambda_h}) \\ \qquad + \mu v(x_h + 1, x_m, p_1) \\ (\eta_0 + \eta_1 + (1 - p_1)\lambda_h) v(x_h, x_m, \frac{\eta_0}{\eta_0 + \eta_1 + (1 - p_1)\lambda_h}) \\ \qquad + \mu v(x_h, x_m + 1, p_1) \end{array} \right\} \tag{4.5}$$

Due to the complicated interaction between the probability transitions, it is difficult to analyze the optimal structure of the policy. Numerical analysis shows optimal total discounted cost for knowing phase information is no higher than knowing the phase probability, as shown in figure 4.3.

4.6 Conclusion and Extension

We model the production-inventory system with a Markov decision process (MDP), and formulate it as a continuous-time infinite-horizon stochastic dynamic program. We consider the situation where the demand variability (as measured by coefficient of variation of inter-arrival times) of class H is higher than that of class M. We use hyper-exponential distribution to model the inter-arrival time of class H products demand. The inter-arrival time for class M products demand is exponential. The stochastic demand for any class is satisfied from the on-hand inventory, and unsatisfied demands are backordered. There are linear costs for holding inventory and delaying orders. The production is preemptive, and we assume there is no set up or switching cost for both products. At any time, the production manager observes the phase information of class H demand, the current inventory for the two products, and then chooses to take one of the production decisions: produce one unit of class M, produce one unit of class H, or not to produce. The objective is to minimize the discounted expected total cost over the infinite-time horizon. Let us further characterize the demand process for class H. The demand inter-arrival times follow a hyperexponential distribution. The corresponding demand arrival process is known as *Interrupted Poisson Process* (IPP). An IPP is characterized by two phases, an on phase with Poisson arrivals and an off phase with no arrivals. The durations of on phase and off phase are exponentially distributed with different rates. The arrival process switches between two states according to an underlying Markov chain. When the phase information is fully observable, our dynamic formulation treats it as an observed state variable.

Then we consider the case where the phase information is only partially known. We model the system as a partially-observable Markov decision process with a prior belief of phase for class H being *on* phase. The belief at next decision epoch is updated according to Bayes rule. No simple structure for the belief model exists due to the belief variable is continuous. Numerical analysis is conducted to compare the total cost of the system under full and partial phase information. We interpret our analytical and numerical results in the real world contexts that motivated us and draw insights for managers in similar situations.

There are many directions for future research. We make the assumption that unmet demands are backordered. It would be interesting to see whether the same results extend to the lost sales case. While our model is from the perspective of production-inventory manager, it is possible to include his interaction with the retailer for information sharing. Extensive literature study the information sharing under static production control policy, but none has dealt with dynamic scheduling.

BIBLIOGRAPHY

[1] Amazon. Amazon ec2 service level agreement, June, 2013. `http://aws.amazon.com/ec2-sla/`.

[2] Necati Aras, Vedat Verter, and Tamer Boyaci. Coordination and priority decisions in hybrid manufacturing/remanufacturing systems. *Production and Operations Management*, 15(4):528–543, 2006.

[3] JS Baras, AJ Dorsey, and AM Makowski. Two competing queues with linear costs and geometric service requirements: The μ c-rule is often optimal. *Advances in Applied Probability*, pages 186–209, 1985.

[4] Z Pelin Bayındır, Nesim Erkip, and Refik Güllü. A model to evaluate inventory costs in a remanufacturing environment. *International Journal of Production Economics*, 81:597–607, 2003.

[5] Richard Bellman and Robert E Kalaba. *Dynamic programming and modern control theory*. Academic Press New York, 1965.

[6] S. Benjaafar and M. ElHafsi. Production and inventory control of a single product assemble-to-order system with multiple customer classes. *Management Science*, pages 1896–1912, 2006.

[7] S. Benjaafar, M. ElHafsi, and F. de Véricourt. Demand allocation in multiple-product, multiple-facility, make-to-stock systems. *Management Science*, pages 1431–1448, 2004.

[8] S. Benjaafar, M. ElHafsi, and T. Huang. Optimal control of a production-inventory system with both backorders and lost sales. *Naval Research Logistics (NRL)*, 57(3):252–265, 2010.

[9] S. Benjaafar, M. ElHafsi, C.Y. Lee, and W. Zhou. Technical note - optimal control of an assembly system with multiple stages and multiple demand classes. *Operations Research*, 59(2):522–529, 2011.

[10] Saif Benjaafar, William L Cooper, and Joon-Seok Kim. On the benefits of pooling in production-inventory systems. *Management Science*, 51(4):548–565, 2005.

[11] Saif Benjaafar, Joon-Seok Kim, and N Vishwanadham. On the effect of product variety in production–inventory systems. *Annals of Operations Research*, 126(1-4):71–101, 2004.

[12] Scott Carr and Izak Duenyas. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. *Operations research*, 48(5):709–720, 2000.

[13] Rolando Cavazos-Cadena and Linn I Sennott. Comparing recent assumptions for the existence of average optimal stationary policies. *Operations research letters*, 11(1):33–37, 1992.

[14] WK Ching, WO Yuen, and AW Loh. An inventory model with returns and lateral transshipments. *Journal of the Operational Research Society*, 54(6):636–641, 2003.

[15] Peng Dai and Daniel S Adviser-Weld. *Decision making under uncertainty: scalability and applications*. University of Washington, 2011.

[16] Marisa P de Brito and Rommert Dekker. Modelling product returns in inventory control¡ªexploring the validity of general assumptions. *International Journal of Production Economics*, 81:225–241, 2003.

[17] Gregory DeCroix, Jing-Sheng Song, and Paul Zipkin. A series system with returns: Stationary analysis. *Operations Research*, 53(2):350–362, 2005.

[18] Rommert Dekker, Moritz Fleischmann, and Karl Inderfurth. *Reverse logistics: quantitative models for closed-loop supply chains*. Springer, 2004.

[19] Shad Dowlatshahi. Developing a theory of reverse logistics. *Interfaces*, 30(3):143–155, 2000.

[20] D.D. Eisenstein and A.V. Iyer. Separating logistics flows in the chicago public school system. *Operations Research*, pages 265–273, 1996.

[21] Mohsen ElHafsi, Herve Camus, and Etienne Craye. Managing an integrated production inventory system with information on the production and demand status and multiple non-unitary demand classes. *European Journal of Operational Research*, 207(2):986–1001, 2010.

[22] Eugene A Feinberg, Adam Shwartz, and Eitan Altman. *Handbook of Markov decision processes: methods and applications*. Kluwer Academic Publishers Boston, MA, 2002.

[23] Wolfgang Fischer and Kathleen Meier-Hellstern. The markov-modulated poisson process (mmpp) cookbook. *Performance evaluation*, 18(2):149–171, 1993.

[24] Marshall Fisher. Strengthening the empirical base of operations management. *Manufacturing & Service Operations Management*, 9(4):368–382, 2007.

[25] M.L. Fisher. What is the right supply chain for your product? *Harvard Business Review*, 75:105–117, 1997.

[26] SDP Flapper, Jean-Philippe Gayon, and Samuel Vercraene. Control of a production-inventory system with returns under imperfect advance return information. *European Journal of Operational Research*, 218(2):392–400, 2012.

[27] Simme Douwe P Flapper, Jo AEE van Nunen, and Luk N Van Wassenhove. *Managing closed-loop supply chains*. Springer, 2005.

[28] Moritz Fleischmann and Roelof Kuik. On optimal inventory control with independent stochastic item returns. *European Journal of Operational Research*, 151(1):25–37, 2003.

[29] Moritz Fleischmann, Roelof Kuik, and Rommert Dekker. Controlling inventories with stochastic item returns: A basic model. *European journal of operational research*, 138(1):63–75, 2002.

[30] Jean-Philippe Gayon. A make-to-stock queue with product returns. In *Information Control Problems in Manufacturing*, volume 12, pages 209–214, 2006.

[31] Jean-Philippe Gayon, Saif Benjaafar, and Francis De Véricourt. Using imperfect advance demand information in production-inventory systems with multiple customer classes. *Manufacturing & Service Operations Management*, 11(1):128–143, 2009.

[32] Google. Efficiency: How we do it, 2013. `http://www.google.com/about/datacenters/efficiency/internal/#servers`; Accessed date: 06/24/2013.

[33] Google. Equipment lifecycle, 2013. `http://www.google.co.uk/about/datacenters/ecosystem.html`.

[34] V Daniel R Guide. Production planning and control for remanufacturing: industry practice and research needs. *Journal of Operations Management*, 18(4):467–483, 2000.

[35] Albert Y Ha. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, 43(8):1093–1103, 1997.

[36] A.Y. Ha. Optimal dynamic scheduling policy for a make-to-stock production system. *Operations Research*, pages 42–53, 1997.

[37] A.Y. Ha. Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Research Logistics (NRL)*, 44(5):457–472, 1997.

[38] Daniel P Heyman. Optimal disposal policies for a single-item inventory system with returns. *Naval Research Logistics Quarterly*, 24(3):385–405, 1977.

[39] Mehmet Ali Ilgin and Surendra M Gupta. Environmentally conscious manufacturing and product recovery (ecmpro): a review of the state of the art. *Journal of Environmental Management*, 91(3):563–591, 2010.

[40] Seyed MR Iravani, Tieming Liu, and David Simchi-Levi. Optimal production and admission policies in make-to-stock/make-to-order manufacturing systems. *Production and Operations Management*, 21(2):224–235, 2012.

[41] A.V. Iyer and A. Jain. The logistics impact of a mixture of order-streams in a manufacturer-retailer system. *Management Science*, pages 890–906, 2003.

[42] A.V. Iyer and A. Jain. Modeling the impact of merging capacity in production-inventory systems. *Management Science*, pages 1082–1094, 2004.

[43] A. Jain. Priority and dynamic scheduling in a make-to-stock queue with hyper-exponential demand. *Naval Research Logistics (NRL)*, 53(5):363–382, 2006.

[44] A. Jain and A.V. Iyer. Asking for it. *International Commerce Review*, 7(2):134–143, 2007.

[45] Jemaı. The influence of demand variability on the performance of a make-to-stock queue.

[46] Fikri Karaesmen, George Liberopoulos, and Yves Dallery. The value of advance demand information in production/inventory systems. *Annals of Operations Research*, 126(1-4):135–157, 2004.

[47] Data Center Knowledge. 2011 data center market insight report. Technical report, 2011. http://www.datacenterknowledge.com/archives/2011/09/13/2011-data-center-market-insight-report/.

[48] Ger Koole. *Monotonicity in Markov reward and decision chains: Theory and applications*, volume 1. Now Pub, 2007.

[49] Aybek Korugan and Surendra M Gupta. Substitution policies for a hybrid system. In *Proceedings of SPIE*, volume 4193, pages 1–6, 2001.

[50] National Renewable Energy Laboratory. Best practices guide for energy-efficient data center design. Technical report, U.S. Department of Energy, Energy Efficiency & Renewable Energy, March, 2011.

[51] Guy Latouche and Vaidyanathan Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*, volume 5. Siam, 1999.

[52] Hau L Lee, Venkata Padmanabhan, and Seungjin Whang. Information distortion in a supply chain: the bullwhip effect. *Management science*, 43(4):546–558, 1997.

[53] Steven A Lippman. Applying a new device in the optimization of exponential queuing systems. *Operations Research*, 23(4):687–710, 1975.

[54] Michael L Littman, Thomas L Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.

[55] Pinar Martin, V Daniel R Guide Jr, and Christopher W Craighead. Supply chain sourcing in remanufacturing operations: an empirical investigation of remake versus buy. *Decision Sciences*, 41(2):301–324, 2010.

[56] Rich Miller. Facebook's $1 billion data center network, February, 2012. http://www.datacenterknowledge.com/archives/2012/02/02/facebooks-1-billion-data-center-network/.

[57] Rich Miller. Google spent $951 million on data centers in 4q, January, 2012. http://www.datacenterknowledge.com/archives/2012/01/23/google-spent-951-million-on-data-centers-in-4q/.

[58] John A Muckstadt and Michael H Isaac. An analysis of single item inventory systems with returns. *Naval Research Logistics Quarterly*, 28(2):237–254, 1981.

[59] J.A. Narus and J.C. Anderson. Rethinking distribution: adaptive channels. *Harvard Business Review*, 74(4), 1996.

[60] Marcel F Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Dover Publications, 1981.

[61] Viên Nguyen. A multiclass hybrid production center in heavy traffic. *Operations research*, 46(3-Supplement-3):S13–S25, 1998.

[62] Hiroshi Ohta, Tomoyuki Hirota, and Abdur Rahim. Optimal production-inventory policy for make-to-order versus make-to-stock based on the m/er/1 queuing model. *The International Journal of Advanced Manufacturing Technology*, 33(1-2):36–41, 2007.

[63] M Ortega and L Lin. Control theory applications to the production–inventory problem: a review. *International Journal of Production Research*, 42(11):2303–2322, 2004.

[64] Takayuki Osogami. *Analysis of multi-server systems via dimensionality reduction of Markov chains*. PhD thesis, Citeseer, 2005.

[65] Agnes Peña Perez and Paul Zipkin. Dynamic scheduling rules for a multiproduct make-to-stock queue. *Operations Research*, 45(6):919–930, 1997.

[66] Carol Prahinski and Canan Kocabasoglu. Empirical research opportunities in reverse supply chains. *Omega*, 34(6):519–532, 2006.

[67] Martin L Puterman. Markov decision processes: Discrete dynamic stochastic programming, 1994.

[68] R Canan Savaskan, Shantanu Bhattacharya, and Luk N Van Wassenhove. Closed-loop supply chain models with product remanufacturing. *Management science*, 50(2):239–252, 2004.

[69] Shekhar Srikantaiah, Aman Kansal, and Feng Zhao. Energy aware consolidation for cloud computing. In *Proceedings of the 2008 conference on Power aware computing and systems*, pages 10–10. USENIX Association, 2008.

[70] Samir K Srivastava. Green supply-chain management: a state-of-the-art literature review. *International journal of management reviews*, 9(1):53–80, 2007.

[71] Shaler Stidham Jr. Optimal control of admission to a queueing system. *Automatic Control, IEEE Transactions on*, 30(8):705–713, 1985.

[72] Ruud Teunter. Lot-sizing for inventory systems with product recovery. *Computers & Industrial Engineering*, 46(3):431–441, 2004.

[73] Ruud H Teunter, Z Pelin Bayindir, and Wilco Van Den Heuvel. Dynamic lot sizing with product returns and remanufacturing. *International Journal of Production Research*, 44(20):4377–4400, 2006.

[74] Martijn Christiaan Thierry, Marc Salomon, JAEE van Nunen, and LN van Wassenhove. Strategic issues in product recovery management. *California management review*, 37(2):114–135, 1995.

[75] L Beril Toktay, Lawrence M Wein, and Stefanos A Zenios. Inventory management of remanufacturable products. *Management Science*, 46(11):1412–1426, 2000.

[76] Donald M Topkis. Optimal ordering and rationing policies in a nonstationary dynamic inventory model with n demand classes. *Management Science*, 15(3):160–176, 1968.

[77] USTIC. Remanufactured goods: An overview of the u.s. and global industries, markets, and trade. Research note 4356, United States International Trade Commission, October 2012.

[78] Erwin Van der Laan, Rommert Dekker, and Marc Salomon. Product remanufacturing and disposal: A numerical comparison of alternative control strategies. *International Journal of Production Economics*, 45(1):489–498, 1996.

[79] M.H. Veatch and L.M. Wein. Monotone control of queueing networks. *Queueing Systems*, 12(3):391–408, 1992.

[80] Michael H Veatch and Lawrence M Wein. Scheduling a make-to-stock queue: Index policies and hedging points. *Operations Research*, 44(4):634–647, 1996.

[81] Samuel Vercraene and Jean-Philippe Gayon. Optimal control of a production-inventory system with product returns. *International Journal of Production Economics*, 2012.

[82] F. Véricourt, F. Karaesmen, and Y. Dallery. Assessing the benefits of different stock-allocation policies for a make-to-stock production system. *Manufacturing & Service Operations Management*, 3(2):105–121, 2001.

[83] Richard R Weber. *Optimization and Control*, pages 32–33. online, 2013. `http://www.statslab.cam.ac.uk/~rrw1/oc/oc2013.pdf`.

[84] Richard R Weber and Shaler Stidham Jr. Optimal control of service rates in networks of queues. *Advances in applied probability*, pages 202–218, 1987.

[85] Ward Whitt. Approximations for the gi/g/m queue. *Production and Operations Management*, 2(2):114–161, 1993.

[86] Sean Work. How loading time affects your bottom line, 2011. `http://blog.kissmetrics.com/loading-time/`.

[87] Jing Xu, Ming Zhao, Jose Fortes, Robert Carpenter, and Mazin Yousif. On the use of fuzzy modeling in virtualized data center management. In *Autonomic Computing, 2007. ICAC'07. Fourth International Conference on*, pages 25–25. IEEE, 2007.

[88] Xue-Ming Yuan and Ki Ling Cheung. Modeling returns of merchandise in an inventory system. *OR Spectrum*, 20(3):147–154, 1998.

[89] Hichem Zerhouni, Jean-Philippe Gayon, and Yannick Frein. Influence of dependency between demands and returns in a reverse logistics system. *International Journal of Production Economics*, 2010.

[90] Y.S. Zheng and P. Zipkin. A queueing model to analyze the value of centralized inventory information. *Operations Research*, pages 296–307, 1990.

[91] P.H. Zipkin. Performance analysis of a multi-item production-inventory system under alternative policies. *Management Science*, pages 690–703, 1995.

Appendix A

APPENDIX OF CHAPTER 2

A.1 Proof of Lemma 2.1

The goal is to show T_1 to T_4 are all preserved under properties A.1 to A.7. By default, $\forall x, y \geq 0$.

A.1.1 Proof of T_1

$$T_1 v(x,y) = \begin{cases} v(x,y) + c_x & \text{if } x = 0 \\ v(x-1,y) & \text{if } x > 0 \end{cases}$$

A.1: $\forall x \geq 0$

$$\underset{[T_1 v(x+1,y) - T_1 v(x,y)]}{D_x T_1 v(x,y)} = \begin{cases} -c_x \geq -c_x & \text{if } x = 0 \\ D_x v(x-1,y) \geq -c_x & \text{if } x > 0 \end{cases}$$

A.2: $\forall x \geq 0, y > 0$

$$\underset{[D_x T_1 v(x+1,y) - D_x T_1 v(x,y)]}{D_{x,x} T_1 v(x,y)} = \begin{cases} D_x v(x,y) + c_x & \text{if } x = 0 \\ D_{x,x} v(x-1,y) & \text{if } x > 0 \end{cases}$$

$$\underset{[T_1 v(x+1,y-1) - T_1 v(x,y)]}{D_{x-y} T_1 v(x,y)} = \begin{cases} -D_y v(x,y-1) - c_x & \text{if } x = 0 \\ D_{x-y} v(x-1,y) & \text{if } x > 0 \end{cases}$$

$$\underset{D_{x-y} T_1 v(x+1,y) - D_{x-y} T_1 v(x,y)}{D_{x,x-y} T_1 v(x,y)} = \begin{cases} D_x v(x,y-1) + c_x \geq 0 & \text{if } x = 0 \\ D_{x,x-y} v(x-1,y) \geq 0 & \text{if } x > 0 \end{cases}$$

$$D_{x,x}T_1v(x,y) - D_{x,x-y}T_1v(x,y) = \begin{cases} D_{x,y}v(x,y-1) \geq 0 & \text{if } x = 0 \\ D_{x,x}v(x-1,y) - D_{x,x-y}v(x-1,y) \geq 0 & \text{if } x > 0 \end{cases}$$

A.3: $\forall x \geq 0, y > 0$

Implied in A.2

A.4: $\forall x, y \geq 0$

$$\underbrace{D_yT_1v(x,y)}_{D_yT_1v(x,y+1)-D_yT_1v(x,y)} = \begin{cases} D_yv(x,y) \geq -c_y & \text{if } x = 0 \\ D_yv(x-1,y) \geq -c_y & \text{if } x > 0 \end{cases}$$

$$\underbrace{D_{x,y}T_1v(x,y)}_{D_yT_1v(x+1,y)-D_yT_1v(x,y)} = \begin{cases} 0 \geq 0 & \text{if } x = 0 \\ D_{x,y}v(x-1,y) & \text{if } x > 0 \end{cases}$$

A.5: $\forall x, y \geq 0$

$$\underbrace{D_{y,y}T_1v(x,y)}_{D_yT_1v(x,y+1)-D_yT_1v(x,y)} = \begin{cases} D_{y,y}v(x,y) \geq 0 & \text{if } x = 0 \\ D_{y,y}v(x-1,y) \geq 0 & \text{if } x > 0 \end{cases}$$

A.6: $\forall x \geq 0, y > 0$

$$\underbrace{D_{y,x-y}T_1v(x,y)}_{D_{x-y}T_1v(x,y+1)-D_{x-y}T_1v(x,y)} = \begin{cases} -D_{y,y}v(x,y-1) \leq 0 & \text{if } x = 0 \\ D_{y,x-y}v(x-1,y) \leq 0 & \text{if } x > 0 \end{cases}$$

A.7: $\forall x, y \geq 0$

Implied in proof of A.4.

A.1.2 Proof of T_2

$$T_2v(x,y) = \begin{cases} v(x,y) + c_y & \text{if } y = 0 \\ v(x,y-1) & \text{if } y > 0 \end{cases}$$

<u>*A.1:*</u> $\forall x \geq 0$

$$\underset{[T_2v(x+1,y)-T_2v(x,y)]}{D_xT_2v(x,y)} = \begin{cases} D_xv(x,y) \geq -c_x & \text{if } y=0 \\ D_xv(x,y-1) \geq -c_x & \text{if } y>0 \end{cases}$$

<u>*A.2:*</u> $\forall x \geq 0, y > 0$

$$\underset{[D_xT_2v(x+1,y)-D_xT_2v(x,y)]}{D_{x,x}T_2v(x,y)} = \begin{cases} D_{x,x}v(x,y-1) & \text{if } y>0 \end{cases}$$

$$\underset{[T_2v(x+1,y-1)-T_2v(x,y)]}{D_{x-y}T_2v(x,y)} = \begin{cases} D_xv(x,y-1)+c_y & \text{if } y=1 \\ D_{x-y}v(x,y-1) & \text{if } y>1 \end{cases}$$

$$\underset{D_{x-y}T_2v(x+1,y)-D_{x-y}T_2v(x,y)}{D_{x,x-y}T_2v(x,y)} = \begin{cases} D_{x,x}v(x,y-1) \geq 0 & \text{if } y=1 \\ D_{x,x-y}v(x,y-1) \geq 0 & \text{if } y>1 \end{cases}$$

$$D_{x,x}T_2v(x,y)-D_{x,x-y}T_2v(x,y)= \begin{cases} 0 \geq 0 & \text{if } y=1 \\ D_{x,x}v(x,y-1)-D_{x,x-y}v(x,y-1) \geq 0 & \text{if } y>1 \end{cases}$$

<u>*A.3:*</u> $\forall x \geq 0, y > 0$

Implied in A.2

<u>*A.4:*</u> $\forall x, y \geq 0$

$$\underset{D_yT_2v(x,y+1)-D_yT_2v(x,y)}{D_yT_2v(x,y)} = \begin{cases} -c_y \geq -c_y & \text{if } y=0 \\ D_yv(x,y-1) \geq -c_y & \text{if } y>0 \end{cases}$$

$$\underset{D_yT_2v(x+1,y)-D_yT_2v(x,y)}{D_{x,y}T_2v(x,y)} = \begin{cases} 0 \geq 0 & \text{if } y=0 \\ D_{x,y}v(x,y-1) & \text{if } y>0 \end{cases}$$

<u>*A.5:*</u> $\forall x, y \geq 0$

$$\underset{D_yT_2v(x,y+1)-D_yT_2v(x,y)}{D_{y,y}T_2v(x,y)} = \begin{cases} D_yv(x,y)+c_y \geq 0 & \text{if } y=0 \\ D_{y,y}v(x,y-1) \geq 0 & \text{if } y>0 \end{cases}$$

A.6: $\forall x \geq 0, y > 0$

$$\underbrace{D_{y,x-y}T_2v(x,y)}_{D_{x-y}T_2v(x,y+1)-D_{x-y}T_2v(x,y)} = \begin{cases} \underbrace{D_{x-y}v(x,y) - D_xv(x,y-1)}_{=-D_yv(x,y-1)-c_y} - c_y \leq 0 & \text{if } y = 1 \\ D_{y,x-y}v(x,y-1) \leq 0 & \text{if } y > 1 \end{cases}$$

A.7:

Implied in proof of A.4.

A.1.3 Proof of T_3

Since $T_3v(x,y) = v(x,y+1)$, all the proof follow automatically from the properties.

A.1.4 Proof of T_4

Proof. $T_4v(x,y) = \begin{cases} min[v(x,y), v(x+1,y)] & \text{if } y = 0 \\ min[v(x,y), v(x+1,y), v(x+1,y-1)] & \text{if } y > 0 \end{cases}$

Since T_4 includes min of three components, we follow the approach by [36]. Let w be a function on $\{0,1,2\} \times Z^+ \times Z^+$ such that

$$\begin{aligned} w(u,x,y) &= \tfrac{1}{2}(1-u)(2-u)v(x,y) + u(2-u)v(x+1,y) + \tfrac{1}{2}u(u-1)v(x+1,y-1) \\ &= \begin{cases} v(x,y) & \text{if } u = 0 \\ v(x+1,y) & \text{if } u = 1 \\ v(x+1,y-1) & \text{if } u = 2 \quad \text{when } y > 0 \end{cases} \end{aligned}$$

Also define

$$g(x,y) = min[v(x,y), v(x+1,y), v(x+1,y-1)] = \min_{u\in\{0,1,2\}} w(u,x,y)$$

So, $D_yw(u,x,y) = \begin{cases} D_yv(x,y) & \text{if } u = 0 \\ D_yv(x+1,y) & \text{if } u = 1 \\ D_yv(x+1,y-1) & \text{if } u = 2 \quad \text{when } y > 0 \end{cases}$.

By property A.4 $D_yw_{u=1} \geq D_yw_{u=0}$, by property A.3 $D_yw_{u=2} \leq D_yw_{u=0}$. Thus $D_yw_{u=1} \geq D_yw_{u=0} \geq D_yw_{u=2}$, when $y > 0$. When $y = 0$, $D_yw_{u=1} \geq D_yw_{u=0}$.

$$\text{Also, } D_{x-y}w(u,x,y) = \begin{cases} D_{x-y}v(x,y) & \text{if } u=0 \\ D_{x-y}v(x+1,y) & \text{if } u=1 \\ D_{x-y}v(x+1,y-1) & \text{if } u=2 \quad \text{when } y>0 \end{cases}.$$

In the same manner, we can determine when $y > 0$, $D_{x-y}w_{u=2} \geq D_{x-y}w_{u=1} \geq D_{x-y}w_{u=0}$ by A.6 and A.3.

The following proof on T_4 is based on the case when $y > 0$. When $y = 0$, the proof is a subset of when $y > 0$. We omit the details for brevity.

A.1: To show $\forall x, y \geq 0$, $D_xT_4v(x,y) \geq -c_x$ is equivalent to show $g(x,y) \leq g(x+1,y) + c_x$

Let $g(x+1,y) = w(u_1, x+1, y)$, where $u_1 \in \{0,1,2\}$

$g(x,y) \leq w(u_1,x,y) \underset{\text{A.1}}{\leq} w(u_1,x+1,y) + c_x = g(x+1,y) + c_x$

A.2: To show $D_{x,x}T_4v(x,y) \geq D_{x,x-y}T_4v(x,y)$ is equivalent to show $g(x+2,y-1) + g(x+1,y) \leq g(x+2,y) + g(x+1,y-1)$.

Let $g(x+2,y) = w(u_1,x+2,y)$, $g(x+1,y-1) = w(u_2,x+1,y-1)$.

Case 1: $D_yw_{u_2} \leq D_yw_{u_1}$, or the ordered pair of (u_1,u_2) are within these cases: $(1,2), (0,2), (1,0)$

$$\begin{aligned} g(x+2,y-1) + g(x+1,y) &\leq w(u_1,x+2,y-1) + w(u_2,x+1,y) \\ \text{By A.2} \quad &\leq w(u_1,x+2,y) + w(u_1,x+1,y-1) - w(u_1,x+1,y) \\ &\quad +w(u_2,x+1,y) \\ \text{By } D_yw_{u_2} \leq D_yw_{u_1} \quad &\leq w(u_1,x+2,y) + w(u_1,x+1,y) - w(u_1,x+1,y) \\ &\quad +w(u_2,x+1,y-1) \\ &= g(x+2,y) + g(x+1,y-1) \end{aligned}$$

Case 2: $D_yw_{u_2} > D_yw_{u_1}$

Case 2.1: $u_1 = 0, u_2 = 1$

$$\begin{aligned} g(x+2,y-1) + g(x+1,y) &\leq w(u_1=0,x+2,y-1) + w(u_2=1,x+1,y) \\ &= v(x+2,y) + v(x+2,y-1) \\ &= g(x+2,y) + g(x+1,y-1) \end{aligned}$$

Case 2.2: $u_1 = 2, u_2 = 1$

$$
\begin{aligned}
g(x+2,y-1)+g(x+1,y) &\le w(u_2=1,x+2,y-1)+w(u_1=2,x+1,y)\\
&= v(x+3,y-1)+v(x+2,y-1)\\
&= g(x+2,y)+g(x+1,y-1)
\end{aligned}
$$

Case 2.3: $u_1 = 2, u_2 = 0$

$$
\begin{aligned}
g(x+2,y-1)+g(x+1,y) &\le w(u_2=0,x+2,y-1)+w(u_1=2,x+1,y)\\
&= v(x+2,y-1)+v(x+2,y-1)\\
\text{By A.2} \quad &\le v(x+3,y-1)+v(x+1,y-1)\\
&= g(x+2,y)+g(x+1,y-1)
\end{aligned}
$$

A.3: To show $D_{x,x-y}T_4v(x,y) \ge 0$ is equivalent to show $g(x+1,y)+g(x+1,y-1) \le g(x+2,y-1)+g(x,y)$.

Let $g(x+2,y-1) = w(u_1,x+2,y-1), g(x,y) = w(u_2,x,y)$.

Case 1: $D_{x-y}w_{u_2} \le D_{x-y}w_{u_1}$, or the ordered pair of (u_1,u_2) are within these cases: $(1,2),(2,0),(1,0)$

$$
\begin{aligned}
g(x+1,y)+g(x+1,y-1) &\le w(u_1,x+1,y)+w(u_2,x+1,y-1)\\
\text{By A.3} \quad &\le w(u_1,x+2,y-1)+w(u_1,x,y)-w(u_1,x+1,y-1)\\
&\quad +w(u_2,x+1,y-1)\\
\text{By } D_{x-y}w_{u_2} \le D_{x-y}w_{u_1} \quad &\le w(u_1,x+2,y-1)+w(u_1,x+1,y-1)\\
&\quad -w(u_1,x+1,y-1)+w(u_2,x,y)\\
&= g(x+2,y-1)+g(x,y)
\end{aligned}
$$

Case 2: $D_{x-y}w_{u_2} > D_{x-y}w_{u_1}$

Case 2.1: $u_1 = 1, u_2 = 2$

$$
\begin{aligned}
g(x+1,y)+g(x+1,y-1) &\le w(u_2=2,x+1,y)+w(u_1=1,x+1,y-1)\\
&= v(x+2,y-1)+v(x+2,y-1)\\
\text{By A.2} \quad &\le v(x+1,y-1)+v(x+3,y-1)\\
&= g(x+2,y-1)+g(x,y)
\end{aligned}
$$

Case 2.2: $u_1 = 0, u_2 = 2$

$$
\begin{aligned}
g(x+1,y)+g(x+1,y-1) &\le w(u_2=2,x+1,y)+w(u_1=0,x+1,y-1)\\
&= v(x+2,y-1)+v(x+1,y-1)\\
&= g(x+2,y-1)+g(x,y)
\end{aligned}
$$

Case 2.3: $u_1 = 0, u_2 = 1$

$$\begin{aligned} g(x+1,y)+g(x+1,y-1) &\le w(u_1=0,x+1,y)+w(u_2=1,x+1,y-1) \\ &= v(x+1,y)+v(x+2,y-1) \\ &= g(x+2,y-1)+g(x,y) \end{aligned}$$

A.4: To show $D_{x,y}T_4v(x,y) \ge 0$ is equivalent to show $g(x+1,y)+g(x,y+1) \le g(x+1,y+1)+g(x,y)$.

Let $g(x+1,y+1) = w(u_1,x+1,y+1), g(x,y) = w(u_2,x,y)$.

Case 1: $D_y w_{u_2} \le D_y w_{u_1}$, or the ordered pair of (u_1,u_2) are within these cases: $(1,2),(0,2),(1,0)$.

$$\begin{aligned} g(x+1,y)+g(x,y+1) &\le w(u_1,x+1,y)+w(u_2,x,y+1) \\ \text{By A.4} \quad &\le w(u_1,x+1,y+1)+w(u_1,x,y)-w(u_1,x,y+1) \\ &\quad +w(u_2,x,y+1) \\ \text{By } D_y w_{u_2} \le D_y w_{u_1} \quad &\le w(u_1,x+1,y+1)+w(u_1,x,y+1)-w(u_1,x,y+1) \\ &\quad +w(u_2,x,y) \\ &= g(x+1,y+1)+g(x,y) \end{aligned}$$

Case 2: $D_y w_{u_2} > D_y w_{u_1}$

Case 2.1: $u_1 = 0, u_2 = 1$

$$\begin{aligned} g(x+1,y)+g(x,y+1) &\le w(u_1=0,x+1,y)+w(u_2=1,x,y+1) \\ &= v(x+1,y)+v(x+1,y+1) \\ &= g(x+1,y+1)+g(x,y) \end{aligned}$$

Case 2.2: $u_1 = 2, u_2 = 1$

$$\begin{aligned} g(x+1,y)+g(x,y+1) &\le w(u_2=1,x+1,y)+w(u_1=2,x,y+1) \\ &= v(x+2,y)+v(x+1,y) \\ &= g(x+1,y+1)+g(x,y) \end{aligned}$$

Case 2.3: $u_1 = 2, u_2 = 0$

$$\begin{aligned} g(x+1,y)+g(x,y+1) &\le w(u_2=0,x+1,y)+w(u_1=2,x,y+1) \\ &= v(x+1,y)+v(x+1,y) \\ \text{By A.2} \quad &\le v(x,y)+v(x+2,y) \\ &= g(x+1,y+1)+g(x,y) \end{aligned}$$

A.5: To show $D_{y,y}T_4v(x,y) \geq 0$ is equivalent to show $2g(x,y+1) \leq g(x,y+2) + g(x,y)$.

Let $g(x,y+2) = w(u_1,x+1,y+1), g(x,y) = w(u_2,x,y)$.

Case 1: $D_y w_{u_2} \leq D_y w_{u_1}$, or the ordered pair of (u_1,u_2) are within these cases: $(1,2),(0,2),(1,0)$.

$$
\begin{aligned}
g(x,y+1)+g(x,y+1) &\leq w(u_1,x,y+1)+w(u_2,x,y+1)\\
\text{By A.5} \quad &\leq w(u_1,x,y+2)+w(u_1,x,y)-w(u_1,x,y+1)\\
&\quad +w(u_2,x,y+1)\\
\text{By } D_y w_{u_2} \leq D_y w_{u_1} \quad &\leq w(u_1,x,y+2)+w(u_1,x,y+1)-w(u_1,x,y+1)\\
&\quad +w(u_2,x,y)\\
&= g(x,y+2)+g(x,y)
\end{aligned}
$$

Case 2: $D_y w_{u_2} > D_y w_{u_1}$

Case 2.1: $u_1 = 0, u_2 = 1$

$$
\begin{aligned}
g(x,y+1)+g(x,y+1) &\leq w(u_1=1,x,y+1)+w(u_2=0,x,y+1)\\
&= v(x+1,y+1)+v(x,y+1)\\
\text{By A.6} \quad &\leq v(x,y+2)+v(x+1,y)\\
&= g(x,y+2)+g(x,y)
\end{aligned}
$$

Case 2.2: $u_1 = 2, u_2 = 1$

$$
\begin{aligned}
g(x,y+1)+g(x,y+1) &\leq w(u_2=1,x,y+1)+w(u_1=2,x,y+1)\\
&= v(x+1,y+1)+v(x+1,y)\\
&= g(x,y+2)+g(x,y)
\end{aligned}
$$

Case 2.3: $u_1 = 2, u_2 = 0$

$$
\begin{aligned}
g(x,y+1)+g(x,y+1) &\leq w(u_2=0,x,y+1)+w(u_1=2,x,y+1)\\
&= v(x,y+1)+v(x+1,y)\\
\text{By A.4} \quad &\leq v(x,y)+v(x+1,y+1)\\
&= g(x,y+2)+g(x,y)
\end{aligned}
$$

A.6: To show $D_{y,x-y}T_4v(x,y) \leq 0$, it is equivalent to show $g(x+1,y)+g(x,y) \leq g(x,y+1)+g(x+1,y-1)$.

Let $g(x,y+1) = w(u_1,x,y+1), g(x+1,y-1) = w(u_2,x+1,y-1)$.

Case 1: $D_{x-y}w_{u_1} \le D_{x-y}w_{u_2}$, or the ordered pair of (u_1, u_2) are within these cases: $(0,1),(0,2),(1,2)$

$$\begin{aligned}
g(x+1,y)+g(x,y) &\le w(u_1,x+1,y)+w(u_2,x,y)\\
\text{By A.6} &\le w(u_1,x,y+1)+w(u_1,x+1,y-1)-w(u_1,x,y)\\
&\quad +w(u_2,x,y)\\
\text{By } D_{x-y}w_{u_1} \le D_{x-y}w_{u_2} &\le w(u_1,x,y+1)+w(u_1,x,y)-w(u_1,x,y)\\
&\quad +w(u_2,x+1,y-1)\\
&= g(x,y+1)+g(x+1,y-1)
\end{aligned}$$

Case 2: $D_{x-y}w_{u_1} > D_{x-y}w_{u_2}$

<u>Case 2.1: $u_1 = 1, u_2 = 0$</u>

$$\begin{aligned}
g(x+1,y)+g(x,y) &\le w(u_2=0,x+1,y)+w(u_1=1,x,y)\\
&= v(x+1,y)+v(x+1,y)\\
\text{By A.5} &\le v(x+1,y+1)+v(x+1,y-1)\\
&= g(x,y+1)+g(x+1,y-1)
\end{aligned}$$

<u>Case 2.2: $u_1 = 2, u_2 = 0$</u>

$$\begin{aligned}
g(x+1,y)+g(x,y) &\le w(u_2=0,x+1,y)+w(u_1=2,x,y)\\
&= v(x+1,y)+v(x+1,y-1)\\
&= g(x,y+1)+g(x+1,y-1)
\end{aligned}$$

<u>Case 2.3: $u_1 = 2, u_2 = 1$</u>

$$\begin{aligned}
g(x+1,y)+g(x,y) &\le w(u_1=2,x+1,y)+w(u_2=1,x,y)\\
&= v(x+1,y)+v(x+2,y-1)\\
&= g(x,y+1)+g(x+1,y-1)
\end{aligned}$$

<u>A.7:</u> To show $D_yT_4v(x,y) \ge -c_y$, it is equivalent to show $g(x,y) \le g(x,y+1)+-c_y$.

Let $g(x,y+1) = w(u_1,x,y+1)$, where $u_1 \in \{0,1,2\}$

$g(x,y) \le w(u_1,x,y) \underset{\text{A.7}}{\le} w(u1,x,y+1)+c_y = g(x,y+1)+c_y$

Since operator T is contraction mapping, with the fixed point theorem on the Banach space we can show any sequence of value functions (v_n) defined as $v_{n+1} = Tv_n$ will converge to the optimal value function v^*, which is the unique solution of $v^* = Tv^*$. By induction, $v^* \in U$. □

Appendix B

APPENDIX OF CHAPTER 4

B.1 Proof of Lemma 4.1

Proof: for any (x_h, x_m), v is super modular and has diagonal dominance.

Proof. Let w be a function on $\{0, 1, 2\} \times Z^2 \times \{0, 1\}$ such that

$$\begin{aligned} w(u, x_h, x_m, i) &= \tfrac{1}{2}(1-u)(2-u)v(x_h, x_m, i) \\ &\quad +u(2-u)v(x_h+1, x_m, i) \\ &\quad +\tfrac{1}{2}u(u-1)v(x_h, x_m+1, i) \\ &= \begin{cases} v(x_h, x_m, i) & \text{if } u = 0 \\ v(x_h+1, x_m, i) & \text{if } u = 1 \\ v(x_h, x_m+1, i) & \text{if } u = 2 \end{cases} \end{aligned}$$

Also define

$$\begin{aligned} g(x_h, x_m, i) &= min\{v(x_h, x_m, i), v(x_h+1, x_m, i), v(x_h, x_m+1, i)\} \\ &= \min_{u \in \{0,1,2\}} w(u, x_h, x_m, i) \qquad \text{where } i \in \{0, 1\} \end{aligned}$$

1. *Sumpermodularity*: The goal is to show for any (x_h, x_m),

$$g(x_h+1, x_m, i) + g(x_h, x_m+i) \leq g(x_h+1, x_m+1, i) + g(x_h, x_m, i)$$

2. *Diagonal dominance*: The goal is to show for any (x_h, x_m),

$$\begin{aligned} g(x_h+1, x_m+1, i) + g(x_h, x_m+1, i) &\leq g(x_h, x_m+2, i) + g(x_h+1, x_m, i) \\ g(x_h+1, x_m+1, i) + g(x_h+1, x_m, i) &\leq g(x_h+2, x_m, i) + g(x_h, x_m+1, i) \end{aligned}$$

Following [36], it is easy to show the supermodularity and diagonal dominance are preserved under the optimal operator. □

Proof: for any (x_h, x_m), $(D_m - D_h)v(x_h, x_m, 0) \leq (D_m - D_h)v(x_h, x_m, 1)$ *is preserved under* T_0 *.*

Proof.

$$(D_m - D_h)w(u, x_h, x_m, i) = \begin{cases} (D_m - D_h)v(x_h, x_m, i) & \text{if } u = 0 \\ (D_m - D_h)v(x_h + 1, x_m, i) & \text{if } u = 1 \\ (D_m - D_h)v(x_h, x_m + 1, i) & \text{if } u = 2 \end{cases}$$

Since v has diagonal dominance, $(D_m - D_h)w_{u=1} \leq (D_m - D_h)w_{u=0} \leq (D_m - D_h)w_{u=2}$. The goal is to show for any (x_h, x_m),

$$g(x_h, x_m + 1, 0) + g(x_h + 1, x_m, 1) \leq g(x_h + 1, x_m, 0) + g(x_h, x_m + 1, 1)$$

Let $u_1, u_2 \in \{0, 1, 2\}$ be such that

$$\begin{aligned} g(x_h + 1, x_m, 0) &= w(u_1, x_h + 1, x_m, 0) \\ g(x_h, x_m + 1, 1) &= w(u_2, x_h, x_m + 1, 1) \end{aligned}$$

There are two cases to consider:

1. When $(D_m - D_h)w_{u_1} \leq (D_m - D_h)w_{u_2}$

Equivalently, the ordered pair (u_1, u_2) are within these cases: $(1, 0)$, $(1, 2)$, $(0, 2)$ or $u_1 = u_2$.

$$\begin{aligned} g(x_h, x_m + 1, 0) + g(x_h + 1, x_m, 1) &\leq w(u_1, x_h, x_m + 1, 0) + w(u_2, x_h + 1, x_m, 1) \\ &\leq w(u_1, x_h + 1, x_m, 0) + w(u_1, x_h, x_m + 1, 1) \\ &\quad -w(u_1, x_h + 1, x_m, 1) + w(u_2, x_h + 1, x_m, 1) \\ &\leq w(u_1, x_m + 1, x_m, 0) - w(u_1, x_h + 1, x_m, 1) \\ &\quad +w(u_1, x_h + 1, x_m, 1) + w(u_2, x_h, x_m, 1) \\ &= g(x_h + 1, x_m, 0) + g(x_h, x_m + 1, 1) \end{aligned}$$

2. When $(D_m - D_h)w_{u_1} \geq ((D_m - D_h)w_{u_2})$

<u>Case 2.1: $u_1 = 2$, $u_2 = 0$</u>

$$
\begin{aligned}
g(x_h, x_m+1, 0) + g(x_h+1, x_m, 1) &\leq w(u_2 = 0, x_h, x_m+1, 0) \\
\text{By definition of } w \qquad &\quad +w(u_1 = 2, x_h+1, x_m, 1) \\
&= v(x_h, x_m+1, 0) + v(x_h+1, x_m+1, 1) \\
\text{By } D_h v(1) \leq D_h v(0) \qquad &\leq v(x_h+1, x_m+1, 0) + v(x_h, x_m+1, 1) \\
&= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m+1, 1) \\
&= g(x_h+1, x_m, 0) + g(x_h, x_m+1, 1)
\end{aligned}
$$

<u>Case 2.2: $u_1 = 2$, $u_2 = 1$</u>

$$
\begin{aligned}
g(x_h, x_m+1, 0) + g(x_h+1, x_m, 1) &\leq w(u_2 = 1, x_h, x_m+1, 0) \\
\text{By definition of } w \qquad &\quad +w(u_1 = 2, x_h+1, x_m, 1) \\
&= v(x_h+1, x_m+1, 0) + v(x_h+1, x_m+1, 1) \\
&= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m+1, 1) \\
&= g(x_h+1, x_m, 0) + g(x_h, x_m+1, 1)
\end{aligned}
$$

<u>Case 2.3: $u_1 = 0$, $u_2 = 1$</u>

$$
\begin{aligned}
g(x_h, x_m+1, 0) + g(x_h+1, x_m, 1) &\leq w(u_1 = 0, x_h, x_m+1, 0) \\
\text{By definition of } w \qquad &\quad +w(u_2 = 1, x_h+1, x_m, 1) \\
&= v(x_h, x_m+1, 0) + v(x_h+2, x_m, 1) \\
\text{By...} \qquad &\leq v(x_h+1, x_m, 0) + v(x_h+1, x_m+1, 1) \\
&= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m+1, 1) \\
&= g(x_h+1, x_m, 0) + g(x_h, x_m+1, 1)
\end{aligned}
$$

Note: This case is numerically tested to be true given:

$(u_1 = 0) \Rightarrow D_m v(x_h+1, x_m, 0) \geq 0$ and $D_h v(x_h+1, x_m, 0) \geq 0$

$(u_2 = 1) \Rightarrow D_h v(x_h, x_m+1, 1) \leq 0$ and $(D_h - D_m)v(x_h, x_m+1, 1) \leq 0$

Need to show:

$(D_m - D_h)v(x_h, x_m, 0) \leq (D_m - D_h)v(x_h+1, x_m, 1)$ $\square$

Proof: for any (x_h, x_m), $D_h v(x_h, x_m, 1) \leq D_h(x_h, x_{m,}0)$ is preserved under T_0 .

Proof. Use the same definition of function w and g, the goal is to show that for any (x_h, x_m)

$$g(x_h+1, x_m, 1) + g(x_h, x_m, 0) \leq g(x_h+1, x_m, 0) + g(x_h, x_m, 1)$$

Let $u_1, u_2 \in \{0, 1, 2\}$ be such that

$$\begin{aligned} g(x_h+1, x_m, 0) &= w(u_1, x_h+1, x_m, 0) \\ g(x_h, x_m, 1) &= w(u_2, x_h, x_m, 1) \end{aligned}$$

Let D_i be the difference operation on phase variable i, e.g:$D_i v(x_h, x_m, 0) = v(x_h, x_m, 1) - v(x_h, x_m, 0)$. We have

$$D_i w(u, x_h, x_m, 0) = \begin{cases} D_i v(x_h, x_m, 0) & \text{if } u = 0 \\ D_i v(x_h+1, x_m, 0) & \text{if } u = 1 \\ D_i v(x_h, x_m+1, 0) & \text{if } u = 2 \end{cases}$$

By property (d), $D_i w_{u=1} \leq D_i w_{u=0}$. By property (c), $D_i w_{u=1} \leq D_i w_{u=2}$. We separate the cases by comparing $D_i w_{u=0}$ and $D_i w_{u=2}$.

1. When $D_i w_{u=1} \leq D_i w_{u=0} \leq D_i w_{u=2}$

Case 1.1: $D_i w_{u_1} \leq D_i w_{u_2}$

The ordered pair of (u_1, u_2) is within these cases of $(1, 0)$, $(1, 2)$, $(0, 2)$ or $u_1 = u_2$.

$$\begin{aligned} & g(x_h+1, x_m, 1) + g(x_h, x_m, 0) && \leq w(u_1, x_h+1, x_m, 1) + w(u_2, x_h, x_m, 0) \\ & \text{By property (d)} && \leq w(u_1, x_h+1, x_m, 0) + w(u_1, x_h, x_m, 1) \\ & && \quad -w(u_1, x_h, x_m, 0) + w(u_2, x_h, x_m, 0) \\ & \text{By } D_i w_{u_1} \leq D_i w_{u_2} && \leq w(u_1, x_h+1, x_m, 0) - w(u_1, x_h, x_m, 0) \\ & && \quad +w(u_1, x_h, x_m, 0) + w(u_2, x_h, x_m, 1) \\ & && = w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\ & && = g(x_h+1, x_m, 0) + g(x_h, x_m, 1) \end{aligned}$$

Case 1.2: $D_i w_{u_1} \geq D_i w_{u_2}$

Case 1.2.1: $u_1 = 2,\ u_2 = 0$

$$\begin{aligned} g(x_h+1, x_m, 1) + g(x_h, x_m, 0) &\leq w(u_2 = 0, x_h+1, x_m, 1) + w(u_1 = 2, x_h, x_m, 0) \\ &= v(x_h+1, x_m, 1) + v(x_h, x_m+1, 0) \\ \text{By (c) and } D_{m,h} \geq 0 \qquad &\leq v(x_h+1, x_m+1, 0) + v(x_h, x_m, 1) \\ &= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\ &= g(x_h+1, x_m, 0) + g(x_h, x_m, 1) \end{aligned}$$

Case 1.2.2: $u_1 = 0,\ u_2 = 1$

$$\begin{aligned} g(x_h+1, x_m, 1) + g(x_h, x_m, 0) &\leq w(u_1 = 0, x_h+1, x_m, 1) + w(u_2 = 1, x_h, x_m, 0) \\ &= v(x_h, x_m+1, 1) + v(x_h+1, x_m, 0) \\ &= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\ &= g(x_h+1, x_m, 0) + g(x_h, x_m, 1) \end{aligned}$$

Case 1.2.3: $u_1 = 2,\ u_2 = 1$

$$\begin{aligned} g(x_h+1, x_m, 1) + g(x_h, x_m, 0) &\leq w(u_1 = 2, x_h+1, x_m, 1) + w(u_2 = 1, x_h, x_m, 0) \\ &= v(x_h+1, x_m+1, 1) + v(x_h+1, x_m, 0) \\ \text{By...} \qquad &\leq v(x_h+1, x_m+1, 0) + v(x_h+1, x_m, 1) \\ &= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\ &= g(x_h+1, x_m, 0) + g(x_h, x_m, 1) \end{aligned}$$

2. When $D_i w_{u=1} \leq D_i w_{u=2} \leq D_i w_{u=0}$

Case 2.1: $D_i w_{u_1} \leq D_i w_{u_2}$

The ordered pair of (u_1, u_2) is within these cases of $(1,0)$, $(1,2)$, $(2,0)$ or $u_1 = u_2$. The remaining is the same as case 1.1.

Case 2.2: $D_i w_{u_1} \geq D_i w_{u_2}$

Case 2.2.1: $u_1 = 0,\ u_2 = 2$

$$
\begin{aligned}
g(x_h+1, x_m, 1) + g(x_h, x_m, 0) &\le w(u_1 = 0, x_h+1, x_m, 1) + w(u_2 = 2, x_h, x_m, 0) \\
&= v(x_h+1, x_m, 1) + v(x_h, x_m+1, 0) \\
\text{By property (c)} \qquad &\le v(x_h+1, x_m, 0) + v(x_h, x_m+1, 1) \\
&= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\
&= g(x_h+1, x_m, 0) + g(x_h, x_m, 1)
\end{aligned}
$$

Case 2.2.2: $u_1 = 0,\ u_2 = 1$

It is identical to case 1.2.2.

Case 2.2.3: $u_1 = 2,\ u_2 = 1$

$$
\begin{aligned}
g(x_h+1, x_m, 1) + g(x_h, x_m, 0) &\le w(u_1 = 2, x_h+1, x_m, 1) + w(u_2 = 1, x_h, x_m, 0) \\
&= v(x_h+1, x_m+1, 1) + v(x_h+1, x_m, 0) \\
\text{By...} \qquad &\le v(x_h+1, x_m, 0) + v(x_h, x_m+1, 1) \\
&= w(u_1, x_h+1, x_m, 0) + w(u_2, x_h, x_m, 1) \\
&= g(x_h+1, x_m, 0) + g(x_h, x_m, 1)
\end{aligned}
$$

Note: for case 1.2.3 and case 2.2.3, when $u_1 = 2$ and $u_2 = 1$, it is tested that numerically these two inequalities hold regardless of the sign of $D_{m,i}v(x_h, x_m, 0)$. The goal is to show $D_{m,i}v(x_h+1, x_m, 0) \le 0$ when

$$
\begin{aligned}
&(u_1 = 2) \text{ for } w(u_1, x_h+1, x_m, 0) \\
&\Rightarrow D_m v(x_h+1, x_m, 0) \le 0,\ (D_m - D_h)v(x_h+1, x_m, 0) \le 0
\end{aligned}
$$

$$
\begin{aligned}
&(u_2 = 1) \text{ for } w(u_2, x_h, x_m, 1) \\
&\Rightarrow D_h v(x_h, x_m, 1) \le 0,\ (D_m - D_h)v(x_h, x_m, 1) \ge 0
\end{aligned}
$$

Since $c(X(t)) = \sum_{j=h,m} h_j X_j(t)^+ + b_j X_j(t)^-$ is separable, it is supermodular and has diagonal dominance. Hence, $c(X(t)) \in V$. Since V is closed under addition and multiplication by positive scalars, and from the fixed point theorem, $v^* \in V$. $\square$

Appendix C
ALGORITHMS

C.1 Value Iteration

C.1.1 Description

Value iteration [5] is a simple yet powerful algorithm to solve Markov decision process (MDP) optimally. The algorithm first initializes the values function for all states arbitrarily, for example, all zeros. Then, in each iteration, value function on each state is updated according to *Bellman's equation* by choosing the action that leads to the successive state with least cost. The Bellman *residual* for a state is defined as the absolute difference of a state value before and after applying Bellman's equation. Value iteration stops when the value function converges, which corresponds to the *error* term, or the largest *residual* term among all the states, becomes less than a pre-defined threshold value δ in the algorithm 1. A Bellman equation is a contraction mapping if for every state, its Bellman residual never increase with the iteration number [15, Chapter 2]. The fixed point theorem ensures the convergence of the algorithm.

C.1.2 Algorithms

Following the definition in [15, Chapter 2], define a MDP $M =< S, A, Ap, T, C >$ and a threshold value δ, where

- S is a finite set of discrete states.
- A is a finite set of all applicable actions.

- $Ap : S \rightarrow P(A)$ is the applicable function. $Ap(s)$ denotes the set of actions that can be applied in state s, and $P(A)$ is the power set of A.

- $T : S \times A \times S \rightarrow [0, 1]$ is the transition function describing the probability of an action execution.

- $C : S \times A \rightarrow \mathbb{R}^+$ is the cost of executing an action in a state.

Value iteration converges to the optimal value function in time polynomial in $|S|$ [54].

Algorithm for Discounted Total Cost

Algorithm 1 Value iteration

Input: an MDP $M =< S, A, Ap, T, C >, \delta$: the threshold value

Initialize V arbitrarily

While true **do**

  $error \leftarrow 0$

  for each state $s \in S$ **do**

    $oldV \leftarrow V(s)$

    $V(s) \leftarrow \min_{a \in Ap(s)}[C(s, a) + \sum_{s' \in S} T_a(s^{'}|s)V(s)]$

    $residual(s) \leftarrow |V(s) - oldV|$

    $error \leftarrow \max(error, residual(s))$

  if $error < \delta$ **then**

    return V

Value Iteration for Average Cost

The idea behind value iteration for the average cost is that the minimal (optimal) average cost is bounded by the limit of $\lim_{n\to\infty}[V(x) - V_{n-1}(x)]$, as seen in theorem 2.2 and [83].

Algorithm 2 Value iteration for average cost

Input: an MDP $M =< S, A, Ap, T, C >, \delta$: the threshold value δ

Initialize V arbitrarily

While true **do**

$error \leftarrow 0$

$m_s \leftarrow \bar{N}$ (N is a large positive number)

$M_s \leftarrow \underline{N}$ (N is a negative number)

for each state $s \in S$ **d**o

$oldV \leftarrow V(s)$

$V(s) = \min_{a \in Ap(s)}\{c(s, a) + \sum_{s' \in S} T_a(s'|s)V(s)\}$

$m_s = \min\{|V(s) - oldV|, m_s\}$

$M_s = \max\{|V(s) - oldV|, M_s\}$

$error \leftarrow |m_s - M_s|$

if $error < \delta$ **then**

return m_s

Appendix D
VALUE ITERATION CODE

Here are the codes for value iteration for the base model in section 2.3, production-inventory system with returns and lost sales. It finds the optimal policy under the total discounted cost scenario. The program will stop when value functions for all the states converge.

Other models, with backorder, non-unitary demand and ect.; or under average cost scenario are implemented in the similar way.

D.1 'state.h'

This file defines the data structure of the state space and state.

```
#ifndef STATE_H
#define STATE_H

#include <vector>

class state
{
private:
        int x1_;
        int x2_;
        double f_;
        double old_f_;
        int best_action_;
        int best_admission_action_;

public:
```

```
        state( int x1, int x2 ) : x1_(x1), x2_(x2), f_(0.0), old_f_(0.0)
            , best_action_(-1), best_admission_action_(-1) {}
        state( int x1, int x2, double f ) : x1_(x1), x2_(x2), f_(f),
            old_f_(f), best_action_(-1), best_admission_action_(-1) {}
        int x1( void ) { return x1_; }
        int x2( void ) { return x2_; }
        double f( void ) { return f_; }
        double old_f( void ) { return old_f_;}
        void change_value( double f ) { f_ = f; }
        void change_old_value(double f) { old_f_ = f;}
        int best_action( void ) { return best_action_; }
        int best_admission_action( void ) {return best_admission_action_
            ;}
        void modify_best_action( int a ) { best_action_ = a; }
        void modify_best_admission_action( int b) {
            best_admission_action_ = b; }
};

class space
{
private:
        int n_;
        int m_;
        state *initial_;
        std::vector<state*> states_;

        //parameters
        double H1; // NEW
        double H2; // OLD
        double MU; // PRODUCTION
        double B1; // LOST SALES FOR NEW
        double B2; // LOST SALES FOR OLD
        double P1; // PRODUCTION COST FOR NEW
```

```
        double P2; // PRODUCTION COST FOR OLD
        double LAMBDA1; // NEW
        double LAMBDA2; // OLD
        double DELTA;   // RETURNED
        double ALPHA; // DISCOUNT RATE

        double NORMALIZECOST;
        double EPSILON;
        double  MAXITER;

public:
        space( int, int );
    long index( int, int );
        double iteration( int );
        int vi( int  );
        int setParameters(double h1, double h2, double mu, double b1,
            double b2, double p1, double p2, double lambda1, double
            lambda2,
                double delta, double alpha){H1=h1; H2=h2; MU=mu; B1=b1;
                    B2=b2; P1=p1; P2=p2; LAMBDA1=lambda1; LAMBDA2=lambda2
                    ; DELTA=delta; ALPHA=alpha;
                NORMALIZECOST=LAMBDA1+LAMBDA2+DELTA+MU+ALPHA; EPSILON=1e
                    -2; MAXITER=1e6; return 1;}

        //for cost rate
        double h( int x1, int x2 );
};
#endif
```

D.2 'value iteration.cpp'

To use the value iteration method, state space needs to be truncated to a finite space. We examine different values of the boundaries until the values / policy is not sensitive to the boundary value. In this case, both x and y are bounded above at 30 and below at 0 (due to lost sales scenario).

The main functions include:

- *iteration* (line 74 - 200): This is one iteration in the value iteration algorithm for all the states. It updates state values from the neighbor states. The new state value is calculated in line 181.

- *vi* (line 204 - 220): This is the value iteration algorithm. It runs *iteration* until all the values in the state space converges.

```
1 // value iteration.cpp : Defines the entry point for the console
     application.
2 //
3
4 #include "stdafx.h"
5 #include "state.h"
6 #include "time.h"
7 #include <string>
8 #include <iostream>
9 #include <fstream>
10 using namespace std;
11
12 double startingValue;
13
14 double
15 space::h( int x1, int x2 )
16 {
17         double currentH = 0;
```

```
        if (x1 < 0)
        {
                x1 = -x1;
                currentH += x1 * B1;
        }
        else
                currentH += x1 * H1;

        if (x2 < 0)
                {
                        x2 = -x2;
                        currentH += x2 * B2;
                }
        else
                currentH += x2 * H2;

        return currentH;
}

space::space( int n, int m ) : n_(n), m_(m)
{
        int i, j, k = 0;

        for ( i = n; i <= m; ++i ) {
                for ( j = n; j <= m; ++j ) {
                        // add boundary conditions here
                        state *s = new state( i, j);
                        // state *s = new state( i, j, 100);
                        states_.push_back(s);
                        if ( i == 0 && j == 0 )
                                initial_ = s;
                        ++k;
                }
```

```
        }
        cout << k << " states were generated\n";
}

long
space::index( int x1, int x2 )
{
        if ( x1 < n_ )
                x1 = n_;
        if ( x1 > m_ )
                x1 = m_;
        if ( x2 < n_ )
                x2 = n_;
        if ( x2 > m_ )
                x2 = m_;

        long ind = x2 - n_;
        ind += ( x1 - n_ ) * ( m_ - n_ + 1);

        return ind;
}

double
space::iteration( int iter )
{
        double error = 0.0;
        std::vector<state*>::iterator si;
        int x1, x2;

        double T1v, T2v, T3v, T4v;
        double best_value_T2 = 0;
        double best_value_T4 = 0;

```

```
        for ( si = states_.begin(); si != states_.end(); ++si ) {
                state *s = (*si);
                x1 = s->x1();
                x2 = s->x2();

                if (x1 == 0 && x2 == 0)
                        startingValue = s->f();

                double old_value = s->old_f();

                double value;
                long index1 = 0, index2 = 0, index3 = 0, index4 = 0,
                    index5 = 0;
                state *s1, *s2, *s3, *s4, *s5;

                // handle boundary conditions
                if ( x1 == m_ && x2 == m_ )
                        continue;

                if ( x1 == m_ ) {
                        index1 = index( m_-1, x2 );
                        s1 = states_[index1];
                        value = s1->old_f() + H1 / ALPHA;
                        //value = (s1->f() + H1 / ALPHA)/NORMALIZECOST;
                        s->change_value( value );
                        continue;
                }

                if ( x2 == m_ ) {
                        index1 = index( x1, m_-1 );
                        s1 = states_[index1];
                        value = s1->old_f() + H2 / ALPHA;
                        //value = (s1->f() + H2 / ALPHA)/NORMALIZECOST;
```

```
                s->change_value( value );
                continue;
            }


            // find all successors
            if (x1 > n_) {
              index1 = index( x1-1, x2 );
            }
            s1 = states_[index1];

            if (x2 > n_) {
              index2 = index( x1, x2-1);
            }
        s2 = states_[index2];

            index3 = index( x1+1, x2 );
            s3 = states_[index3];

            index4 = index( x1, x2+1);
            s4 = states_[index4];

            if (x2 > n_) {
              index5 = index( x1+1, x2-1);
            }
            s5 = states_[index5];

            double best_value = 1e50;

            if (x1 == 0)
                T1v = s->old_f() + B1;
            else {
                if (x1 > 0)
```

```
                T1v = s1->old_f();
        }

        if (x2 == 0)
                T2v = s->old_f() + B2;
        else
                T2v = s2->old_f();

        T3v = s4->old_f();

        // Update T4v
        // First, test when x2 == 0
        if (s->old_f() < s3->old_f())
        {
                T4v = s->old_f();
                s->modify_best_action(0);
        }
        else
        {
                T4v = s3->old_f();
                s->modify_best_action(1);
        }

        if (x2 > 0)
        {
                if (s5->old_f() < T4v)
                {
                        s->modify_best_action(2);
                        T4v = s5->old_f();
                }
        }

        value = (h( x1, x2 ) + LAMBDA1 * T1v + LAMBDA2 * T2v +
```

```
                DELTA * T3v + MU * T4v)/NORMALIZECOST;

            s->change_value( value );

            double diff = value - old_value;
            if ( diff < 0 )
                diff = -diff;

            if ( diff > error )
                error = diff;
        }
        cout << "Iteration " << iter << " " << initial_->f() << " ends.
            error: " << error << std::endl;

        //replace all the old_f_ with f_
        for ( si = states_.begin(); si != states_.end(); ++si ) {
            state *s = (*si);
            s->change_old_value(s->f());
        }
        return error;
}


int
space::vi( int fileIndex )
{
        int iter = 0;
        double error = 1.0;

        // Value iteration for optimal cost function
        while ( error > EPSILON && iter < MAXITER) {
            error = iteration( ++iter );
        }
```

```
        cout << "Optimal value function: " ;
        if (iter >= MAXITER)
                cout << "Maximum number of iteration reached!" << endl;
        else
                cout <<"Number of iterations is " << iter << endl;
        cout << "f(0,0) = " << startingValue << endl;
        return iter;
}


int main (int argc, char * const argv[])
{
        clock_t starting = clock();
        char c;
        startingValue = 0;
        space *sp = new space(0, 30);

        //H1, H2, MU, B1, B2, P1, P2, LAMBDA1, LAMBDA2, DELTA, ALPHA
        sp->setParameters(2, 1, 1, 20, 10, 8, 4, 0.85, 0.4, 0.6, 0.05);
        sp->vi(1);
        return 0;
}
```